THE PDP-9 MINI TIME-SHARING SYSTEM

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Master of Science

by

Robert W. Blean

Thayer School of Engineering
Dartmouth College
Hanover, New Hampshire

June 1972

Examining Committee:

[illegible]
Chairman

[illegible]

[illegible]
Director of Graduate Study

[illegible]

This research was supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Air Force Office of Scientific Research under Contract No. F44620-68-C-0015.

THAYER SCHOOL OF ENGINEERING

DARTMOUTH COLLEGE

The PDP-9 Mini Time-Sharing System

by

Robert W. Blean

Master of Science

JUNE 1972

ABSTRACT

The PDP-9 Mini Time-Sharing System (MTSS) is a small general purpose time-sharing system running on a Digital Equipment Corporation PDP-9 computer with 8K of core memory and three Teletype terminals. MTSS demonstrates the practicality of implementing a general purpose time-sharing system on such a small computer, and illustrates one way in which this can be done.

MTSS provides the facility for three users simultaneously to load or create, debug, run, and save arbitrary machine language programs. These machine language programs may exceed (slightly) 7K in length, or may be arbitrarily large if assembled in 7K segments.

MTSS imposes minimum constraint on the content of such programs. Physical paper tape, Teletype, and disk input/output are available to users using standard IOT instructions. Physical disk and DECtape input/output are available to users by means of Special IOT instructions. Logical file operations on disk or the user's own DECtape are also available by Special IOT instructions.

MTSS is designed to permit additional system programs and features to be easily added.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Professor Miles V. Hayes, for his help, encouragement, and time spent as a sounding board. In addition, I am grateful for the advice of my other committee members.

My thanks also go to Thayer School and to the College for the resources provided which enabled me to do this project. Finally, my thanks to those students who maintain the service software I used in DTSS and on the PDP-9 for their unending patience with my allegations of bugs in their programs (and prompt repairs when my allegations were correct).

TABLE OF CONTENTS

FIGURES

APPENDICES

PROGRAM LISTINGS

- A) Initialization
- B) Executive -- Resident
- C) Executive -- Overlays
  - 1) Memory Protection #1
  - 2) Memory Protection #2
  - 3) Special IOT handler
  - 4) Swapper
- D) Phantom programs
  - 1) MONITOR/system message output
  - 2) LOADER
- E) System-Supplied User programs
  - 1) DDT
  - 2) BASIC Interpreter

INTRODUCTION

The PDP-9 Mini Time-Sharing System (MTSS) is a general purpose time sharing system running on a Digital Equipment Corporation (DEC) PDP-9 computer with 8K of core memory. MTSS is capable of handling arbitrary user-originated machine-language programs whose length may (slightly) exceed 7K without segmenting, or may be arbitrarily large if assembled in 7K segments.

MTSS is a two-fold project. It serves the educational purpose of illustrating one way in which a general purpose time-sharing system can be implemented. It is also an exploration of the practicality of a general purpose time-sharing system on such a small computer.

Educational Role

MTSS provides a relatively simple system running on a relatively simple hardware setup for students to work with. In contrast to a larger public-utility type of system MTSS is designed to be easily altered by students to see how changes affect the system. MTSS is simple enough for this to be practical; it is written flexibly enough to permit easy addition of system features or system programs. The small core size of the PDP-9 proves to be a much less serious limitation than one might suppose. One could implement on

MTSS almost any feature of a larger system he wished to.

Practicality

It is interesting to note that the PDP-9 memory protection hardware was engineered assuming that it would not be used on an 8K machine (see section 4.3: Memory Protection). In spite of that, MTSS is a highly flexible system capable of giving good service to a limited number of users. The main limitation on the number of users is that it is difficult to fit very many Teletype line buffers into a Resident Executive which is itself only about seven hundred (octal) words long.

MTSS Hardware

MTSS uses:

one minimum configuration PDP-9 including:

8K of 18 bit core memory

300 cps paper tape reader

50 cps paper tape punch

console Teletype

16.7 msec real time clock

high speed I/O data channels

program interrupt control

memory protection option
disk controller with one 256K disk platter
DECtape controller with two DECtape transports
Teletype controller with two model KSR33 Teletypes

These numbers are in no way system limits. In addition to the current hardware, MTSS is capable of supporting additional disk platters, DECtapes (DEC's block addressable magnetic tape), and additional terminals. MTSS could be considerably expanded, should the hardware be purchased.

System Transparency

MTSS is designed to permit machine language programs to operate as nearly as possible as they would on a dedicated machine (i.e. MTSS is designed to permit maximum practical system transparency.) Also the user can control from his Teletype nearly all of the functions he could normally control from the console switches. Transparency makes the system especially valuable to beginning machine-language programmers.

System transparency permits programs written for the PDP-9 to be run under MTSS with minimal alterations. Illustrating this, MTSS currently runs the GROWTH system DEBUGGER, LOADER, standard Teletype handler and standard

catalog routines. These are run with little change from the originals, except to add a few features desirable for MTSS users. The BASIC Interpreter is also run this way.

As an extension to its transparent mode of operation MTSS offers users certain Executive services. The user requests such services by using "Special IOT instructions".

MTSS System Programs

MTSS system programs are of three types. The first type is the Executive program. The Executive consists of one Resident Executive program and (currently) four Executive overlays. The overlays are SWAPPER, two Memory Protection overlays, and a Special IOT handler. (Special IOT'S are MTSS EXECUTIVE service calls.)

The second type of system program is the Phantom program. Phantom programs may be thought of as parts of the Executive that run in place of the user for whom they are doing a service. Except that they have privileges normally reserved for the Executive, and except that they alter neither the user's core nor his registers, they are in all respects a user program. The name is derived from the fact that apart from the service it performs, the user cannot detect the presence of a Phantom program.

The third type of system program is the System-Supplied User Program (S-user program). These programs are, at the user's request, overlaid on his core.

User Programs

MTSS permits a user to load (using LOADER) or to create (by patching with DDT) his machine language program. The user can then debug the program (using DDT) or he can run it (from MONITOR or from DDT). Upon completion he can save to disk or to DECtape his core image and/or his "physical disk" (using LOADER).

Alternatively he can create and run a BASIC program using the BASIC Interpreter.

MTSS Disk and DECtape Usage

MTSS uses the disk for program storage and for swapping. It keeps a copy of each of its files (except the initialization and Resident Executive programs) on the disk. Each time it needs a program it reads in a fresh copy.

Also on the disk, for each user, is a place to which he can be swapped, a place to which the non-pure portion of any Phantom program he caused to run for him can be swapped, and

an area referred to as the user's "physical disk". The user's "physical disk" is the only area of the disk he can reference physically. He can do this either with standard IOT instructions or with a Special.

MTSS files are cataloged in the GROWTH system catalog. It is possible for a user to catalog files in the same catalog. This practice is strongly discouraged, however, else the disk would soon be full. It is preferred that the user save his files only on his own DECtape. Files saved on either the disk or on the user's own DECtape are completely compatable with the major operating system on the dedicated machine (the GROWTH system). Likewise, files saved under GROWTH are accessible to MTSS.

System Performance and Verification

MTSS verification has been somewhat informal. One type of verification is that the system runs. Both the Phantom programs (MONITOR, LOADER, DDT, and CATALOG) and the S-user programs (DEBUGGER and BASIC) run successfully. This set of programs includes three written for the system, and three written previously with no thought of the system. That all of these run successfully is regarded as verification of the system's ability to run arbitrary machine-language programs.

The system's ability to handle infinite loops and large amounts of output has been demonstrated by setting from one to three Teletypes doing either an infinite loop (and verifying that each Teletype was getting it's share of service) or doing an infinite core dump to the Teletype.

User response has always been good. In none of the above trials, nor in any specially patched test programs has it been possible to deprive any user of his share of processor time, to prevent him from returning to MONITOR at will, or to prevent him from keeping his Teletype continuously busy with output if that was what his program was trying to do. It has not been possible to drive user response to be slower than one and a half or two seconds, and response is generally faster than that.

The sole exception to these response figures is that if one user is doing a DECtape operation, it temporarily suspends all system functions except buffered Teletype I/O. This produces no noticeable effect generally, but is noticeable if a long DECtape operation is involved. A long DECtape operation is felt to be rare enough that this is not a very significant shortcoming.

Scope

This thesis project establishes a powerful and flexible structure for MTSS. It does not attempt to completely fill in this structure more than is necessary to demonstrate the important features of it. Thus further program level system services would be provided as additional Special IOT instructions, possibly requiring additional Special overlays or Phantom programs. Improved catalog structure, file-building and editing, compilers, and assemblers would all be provided as additional Phantom programs. The DATAPHONE and Graphics II peripherals could be enabled, but would probably each require an additional memory protection overlay. The Automatic Priority Interrupt system could be enabled in the same manner as the Program Interrupt system is now, but would require a larger Resident Executive, which would slightly exceed 1K, pushing the Executive size up to the next boundary, 2K. In the current implementation it was felt desirable to allow maximum size user programs.

Comparison

It is difficult to compare MTSS directly with other small machine time-sharing systems. Most such systems require from 12K to 20K minimum core size, and it is not

always clear how many users this size can handle. Most such systems are then limited to running interpretive programs. Perhaps better than most is DEC's PDP-8 system which can run user programs of up to 4K in length in a machine with a minimum of 12K of core.

By contrast, MTSS can handle arbitrary programs for users. MTSS allows 7K user programs to be run in an 8K machine, and full size programs to also be debugged. With its Phantom and S-user system program structure added to its capability for running arbitrary user-mode machine-language programs, MTSS is particularly easy to expand to add system services as desired. Adding system services does not make MTSS any larger than now. It merely requires more disk storage.

Future Ideas

There are several minor additions that could be made. One could make a virtue of the software echo required for the remote Teletypes by adding a mode to refrain from doing the echoing, thus concealing input.

The catalog structure could be changed to permit filenames of more than three characters, as well as passwords and access bits.

There are also some more important additions. To conserve core space with a lot of Teletypes, a denser packing scheme could be used for Teletype buffers. To minimize Teletype buffer size, an improved scheduling algorithm could be developed to give a priority for core time to a job which has just had it's I/O roadblock status cleared.

A number of MONITOR commands, including specifically the resource allocation and de-allocation commands could be handled from programs, possibly.

If the Resident Executive is allowed to grow, and if the Executive overlay area is allowed to grow, the Graphics II scope and the DATAPHONE could be enabled, and user API service could also be enabled.

A very useful thing would be to have a fault vector for user programs. This should be optionally enabled by either a MONITOR command or a Special IOT instruction. It would, for example, permit system programs such as DDT to retain control when the user does an illegal operation. It would mean that not all currently fatal errors need be so.

Special IOT instructions for bulk data transfers could be easily amended to specify a device table entry rather than an actual device for the transfer. Then this table

could be set either by a MONITOR command or by a Special IOT, thus giving programs full device independence. The transfer commands could also be expanded to include paper tape operations and Teletype operations.

User CAL instructions should be made legal. When that is done then the DEC V5A operating system can be added to the system as a set of S-user programs. There is no reason other than the CAL instruction why V5A cannot be run so far as it fits into 7K.

MTSS is not operating in a vacuum. To maximize its usefulness and to minimize inconvenience for all machine users it must take into account established local patterns of usage as well as available hardware.

4.1) Software Environment

At Dartmouth the PDP-9 is used mainly for two purposes. One purpose is to allow actual hands-on machine experience for beginning machine language programmers. The other (lower priority) purpose is the playing of real-time games on the Graphics II display scope. The locally programmed GROWTH operating system is commonly used; the DEC V5a operating system usage is small and declining.

Most PDP-9 users do all of their file building, editing, and assembling operations under the Dartmouth Time-Sharing System for two reasons. First, DTSS has more familiar and more powerful facilities for these things than does the PDP-9. Second, this frees the PDP-9 from unnecessary work, allowing it to do those things which can be done only on it -- the debugging and running of PDP-9 machine language programs. MTSS, therefore, accepts all forms of output from the DTSS assembler (9MAP).

To permit users to easily switch operating systems without doing frequent disk saves to and restores from DECtape all three systems share the disk (see section 13). Since GROWTH is more commonly used than V5A, MTSS and GROWTH catalogs and files are interchangeable (see section 11).

There are two other fundamental constraints underlying MTSS design. One is that, since most machine usage is to gain familiarity with machine language programming, maximum system transparency should be retained. That is, so far as possible, a program running under MTSS should produce the same results it would on the bare machine. The other constraint is that since we have only 8K of core, users should be allowed as much core as possible. Due to the way the memory protection hardware works this means restricting the operating system to 1K.

4.2) Relationship Between MTSS, GROWTH, and DTSS

Neither GROWTH nor MTSS has its own assembler. To create a program for the PDP-9 one should create a source file in DTSS, assemble it in DTSS using 9MAP, and either punch the object file on a paper tape or write it on a DECtape. Copy either version onto a DECtape with a GROWTH catalog if you wish to. (If you do not know how to do these things consult

the GROWTH system documentation which is available both in the DTSS file PDP9LIB***:DOCUMENT and in a line printer listing binder in the PDP-9 machine room.)

Files written in any format onto paper tape, disk, or DECtape by the GROWTH system are acceptable to MTSS; and files created by MTSS are acceptable to GROWTH.

4.3) Hardware Environment

MTSS runs on a DEC PDP-9 computer with 8K of core memory. Core memory is (optionally) expandable to 32K by 8K increments. Memory cycle time is one micro-second. For high speed data transfers the memory is accessed by a data channel facility.

Clock

Timing is controlled by a sixty pulse per second real-time clock. The clock can be enabled or disabled under program control.

Program Interrupt System

Coordination is achieved through the Program Interrupt control (PI), a single level interrupt facility which can be

enabled or disabled under program control. All input/output devices are interfaced to the PI, as are the clock and memory protection violations. Disk and DECtape can be disconnected from the PI under program control. Upon receipt of an interrupt a flag search must be made to determine which device generated the interrupt.

Memory Protection

The memory protection option permits a boundary to be set which specifies protected core (locations 0 to boundary - 1) and unprotected core (locations boundary to the maximum core address). This boundary can be set, under program control, in 2000(8) word increments. Memory protection violations are caused by an attempt to execute any IOT instruction, to halt, to OR the accumulator switches into the accumulator, to do a memory reference instruction whose effective address is below the boundary, or to do an XCT of an XCT. The last is because chained XCT's form a non-interruptable sequence. Note that indirect memory references where the effective address is fetched from a location below the boundary are quite legal if the effective address is itself above the boundary. In particular this allows the user to reference the auto-index registers indirectly (thus altering them), so the Executive may as well

allow the user full and correct use of them.

References to non-existant memory, including disk or DECtape data channel transfers past the end of core are supposed to be trapped with the non-existant memory reference violation flag set. On an 8K PDP-9, however, this flag is not enabled. Non-existant memory references instead wrap core (i.e. they are used mod 8K). The result is that memory reference instructions will either cause an ordinary memory protection violation, or if they are far enough off the end of core will be legal. But data channel transfers that go off the end of core will not trap. If such a data channel transfer were writing to core it could destroy the resident Executive. Therefore the system must do enough error checking to never let such a transfer get under way.

User mode can be enabled under program control. It is disabled by a program interrupt, by a memory protection violation, by a CAL instruction, or by a CAL (indirect) instruction. The system must be careful or else a CAL (indirect) could seize control of the system.

Peripheral Devices

Standard PDP-9 peripheral devices used in MTSS are a 300 cps paper tape reader, a 50 cps paper tape punch, and a KSR33

Teletype. MTSS currently also has one multi-station Teletype control and two Teletypes connected to it. The PDP-9 can accept up to four Teletype controls, each of which can control up to four Teletypes. For these remote Teletypes software must echo each character typed. The console Teletype has a hardware echo.

DECtape is DEC's block addressable magnetic tape. A single block anywhere on the tape can be read or written. Standard DECtape format is 1100(8) blocks of 400(8) words each. The DECtape can be attached to or detached from the PI system under program control. MTSS has one DECtape control and two DECtape transports. The control is capable of handling up to eight transports.

MTSS has one disk control with one 256K fixed head disk platter. The disk can be attached to or detached from the PI SYSTEM under program control. The control can handle up to eight platters.

Except for Teletypes it is not practical to make a permanent assignment of any resource to a particular user. For one thing there would not be enough resources for all users. In the second place, even if there were, someone would be apt to have a legitimate need for more than his share of some resource. The way MTSS handles resource allocation is that it permits a user to specify to MONITOR the resources he needs. If the resource specified is free it is assigned. Otherwise a warning message is printed. When a user is done with a resource he must have MONITOR de-allocate it. No one can de-allocate anyone else's resource unless he is on a control line. An allocation request is a MONITOR command:

ON <resource>

A de-allocation is of the form:

OFF <resource>

Resources can neither be allocated nor de-allocated under program control. Available resources and their abbreviations are:

abbreviation	resource name
ACS	hardware accumulator switches
DTn	DECtape handler #n
PTR	paper tape reader
PTP	paper tape punch
CNT	control line

When a job tries to execute an IOT instruction pertaining to a device not previously allocated to it, the job is terminated with an appropriate error message and control is returned to MONITOR.

All flags of devices not allocated to a job will be turned off in its software I/O device status word.

TELETYPES

Teletype I/O is as nearly transparent to the user as is practical. The only character the user cannot use indiscriminately is a null (control shift 'p'). If the user types one it will kill his running program, stop all I/O, and return control to MONITOR. No character other than null has any meaning to the MTSS system when a user program is running.

In order to insure that Teletype I/O will be continuous, even while the user is swapped out for a maximum length of time, The Resident Executive keeps an I/O buffer and a one character buffer for each Teletype. The I/O buffer is used as either an input buffer or an output buffer at any given time. Its purpose is to permit uninterrupted Teletype I/O while the running program is swapped out of core.

On input, whenever the user types a character other than null (or control 'x' if the delete option is on and output is in progress), his software keyboard flag is set, his keyboard I/O roadblock flag is cleared and the character overwrites the previous character in the one-character buffer. If no teleprinter output is in progress the character is also placed in the next slot in the line buffer. If the buffer is full the last character in it is overwritten.

When the user does a keyboard read buffer (KRB) instruction his accumulator will be supplied with a buffered input character. If output is in progress, this will be the character from his one-character buffer and his software keyboard flag will be cleared. If output is not in progress this will be the oldest character from his line buffer; the software keyboard flag will be cleared only if it is the last character from this buffer. Thereafter, continued KRB's will continue to read the same character if there is no more input typed.

When the user program does a keyboard skip on flag (KSF) instruction his program is made to skip if and only if his software keyboard flag is set. Keyboard I/O roadblock occurs when a program does a KSF which fails to skip and is followed by a <JMP .-1>. It is relieved whenever the user

next sets his software keyboard flag. Thus input is character oriented rather than line oriented. This causes an inefficiency which is tolerable to a system the size of MTSS and which is necessary in order to have reasonable system transparency.

When the user program does a teleprinter send (TLS) instruction and the output-in-progress flag is already set, its output is placed in the next available character position in the line buffer. If the buffer is already full the previous character is overwritten. If output is not in progress any unread input is destroyed, the character is sent to the teleprinter, the software teleprinter flag is set, and the output-in-progress flag is set.

When a program does a TSF the skip will occur if the teleprinter flag is set and if there is also room in the output buffer. Teleprinter I/O roadblock occurs if the program does a TSF which fails to skip and is followed by a <JMP .-1>. It is relieved the next time a character is printed reducing the amount left in the I/O buffer below the cutoff point necessary to maintain comtinuous output.

Whenever the Resident Executive receives a teleprinter interrupt it checks to see whether or not output is already in progress. If not, it exits. If so, it tries to get

another character to print. If it gets one it prints it; but if the buffer is now empty it changes the I/O buffer to an input buffer and then checks the software keyboard flag. If that is set it copies the character from the one-character buffer into the line buffer.

PAPER TAPE

Paper Tape Reader and Paper Tape Punch are handled as two distinct devices. Each may be assigned to only one job at a time. They may be assigned to different jobs or both to the same job. Since neither device cares much about real time, a software image exists only of their flags. Remaining information is kept in the hardware.

Whenever a paper tape interrupt occurs the Executive sets the proper software flag, clears the hardware flag, and exits.

When a job issues a paper tape command an error message is printed if the resource has not been allocated to that job. Otherwise if the command pertains to the device flag it is simulated based on the software flag. Any other paper tape command is executed.

Paper tape I/O is not buffered, so it pauses whenever

the job using it is swapped out of core.

DISK

Disk I/O can be permitted to all user's simultaneously, since each user has his own 'physical disk' and his own set of software disk registers and flags. The Executive cannot overlap much of anything with disk operations, anyway, so it uses non-interrupting disk operations. User programs may use either interrupting or non-interrupting disk operations.

Whenever the Executive gets a disk interrupt it copies all of the hardware information into the current job's software images and exits.

A user program may use either standard disk IOT instructions or Special IOT instructions to cause transfers of data between disk addresses 0 - 17777, and core addresses 2000 -17777. All disk operations will be simulated by the Executive so that they are transparent to the user program. The executive will map the disk addresses onto the job's "user physical disk". Any attempt to violate either the disk addressing or core addressing constraints will cause an appropriate error message to be printed and the job to be terminated with control returned to MONITOR.

DECTAPE

DECtape I/O is not conducted in a transparent manner. It is handled solely by Specials. For a DECtape Special to be legal it must request the transfer of data between unprotected core and a DECtape handler previously allocated to the job. If the Special is legal, the transfer will be carried out and then control will be returned to the job. Otherwise the job will be terminated, an appropriate error message will be printed, and control will be returned to MONITOR.

ACCUMULATOR SWITCHES

The hardware accumulator switches can be read only by a job to which they have been allocated. For any other job such instructions will be simulated on its software "accumulator switches register".

OTHER

Other I/O devices and commands are illegal. The problems involved in implementing them are mainly concerned with a lack of core room for the necessary buffers and flags. If they were to be made legal, the Executive would have to have

software versions of all of their flags, and in some cases would have to buffer information for them. It was felt to be better for now to keep the compact Resident Executive that MTSS has than to try to include all of these extra devices. In addition they establish little or nothing in principle that MTSS does not already have.

If the Executive gets an interrupt it does not recognize, it just clears all other possible interrupts and exits.

6.) How to Initialize MTSS

1) Find the DECtape labelled "GROWTH"

2) Mount it on a DECtape handler: choose a DECtape handler. Turn the LOCAL-OFF-REMOTE switch to LOCAL. Rewind onto the left-hand take-up reel whatever DECtape is currently mounted. Turn the switch to OFF. Remove the old DECtape. Put the "GROWTH" DECtape on the left-hand spindle. Take two or three turns of tape around the right-hand take-up reel. Turn the switch to REMOTE. Turn the WRITE LOCK - WRITE ENABLED switch to WRITE LOCK to protect the system DECtape from accidental damage.

3) Open the door to the disk cabinet. Check to be sure that the WRITE LOCKOUT switches are all DISABLED except those numbered 54, 60, 64, 70, and 74 should be ENABLED. Close the door.

4) Turn the LINE-OFF-LOCAL switch on the console Teletype to LINE.

5) Carefully mount the first part of the paper tape labelled "GROWTH bootstrap" in the paper tape reader. Be

sure the feed holes in the paper tape are correctly seated on the drive sprocket. Set the address switches to 100(8) (all of the white console switches should be down, except the left-hand bit 11 switch should be up). Depress and release the I/O RESET toggle. Briefly depress the little white button on the paper tape reader. Depress the READ IN toggle switch.

6) The paper tape will be read in and a user number will be requested on the console Teletype. Type any six digits followed by a carriage return. Next a dollar sign ($) will be printed on the console Teletype. You type "TPn:INT", substituting for n the number of the handler on which you mounted the MTSS DECtape. (If it is handler 8, call it handler 0.)

7) Relax and watch the DECtape move back and forth for a short while. It is busy initializing MTSS on the disk. When done it will type a completion message followed by:

```
TSSMON
#
```

The system is now up and running.

6.1 What Actually Happens

The INITIALIZATION program has internal lists of the required system programs. When it runs, it first unsaves from the disk any existing versions of them. Then it purges the disk to compact storage. Next it copies all system programs from the system DECtape to the disk. This insures that all system files on the disk have the right parameters and are current.

During initialization the correct RESIDENT and SWAPPER catalogs are set up.

6.2) Adding New System Programs To MTSS

To insert a new program into MTSS save it on the MTSS library DECtape and:

1 -- Overlay programs: add its name to the list of overlay programs (OFILES). In the OFILES list the memory protection overlays must be listed consecutively. Other than that, order is immaterial.

2 -- User-type system programs: add its name to the list of user-type system programs (UFILES). Order is immaterial.

3 -- Phantom-type system programs: add its name to the list of Phantom-type user programs (PFILES). Order is immaterial.

Note that all program names must already be defined in the DEFINS program. That is to make them available to all MTSS programs.

LOADING AND RUNNING A PROGRAM

7.1) System Programs

To start (or to return to) MONITOR type a break. To load any other system program type its name in response to MONITOR's sharp sign (#). System programs start automatically.

7.2) User Supplied Programs

To load a user-supplied program use LOADER. Using LOADER, replace the file 'COR' with the desired user file, and EXIT. The user physical disk may be loaded in the same way by replacing the file 'DIS'. (See the LOADER description.) To run a program in user core, TRANSFER to the start or CONTINUE from either MONITOR or from DDT.

To save the contents of user core use LOADER. Save or replace the disired file from the file 'COR'. The user physical disk may be saved in the same way by using 'DIS' as the source file. Then EXIT.

8.) MTSS Conventions

MTSS has few input conventions. A user can at any time type a break or a null (control shift 'p'). That will stop his running program, stop further Teletype output, and transfer control to MONITOR.

For all system programs a control 'x' will delete the remaining output. This convention is optionally available to user programs (see MONITOR Delete command).

A letter may be either upper case or lower case; a digit may be either an octal digit (0,1,...7) or a decimal digit (0,1,...,9); a delimiter is any character that is neither a letter nor a digit; a number is any sequence of one or more digits followed by a delimiter; a word is any sequence of letters and/or numbers followed by a delimiter. Note that a trailing space is a legal delimiter. Leading spaces are ignored -- therefore multiple spaces have precisely the same effect as one space.

On a cataloged disk or DECtape a filename is of the form <device>:<name> where device is DTn, TPn or DKn, and the name is a three character or shorter name for the file. The filename of an uncatalogued file is either PPT (if it is on a paper tape) or else it is of the form <device>,<starting

block number>. In addition, core (COR) and user disk (DIS) may be used as filenames.

Whenever MONITOR wants input it will print a sharp sign (#). Whenever any other system program wants input it will print a question mark (?). This is so that the user need never be confused as to whether he is in MONITOR or whether he is in some other program.

PROGRAM STRUCTURE

The program structure used by MTSS is one of its most important features. It is the key to MTSS' success in offering a very large amount of service on a very small machine. It is also the key to MTSS' great flexibility and power. Basically MTSS considers programs to be of five types:

1) Resident Executive
2) Executive overlays
3) Phantom system programs
4) System-Supplied user programs
5) User-Supplied user programs

The first four of these are types of programs supplied by MTSS. Which system services will be supplied in which manner is in most cases a design choice, though in a few cases choice is dictated by logical or practical necessity. The fifth type of program is that which the ordinary user wishes to run under MTSS.

9.1) Resident Executive

The Resident Executive is the only core-resident program in MTSS. Since it is core-resident, it uses a very scarce resource on our 8K PDP-9 -- core space. Consequently it must be kept as small as possible. The Resident

Executive is responsible for supplying the storage and those service routines which logically must be in core at all times. If it then has any further room it may supply storage and routines intended to make the system more efficient.

As an example of this philosophy, the routines to initiate handling of program interrupts and to retrieve SWAPPER from the disk must be in core at all times. However SWAPPER itself is not core-resident. Teletype I/O buffers and their handling routines are core-resident only as a matter of system efficiency.

The Resident Executive is a collection of largely disjoint routines and storage areas. (See figures 12.1, 12.2.) The Resident Executive contains:

a) temporary storage
b) I/O parameters for each user
c) an I/O buffer for each Teletype
d) routines to handle Teletype I/O
e) routine to handle input of a null
f) allocation records for system resources
g) state-of-the-system information
h) routines for all legal I/O devices
i) information to retrieve SWAPPER
j) routines to service all PI interrupts

k) CAL service

9.2) Executive Overlays

Executive Overlays provide those user services not provided by the Resident Executive but which either logically or for efficiency must be provided by a routine co-resident with the user. The Executive Overlay area always contains one of the Executive Overlays. (See figures 12.1, 12.2.)

9.2.1 SWAPPER

It would be nice to make SWAPPER a part of the Resident Executive; the system would run faster if this were done. However this is not logically necessary if the Resident Executive has enough information to call SWAPPER from the disk as needed. The reason SWAPPER was made an Executive Overlay in MTSS is that otherwise the Executive would exceed 1K, and therefore have to be enlarged to 2K. This would be a reasonable and desirable change to make in MTSS if additional core were purchased.

Another logical alternative for SWAPPER would be to be a Phantom program. This was rejected because it would be inefficient and would also require an enlarged Resident Executive to allow user core to be swapped out before SWAPPER

WAS read in over it.

SWAPPER does all swapping for the system. It can be initiated by the Resident Executive, any Executive Overlay, or by any running program. In each of these cases the actual SWAPPER fetch is accomplished by the Resident Executive, using its resident disk handler and resident catalog. No matter who calls SWAPPER, it is entered with its flag set so that further interrupts will not actually call SWAPPER prematurely. Instead, their occurence will be noted for later action.

If the Resident Executive initiates a SWAPPER call, it is the result of a PI interrupt. Either a user has typed a null to request the killing of his current program and the calling of MONITOR, or else the clock has run out and there is another user ready to run.

If a Memory Protect Overlay initiates a SWAPPER call it is because it has detected an error condition and is calling the error message output routine. A Special IOT Overlay may initiate a SWAPPER call either for this reason or as the result of a Special.

A user program can call SWAPPER only indirectly, as the result of an error or of a Special call. A Phantom program

can generate a SWAPPER call this way, or it may intentionally call SWAPPER (e.g. MONITOR uses SWAPPER to call DDT).

Each of these types of SWAPPER calls except the last has its own special entry to SWAPPER. These special entrances allow SWAPPER to set up its own parameters, thus simplifying SWAPPER modifications. They also minimize the required resident or overlay code.

SWAPPER runs as much as possible with the PI system enabled to permit continuous Teletype I/O. The generalized sequence of SWAPPER actions is:

a) if a special entrance, do necessary setup
b) prepare to turn on the interrupt system
c) turn on the interrupts
d) do the requested swap activity

The requested swap activity is determined by a bit-coded word directing SWAPPER to do, in order, any or all of the following things:

a) swap out the current user's core
b) swap out the current user's job table
c) set the current user to also be the next user
d) read in the next user's job table
e) see if next user has an outstanding MONITOR request
f) read in the next user's core

g) set up the named Phantom program for the next user

h) set up the named S-user program for the next user

i) record the new core user

j) override the restart address

k) restore the user's registers, etc., and go

Swap out the current user's core: copies all of user core to the user's core image on the disk if the current program type is USER. If the current program type is PHANTOM, the non-pure code portion of core is copied out to the user's Phantom core image.

Swap out the current user's job table: is ignored if the current user is also the next user.

Read in the next user's job table is also ignored if the current and next users are the same.

See if the next user has an outstanding MONITOR request: replaces whatever the next user's job was to have been with MONITOR if he has typed a null since the last time he was in core.

Read in the next user's core: copies in the saved copy. It then checks for a Phantom program, and if it finds one copies in the pure code portion provided it is not already in core. Typically the previous user will have been using

MONITOR and the next user will want to. In this case the pure code copy is unnecessary and is not done.

Alternatively SWAPPER can set up a specified Phantom or S-user program for the user.

The last thing that SWAPPER does before starting the new user is to record the new user's name to update its internal records, and then to copy itself out to the disk to record the update.

9.2.2) Memory Protection

The memory protection overlays handle all memory protection violations, either returning to the user after taking appropriate action or printing an appropriate error message and returning to MONITOR.

Logically, the handling of memory protection violations need not be in the Executive. The Executive could note the violating instruction, swap out the user, and swap in a Phantom program to analyse the violation. However, each violation would then cause several disk operations for a total of about 16K words transferred. Memory protection violations are expected to be frequent enough to make this impractical, so MTSS handles memory protection violations with Executive overlays which are very carefully laid out to minimize required overlay exchanges.

Memory Protection is designed, so far as possible, to allow the user to run an arbitrary machine-language program involving any of the system resources except DECtape with the same results that program would produce on a dedicated machine. This is felt to be an important goal for one major class of Dartmouth PDP-9 users -- beginning machine-language programmers.

9.2.2.1) Memory Reference Instructions

The memory protection boundary is set to 2000(8). Therefore any memory reference to locations 2000-17777 is legal. Programs should be assembled to run above 2000 because this machine has no hardware address relocation. In addition, the Executive will allow direct memory references to the locations 0-37. For various hardware reasons, the user's contents of 0 and 10-17 are stored in their true locations, but 1-7 and 20-37 are stored in a table. This means that both indirect and direct program references to 0 and 10-17 will work, but that only direct program references to 1-7 and 20-37 will work. No direct memory references to 40-1777 will work and undefined results are produced by indirect memory references through these locations. No instruction that results in a transfer to any location below the boundary will work. Note that this means CAL and CAL (indirect) are also illegal instructions. The user may use all program accessible registers.

9.2.2.2) Operate Instructions

Operate instructions which get trapped are those which have either the halt or the or-the-accumulator-switches bit set. The actual accumulator switches are considered a system

resource and are assigned to users just as any other resource, such as the paper tape reader. Each user has a software "accumulator register" which he can examine or alter through MONITOR. When an OAS instruction is encountered it is carried out by the memory protection overlays using either the actual switches (if that resource is assigned to the user) or else using the value in the user's software "register". A HLT instruction causes a return to MONITOR, with an appropriate message printed.

9.2.2.3) Input/Output Instructions

All IOT instructions are detected by Memory Protection. Certain IOT instructions not otherwise used are used as Special Executive Service requests. (See section 9.2.3.)

The memory protection overlays contain a table of legal IOT instructions. Any IOT instruction other than a Special or one of these will cause a return to MONITOR with an appropriate error message being printed. The following IOT instructions, including all of their microcoded variations are legal:

basic IOT instructions (IOT, IORS, CAF, TTS, SKP7)
program interrupt instructions (IOF, ION)
console Teletype instructions (KSF, KRB, TSF, TCF,

TLS)

DBK and DBR

paper tape reader instructions (RSF, RCF, RSA, RRB, RSB)

paper tape punch instructions (PSF, PCF, PSA, PSB)

disk instructions (DSSF, DSCC, DRAL, DRAH, DLAL, DLAH DSCF, DSFX, DSCN, DLOK, DSCD, DSRS)

The Executive keeps in the user's job table a record of all of the user's IOT operations and of their results. When the user attempts an IOT instruction its effect is simulated as transparently as possible, based on the current state of the user's IOT records. In most cases, the user will be unable to detect this simulation.

Basic IOT Instructions

All basic IOT instructions are implemented in a totally transparent manner for user programs.

Program Interrupt Instructions

Both program interrupt instructions are implemented in a totally transparent manner for user programs.

DBK and DBR

Both instructions are implemented in a totally

transparent manner for user programs except that a terminal error message is caused by a DBR which is not followed by a <JMP Y,X>. After a DBR instruction, the jump will occur as if bit 2 of word Y were turned on (regardless of its actual state), but the bit itself will be unaffected.

Paper Tape Instructions

All paper tape reader and paper tape punch instructions are implemented in a totally transparent manner for user programs. Since paper tape I/O is not buffered by the Executive, tape operations can occur only while the user's program is actually in core. Any paper tape instruction by a user who has not previously been allocated the device will result in a terminal error message.

Disk Instructions

All disk instructions are implemented in a totally transparent manner for user programs except that data transfers involving a core address less than 2000(8) or a disk address greater than 8K will result in a terminal error message.

Console Teletype Instructions

Console Teletype instructions are handled in as nearly

transparent a manner as practical. Additional transparency is possible only if one were willing to have all Teletype I/O stop whenever the user was swapped out. The chief variation from transparency is best shown by the following example from the GROWTH system Teletype handler.

In this handler, message output involves the following sequence of instructions:

```
TSF
JMP .-1
KSF
TLS
```

The effect is to delete further message output after any key is struck. Under MTSS, the program will delete all further output after the remainder of the I/O buffer is printed.

9.2.3) Special IOT Instructions

Special IOT instructions are a particular block of otherwise unused IOT instructions. They are used by a program running under MTSS to call upon the Executive for system services. The fundamental Special IOT instruction is 705000. Possible Specials range from Special+0 to Special+377. Only a few of these are currently enabled, leaving this as one area for major future system expansion.

When the user executes a Special, the Special handler overlay is called; it either services the user's request or else it prints an error message and returns control to MONITOR.

Specials could logically handled by a Phantom program. It does seem more efficient, however, to do a disk operation as a core-to-disk operation using an overlay than to have to do a buffered disk-to-disk operation using a Phantom program.

9.2.3.1) MPOFF

MPOFF (705000) is legal only for Phantom programs. Control is returned to the user at the next instruction after the Special, with the state of the machine unchanged except

that user mode is disabled. The program itself should re-enable user mode as soon as possible by issuing an MPEU (701742) instruction to guard against its own bugs crashing the system.

MPOFF should be disallowed as soon as practicable by adding enough Executive services to the system to make it unnecessary. This will greatly enhance system reliability.

9.2.3.2) TERMINATE

TERMINATE (705001) is legal for all programs. Its effect is exactly the same as if a HLT instruction were encountered in the running program except it returns control to MONITOR without the error message "HALTED AT. . . " being printed. In either case, if MONITOR is requested to 'CONTINUE', program execution will be resumed at the next instruction with registers unaltered.

DISK AND DECTAPE SPECIALS

The disk and DECtape Specials make use of a modified standard GROWTH system disk/DECtape handler; to simplify modifying stand-alone programs to run under MTSS, the format used by the Specials is the same one that the handler normally uses.

The disk/DECtape Specials provide all programs with the capabilities of:

1) reading or writing in a logical-block-addressed format the program's DECtapes or "user "physical disk".
2) reading in a logical-block-addressed format the actual physical disk.

In addition, Phantom programs can write in a logical-block-addressed format the actual physical disk. These capabilities allow device independent programming with respect to disk and DECtape.

All disk/DECtape Specials are executed with the AC containing a pointer to a list of parameters of the following form:

word1: bits 0-2 are the DECtape handler number or the physical disk number, as appropriate.
bit 3 = 0 for a DECtape operation; = 1 for a disk operation.
bits 8-17 contain the block number for the start of the data transfer.
word2: core address for the start of the data transfer.
word3: word count to be transferred.

The disk/DECtape Specials perform the following checks:

1) An attempt to read or write off the end of a DECtape or disk generates an error message for the user.
2) An attempt to transfer data to or from a core address in excess of 8k generates an error message for the user.
3) A core address below the memory protect boundary is legal only for Phantom programs. If a user program attempts a data transfer to or from such an address, an error message is generated for the user.
4) An attempt to transfer data to/from a non-existant disk generates an error message for the user.
5) An attempt to transfer data to/from a DECtape not assigned to the user generates an error message.
6) An attempt by a user program to write to the physical disk generates an error message.

Return of control to the user:

1) If the disk/DECtape transfer is successfully completed control is returned to the user at the address the user passed in the MQ.
2) If a device error was encountered control is returned to the user one location past the Special.

3) If a user software error is encountered an error message is printed on his Teletype and control is returned to MONITOR.

Some possible causes of a "device error" are:

1) a disk or DECtape hardware malfunction
2) a DECtape called which has not been remote-enabled.
3) a DECtape not wound far enough onto the spool to start.

9.2.3.3) READ and WRITE

READ (705002) and WRITE (705004) are legal for all programs. These Specials use the standard disk/DECtape format (see above.) They cause the operation indicated by their parameters to be attempted to/from the DECtape or "user physical disk".

1) if the READ/WRITE is to/from DECtape, it is passed along unaltered.
2) if the READ/WRITE is to/from the disk, the block number is understood to refer to the block desired on the user's "physical disk".

9.2.3.4) PREAD and PWRITE

PREAD (705003) and PWRITE (705005) are identical to READ AND WRITE except that:

1) disk references are to the actual physical disk instead of to the "user physical disk".
2) PWRITE is illegal for user programs

9.2.3.5) OPEN

OPEN (705018) is legal for all programs. The disk file whose name is passed in the AC is located and its parameters are stored in the user's job table. On entrance the following parameters are passed:

AC: filename to be opened
word1: OPEN
word2: bits 0-2 handler number
bit 3 is 0 for DECtape; 1 for disk

Return to the user is:

+1 for a hardware error
to an error message and MONITOR for a software error
+2 for success

In addition to any applicable error message which can be caused by a disk/DECtape Special, OPEN can also cause a "file not found" message.

9.2.3.6) COPY

COPY (705019) is legal for all programs. It provides core-to-device and device-to-core copies to or from files on DECtape or on the system disk. Only Phantom programs are allowed to copy from core to any disk file other than 'DIS' (user "physical disk"). On entrance, the parameters passed are:

AC: bit 0 : = 0 for device-to-core copy
= 1 for core-to-device copy

MQ: bits 5-17: user's desired restart address

word1: copy

word2: bits 5-17: starting core address for the copy

word3: length of the copy

Control is returned to the user after a succesful copy at the user-specified restart address. This allows a 100% overlay. An error message is printed and control is returned to MONITOR if for any reason the copy was unsuccessful. This is because that is what should happen for a software error on the part of the user. If the error was a hardware error, it is probably unrecoverable, anyway.

9.3) System services -- Phantom programs

System services not provided by either the Resident Executive or an Executive Overlay are provided by Phantom programs. A Phantom program is a system-supplied program which runs in place of the user for whom it is doing a service -- it occupies his place on the job queue, and if billing were being done its use of resources would be billed to him. A Phantom program runs mainly in user mode both to permit it to access the full range of executive services and to protect the Executive if the Phantom should have a bug. A Phantom program is so named because except for the service it performs its presence is undetectable by the user; it alters neither the user's core nor his registers.

When a Phantom program is swapped out only its non-pure portion (2000-3677) is swapped. That portion is swapped to the user's Phantom core image, preserving the actual core image.

Available Phantom programs are MONITOR, LOADER, DDT, and CATALOG.

9.3.1. MONITOR

MONITOR is the user's main communication with the

system. It is a phantom program which can perform certain services for the user, such as initiating or terminating his session. It can call any other system program, or it can initiate running the user's program.

9.3.1.1 MONITOR Commands

Any command which is longer than three characters can be abbreviated to its first three characters.

ON <resource> requests that the system allocate to the requesting user's sole use the specified <resource>. To prevent an error message this must be done prior to running a program requiring the <resource> in question. When one user has been allocated a <resource>, no other user can use or be allocated that particular <resource>.

OFF <resource> tells the system that the <resource> is no longer needed and can be de-allocated. The <resource> is now available once more to other users. A user may not do an OFF on another user's <resource>. Available resources are:

ACS -- hardware accumulator switches

CNT -- control line

DTn -- DECtape handler #n

PTP -- paper tape punch

PTR -- paper tape reader

TPn -- same as DTn

HELLO tells MONITOR that the user is a new user. The command results in all of the user's flags being cleared, his resources being de-allocated, and his core and disk being zeroed.

GOODBYE OR BYE tells MONITOR that the user has finished his session. The result is the same as if he had typed HELLO.

EXPLAIN produces a list of legal MONITOR commands.

CAF stands for CLEAR ALL FLAGS. It does just that.

VALIDATE OR V requests MONITOR to underprint an area for the user to type a password. If MONITOR recognizes the password, the user will be allowed certain added privileges, such as access to system core and write access to the physical disk. This validation is needed for no normal user function.

XDUMP is a simple physical core dump, requiring validation.

XPATCH is a simple physical core patch, requiring validation.

DDT calls the DDT phantom program for the user.

DEBUGGER calls the S-user debugger program.

BASIC calls the BASIC Interpreter for the user.

CATALOG calls the CATALOG module for the user.

LDR calls the LOADER for the user

GROWTH aborts MTSS and restarts the GROWTH system. Validation is needed to do this.

TRA or T or JMP or J <address> starts the user program running using the current values of its registers. These values are available through DDT.

ZER COR sets the entire user core area to zero

ZER DIS sets the entire user "physical disk" to zero

9.3.1.2 Validation

A user who wants to be validated types VAL or V. This command will be overprinted. The user then types his password on the underprinted area. If MONITOR recognizes the password, appropriate validation privileges are granted. Validation is necessary for a few activities, such as any disk write activity not to the user disk nor accessing actual

machine core.

No normal user activity requires any form of validation.

9.3.1.3 Control Line

Any user can get a control line by typing the command 'ON CNT'. The reason for having the control line convention at all is to insure that certain potentially damaging things are done knowingly, not by accident. In accordance with this idea, the control line request is valid for the remainder of the current command line only. For this feature to be of any practical use it must be used as a part of a multiple command line, and typed on the line prior to any other command requiring control line permission.

With a control line enabled, a user can de-allocate any resource, even if it is currently allocated to some other user. He can also give the GROWTH command, which bootstraps the GROWTH system, aborting MTSS.

9.3.1.4 MONITOR Error Messages

Nearly all MONITOR error messages indicate the word which MONITOR was trying to interpret when it discovered an

error. For this purpose each delimiter is considered to end a word.

ALL DECTAPE HANDLERS ALREADY ALLOCATED -- an attempt has been made to allocate to a user a DECtape handler when all possible dectape handlers are already allocated.

FORMAT ERROR -- can indicate a variety of things, such as a decimal digit included in an octal number, or a letter included in any number.

NOT YOUR RESOURCE -- indicates an attempt to de-allocate a resource belonging to some other user.

RESOURCE ALREADY ALLOCATED -- indicates an attempt to allocate a resource already belonging to some other user.

VALIDATION ERROR -- an incorrect password has been given in a validation attempt.

9.3.1.5 Run-Time Error Messages

Nearly all runtime error messages print an address with themselves. This is the address of the instruction causing the error message, except that with a transfer-type instruction it is the address the program was attempting to transfer to.

BAD ADDRESS: <address> -- a memory reference instruction whose effective address was in the range 40-BOUNDARY was attempted.

CAL: <address> -- for now, all CAL instructions are illegal.

CHAINED XCT'S -- chained XCT'S form a non-interruptible sequence, and hence are illegal.

CORE OVERFLOW: <address> -- a data channel transfer was attempted which would result in core addresses being generated beyond the maximum core address.

DATA TRANSFER TO/FROM PROTECTED MEMORY: <address> -- a data channel transfer was attempted which would have resulted in transfering data to/from illegal core addresses (40-BOUNDARY).

DEVICE OVERFLOW: <address> -- a data channel transfer was attempted which would have resulted in transferring data past the end of the physical peripheral for which it was intended -- user disk or DECtape.

FILE NOT FOUND -- an attempt was made to open a file which the system was unable to find.

ILLEGAL IOT INSTRUCTION <instruction> AT <address> --an

IOT instruction was issued which is not recognized by the system.

ILLEGAL TRANSFER TO <address> -- either a JMP or JMS to the indicated address was attempted, or else something was done to result in an attempt to resume the program in a protected address. Unfortunately the hardware does not detect the error until after the PC is changed, so the actual instruction generating the error is not available.

NON-EXISTANT DISK REFERENCED: <address> -- an attempt was made to transfer data to/from a non-existant physical disk. (Currently the only legal disk is disk #0.)

PROGRAM HALTED: <address> -- an operate instruction with the halt bit set was encountered at the given address. Of course this is not necessarily an error. However a cleaner program termination can be made using the TERMINATE Special IOT instruction.

UNASSIGNED DEVICE REQUESTED: <address> -- the program attempted to use a physical device not allocated to it.

9.3.2. LOADER

The LOADER is the general means of manipulating MTSS files. LOADER will create, update, or delete files from MTSS DISK or DECtapes. It will accept several data formats.

9.3.2.1) Formats

GROWTH -- this is a core-image binary format file. It is the standard format for a cataloged file.

ABSOLUTE -- this is absolute loadstring binary, which is the format output by the assembler. It is the format a paper tape or non-cataloged dectape of an assembly will be in.

BINARY -- straight binary. If the device is paper tape a hardware read-in format tape will be read or punched, as appropriate.

9.3.2.2 Filenames

See Section 8 (MTSS Conventions).

9.3.2.3) Commands

Only the first three characters of any word in any

loader command are significant and need be typed.

CLEAR <device>

CLEAR creates for the device a new catalog containing only the catalog itself.

EXIT

EXIT should always be used to terminate the LOADER in order to insure that the device catalog is brought up to date.

IDUMP <device1> <device2>

Incremental Dump copies all files from device2 onto device1 without destroying anything that was previously on device1. In the case of duplicate filenames, the previous file on device1 will be lost. An automatic PURGE is done on device1 at the completion of the dump.

LDUMP <device1> <device2>

Logical Dump creates a new catalog for device1 and then copies all files from device2 onto device1 without affecting device2. This is useful for creating backup tapes.

PUNCH <filename>

PUNCH punches a hardware read-in format paper tape from the file named.

PURGE <device>

PURGE is merely an LDUMP where devicel and device2 are the same device. The effect of this is to recover storage lost by previous UNSAVE commands. This may be fairly time consuming if the device is a DECtape.

REPLACE <filename><format><filename>

REPLACE is the same as SAVE except that the destination filename is assumed to already be saved, and therefore not need the full specification as given in the save command.

SAVE <fnl><start address><end address><format><fn2>

<Fnl> must be a catalog fn. <Fn2> is of a type determined by the format. It can ba a non-cataloged filename if the format is ABS; it can be a cataloged filename if the format is GRO.

SAVE creates a cataloged filenamel and copies into it, subject to the stated start and end addresses the file from filename2. An error message is printed if the destination filename is already used.

Common SAVE formats are:

```
SAV DTn:NAM 100,1000 ABS PPT
        or
SAV DTn:NAM 100,1000 ABS DT1,1
```

to save an assembly, or

```
SAV DTn:NAM 100,1000 GRO DTq:ABC
```

to copy file ABC from DECtape #q onto DECtape #n as file NAM.

UNSAVE <filename>

UNSAVE deletes from the device catalog the given filename. The remaining storage is not automatically compacted because this is time-consuming on a DECtape. The storage can be recovered via the PURGE command when desired.

9.3.3.) DDT

DDT provides the user with an aid for debugging his machine language programs. DDT enables a user to exercise all of the control over his program which he would have at the console of a dedicated machine. DDT also provides a number of other useful and powerful features.

DDT is obtained by giving MONITOR the command "DDT". DDT will respond with the message:

```
DDT HERE
?
```

DDT will then await a command.

9.3.3.1) DDT Format

A DDT command is of the form:

@<file>@<range>:<mode><delim><command><delim><arguments>

A DDT command line is of the form:

<command>;<command>;<command>...<command><carriage return>

After a carriage return has been typed DDT will type a line feed and begin to process the command line.

9.3.3.2) FILE

The currently open file is initialized to be the user's

core each time DDT is entered. This is true even if DDT is entered by a breakpoint occurence. A <file> need be given only when the user wishes to open some file other than the currently open one. Before DDT will open another file, it will close the currently open file so that any alterations which have been made get recorded.

Note that a <file> will not normally be included in a command line. If it is not included, then the delimiters surrounding it also need not be included. Thus the command line will start with <range>, if a <range> is being given (see below).

Any logical or physical disk or DECtape, or any logical file on a disk or DECtape can be opened as a <file>. Users without proper validation will be permitted to read any of these, but to write to only their own user core, user disk, or a DECtape which has been allocated to them in response to a previous ON DTn command to MONITOR. Possible files are:

DKn -- physical disk #n

DTn -- physical DECtape #n

TPn -- physical DECtape #n (identical to DTn)

DKn:<filename> -- a logical file on physical disk #n

DTn:<filename> -- a logical file on DECtape #n

TPn:<filename> -- a logical file on DECtape #n

CORE -- user core

DISK -- user disk

SYSTEM -- MTSS system logical disk

V5A:n -- V5A logical disk #n

XCORE -- actual machine core

PREVIOUS -- the <file> open before actual core was opened

9.3.3.3) RANGE

<Range> tells DDT to which addresses in the open file the command refers. For exceptions and default conventions see the specific commands. Operation codes cannot be used in <range> specification. <Range> can be given in three ways:

<Symbolic expression>,<symbolic expression> indicates the lower and upper bounds of the interval

<Symbolic expresion><space><symbolic expression> indicates the lower bound and the length of the interval

<Symbolic expression> alone indicates identical upper and lower bounds to the interval.

9.3.3.4) MODE

<Mode> sets the dump format. <Mode> may be specified

separately for registers, addresses, and contents. Each type will retain its current setting until a new <mode> specification of that type is given. <Mode> is always optional, and more than one type may be given in any command. The modes listed below affect the printing of words from the currently open file. Prefixing one with an 'A' sets the address printing, while prefixing it with an 'R' sets the format by which to print registers. Legal <mode>s are:

O -- octal

A -- ACI6 ASCII (8-bit ASCII - 240)

6 -- trimmed sixbit ASCII

H -- two ASCII characters (bits 0-8 & 9-17)

7 -- two ASCII characters (bits 4-10 & 11-17)

8 -- one ASCII chracter (bits 10-17)

D -- decimal

S -- symbolic

After setting a mode switch, DDT will look for another <mode> (preumably, but not necessarily, for another switch). It will continue to look for more mode settings until it encounters the first command.

NUMBERS

All numbers are integers. Numbers are considered to be octal unless they contain a trailing decimal point, in which

case they are decimal. An octal number may not contain either of the digits 8 or 9.

OPERATORS

All operators are of equal precedence, and are taken strictly from left to right. Legal operators are:

+ addition

- subtraction

* multiplication

/ division

! logical inclusive or

/ logical exclusive or

& logical and

Λ "contents pointed to by this number". This indirection can be stacked as deeply as desired, provided that all addresses generated are legal.

, If followed by 'X<delimiter' or by 'I<delimiter>'. This puts the indirect addressing bit in the number. Otherwise it is taken to be a delimiter rather than an operator.

A blank space between a permanent symbol (op code) and a non-op code is the same as a + of the non-rp code masked to the last 13 bits. Otherwise the space is taken to be a delimiter.

EXPRESSIONS

<value> := <number> or <user-defined symbol>

<op code> := <permanent symbol> or <op code> or

<operator><value>

<expression> := <value> or <op code>

<se> := <symbolic expression> := <expression> or

<exp><opertor><se>

REGISTERS

AC -- accumulator

ACS -- software accumulator switches (kept by MONITOR)

ALL -- all registers

LK -- link

MQ -- multiplier/quotient

PC -- program counter

SC -- step counter

STS -- program interrupt status register

9.3.3.5) COMMANDS

ADS -- not yet implemented

ALT <register><se><register>...<se>

Each <register><symbolic expression> pair causes the

named when the user's next run is started.

BAS

The base address, relative to which all input and output addresses are considered, is set to the low end of the given <range>.

BRE <se> <se> ... <se>,<option #1> <opt #2> ... <opt #n>

Breakpoint is an extremely powerful command. This format yields not only conventional breakpoints, but also a trace and cascading breakpoints. Breakpoint allows the user a method of completely controlling his program, including stopping it wherever he wishes to, examining its progress, and continuing if he wishes to.

A breakpoint will be set for each <se>. All breakpoints set in one command will have the same option list. When a breakpoint is encountered during the execution of the user's program, control is returned to DDT and a breakpoint message is printed telling the user which breakpoint has occurred. At that time the user can give any DDT command he wishes to. If he opens another file it will automatically be closed if he does a transfer or a continue. When the breakpoint message is printed, the user's core is automatically made the open file.

All breakpoint arguments are optional. If there are any <option>s at all, the first of them must be preceded with a comma. The order of the <se>s and of the <option>s is not important, so long as all options come after all <se>s. Legal breakpoint options (and their permissible abbreviations) are:

<register name>

CON (c)

DUM (d)

ERA (e)

FUT (f)

RES (r)

SET (s)

<count>

n<option>

<Register name> -- the named register will be printed when the breakpoint message is printed. All is a legal register name.

CON -- conditions DDT to automatically return to the user program (do a CONTINUE) after first carrying out all other options. This can be used to effect a "tracepoint" or "checkpoint" mode. It will print the specified locations and/or registers every <count> times through the breakpoint.

DUM -- dumps user core within the <range> given in the breakpoint command. Current dump formats are used.

ERA -- automaticaly erases the breakpoint the first time it is encountered.

FUT -- marks the breakpoint in the table (and uses up one of the table entries), but does not set it in the user program. It will be set at some future time by another breakpoint with the SET option enabled.

RES -- restore the breakpoint count. If this option is not employed, the breakpoint will give its message each time through from now on.

SET <se> -- sets the breakpoint indicated by the <se> This is useful for use with overlays, or checking common subroutines when enterd after some specific other event.

<Count> -- allows the breakpoint to be reached <count> times before any action except the counting is taken. <Count> is a <symbolic expression>.

N<option> -- unsets the specified option.

CLO -- closes the curently open file.

CON -- continues executing the user's program at the current value of his registers. If DDT was entered on a breakpoint, and if the user has not altered register PC the breakpointed instruction will be executed.

CRS -- is not yet implemented.

DSM -- is not yet implemented.

DUM -- dumps from the currently open file, according to the current formats, the words specified by the range. Default range is the next location.

EXI -- exits to MONITOR after first closing the currently open file.

EXP -- explains about DDT. Lists legal DDT commands, modes, etc

JMP <se> commences to execute the user's program at the address <se>. The current values of all of his registers are used.

LDS -- not yet implemented

LIM -- sets the limit on the maximum offset allowed from

a symbol table value in a symbolic printing. If the offset is greater than the limit, the octal value is printed instead of <symbol>+<offset>.

LIS <option> -- lists the value of the option:
LIM -- lists the value of the limit
MAS -- lists the value of the search mask
BRE -- lists the locations of the set and future breakpoints.

LOA -- loads the user's core from the user disk. This presupposes a previous SAVE command.

MAS <se> -- sets the search mask to the <se> value

NON -- null command to suppress the otherwise implicit DUMP.

PAT <se><se>...<se>,<se><se>...<se>
PATCH replaces the contents of <range> locations in the open file with the first <se>, the next location with <se#2>, etc. until reaching the comma. The first <se> after the comma is the next location to patch. The next <se> is put there, the following one in the next location, etc. As many commas as desired can be used in one command.

PRE -- is only meaningful with actual core open. Then

it opens the file open prior to opening actual core.

REG <register name><register name>...<register name>. REGISTER is a useless command, since omitting it and just typing the register names themselves has the same effect: the register value is printed.

REP -- is part of a binary editor. It will replace one binary quantity by another. NOT yet implemented.

SAV -- copies user core into the user disk. Useful for debugging. To save into a catalogued file, LOADER must be used.

SEA -- searches the specified <range> for all words for which the <se>&<mask> is identical with <contents>&<mask>. For each location for which this is true, the location and contents are printed out in the current <mode>s.

UNB <se><se>...<se> removes the breakpoint at each <se>. An error message occurs if there is none.

VAL -- requests underprinting to protect a password. If the password is approved, the user is validated. Identical to the MONITOR command of the same name.

9.3.4) CATALOG

CATALOG is a Phantom program to list on the Teletype the catalog information about either selected file(s) from a disk or DECtape or about all of the files on the device. The information can be listed either with or without a header and trailer.

9.4) System-Supplied User programs

System-Supplied User programs (S-user programs) are those programs of which the Executive will give the user a copy. This means that S-user programs are overlaid onto user core, and the user can alter them at will. S-user programs save the user from having to write many of his own service routines (e.g. a Teletype handler would be supplied as an S-user program) or they may be used when the user does not need his own core anyway (e.g. the BASIC Interpreter).

Available S-user programs are BASIC and DEBUGGER.

DEBUGGER is the old GROWTH system debugger re-assembled for location 2000 and running as an S-user program. For complete documentation on it either list the DTSS file PDP9LIB***:DDTDOC or else refer to the copy of this file kept in the PDP-9 machine room.

9.4.1) BASIC Interpreter

BASIC Interpreter was written by Ron Harris '71 as a quick course project. It is an illustration of the point that MTSS is capable of running arbitrary machine language programs, including new system programs to give the system improved capabilities, such as higher languages. The interpreter should not, therefore, be judged on the subset of BASIC it recognizes.

BASIC interpreter is running unchanged under MTSS, except that it is now assembled to start at location 2000 to stay out of trouble with memory protect.

A negative accumulator switches value restarts it, so it will not run unless either the user's software ACS register is positive or else he has ON ACS and the hardware switches are positive.

BASIC interpreter recognizes the following BASIC statements: IF, GOTO, PRINT, LET, and END. It recognizes the following commands: RUN, LIST, and EXIT.

9.5) General Discussion

It is rarely clear which system services should be provided by which class of program. For maximum speed one would like to have all Executive routines, or even all system services core resident. Either of these costs a lot of core space -- a strong consideration in an 8K PDP-9. Overlays are fast, provided the proper overlay is in core at the proper time. But the more overlays a system uses, the more likely it is not to have the right one in core at the right time.

Overlays are restricted in length to 700(8) words. For some system services that handicap overcomes any speed advantage. Such services are put into Phantom programs. The length of a Phantom program can be up to 1600(8) words of unrestricted code plus the balance of 7K words of pure code. MONITOR, for example, suffers little or no penalty for being a Phantom program. In fact, its normal access to Executive services is an asset. MONITOR does not often need to refer to user core, and then only to overlay an S-user program or to swap a user in and start him running. These things are provided for MONITOR as Executive services.

DDT is an example of a system program which could logically be run either as a Phantom program or as an S-user program. There are advantages and disadvantages each way. If

DDT were a Phantom program it would put more pressure on system resources and it would run somewhat slower because each reference to the user's core would require a disk operation. DDT could let the user run full sized programs, though, and it could be granted full Phantom program privileges. As an S-user program DDT cannot be granted full privileges or a user could take over the system. Also it does take up a sizeable amount of user core, especially if the user loads his symbol table.

Scheduling

A user is considered to be ready to run if he is not I/O roadblocked. To find the next user ready to run, users are checked in a circular manner starting with the first user after the current one. The first one who is found who is not I/O roadblocked is the next user ready to run. If all users other than the current user are roadblocked, then the current user is also the next user ready to run.

When a user is interrupted by the clock the next user ready to run is swapped into core and started. If a user becomes I/O roadblocked his job is suspended and the next user who is ready to run is started. No swap is made unless necessary.

FILE STRUCTURE

MTSS and GROWTH share a common catalog and file structure. Block 1 of the MTSS/GROWTH logical disk, or block 1 of a physical DECtape contains the catalog for that device. (For a detailed description of catalog format consult the documentation with the standard catalog handler in the initialization program assembly listing.)

As a result, files created under GROWTH are completely accessable under MTSS and files created under MTSS are completely accessable under GROWTH.

Loadstring binary format files cannot be created under MTSS. They can be created in DTSS on paper tape from 9MAP assembler output. Under GROWTH they can be created either on paper tape or on DECtape from 9MAP assembler output. MTSS can read these files.

MTSS files are not compatable with the V5A operating system.

CORE STRUCTURE

The area between locations 0-1777 is the same no matter which type of program may be in upper core. Locations 0-777 contain the Resident Executive (see section 9.1) and locations 1000-1677 contain whichever Executive overlay is currently in core (see section 9.2).

Locations 1700-1777 contain the user job table. This job table is the same one whether a system-supplied program or a user-supplied program is running. Phantom programs will store internally anything they may alter so that they can restore it before returning to the user.

The memory protection boundary is set at location 2000. The contents of the area between there and the end of core depends on what type of program is running. If it is a user-type program, then this area will contain all user-supplied and system-supplied user programs which the user has called since the last time he cleared core. They will be overlaid on top of each other in the order he called them.

MTSS CORE MAP
USER OR S-USER PROGRAM

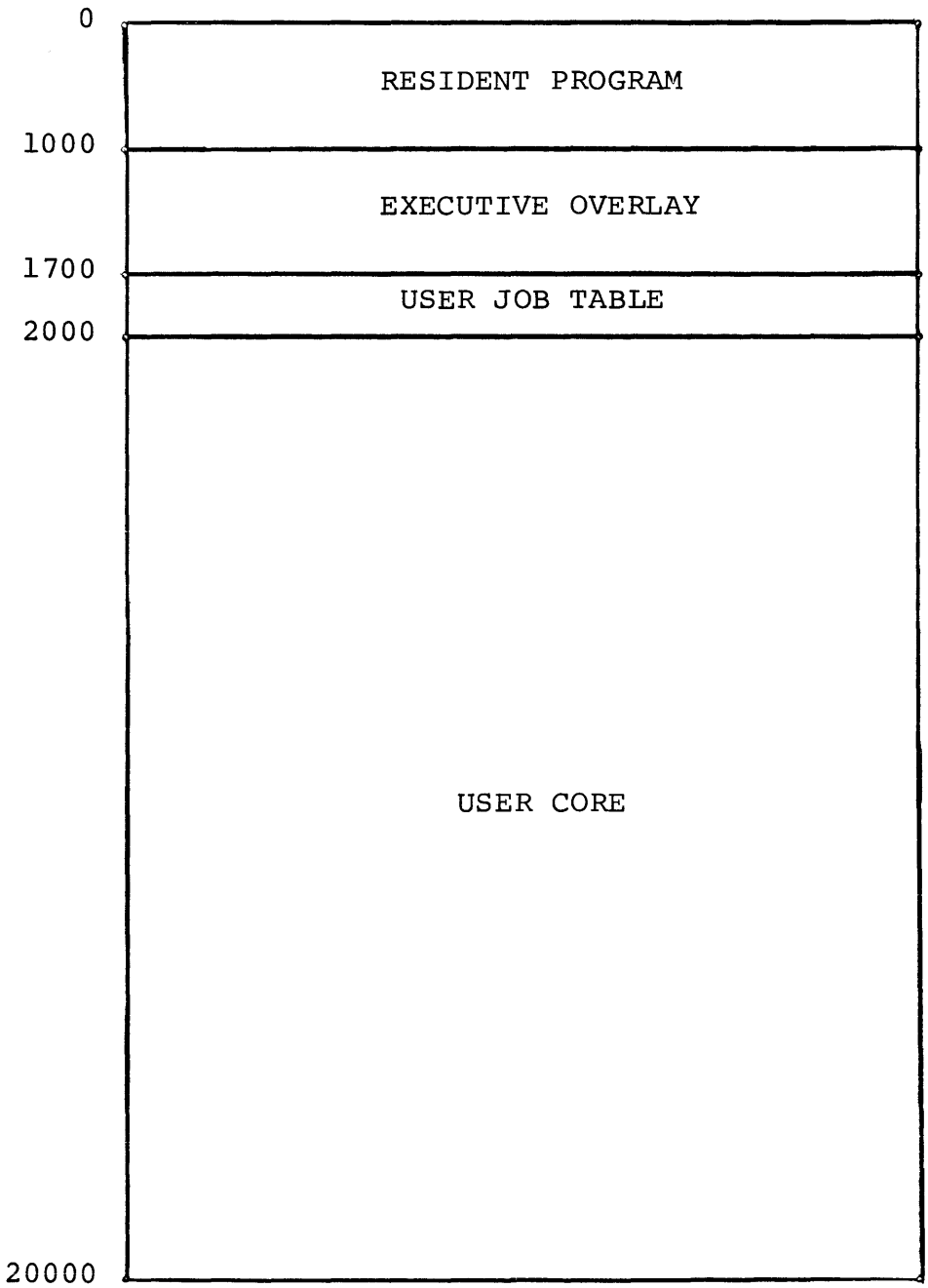


figure 12.1

If this area contains a Phantom program, it will be divided as shown in figure 12.2. Common is below all Phantom programs. As a result, it is capable of holding certain user data (e.g. DDT dump format switch setting or data saved to allow Phantom programs to remain invisible) regardless of which programs may get called.

Non-pure code is the area from the standard Phantom program starting location through 3577. Each Phantom program must use this area for any non-pure locations. (N.b. on the PDP-9 all subroutine entrances are non-pure.) The remainder of core contains the rest of the Phantom program. This must be entirely in pure code.

When a Phantom program is swapped out, only the common and non-pure code areas are copied into the Phantom core image. Pure code is ignored. The user job table is copied to the user job table image. When the Phantom program is swapped in, the user job table, common, and non-pure code areas are all copied in. The pure code area is copied from the system's program if and only if it was not already in core for the previous user.

MTSS CORE MAP
PHANTOM PROGRAM

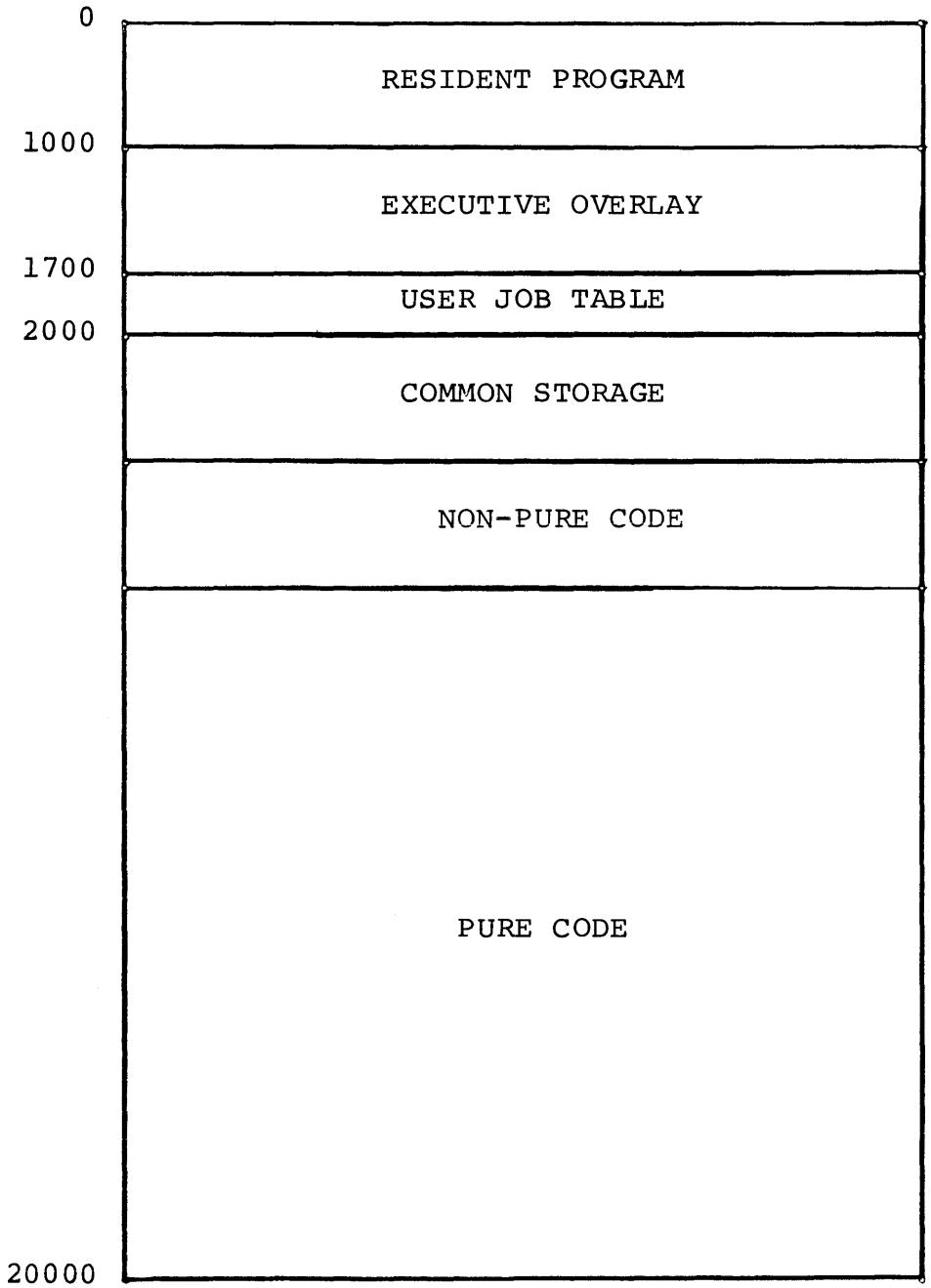


figure 12.2

MTSS does not have exclusive use of the disk. It was felt that the most convenient thing for all users of the PDP-9 would be to minimize the number of times anyone would have to copy the disk onto two DECtapes before using it and then copy it back when done. To do this the DEC V5A system has been allowed to retain the first half of the disk. The V5A user file catalog has been altered to make the V5A system think that the MTSS/GROWTH and scratch areas of the disk are already full.

MTSS and GROWTH share a common catalog in the second block of their area of the disk and share the remainder of their area for system files. It is currently possible for users to also save files in this area; but this is discouraged for lack of room. The V5A system usage is small and declining, so it is hoped that before long the V5A user file area can be further cut back, and the extra space given to MTSS/GROWTH. Then it will be practical to accomodate a limited number of user files in this area.

The last 48K of the disk is permanently reserved as a scratch area. This permits MTSS to have a swap area, and individual users of the dedicated machine to have a reasonable amount of physical disk space without worrying about destroying anyone's files.

MAJOR DIVISIONS OF THE DISK

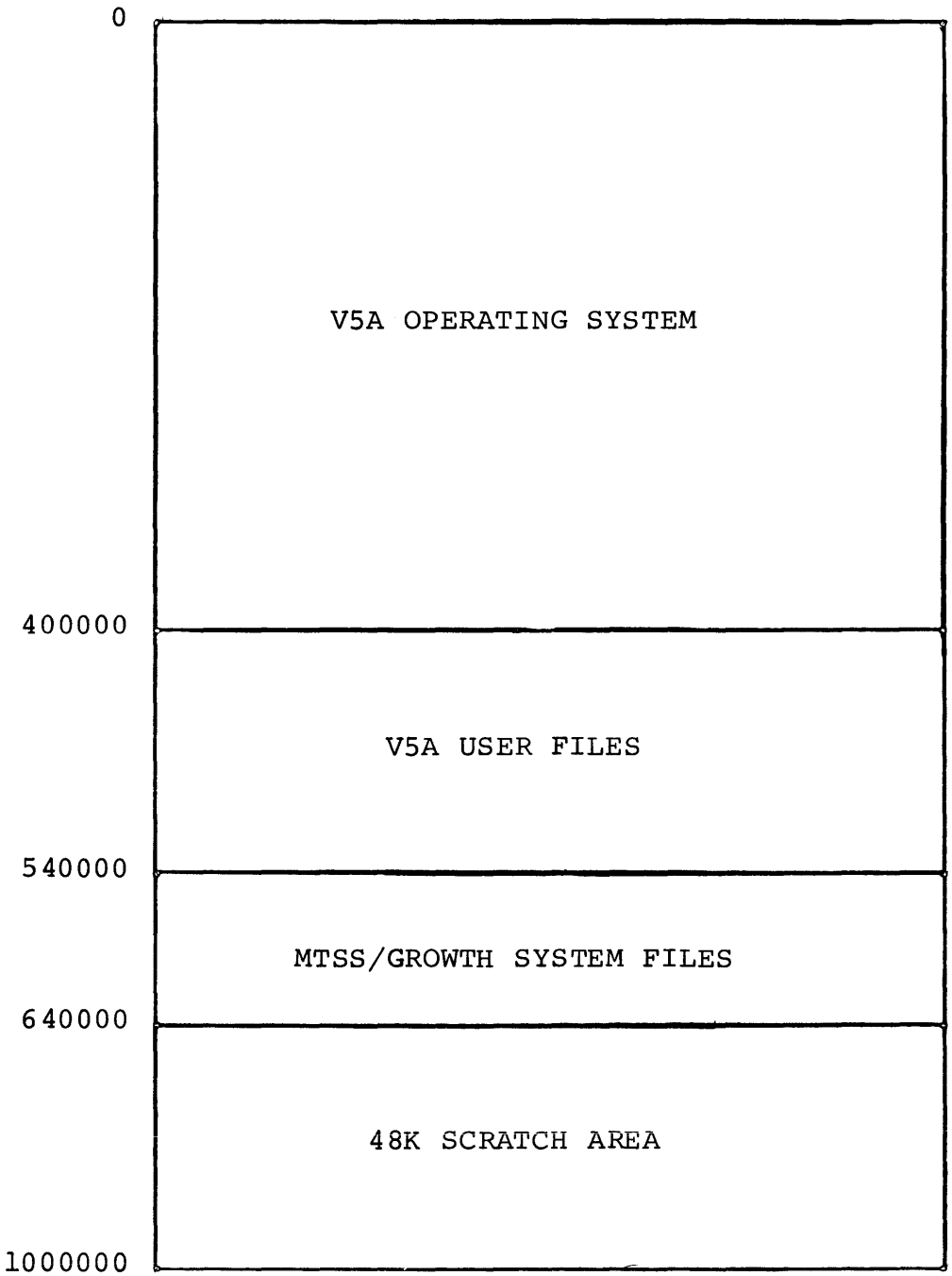


figure 13.1

16K of scratch area is allocated to each Teletype, divided as shown in figure 13.2. The user job table occupies 100 words, the Phantom core image occupies 1700(8) words and the "user physical disk" occupies 20000(8) words. If more Teletypes were added to the system, the scratch area would have to be expanded.

TYPICAL USER'S DISK AREA

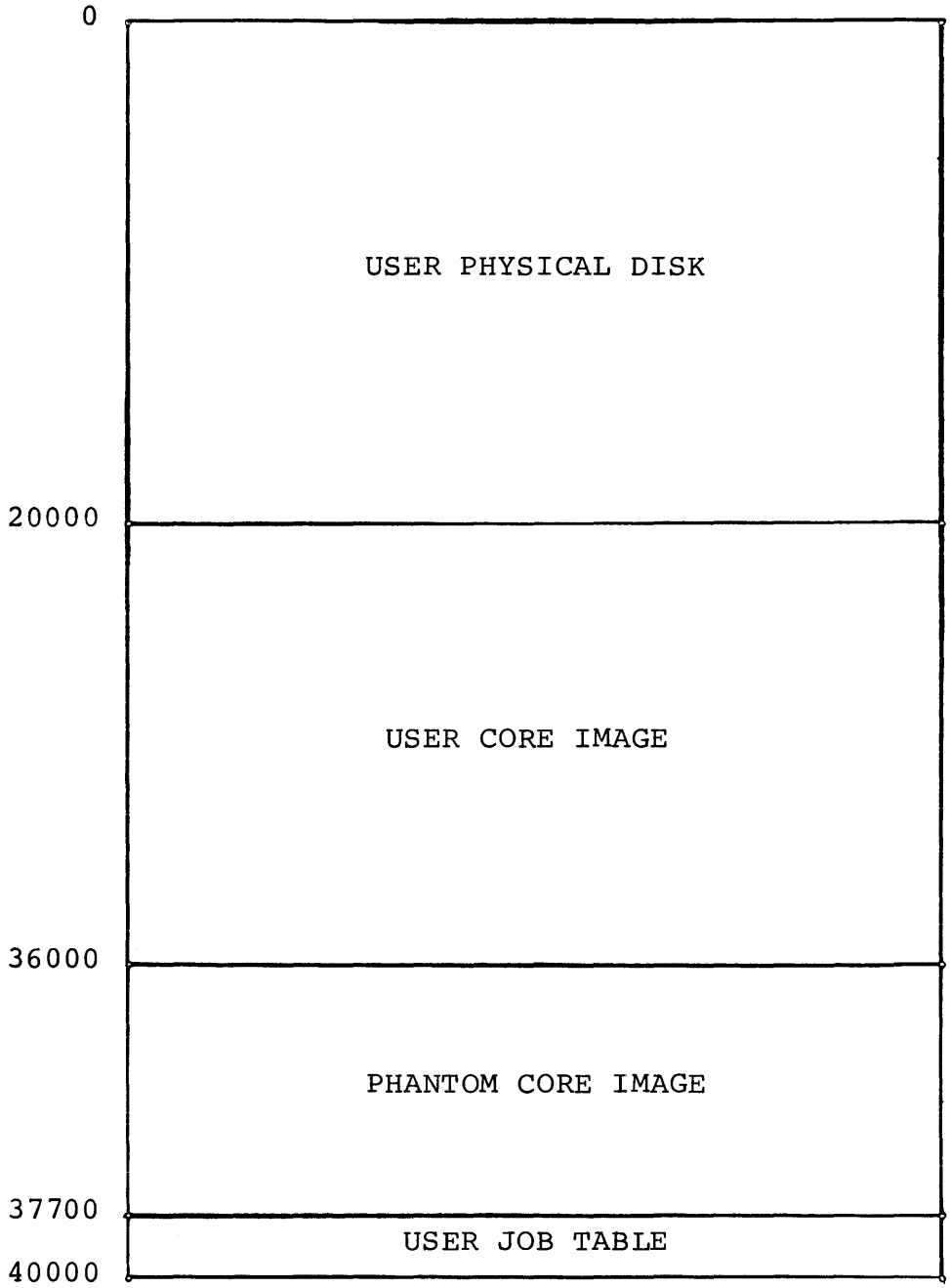


figure 13.2

MAJOR DIVISIONS OF THE
MTSS/GROWTH PORTION OF THE DISK

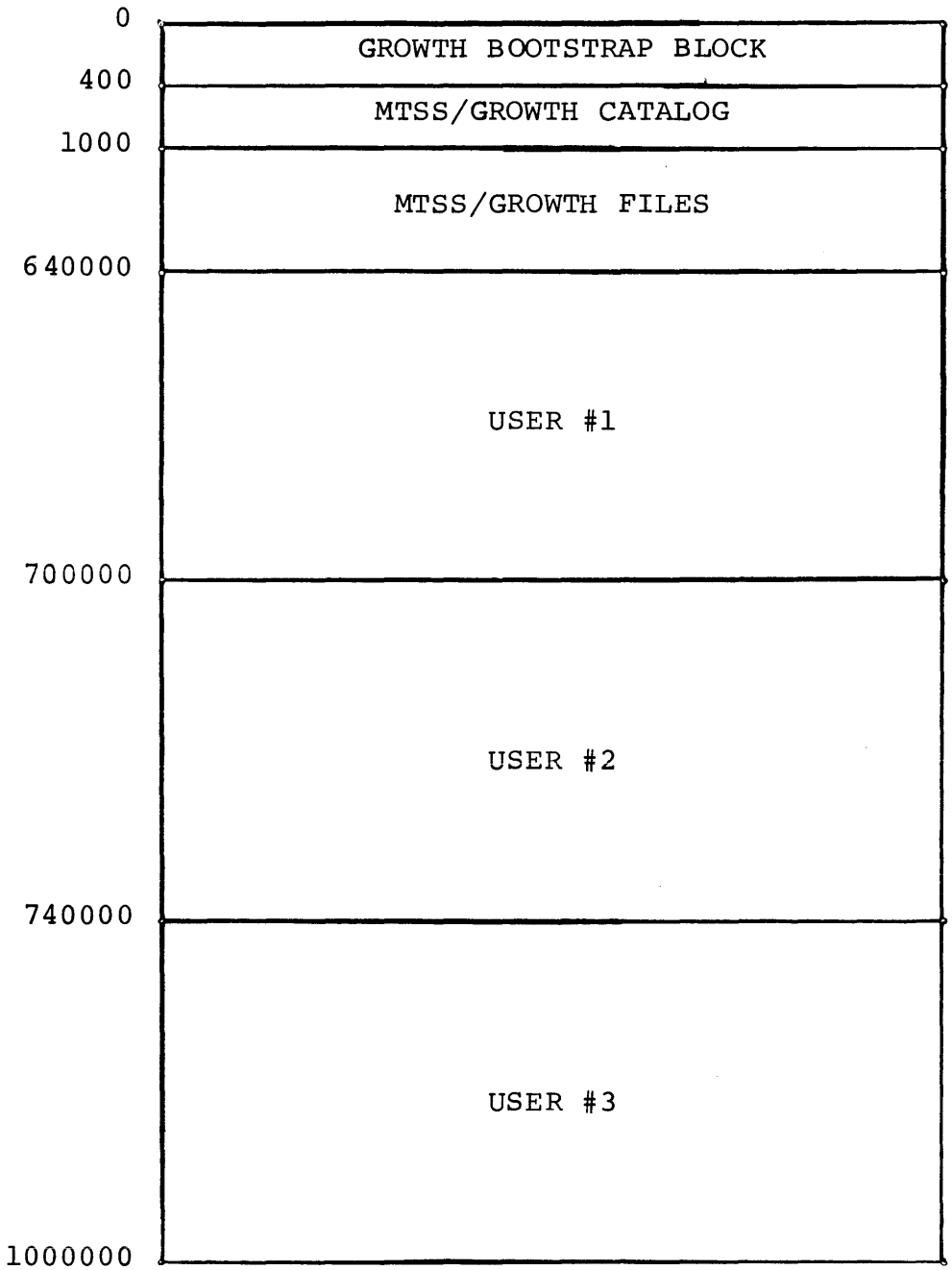
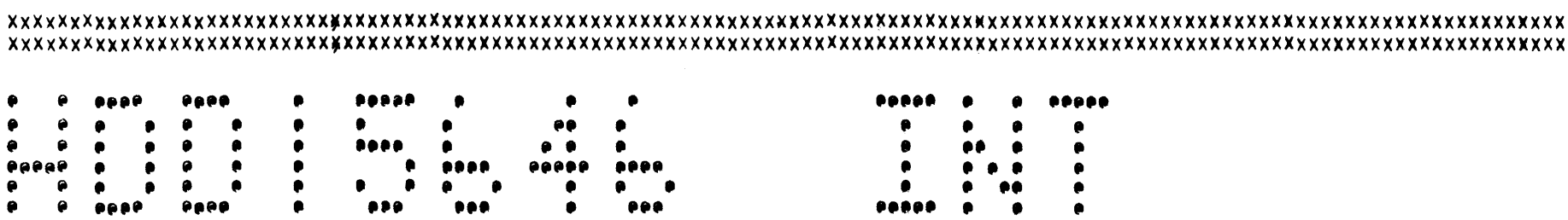


figure 13.3



05/31/72 01:55:19

```
**********************************************
**********************************************
**                                          **
**        PDP-9 MINI TIME-SHARING SYSTEM    **
**        SYSTEM INITIALIZATION PROGRAM     **
**                  DTSSI INT               **
**                  MTSSI INT               **
**                                          **
**********************************************
**********************************************
```

```
100             .TITLE  PDP-9 MINI TIME-SHARING SYSTEM INITIALIZATION PROGRAM
110             .NAME   INT--INT
120     *
130     *       **********PROGRAMS USED IN MTSS**********
140     *
150     *       INITIALIZATION PROGRAM
160     *       DARTMOUTH TIME-SHARING SYSTEM (SOURCE FILE) NAME: INT
170     *       MINI TIME-SHARING SYSTEM (NON-RELOCATABLE BINARY FILE) NAME: INT
180     *
190     *       EXECUTIVE -- RESIDENT PROGRAM
200     *       DTSS NAME: RES
210     *       MTSS NAME: B01
220     *
230     *       EXECUTIVE -- SWAPPER OVERLAY
240     *       DTSS NAME: SWP
250     *       MTSS NAME: B02
260     *
270     *       EXECUTIVE -- MEMORY PROTECTION OVERLAY #1
280     *       DTSS NAME: MP1
290     *       MTSS NAME: B03
300     *
310     *       EXECUTIVE -- MEMORY PROTECTION OVERLAY #2
320     *       DTSS NAME: MP2
330     *       MTSS NAME: B04
340     *
350     *       EXECUTIVE -- SPECIAL IOT (EXECUTIVE CALL) HANDLER #1 OVERLAY
360     *       DTSS NAME: SPL
370     *       MTSS NAME: B12
380     *
390     *       PHANTOM PROGRAM -- SYSTEM MONITOR AND MESSAGE OUTPUT
400     *       DTSS NAME: MTR
410     *       MTSS NAME: B05
420     *
430     *       PHANTOM PROGRAM -- SYSTEM LOADER PROGRAM
440     *       DTSS NAME: LDR
450     *       MTSS NAME: B06
460     *
470     *       $-USER PROGRAM -- DEBUGGER
480     *       DTSS NAME: DDT
490     *       MTSS NAME: B07
500     *
510     *       $-USER PROGRAM -- BASIC INTERPRETER
520     *       DTSS NAME: BAS
530     *       MTSS NAME: B08
540     *
550     *       USER PROGRAM -- PHYSICAL TELETYPE HANDLER SUBROUTINES PACKAGE
560     *       DTSS NAME: LIBTTY
570     *       MTSS NAME: B10
580     *
590     *       USER PROGRAM -- GROWTH CATALOG HANDLING SUBROUTINES PACKAGE
600     *       DTSS NAME: GROCAT
610     *       MTSS NAME B11
```

```
                .EJECT
*
*       CORE LAYOUT FOR THE INITIALIZATION PROGRAM
*
*       ********************************************************************
*       *                                                                  *
*       * LOCATIONS BUF TO BUF+4K ARE USED AS A BUFFER FOR COPYING FILES  *
*       * FROM THE LIBRARY DECTAPE TO THE SYSTEM DISK.                     *
*       *                                                                  *
*       ********************************************************************
*       *                                                                  *
*       * LOCATIONS BASE-LCATL ARE USED FOR THE MAIN INITIALIZATION PROGRAM *
*       *                                                                  *
*       ********************************************************************
*       *                                                                  *
*       * LOCATIONS LCATL-TSWORDB ARE USED FOR MISCELLANEOUS INITIALIZATION *
*       * PROGRAM SUBROUTINES.                                             *
*       *                                                                  *
*       ********************************************************************
*       *                                                                  *
*       * LOCATIONS TSWORDB-CSCTEM1 ARE USED FOR THE MTSS STANDARD TELETYPE *
*       * HANDLER.                                                         *
*       *                                                                  *
*       ********************************************************************
*       *                                                                  *
*       * LOCATIONS CSCTEM1-ISOFILES ARE USED FOR THE GROWTH SYSTEM         *
*       * STANDARD CATALOG ROUTINES.                                       *
*       *                                                                  *
*       ********************************************************************
*       *                                                                  *
*       * LOCATIONS ISOFILES-ISLCAT ARE USED FOR A LIST OF MTSS LIBRARY    *
*       * FILES, ASSORTED CONSTANTS, AND AN INITIALIZATION PROGRAM PHYSICAL *
*       * DISK CATALOG.                                                    *
*       ********************************************************************
*       *                                                                  *
*       * LOCATIONS LCAT-LCAT+377 ARE USED TO HOLD A COPY OF THE LIBRARY   *
*       * DECTAPE CATALOG.                                                 *
*       *                                                                  *
*       ********************************************************************
*       *                                                                  *
*       * LOCATIONS SCATLOG-SCATLOG+377 ARE USED FOR THE STANDARD GROWTH   *
*       * CATALOG, SINCE THIS INITIALIZATION PROGRAM IS LOADED FROM THE    *
*       * MTSS LIBRARY DECTAPE BY THE GROWTH SYSTEM MONITOR, AT START-UP   *
*       * TIME THIS CATALOG WILL BE A CATALOG OF THE MTSS LIBRARY DECTAPE.  *
*       *                                                                  *
*       ********************************************************************
```

```
1080            .EJECT
1090     *
1100     *      THE INITIALIZATION PROGRAM WILL INITIALIZE MTSS FOR ANY NUMBER
1110     *      OF FILES OF ANY TYPE, AND FOR ANY NUMBER OF TELETYPES/USERS.
1120     *
1130     *      THE SEQUENCE OF INITIALIZATION ACTIONS IS:
1140     *
1150     *           1 -- INITIALIZE THE HARDWARE
1160     *
1170     *           2 -- INITIALIZE THE SOFTWARE FLAGS
1180     *
1190     *           3 -- COPY THE CURRENT GROWTH CATALOG (IT SHOULD BE THE MTSS
1200     *                LIBRARY DECTAPE CATALOG) INTO LCAT, SINCE IT IS REPEATEDLY
1210     *                ACCESSED BY THIS PROGRAM.
1220     *
1230     *           4 -- COPY THE CATALOG FROM THE SYSTEM DISK INTO THE STANDARD
1240     *                GROWTH CATALOG BLOCK SO THE STANDARD CATALOG ROUTINES
1250     *                WILL WORK WITH IT.
1260     *
1270     *           5 -- UNSAVE FROM THE SYSTEM DISK ALL FILES ON THE LIST OF SYSTEM
1280     *                FILES WHICH BEGINS AT OFILES. DO NOT WORRY ABOUT ANY WHICH ARE
1290     *                NOT SAVED TO BEGIN WITH.
1300     *
1310     *                NOTE THAT OVERLAY FILES, USER-TYPE SYSTEM FILES, AND
1320     *                PHANTOM-TYPE SYSTEM FILES ARE UNSAVED IN DISCRETE OPERATIONS.
1330     *                THIS IS SO THAT IF IT BECOMES DESIRABLE TO HANDLE THEM IN
1340     *                DIFFERING MANNERS IN THE FUTURE, IT CAN BE EASILY DONE.
1350     *
1360     *           6 -- PURGE THE DISK TO COMPACT THE RESULTING CATALOG AND THE
1370     *                RESULTING DISK SAVED STORAGE.
1380     *
1390     *           7 -- COPY ALL FILES ON THE LIST OF SYSTEM FILES WHICH
1400     *                BEGINS AT OFILES FROM THE MTSS LIBRARY DECTAPE TO THE SYSTEM
1410     *                DISK.
1420     *
1430     *                NOTE THAT OVERLAY FILES, USER-TYPE SYSTEM FILES, AND
1440     *                PHANTOM-TYPE SYSTEM FILES ARE COPIED IN DISCRETE OPERATIONS.
1450     *                THIS IS SO THAT IF IT BECOMES DESIRABLE TO HANDLE THEM IN
1460     *                DIFFERING MANNERS IN THE FUTURE, IT CAN EASILY BE DONE.
1470     *
1480     *           8 -- THE ENTRIES IN THE SYSTEM DISK CATALOG REFER TO LOGICAL
1490     *                DISK BLOCK NUMBERS RELATIVE TO THE START OF THE GROWTH
1500     *                SYSTEM ON THE SYSTEM DISK, AND TO ACTUAL CORE ADDRESSES.
1510     *                COPY INTO INTCAT THE ENTRIES FOR ALL SWAPPABLE SYSTEM
1520     *                FILES (THOSE LISTED IN THE LIST BEGINNING AT OFILES)
1530     *                CHANGING LOGICAL DISK ADDRESSES TO PHYSICAL DISK ADDRESSES
1540     *                AND ACTUAL CORE ADDRESSES TO CORE ADDRESSES -1. ALSO CREATE
1550     *                THE ENTRIES FOR THE PURE CODE PORTION OF PHANTOM PROGRAMS.
1560     *
1570     *                NOTE THAT OVERLAY FILES, USER-TYPE SYSTEM FILES, AND
1580     *                PHANTOM-TYPE SYSTEM FILES ARE COPIED IN DISCRETE OPERATIONS.
1590     *                THIS IS BECAUSE THEY ARE NOT HANDLED IDENTICALLY.
```

```
1600    *               FOR PHANTOM PROGRAMS AN ENTRY MUST BE MADE FOR THE
1610    *               PURE CODE PORTION
1620    *
1630    *            9 -- CERTAIN OVERLAYS HAVE AN INTERNAL CATALOG THEY DEPEND
1640    *                 ON (E.G, SWAPPER, OR EACH MEMORY PROTECTION OVERLAY CALLS
1650    *                 THE NEXT ONE DIRECTLY), NOW THESE OVERLAYS ARE READ INTO
1660    *                 THE OVERLAY AREA, ONE AT A TIME, AND EACH HAS HIS CATALOG
1670    *                 INITIALIZED FOR HIM. THEN THE CORRECTED COPY IS READ BACK
1680    *                 OUT ONTO THE SYSTEM DISK.
1690    *
1700    *           10 -- THE RESIDENT PROGRAM IS READ INTO RESIDENT CORE AND ITS
1710    *                 RESIDENT CATALOG (WHOSE ONLY ENTRY IS THE SWAPPER POINTERS)
1720    *                 IS SET UP ACCORDING TO THE SYSTEM DISK CATALOG.
1730    *
1740    *           11 -- USER AND PHANTOM JOB TABLES ARE INITIALIZED TO INSURE
1750    *                  THEY DON'T CONTAIN EITHER GARBAGE OR RANDOM PERMISSIONS.
1760    *
1770    *           12 -- WITH INITIALIZATION COMPLETE, A MESSAGE IS PRINTED ON THE
1780    *                 CONSOLE TELETYPE, FINAL HARDWARE TIDYING UP IS DONE, AND
1790    *                 THE MONITOR IS CALLED FOR THE CONSOLE TELETYPE.
```

```
1800            .EJECT
1810
1820    *
1830    *
1840    *       TO INSERT A NEW PROGRAM INTO MTSS SAVE IT ON THE MTSS LIBRARY DECTAPE AND:
1850    *
1860    *            1 -- OVERLAY PROGRAMS: ADD ITS NAME TO THE LIST OF OVERLAY
1870    *                 PROGRAMS (OFILES). IN THE OFILES LIST THE MEMORY PROTECTION
1880    *                 OVERLAYS MUST BE LISTED CONSECUTIVELY. OTHER THAN THAT,
1890    *                 ORDER IS IMMATERIAL.
1900    *
1910    *            2 -- USER-TYPE SYSTEM PROGRAMS: ADD ITS NAME TO THE LIST OF
1920    *                 USER-TYPE SYSTEM PROGRAMS (UFILES). ORDER IS IMMATERIAL.
1930    *
1940    *            3 -- PHANTOM-TYPE SYSTEM PROGRAMS: ADD ITS NAME TO THE LIST OF
1950    *                 PHANTOM-TYPE USER PROGRAMS (PFILES). ORDER IS IMMATERIAL.
1960    *
1970    *       NOTE THAT ALL PROGRAM NAMES MUST ALREADY BE DEFINED IN THE DEFINS
1980    *       PROGRAM. THAT IS TO MAKE THEM AVAILABLE TO ALL MTSS PROGRAMS.
```

```
                                        DEFINITIONS LOCAL TO THE INITIALIZATION PROGRAM

                    1990                ,STITL  DEFINITIONS LOCAL TO THE INITIALIZATION PROGRAM
                    2000
000001              2010    DEFINS      ,EQU    1               TURN THE LISTING ON FOR THE DEFINITIONS INSERT
000001              2020    DEBUG       ,EQU    1               TURN THE LISTING ON FOR ALL OTHER INSERTS
012000              2030    NEXTL       ,EQU    I$START         RESTART ADDRESS FROM DISK/DECTAPE HARDWARE ERROR
012000              2040    FORMAT      ,EQU    I$START         AVOIDS ERROR FLAGS FROM THE GROWTH INSERTS
                    2050                ,HEAD   I
                    2060                ,PMC    ON              PRINT ALL MACRO CODE
                    2090                ,INSRT  DEFINS
```

MTSS SYSTEM DEFINITIONS

```
                    140             .HEAD                   MAKE SURE NO HEAD SYMBOL IS ON
                    150     *       LIBPW = PLBPROGS
                    160     *       DOCPW = BLN
                    170     *       RSTPW = CRC
                    180     *         CPW = UVLQ
                    190     *
                    200     *
                    210     *       THE FOLLOWING OPDEF STATEMENTS ARE FOR CONVENIENCE ONLY
                    220     *
                    230     INX     .OPDEF  ISZ             USED WHEN THE INCREMENT SHOULD NEVER SKIP
                    240     RET     .OPDEF  JMP             USED FOR SUBROUTINE EXITS
                    250     *
                    260     *       MTSS PROGRAMS ARE ASSIGNED SERIAL NAMES INSTEAD OF MORE MNEMONIC
                    270     *       NAMES TO MINIMIZE CONFUSION WITH OTHER USERS' PROGRAMS STORED ON THE
                    280     *       SYSTEM DISK UNDER THE GROWTH SYSTEM.
                    290     *
                    300     *       DEFINE THE MTSS SYSTEM PROGRAM NAMES
                    310     *
422020              320     INT     .EQU    422020          SYSTEM NAME IS B00
422021              330     RES     .EQU    422021          B01
422022              340     SWP     .EQU    422022          B02
422023              350     MP1     .EQU    422023          B03
422024              360     MP2     .EQU    422024          B04
422025              370     MTR     .EQU    422025          PHANTOM (ENTIRE CODE) NAME IS B05
602025              380     PMTR    .EQU    602025          PHANTOM (PURE CODE) NAME IS P05
422026              390     LDR     .EQU    422026          B06
602026              400     PLDR    .EQU    602026          P06
422027              410     DDT     .EQU    422027          B07
422030              420     BAS     .EQU    422028          B08
422122              430     SPL     .EQU    422122          B12
                    440     *
                    450     *       MTSS MUST BE ASSEMBLED FOR A SPECIFIC MAXIMUM NUMBER OF USERS IN
                    460     *       ORDER TO ALLOCATE INTERNAL STORAGE AND DISK STORAGE CORRECTLY.
                    470     *
                    480     *       HARDWARE DEVICE NAMES
                    490     *
606462              500     PTR     .EQU    606462          ACI6 PTR
606460              510     PTP     .EQU    606460          ACI6 PTP
606064              520     PPT     .EQU    606064          ACI6 PPT
445320              530     DK0     .EQU    445320          ACI6 DK0
646000              540     TP.     .EQU    646000          ACI6 *TP *
00646₀              550     .TP     .EQU    00646₀          ACI6 * TP*
446400              560     DT.     .EQU    446400          ACI6 *DT *
004464              570     .DT     .EQU    004464          ACI6 * DT*
```

MTSS SYSTEM DEFINITIONS

```
                    600     *
                    610     *
                    620     *
                    630     *            PDP-9 MINI TIME-SHARING SYSTEM CORE LAYOUT
                    640     *
                    650     *       0         ************************************
                    660     *                 *                                  *
                    670     *                 *  EXECUTIVE -- RESIDENT PROGRAM   *  RESLEN
                    680     *                 *                                  *
                    690     *       OVSTRT    ************************************
                    700     *                 *                                  *
                    710     *                 *  EXECUTIVE -- OVERLAY AREA       *  OVLEN
                    720     *                 *                                  *
                    730     *       JTSTRT    ************************************
                    740     *                 *                                  *
                    750     *                 *  USER JOB TABLE                  *  JTLEN
                    760     *                 *                                  *
                    770     *       BOUNDARY  ************************************
                    780     *                 *                                  *
                    790     *                 *                                  *
                    800     *                 *                                  *
                    810     *                 *  USER PROGRAM AREA               *  USLEN
                    820     *                 *                                  *
                    830     *                 *                                  *
                    840     *                 *                                  *
                    850     *       CORMAX    ************************************
                    860     *
                    870     *
                    880     *
                    890     *       MTSS CORE LAYOUT DEFINITIONS
                    900     *
016000              910     CORMAX  ,EQU    8K
001000              920     RESLEN  ,EQU    OVSTRT
001000              930     OVSTRT  ,EQU    1000
000700              940     OVLEN   ,EQU    JTSTRT-OVSTRT
001700              950     JTSTRT  ,EQU    1700
000100              960     JTLEN   ,EQU    BOUNDARY-JTSTRT
002000              970     BOUNDARY,EQU    2000
014000              980     USLEN   ,EQU    CORMAX-BOUNDARY
001700              990     IMPLEN  ,EQU    PURSTRT-BOUNDARY
003700              1000    PURSTR  ,EQU    JTSTRT+BOUNDARY
012100              1010    PURLEN  ,EQU    8K-PURSTRT
016000              1020    8K      ,EQU    16000           FOR DEBUGGING PURPOSES ONLY --- WILL BE LENGTHENED TO 20000
014000              1030    7K      ,EQU    14000           FOR DEBUGGING PURPOSES ONLY -- WILL BE LENGTHEDED TO 16000
```

MTSS SYSTEM DEFINITIONS

```
1050
1060
1070
1080    *                    PHANTOM CORE LAYOUT
1090    *
1100    *       BOUNDARY ********************************
1110    *                *                              *
1120    *                *  TEMPORARY VARIABLES         *
1130    *                *                              *
1140    *       USTORE   ********************************
1150    *                *                              *
1160    *                *  USER REGISTER STORAGE       *
1170    *                *                              *
1180    *       PHSTOR   ********************************
1190    *                *                              *
1200    *                *  PHANTOM REGISTER STORAGE    *
1210    *                *                              *
1220    *       DBSTOR   ********************************
1230    *                *                              *
1240    *                *  DDT STORAGE                 *
1250    *                *                              *
1260    *       COMSTOR  ********************************
1270    *                *                              *
1280    *                *  COMMON PHANTOM STORAGE      *
1290    *                *                              *
1300    *       BCNTRL   ********************************
1310    *                *                              *
1320    *                * FILE BUFFER CONTROL          *
1330    *                *                              *
1340    *       BUFFER   ********************************
1350    *                *                              *
1360    *                *                              *
1370    *                *  CORE BUFFER                 *
1380    *                *                              *
1390    *                *                              *
1400    *       IMPSTRT  ********************************
1410    *                *                              *
1420    *                *                              *
1430    *                *  IMPURE PHANTOM CODE         *
1440    *                *                              *
1450    *                *                              *
1460    *       PURSTRT  ********************************
1470    *                *                              *
1480    *                *                              *
1490    *                *                              *
1500    *                *  PURE PHANTOM CODE           *
1510    *                *                              *
1520    *                *                              *
1530    *                *                              *
1540    *       CORMAX   ********************************
1550    *
1560    *
```

MTSS SYSTEM DEFINITIONS

```
                    1570    *       PHANTOM CORE LAYOUT DEFINITIONS
                    1580    *
                    1590    *
                    1600    *       TEMPORARY VARIABLES
                    1610    *
                    1620            .HEAD   0,M,C,T,D
002000              1630    TEMP0   .EQU    BOUNDARY
002001              1640    TEMP1   .EQU    TEMP0+1
002002              1650    TEMP2   .EQU    TEMP1+1
002003              1660    TEMP3   .EQU    TEMP2+1
002004              1670    TEMP4   .EQU    TEMP3+1
002005              1680    TEMP5   .EQU    TEMP4+1
002006              1690    TEMP6   .EQU    TEMP5+1
002007              1700    TEMP7   .EQU    TEMP6+1
002010              1710    TEMP8   .EQU    TEMP7+1
002011              1720    TEMP9   .EQU    TEMP8+1
002012              1730    TEMP10  .EQU    TEMP9+1
002013              1740    TEMP11  .EQU    TEMP10+1
002014              1750    TEMP12  .EQU    TEMP11+1
                    1760    *
                    1770    *       USER REGISTER STORAGE
                    1780    *
                    1790            .HEAD   0,D,M
002015              1800    USTORE  .EQU    TEMP12+1
002015              1810    ACSAVE  .EQU    USTORE
002016              1820    MQSAVE  .EQU    ACSAVE+1
002017              1830    PCSAVE  .EQU    MQSAVE+1
002020              1840    STSAVE  .EQU    PCSAVE+1
002021              1850    SCSAVE  .EQU    STSAVE+1
002022              1860    ACSW    .EQU    SCSAVE+1
002023              1870    10SAVE  .EQU    ACSW+1
002024              1880    11SAVE  .EQU    10SAVE+1
                    1890    *
                    1900    *       PHANTOM REGISTER STORAGE
                    1910    *
002025              1920    PHSTOR  .EQU    11SAVE+1
002025              1930    PACSAV  .EQU    PHSTOR
002026              1940    PMQSAV  .EQU    PACSAV+1
002027              1950    PPCSAV  .EQU    PMQSAV+1
002030              1960    PSTSAV  .EQU    PPCSAV+1
002031              1970    PSCSAV  .EQU    PSTSAV+1
002032              1980    PACSW   .EQU    PSCSAV+1
002033              1990    P10SAV  .EQU    PACSW+1
002034              2000    P11SAV  .EQU    P10SAV+1
                    2010    *
                    2020    *       DEBUGGER STORAGE
                    2030    *
                    2040            .HEAD   D
002035              2050    DBSTOR  .EQU    P11SAV+1
002035              2060    REGSW   .EQU    DBSTOR
002036              2070    ADRSW   .EQU    REGSW+1
002037              2080    DUMSW   .EQU    ADRSW+1
```

```
                D                       MTSS SYSTEM DEFINITIONS

002040      2090    PATSW   ,EQU    DUMSW+1
002041      2100    LIMIT   ,EQU    PATSW+1
002042      2110    LOC     ,EQU    LIMIT+1
002043      2120    PC      ,EQU    LOC+1
002044      2130    LOCOR   ,EQU    PC+1
002045      2140    HICOR   ,EQU    LOCOR+1
002046      2150    MASK    ,EQU    HICOR+1
002047      2160    RELOC   ,EQU    MASK+1
002050      2170    INDIR   ,EQU    RELOC+1
002051      2180    PCMSK   ,EQU    INDIR+1
002052      2190    REGBR   ,EQU    PCMSK+1
002053      2200    COMFLG  ,EQU    REGBR+1
002054      2210    BKTAB   ,EQU    COMFLG+1
000024      2220    BKNUM   ,EQU    20.             NUMBER OF BREAKPOINT CELLS
            2230    *
            2240    *       PHANTOM COMMON STORGAGE
            2250    *
            2260            ,HEAD   0
002150      2270    COMSTOR ,EQU    3+DSBKNUM+DSBKTAB
002150      2280    PHFLAG  ,EQU    COMSTOR
            2290    *
            2300    *       FILE BUFFER CONTROL STORAGE
            2310    *
            2320            ,HEAD   D
002151      2330    BCNTRL  ,EQU    PHFLAG+1
002151      2340    FTYPE   ,EQU    BCNTRL
002152      2350    OFTYP   ,EQU    FTYPE+1
002153      2360    BDA     ,EQU    OFTYP+1
002154      2370    BCA     ,EQU    BDA+1
002155      2380    BLEN    ,EQU    BCA+1
002156      2390    BALT    ,EQU    BLEN+1
002157      2400    BMIN    ,EQU    BALT+1
002160      2410    MBMIN   ,EQU    BMIN+1
002161      2420    BMAX    ,EQU    MBMIN+1
002162      2430    BPTR    ,EQU    BMAX+1
002163      2440    FDA     ,EQU    BPTR+1
002164      2450    MFDA    ,EQU    FDA+1
002165      2460    FMIN    ,EQU    MFDA+1
002166      2470    MFMIN   ,EQU    FMIN+1
002167      2480    FMAX    ,EQU    MFMIN+1
002170      2490    BUFFER  ,EQU    FMAX+1
001000      2500    BUFLEN  ,EQU    1000
            2510    *
            2520    *       ACTUAL CODE CONTROL
            2530    *
            2540            ,HEAD   M
003170      2550    IMPSTRT ,EQU    BUFFER+BUFLEN
003700      2560    PURSTR  ,EQU    SPURSTR
            2570            ,HEAD
```

MTSS SYSTEM DEFINITIONS

```
                    2590    *
                    2600    *
                    2610    *       MTSS DISK LAYOUT DEFINITIONS
                    2620    *
000100              2630    TABLEN  ,EQU    100             LENGTH OF A JOB TABLE
001700              2640    PHLEN   ,EQU    8K-USLEN-TABLEN  MAXIMUM LENGTH OF IMPURE PHANTOM CODE
016000              2650    DKLEN   ,EQU    8K              LENGTH OF EACH "USER PHYSICAL DISK"
000003₄             2660    DKLENB  ,EQU    DKLEN/400       LENGTH OF "USER PHYSICAL DISK" IN BLOCKS
640000              2670    SCRSTR  ,EQU    640000
```

MTSS SYSTEM DEFINITIONS

```
                2700    *       SYSTEM-WIDE CONSTANTS FOR THE PDP-9 TIME-SHARING SYSTEM.
                2710    *
575600          2720    ON      ,EQU    575600          ACI6 ON
574646          2730    OFF     ,EQU    574646          ACI6 OFF
000036          2740    DKWC    ,EQU    36              HOLDS TWO'S COMPLEMENT WORD COUNT FOR DISK READS
000037          2750    DKCA    ,EQU    37              CORE ADDRESS LOCATION FOR DISK READS
000002          2760    DKRD    ,EQU    2               NON-INTERRUPTING DISK READ COMMAND
000004          2770    DKWRT   ,EQU    4               NON-INTERRUPTING DISK WRITE COMMAND
000001          2780    PHANTOM ,EQU    1               FLAG FOR A PHANTOM PROGRAM
000000          2790    USER    ,EQU    0               FLAG FOR A USER PROGRAM
001300          2800    SYSBAS  ,EQU    1300            STARTING (BASE) BLOCK OF SYSTEM LOGICAL DISK
041300          2810    SYSDA   ,EQU    040000+SYSBAS
001777          2820    SYSMAX  ,EQU    1777            MAXIMUM BLOCK OF THE SYSTEM LOGICAL DISK
300000          2830    IOBLK   ,EQU    300000          MASK TO KEEP JUST THE I/O ROADBLOCK FLAGS
000050          2840    CLKMAX  ,EQU    40.             2/3 SECOND TIMER (60 PER SECOND CLOCK)
000003          2850    USERS   ,EQU    3               MAXIMUM NUMBER OF SIMULTANEOUS JOBS
                2860    *
                2870    *       ASCII CONSTANTS
                2880    *
000243          2890    SHARP   ,EQU    243             #
000300          2900    ATSGN   ,EQU    300             @
000275          2910    EQUAL   ,EQU    275             =
000274          2920    LESS    ,EQU    274             <
000276          2930    GREAT   ,EQU    276             >
000336          2940    UpARR   ,EQU    336             Up-ARROW
000300          2950    AT      ,EQU    300             @
                2960    *
                2970    *
                2980    *       TELETYPE INPUT/OUTPUT BUFFERS MUST BE OF A CERTAIN MINIMUM SIZE
                2990    *       IN ORDER TO PROVIDE UN-INTERRUPTED OUTPUT TO ALL TELETYPES.
                3000    *       A TELETYPE I/O BUFFER MUST AT NO TIME BE EMPTIED PAST THE POINT
                3010    *       WHERE IT HAS ENOUGH REMAINING OUTPUT TO TAKE UP THE TIME UNTIL
                3020    *       ITS JOB'S NEXT CORE SHOT, EVEN IF ALL OTHER USER'S WERE TO USE THEIR
                3030    *       MAXIMUM CORE ALLOWANCES.
                3040    *
                3050    *       WHEN THE TELETYPE I/O BUFFER HAS MORE THAN THE MINIMUM NUMBER OF CHARACTERS
                3060    *       IN IT, ITS JOB CAN BE SUSPENDED FROM RUNNING (I/O ROADBLOCKED).
                3070    *       IT FOLLOWS THAT FOR SYSTEM EFFICIENCY, THE BIGGER THE TELETYPE I/O
                3080    *       BUFFERS CAN BE, THE BETTER, BECAUSE A JOB THAT IS I/O ROADBLOCKED
                3090    *       COSTS VIRTUALLY NO PROCESSOR TIME.
                3100    *
                3110    *       TELETYPE I/O BUFFER CONSTANTS TO DETERMINE MINIMUM PERMISSIBLE BUFFER SIZE
                3120    *
000002          3130    CHRPAK  ,EQU    2               NUMBER OF CHARCTERS PACKED PER WORD IN THE TTY I/O BUFFER
000003          3140    TTYNUM  ,EQU    3               MAXIMUM NUMBER OF TELETYPES ON THE SYSTEM
000010          3150    TTYSPD  ,EQU    10              NUMBER OF CHARACTERS PER SECOND OF THE FASTEST TERMINAL ON THE SYSTEM
000060          3160    CLKSPD  ,EQU    60              NUMBER OF CLOCK PULSES PER SECOND
000006          3170    TTYCLK  ,EQU    CLKSPD/TTYSPD   NUMBER OF CLOCK COUNTS PER TTY OUTPUT CHARACTER
000006          3180    CHRMAX  ,EQU    CLKMAX/TTYCLK   MAXIMUM NUMBER OF CHARACTERS PRINTED DURING ONE STANDARD CPU SHOT
000002          3190    FUDGE   ,EQU    2               FUDGE FACTOR ON BUFFER SIZE
000010          3200    MINBUFF ,EQU    USERS-1*CHRMAX/CHRPAK+FUDGE  MINIMUM TTY BUFFER SIZE FOR CONTINUOUS PRINTING
```

MTSS SYSTEM DEFINITIONS

```
                    3230    *       DEFINITIONS OF LABELS REQUIRED FOR INTER-MODULE COMMUNICATIONS.
                    3240    *       EXCEPT FOR THE RESIDENT PROGRAM THE ADDRESS GIVEN IS THAT OF A
                    3250    *       POINTER TO THE ITEM WITHIN THE MODULE. IN THE CASE OF THE RESIDENT
                    3260    *       PROGRAM THE ADDRESS GIVEN IS THE ACTUAL ADDRESS OF THE ITEM IN QUESTION. THIS
                    3270    *       IS BECAUSE THE RESIDENT PROGRAM HAS NO ROOM FOR THE SIZEABLE
                    3280    *       TRANSFER VECTOR THAT WOULD BE REQUIRED OTHERWISE.
                    3290    *
                    3300    *       THE LABELS DEFINED HERE ARE THE SAME AS THE LABELS DEFINED IN THE
                    3310    *       MAIN PROGRAM, EXCEPT THAT HERE THEY ARE NOT UNDER A HEAD SYMBOL.
                    3320    *
                    3330    *       RESIDENT PROGRAM LABELS
                    3340    *
000002              3350    3TM21   ,EQU    2
000003              3360    3TM22   ,EQU    3
000005              3370    3AC     ,EQU    5
000006              3380    CNTRL   ,EQU    6
000006              3390    RCNT    ,EQU    CNTRL
000026              3400    .310    ,EQU    26
000027              3410    .311    ,EQU    27
000032              3420    RDT0    ,EQU    32
000033              3430    RDT1    ,EQU    33
000034              3440    RACS    ,EQU    34
000035              3450    RCORE   ,EQU    35
000040              3460    SWPS    ,EQU    40
000040              3470    RESCAT  ,EQU    SWPS
000044              3480    CSWP    ,EQU    RESCAT+4
000045              3490    CMP1    ,EQU    CSWP+1
000046              3500    CMP2    ,EQU    CMP1+1
000047              3510    CSPL    ,EQU    CMP2+1
000050              3520    3TM20   ,EQU    CSPL+1
000051              3530    3TEM0   ,EQU    3TM20+1
000052              3540    3TEM1   ,EQU    3TEM0+1
000053              3550    3TEM2   ,EQU    3TEM1+1
000054              3560    3TEM3   ,EQU    3TEM2+1
000055              3570    3TEM4   ,EQU    3TEM3+1
000056              3580    3TEM5   ,EQU    3TEM4+1
000057              3590    3TEM6   ,EQU    3TEM5+1
000060              3600    CTBFR   ,EQU    3TEM6+1
000016              3610    KBLEN   ,EQU    MINBUF+6
000010              3620    KBNUM   ,EQU    8.
000076              3630    LOLOK   ,EQU    CTBFR+KBLEN
000100              3640    CTBIN   ,EQU    CTBFR+KBLEN+2
000102              3650    CTFLG   ,EQU    CTBIN+2
000104              3660    CTNAM   ,EQU    CTFLG+2
000107              3670    L1BFR   ,EQU    CTBIN+KBNUM-1
000125              3680    L1LOK   ,EQU    L1BFR+KBLEN
000127              3690    L1BIN   ,EQU    L1BFR+KBLEN+2
000131              3700    L1FLG   ,EQU    L1BIN+2
000133              3710    L1NAM   ,EQU    L1FLG+2
000136              3720    L2BFR   ,EQU    L1BIN+KBNUM-1
000154              3730    L2LOK   ,EQU    L2BFR+KBLEN
000156              3740    L2BIN   ,EQU    L2BFR+KBLEN+2
```

MTSS SYSTEM DEFINITIONS

```
000160  3750  L2FLG   ,EQU   L2BIN+2
000162  3760  L2NAM   ,EQU   L2FLG+2
000227  3770  PFLAG   ,EQU   L2NAM+45
000230  3780  RPTP    ,EQU   PFLAG+1
000234  3790  RFLAG   ,EQU   RPTP+4
000235  3800  RPTR    ,EQU   RFLAG+1
000241  3810  PBFLAG  ,EQU   RPTR+4
000242  3820  RSCO    ,EQU   PBFLAG+1
000266  3830  DKLOK   ,EQU   RSCO+20.
000270  3840  PIDON   ,EQU   DKLOK+2
000274  3850  PIDN2   ,EQU   PIDON+4

000303  3860  PIOUT   ,EQU   PIDN2+7
000305  3870  3REST   ,EQU   PIOUT+2
000335  3880  SWAP    ,EQU   3REST+24.
000336  3890  SWAP1   ,EQU   SWAP+1
000340  3900  SWAP3   ,EQU   SWAP1+2
000513  3910  IO.IN   ,EQU   SWAP3+153
000525  3920  IO.OT   ,EQU   IO.IN+10.
000540  3930  NEWBR   ,EQU   IO.OT+11.
000546  3940  PUTIN   ,EQU   NEWBR+6
000602  3950  FGET    ,EQU   PUTIN+28.
000623  3960  NXPTR   ,EQU   FGET+17.
000634  3970  BIT0    ,EQU   NXPTR+9.
000635  3980  BIT36   ,EQU   BIT0+1
000636  3990  BIT5    ,EQU   BIT36+1
000637  4000  BIT6    ,EQU   BIT5+1
000640  4010  BIT7    ,EQU   BIT6+1
000641  4020  BIT17   ,EQU   BIT7+1
000642  4030  BL7     ,EQU   BIT17+1
000643  4040  BL8     ,EQU   BL7+1
000644  4050  CB0     ,EQU   BL8+1
000645  4060  CB1     ,EQU   CB0+1
000646  4070  CB5     ,EQU   CB1+1
000647  4080  CB7     ,EQU   CB5+1
000650  4090  CBL8    ,EQU   CB7+1
000651  4100  ADRSS   ,EQU   CBL8+1
000652  4110  JMP     ,EQU   ADRSS+1
000653  4120  DBK     ,EQU   JMP+1
000654  4130  DQ      ,EQU   DBK+1
000662  4140  DQ2     ,EQU   DQ+6.
000663  4150  DQ3     ,EQU   DQ2+1
000672  4160  DKOVR   ,EQU   DQ3+7.
000675  4170  DKDON   ,EQU   DKOVR+3.
000702  4180  QC0     ,EQU   DKDON+5.
000703  4190  QC1     ,EQU   QC0+1
000704  4200  QC2     ,EQU   QC1+1
000705  4210  QC3     ,EQU   QC2+1
        4220  *
        4230  *       DEFINE THE NAMES ASSOCIATED WITH EACH POSSIBLE USER
        4240  *
000076  4250  US0     ,EQU   CTBIN+2          NAME OF THE USER PROGRAM FOR USER #0
000077  4260  PH0     ,EQU   US0+1            NAME OF THE PHANTOM PROGRAM DISK STORAGE SPACE FOR USER #0
```

MTSS SYSTEM DEFINITIONS

```
000100      4270    DK0     ,EQU    PH0+1           NAME OF THE USER "PHYSICAL DISK" DISK STORAGE SPACE FOR USER #0
000075      4280    UT0     ,EQU    US0-1
000125      4290    US1     ,EQU    L1BIN-2
000126      4300    PH1     ,EQU    US1+1
000127      4310    DK1     ,EQU    PH1+1
000124      4320    UT1     ,EQU    US1-1
000154      4330    US2     ,EQU    L2BIN-2
000155      4340    PH2     ,EQU    US2+1
000156      4350    DK2     ,EQU    PH2+1
000153      4360    UT2     ,EQU    US2-1
            4370    *
            4380    *       JOB TABLE LABELS
            4390    *
001700      4400    FRDA    ,EQU    JTSTRT
001701      4410    FRCA    ,EQU    FRDA+1
001702      4420    FRLEN   ,EQU    FRCA+1
001703      4430    FRSTA   ,EQU    FRLEN+1
001704      4440    UTEM0   ,EQU    FRSTA+1
001705      4450    UTEM1   ,EQU    UTEM0+1
001706      4460    UTEM2   ,EQU    UTEM1+1
001707      4470    UTEM3   ,EQU    UTEM2+1
001710      4480    UTEM4   ,EQU    UTEM3+1
001711      4490    UTEM5   ,EQU    UTEM4+1
001712      4500    UTEM6   ,EQU    UTEM5+1
001713      4510    .0      ,EQU    UTEM6+1
001753      4520    AC      ,EQU    .0+40
001754      4530    MQ      ,EQU    AC+1
001755      4540    SC      ,EQU    MQ+1
001756      4550    ACS     ,EQU    SC+1
001757      4560    CLOCK   ,EQU    ACS+1
001760      4570    IORS    ,EQU    CLOCK+1
001761      4580    DFLAG   ,EQU    IORS+1
001762      4590    DAP0    ,EQU    DFLAG+1
001763      4600    DAP1    ,EQU    DAP0+1
001764      4610    DFN     ,EQU    DAP1+1
001765      4620    DSTAT   ,EQU    DFN+1
001766      4630    UCORE   ,EQU    DSTAT+1
001767      4640    UDISK   ,EQU    UCORE+1
001770      4650    VALID   ,EQU    UDISK+1
001771      4660    NUMBR   ,EQU    VALID+1
001772      4670    NAME    ,EQU    NUMBR+1
001773      4680    OVER    ,EQU    NAME+1
001774      4690    TYPE    ,EQU    OVER+1
001775      4700    PURNM   ,EQU    TYPE+1
001776      4710    RSTRT   ,EQU    PURNM+1
            4720    *
            4730    *       SWAPPING PROGRAM POINTER ADDRESSES
            4740    *
001000      4750    SWCAT   ,EQU    OVSTRT
001001      4760    SWPPR   ,EQU    SWCAT+1
001002      4770    SWMTR   ,EQU    SWPPR+1
001003      4780    SWCLK   ,EQU    SWMTR+1
```

MTSS SYSTEM DEFINITIONS

```
001004   4790   SWERR   .EQU    SWCLK+1
001005   4800   SWSPL   .EQU    SWERR+1
001006   4810   SXSPL   .EQU    SWSPL+1
001007   4820   SWMP1   .EQU    SXSPL+1
001010   4830   SWMP2   .EQU    SWMP1+1
001011   4840   SWOPR   .EQU    SWMP2+1
         4850   *
         4860   *       MEMORY PROTECTION PROGRAM POINTER ADDRESSES
         4870   *
001000   4880   MPST    .EQU    OVSTRT
001001   4890   PINT    .EQU    MPST+1
001002   4900   IOT0    .EQU    PINT+1
001003   4910   RDBLK   .EQU    IOT0+1
001004   4920   MPOPR   .EQU    RDBLK+1
         4930   *
         4940   *       SPECIAL IOT (EXECUTIVE CALL) HANDLER POINTER ADDRESSES
         4950   *
001000   4960   SPLST   .EQU    OVSTRT
         4970   *
         4980   *       DEBUGGER PROGRAM POINTER ADDRESSES
         4990   *
012000   5000   DDTST   .EQU    12000
         5010   *
         5020   *       MTSS LOADER PROGRAM POINTER ADDRESSES
         5030   *
002000   5040   LDRST   .EQU    BOUNDARY
         5050   *
         5060   *       MTSS MONITOR/MESSAGE OUTPUT PROGRAM POINTER ADDRESSES
         5070   *
002000   5080   MTRST   .EQU    BOUNDARY
         5090   *
         5100   *       THE FOLLOWING MACROS ARE USED ONLY IN PURE-CODED PHANTOM PROGRAMS,
         5110   *       THEY HELP TO KEEP IMPURE CODE SEPARATE FROM PURE CODE, AND TO
         5120   *       PUT ONLY THE NECESSARY THINGS IN THE IMPURE AREA.
         5130   *
         5280   ENTER   .DEFIN
         5290           .PMC    SAVE,ON
         5300   #1      XX
         5310           .PMC    RESTORE
         5320           .ENDM
```

MTSS SYSTEM DEFINITIONS

```
                5350    *       MTSS EXECUTIVE SERVICES ARE REQUESTED USING A SPECIAL SET OF
                5360    *       OTHERWISE UNUSED IOT INSTRUCTIONS (IOT+5000 - IOT+5377).
                5370    *       HENCE THE NAME 'SPECIALS'
                5380    *
777400          5390    SPMSK   ,EQU    777400          MASK TO RETAIN THE "SPECIAL" BITS
                5400    SPECIAL ,OPDEF  IOT+5000
000377          5410    SPCOD   ,EQU    377             MASK TO RETAIN JUST THE BIT-CODE FROM THE SPECIAL
                5420
                5430    MPOFF   ,DEFIN
                5440            ,PMC    SAVE,ON
                5450            SPECIAL+0               TURN OFF MEMORY PROTECT
                5460            ,PMC    RESTORE
                5470            ,ENDM
                5480    TERMINATE ,OPDEF SPECIAL+1
                5490    READ    ,OPDEF  SPECIAL+2
                5500    PREAD   ,OPDEF  SPECIAL+3
                5510    WRITE   ,OPDEF  SPECIAL+4
                5520    PWRITE  ,OPDEF  SPECIAL+5
000375          5530    TRCON   ,EQU    375             DDT TRACE ON SPECIAL
000376          5540    TRCOFF  ,EQU    376             DDT TRACE OFF SPECIAL
000377          5550    BRK     ,EQU    377             DDT BREAKPOINT
                5560    *
                5570    *
                5580    *
                5590    *       MTSS EXECUTIVE CALL MACROS
                5600    *
                5610    SWAP    ,DEFIN                  CONTROL,NAME,RESTART,NUMBER,(.....)
                5620            SPECIAL 1               SPECIAL IOT SWAPPER REQUEST
                5630            #1                      SWAPPER CONTROL WORD
                5640            #2                      SYSTEM FILENAME
                5650            #3                      RESTART OVERRIDE
                5660            #4                      PASSED PARAMETER COUNT
                5670            ,IDRP   #5
                5680            #5
                5690            ,IDRP
                5700            ,ENDM
                5710
                5730            ,END
                2100            ,INSRT  :DLIBRARY:PDP9LIB:BRODEFIN
```

GROWTH SYSTEM STANDARD DEFINITIONS

```
                140
                150     *       PROGRAMMED BY ROBERT W. BLEAN
                160
                170     *       LATEST REVISION 20 JAN 1971
                180
                190     *       ASCII CHARACTERS
                200
000212          210     LF      .EQU    212
000215          220     CR      .EQU    215
000230          230     CONTX   .EQU    230
000337          240     BKARR   .EQU    337
000240          250     SPACE   .EQU    240
000241          260     EXCLAM  .EQU    241             EXCLAMATION POINT
000243          270     NUMSGN  .EQU    243
000244          280     DOLLAR  .EQU    244             $
000246          290     AMPRSN  .EQU    246             &
000252          300     STAR    .EQU    252             ASTERISK (*)
000253          310     PLUS    .EQU    253
000254          320     COMMA   .EQU    254
000255          330     MINUS   .EQU    255
000256          340     PERIOD  .EQU    256             .
000256          350     POINT   .EQU    PERIOD
000257          360     SLASH   .EQU    257
000272          370     COLON   .EQU    272
000273          380     SCOLON  .EQU    273
000334          390     BSLASH  .EQU    334             BACK SLASH (\)
                400
                410     *       CONSTANTS
                420
017777          430     ADRSS   .EQU    17777           ADRESS FIELD MASK
002000          440     BOUNDA  .EQU    2000            TSS USER CORE START
017500          450     TAPIN   .EQU    17500
017502          460     TAPOT   .EQU    17502
017505          470     RECOV   .EQU    17505
017777          480     VFLAG   .EQU    17777
000010          490     INDEX   .EQU    10              GENERAL PURPOSE AUTO-INDEX REGISTER
000011          500     CATX    .EQU    11              CATALOG ROUTINES' AUTO-INDEX REGISTER
000012          510     CMDX    .EQU    12
017740          520     BOOT    .EQU    17740           BOOTSTRAP LOADER STARTING ADDRESS
017735          530     SYSDEV  .EQU    BOOT-3          HOLDS DEVICE ADDRESS OF CATALOG BLOCK ON THE SYSTEM DEVICE
017000          540     CATLOG  .EQU    17000           START OF THE RESIDENT CATALOG BLOCK
000001          550     CATBLK  .EQU    1               CATALOG IS AT LOGICAL BLOCK 1 OF ANY DEVICE
000400          560     CATLEN  .EQU    400             CATALOG LENGTH IS 400 WORDS MAXIMUM
000005          570     FCBLEN  .EQU    5               FILE CONTROL BLOCK IS FIVE WORDS LONG
000004          580     HDRLEN  .EQU    4               CATALOG HEADER IS FOUR WORDS LONG
017005          590     CPARAM  .EQU    CATLOG+5        POINTER TO PARAMETERS FOR CATALOG READ/WRITE
740000          600     DVCMSK  .EQU    740000          MASK TO EXTRACT HANDLER NUMBER AND TYPE FROM DEVICE ADDRESS
001777          610     BLKMSK  .EQU    1777            MASK TO RETRIEVE DEVICE BLOCK NUMBER
777716          620     CATMAX  .EQU    -50.            MAXIMUM NUMBER OF FILE CONTROL BLOCKS IN A CATALOG
000400          630     BLKLEN  .EQU    400             NUMBER OF WORDS IN ONE LOGICAL BLOCK
776701          640     DTMAX   .EQU    -1077           MAXIMUM NUMBER OF USABLE BLOCKS ON A DECTAPE
777601          650     DKMAX   .EQU    -177            MAXIMUM NUMBER OF USABLE BLOCKS ON A LOGICAL DISK
```

GROWTH SYSTEM STANDARD DEFINITIONS

```
                    660
                    670     *       DEVICE NAMES
                    680
606064              690     PPT     .EQU    606064
606462              700     PTR     .EQU    606462
606460              710     PTP     .EQU    606460
446400              720     DT.     .EQU    446400
646000              730     TP.     .EQU    646000
445300              740     DK.     .EQU    445300
004464              750     .DT     .EQU    004464
006460              760     .TP     .EQU    006460
004453              770     .DK     .EQU    004453
445320              780     DK0     .EQU    445320
                    790
                    800     *       FILENAMES
436454              810     CTL     .EQU    436454          CATALOG BLOCK
                    820
                    830     *       FORMATS
                    840
414263              850     ABS     .EQU    414263          LOADSTRING BINARY
425156              860     BIN     .EQU    425156          BINARY
476257              870     GRO     .EQU    476257          GROWTH SYSTEM FORMAT (CORE IMAGE)
435762              880     COR     .EQU    435762          CORE
                    890
                    900     *       MACROS
                    910
                    920     ENTER   .DEFIN
                    930     #1      XX
                    940             .ENDM
                    950
                    960     LOOP    .DEFIN
                    970             ISZ     #1
                    980             JMP     #2
                    990             .ENDM
                   1000
                   1010     NEG     .DEFIN
                   1020             CMA
                   1030             TAD     (1              )
                   1040             .ENDM
                   1050
                   1060     FORMAT  .DEFIN
                   1070             JMP     FORMAT
                   1080             .ENDM
                   1090
                   1100     START   .DEFIN                  STANDARD INITIALIZATION MACRO FOR THE GROWTH SYSTEM
                   1110             .PMC    SAVE,ON         PRINT THIS ONE MACRO, AT LEAST
                   1120             CAF
                   1130             IOF!CLOF
                   1140             LAC     (700000         )
                   1150             ISA                     API ON, NO PAPER TAPE READER ATTACHED
                   1160             TLS+10
                   1170             DLP                     DISABLE THE LIGHT PEN, ON GENERAL PRINCIPLES
```

GROWTH SYSTEM STANDARD DEFINITIONS

```
1180            DZM     CATALT          WE WON'T MESS WITH SOMEONE ELSE'S ALTERED CATALOG
1190            MESS    <#1             HERE>,#2-5
1200    NEXTL   MESS    <               )>,1 "PRINT THE INPUT REQUEST
1210            LINE                    GET THE USER'S INPUT
1220            .PMC    RESTORE
1230            .ENDM
1240
1260            .END
2110            .INSRT  :DLIBRARY:PDP9LIB:LIBMACRO
140
150
160     *
170     *       THESE MACROS ARE FOR USE WITH THE PROGRAM PDP9LIB***:TTY-NON
180     *       TTY-NON IS A NON-INTERRUPT DRIVEN TELETYPE HANDLER FOR THE CONSOLE
190     *       TELETYPE ON THE PDP-9.
200     *
210     *       LINE INPUT MACRO IS:
220     *
230     *            LINE  -- GETS THE NEXT LINE FROM THE TELETYPE, PACKS IT IN THE
240     *                     INCLUDED LINE BUFFER, AND RETURNS TO THE USER, USE BACK-ARROW
250     *                     FOR CHARACTER DELETION AND CONTROL X FOR LINE DELETION.
260     *                     THE ROUTINE PROTECTS AGAINST BUFFER UNDERFLOW OR OVERFLOW.
270     *
280     *       WORD INPUT MACROS ALL DELETE LEADING BLANKS, RETURNING TO THE USER
290     *       AT +1 WITH THE DELIMITER IN THE AC IF A DELIMITER IS THE FIRST NON-
300     *       BLANK CHARACTER, THEY ALL UTILIZE WORDB AND WORDB+1 FOR STORAGE, AND
310     *       ANY VALUE ACCUMULATED THERE REMAINS UNTIL THE NEXT TIME A WORD-PACKING
320     *       MACRO IS USED ('WORD' OR 'NUM'). THE DELIMITER THAT ENDED THE WORD
330     *       IS STORED IN DLMTR UNTIL THE NEXT TIME A WORD PACKING MACRO IS USED
340     *       OR UNTIL THE USER PROGRAM USES THE ROUTINE 'CHRID'.
350     *       THE AVAILABLE MACROS ARE:
360     *
370     *            WORD  -- PACKS CHARACTERS, IN A LEFT-JUSTIFIED SIXBIT PACK,
380     *                     INTO WORDB, WORDB+1, ..., RETURNS THE FIRST THREE (OR
390     *                     FEWER) CHARACTERS LEFT JUSTIFIED IN THE AC.
400     *
410     *            NUM   -- GETS A NUMBER, AND RETURNS IT IN THE AC. A FORMAT ERROR
420     *                     IS CAUSED BY A LETTER BEING FOUND OR BY A DECIMAL DIGIT
430     *                     (8 OR 9) BEING FOUND WITHOUT A TRAILING DECIMAL POINT,
440     *                     THAT THE DECIMAL VALUE IS DESIRED IS SIGNALLED BY THE
450     *                     DELIMITER BEING A PERIOD. OTHERWISE THE OCTAL VALUE IS
460     *                     RETURNED. THE VALUE RETURNED REMAINS AVAILABLE IN WORDB.
470     *                     THIS IS THE VALUE FOUND MOD 2+18 -- I.E. OVERFLOW IS LOST.
480     *
490     *                     RETURN IS:
500     *                          +1 WITH LINK = 0 FOR A FORMAT ERROR
510     *                          +1 WITH LINK = 1 FOR THE FIRST NON-BLANK CHARACTER A DELIMITER
520     *                          +2 FOR SUCCESS
530     *
540     *            WORD1 -- GETS THE CONTENTS FROM WORDB. THIS IS THE FIRST THREE
550     *                     SIXBIT CHARACTERS OR THE VALUE.
560     *            WORD2 -- GETS THE CONTENTS OF WORDB+1. THIS IS THE SECOND THREE
```

GROWTH SYSTEM STANDARD DEFINITIONS

```
570     *                   SIXBIT CHARACTERS OR THE "DECIMAL" VALUE. NOTE THAT THE
580     *                   "DECIMAL" VALUE WILL BE GARBAGE IF AN OCTAL NUMBER WAS INPUT.
590     *
600     *                   IN THE CASE OF SIXBIT INPUT, FURTHER INPUT WILL BE LOST.
610     *
620     *         COUNT -- GETS THE OCTAL COUNT OF THE NUMBER OF TIMES 'WORD' AND
630     *                   'NUM' HAVE BEEN CALLED SINCE THE LINE WAS INPUT, THIS
640     *                   IS THE COUNT OF THE NUMBER OF WORDS EXTRACTED SO FAR
650     *                   FROM THE CURRENT LINE BUFFER.
660     *
670     *         DELIM -- GETS THE LAST DELIMITER SEEN BY 'CHRID'. THIS WILL BE
680     *                   THE DELIMITER THAT ENDED THE LAST WORD FETCHED UNLESS
690     *                   THE USER PROGRAM IS ACCESSING 'CHRID' ITSELF.
700     *
710     *       MISCELLANEOUS CHARACTER-ORIENTED MACROS:
720     *
730     *            CHAR  -- GETS THE OLDEST REMAINING CHARACTER FROM THE LINE BUFFER.
740     *                     THIS PERMITS THE USER PROGRAM TO EXAMINE THE ENTIRE INPUT
750     *                     STRING, WHICH IS A HARD THING TO DO USING 'WORD'.
760     *                  RETURNS +1 WITH THE CHARACTER IN THE AC
770     *
780     *            CRLF  -- PRINTS A CARRIAGE RETURN AND LINE FEED. IT DISTURBS NO
790     *                     STORAGE OR POINTERS.
800     *
810     *            CHROT -- PRINTS THE SINGLE ASCII CHARACTER IN THE AC.
820     *
830     *
840     *       OUTPUT MACROS ARE:
850     *
860     *            OCT   -- OUTPUTS AS SIX DIGIT OCTAL THE CONTENTS OF THE AC.
870     *
880     *            OCTZ  -- OUTPUTS AS OCTAL WITH LEADING ZEROES SUPPRESSED THE CONTENTS OF THE AC.
890     *
900     *            MESS <TEXT>,<CHARACTER COUNT> USES SIXBIT FORMAT TO OUTPUT THE
910     *                     CARRIAGE RETURN AND LINE FEED, FOLLOWED BY THE TEXT, IT
920     *                     FIRST DOES A 'KRB' INSTRUCTION TO CLEAR ANY PRINT-INHIBIT.
930     *
940     *            MESSR <TEXT>,<CHARACTER COUNT> IS THE SAME AS 'MESS', BUT NO
950     *                     'KRB' IS SUPPLIED. THIS PERMITS CONTINUATION OF A SINGLE
960     *                     MESSAGE.
970     *
980     *            NMESS <TEXT>,<CHARACTER COUNT> IS THE SAME AS 'MESSR' EXCEPT
990     *                     NO CARRIAGE RETURN NOR LINE FEED IS SUPPLIED. THIS PERMITS
1000    *                     CONTINUING THE MESSAGE ON THE SAME LINE.
1010    *
1020    *       HITTING ANY KEY ON THE TELETYPE DURING OUTPUT WILL INHIBIT THE ACTUAL
1030    *       PRINTING OF THE REST OF THE MESSAGE UNTIL THE NEXT 'MESS' OR KRB
1040    *       INSTRUCTION. NOTE THAT EXCEPT THE CHARACTER IS NOT PRINTED, THE REST
1050    *       OF THE PROGRAM CARRIES ON AS USUAL.
1060    *
1070
1080
```

GROWTH SYSTEM STANDARD DEFINITIONS

```
1090
1100    LINE    .DEFIN
1110            JMS     T$INLIN
1120            .ENDM
1130
1140    WORD    .DEFIN
1150            JMS     T$SIXIN
1160            .ENDM
1170
1180    WORD1   .DEFIN
1190            LAC     T$WORDB
1200            .ENDM
1210
1220    WORD2   .DEFIN
1230            LAC     T$WORDB+1
1240            .ENDM
1250
1260    NUM     .DEFIN
1270            JMS     T$NUMIN
1280            .ENDM
1290
1300    CRLF    .DEFIN
1310            JMS     T$CRLF
1320            .ENDM
1330
1340    CHROT   .DEFIN
1350            JMS     T$TTYOT
1360            .ENDM
1370
1380    CHAR    .DEFIN
1390            JMS     T$FGET
1400            .ENDM
1410
1420    DELIM   .DEFIN
1430            LAC     T$DLMTR
1440            .ENDM
1450
1460    COUNT   .DEFIN
1470            LAC     T$COUNT
1480            .ENDM
1490
1500
1510
1520    MESSR   .DEFIN
1530            .CRSM   SAVE,ON
1540            LAW     -#2-2
1550            JMS     T$SIXOT
1560            .PMC    SAVE,OFF
1570    #5      .ACI6   +[]#1+
1580            .PMC    RESTORE
1590            .CRSM   RESTORE
1600            .ENDM
```

GROWTH SYSTEM STANDARD DEFINITIONS

```
1610
1620    MESS    .DEFIN
1630            KRB
1640            MESSR   <#1>,#2
1650            .ENDM
1660
1670    NMESS   .DEFIN
1680            .CRSM   SAVE,ON
1690            LAW     -#2
1700            JMS     TSSIXOT
1710    #5      .ACI6   +#1+
1720            .CRSM   RESTORE
1730            .ENDM
1740
1750    EMESS   .DEFIN
1760            .CRSM   SAVE,ON
1770            MESS    <#1             WORD #>,#2-7
1780            COUNT
1790            OCTZ
1800            .CRSM   RESTORE
1810            .ENDM
1820
1830
1840    OCTZ    .DEFIN                  OCTAL PRINTOUT OF THE AC WITH LEADING ZEROES SUPPRESSED
1850            STL
1860            JMS     TSOCTOT
1870            .ENDM
1880
1890    OCT     .DEFIN                  OCTAL PRINTOUT OF THE AC
1900            CLL
1910            JMS     TSOCTOT
1920            .ENDM
1930
1950            .END
```

MAIN PROGRAM

```
                                2130    SYSDSK  ,EQU    DK0             DISK 0 IS OUR ONLY DISK, AND IS THE SYSTEM DISK
                                2140    RET     ,OPDEF  JMP             CORRECT GROWTH DEFINITION
                                2150    LIBDEV  ,EQU    ISLCAT
                                2160            ,HEAD   I
                                2170
                                2180    *
                                2190    *       INITIALIZE THE HARDWARE
                                2200    *
                                2210            ,LOC    12000
012000  703302                  2220    START   CAF
012001  700006                  2230            IOF!CLOF
012002  705514                  2240            ISA+10
012003  707074                  2250            DLAH+10
012004  214374                  2260            LAC     (BOUNDARY)
012005  701704                  2270            MPLD
012006  700721                  2280            DLP
012007  700416                  2290            TLS+10
                                2300    *
                                2310    *       INITIALIZE THE SOFTWARE
                                2320    *
                                2330    *       COPY THE LIBRARY DECTAPE CATALOG INTO LCAT.
                                2340    *
012010  776777                  2350            LAW     SCATLOG-1
012011  040010                  2360            DAC     10              SET THE START ADDRESS OF THE REGULAR BLOCK
012012  773773                  2370            LAW     LCAT-1
012013  040011                  2380            DAC     11              SET THE START ADDRESS OF THE SPECIAL BLOCK
012014  777400                  2390            LAW     -SCATLEN
012015  053651                  2400            DAC     TEMP2           SET THE LENGTH TO COPY
012016  220010                  2410    INT00   LAC     10,X
012017  060011                  2420            DAC     11,X            TRANSFER THE NEXT  WORD
012020  453651                  2430            ISZ     TEMP2           COUNT THE TRANSFER
012021  612016                  2440            JMP     INT00           NOT DONE YET -- LOOP
                                2450    *
                                2460    *       COPY THE SYSTEM DISK CATALOG INTO SCATLOG
                                2470    *
012022  772027                  2480            LAW     INT01
012023  053516                  2490            DAC     CSDEVCV         SET THE RETURN ADDRESS
012024  214375                  2500            LAC     (SYSDSK)        GET THE SYSTEM DISK MNEMONIC
012025  052634                  2510            DAC     T$WORDB         PASS THE MNEMONIC TO THE SUBROUTINE
012026  613523                  2520            JMP     CSDEVC3         CONVERT IT TO DEVICE ADDRESS FORMAT
012027  740040                  2530    INT01   XX                      FATAL ERROR IF CONVERSION THINKS IT IS A PAPER TAPE
012030  053657                  2540            DAC     SYSDVC          SET THE SYSTEM DEVICE ADDRESS
012031  113306                  2550            JMS     CSRCA"          GET THE SYSTEM DISK CATALOG
```

I MAIN PROGRAM

```
                    2570    *
                    2580    *       UNSAVE ALL SYSTEM FILES FROM THE DISK.
                    2590    *
012032  773634      2600            LAW     UFILES
012033  040010      2610            DAC     10              SET A POINTER TO THE LIST OF FILES TO UNSAVE
012034  112336      2620            JMS     UNSAVE          UNSAVE ALL OF THE OVERLAY FILES
012035  112336      2630            JMS     UNSAVE          UNSAVE ALL OF THE USER-TYPE SYSTEM PROGRAMS
012036  112336      2640            JMS     UNSAVE          UNSAVE ALL OF THE PHANTOM-TYPE SYSTEM PROGRAMS
012037  453305      2650            INX     SCATALT         DEBUGGING INSTRUCTION
012040  113354      2660            JMS     CSFORCE         DEBUGGING INSTRUCTION
                    2670    *
                    2680    *
                    2690    *       PURGE THE DISK TO CLEAN UP ITS CATALOG AND TO COMPACT ANY STORAGE ON IT
                    2700    *
012041  213657      2710            LAC     SYSDVC
012042  052421      2720            DAC     INDA            SET THE INPUT DEVICE
012043  052422      2730            DAC     OUTDA           SET THE OUTPUT DEVICE ADDRESS
012044  217002      2740            LAC     SCATLOG+2
012045  053650      2750            DAC     TEMP1           SET THE FILE CONTROL BLOCK COUNT
                    2760    *
                    2770    *       PUT A NEW HEADER ON THE CATALOG
                    2780    *
012046  217005      2790            LAC     SCPARAM         LOAD THE DEVICE ADDRESS OF THE CATALOG
012047  354376      2800            TAD     (1)
012050  057000      2810            DAC     SCATLOG         RESET THE DEVICE ADDRESS OF THE FIRST FREE BLOCK
012051  777010      2820            LAW     SCATLOG+10
012052  057001      2830            DAC     SCATLOG+1       RESET THE POINTER TO THE FIRST FREE FCB
012053  040012      2840            DAC     SCMDX
012054  040011      2850            DAC     SCATX
012055  777777      2860            LAW     -1
012056  057002      2870            DAC     SCATLOG+2       RESET THE FCB COUNT
012057  453305      2880            INX     SCATALT         RESET THE CATALOG ALTERED FLAG
                    2890    *
                    2900    *       NOW RECOPY THE FILES, COMPACTING CATALOG AND STORAGE
                    2910    *       SCMDX RUNS DOWN THE OLD DEVICE CATALOG
                    2920    *       CATX RUNS DOWN THE NEW DEVICE CATALOG
                    2930    *
012060  453650      2940    PURL    ISZ     TEMP1           CHECK FOR DONE
012061  741000      2950            SKP
012062  612115      2960            JMP     INT03
012063  220012      2970            LAC     SCMDX,X         GET THE NEXT FILE
012064  741200      2980            SNA
012065  612111      2990            JMP     PURZ            NOT THERE
012066  113555      3000            JMS     CSSAVE          SAVE IT
012067  740040      3010            HLT                     %#&*!  THE FILE CANNOT POSSIBLY BE SAVED  !*&#%
012070  220012      3020            LAC     SCMDX,X
012071  052421      3030            DAC     INDA            SET THE INPUT FILE'S CURRENT DEVICE ADDRESS
012072  220012      3040            LAC     SCMDX,X
012073  053651      3050            DAC     TEMP2           SAVE THE FILE'S CORE ADDRESS
012074  220012      3060            LAC     SCMDX,X
012075  052423      3070            DAC     LEN             SAVE THE FILE'S LENGTH
012076  113604      3080            JMS     CSALC           ALLOCATE SPACE ON THE DEVICE FOR IT
```

I MAIN PROGRAM

```
012077  060011  3090          DAC     $CATX,X         SET ITS NEW DEVICE ADDRESS
012100  052422  3100          DAC     OUTDA           SAVE FOR OUTPUT
012101  213651  3110          LAC     TEMP2
012102  060011  3120          DAC     $CATX,X         SET IT'S CORE ADDRESS
012103  212423  3130          LAC     LEN
012104  060011  3140          DAC     $CATX,X         SET IT'S LENGTH
012105  220012  3150          LAC     $CMDX,X
012106  060011  3160          DAC     $CATX,X         SET ITS TRANSFER CARD
012107  112424  3170          JMS     COPY
012110  612060  3180          JMP     PURL            LOOP
012111  214377  3190  PURZ    LAC     (FCBLEN-1)
012112  340012  3200          TAD     $CMDX

012113  040012  3210          DAC     $CMDX           SAVE NEW POSITION
012114  612060  3220          JMP     PURL            LOOP
                3230  *
                3240  *       THE DISK PURGE IS NOW DONE -- NOW COPY THE SYSTEM FILES TO THE DISK.
                3250  *
012115  113354  3260  INT03   JMS     CSFORCE         DEBUGGING AID
012116  773634  3270          LAW     UFILES
012117  040010  3280          DAC     10              SET A POINTER TO THE LIST OF FILES TO SAVE
012120  112350  3290          JMS     SAVE            SAVE ALL OF THE OVERLAY FILES
012121  112350  3300          JMS     SAVE            SAVE ALL OF THE USER-TYPE SYSTEM PROGRAMS
012122  112350  3310          JMS     SAVE            SAVE ALL OF THE PHANTOM-TYPE SYSTEM PROGRAMS
012123  113354  3320          JMS     CSFORCE         FORCE THE DISK CATALOG NOW
```

I MAIN PROGRAM

```
                    3340    *
                    3350    *
                    3360    *       INTCAT -- NOW COPY THE ENTRIES FROM THE SYSTEM DISK CATALOG INTO INTCAT
                    3370    *       ADJUSTING THE DISK ADDRESS TO BE PHYSICAL DISK ADDRESSES AS WE GO.
                    3380    *
012124  773634      3390    INT10   LAW     UFILES
012125  040010      3400            DAC     10              SET A POINTER TO THE FILES WHOSE CATALOG ENTRIES ARE TO BE COPIED
012126  773723      3410            LAW     UCAT
012127  040012      3420            DAC     12              SET A POINTER TO THE CATALOG INTO WHICH TO COPY THEM
012130  112464      3430            JMS     CORCPY          COPY THE USER-TYPE SYSTEM FILES
012131  112514      3440            JMS     PHCRCP          COPY THE PHANTOM-TYPE SYSTEM PROGRAMS, SETTING THEIR PURE CODE ENTRIES
012132  112547      3450            JMS     OVCRCP          COPY THE OVERLAY FILES
                    3460    *
                    3470    *       OVERLAYS -- SWAPPER
                    3480    *
012133  214400      3490            LAC     (SSWP)
012134  112573      3500            JMS     READ            GET THE SWAPPER OVERLAY
012135  201000      3510            LAC     SSWCAT
012136  040010      3520            DAC     10              SET THE POINTER TO THE SWAPPER CATALOG
012137  773657      3530            LAW     INTCAT-1
012140  040012      3540            DAC     12              SET THE POINTER TO THE INITIALIZATION CATALOG
012141  777700      3550            LAW     -CLEN
012142  053650      3560            DAC     TEMP1           SET THE CATALOG LENGTH
012143  220012      3570    INT05   LAC     12,X
012144  060010      3580            DAC     10,X            COPY THE NEXT CATALOG ENTRY
012145  453650      3590            ISZ     TEMP1           COUNT IT; SKIP IF DONE
012146  612143      3600            JMP     INT05           ELSE LOOP
012147  214400      3610            LAC     (SSWP)
012150  112603      3620            JMS     WRITE           DONE -- COPY THE CORRECTED SWAPPER BACK OUT
```

I MAIN PROGRAM

```
                    3640    *
                    3650    *
                    3660    *       INITIALIZE THE RESIDENT PROGRAM -- FIRST READ IT INTO CORE
                    3670    *
        012151      3680    INT50   ...
012151  214401      3690            LAC     ($RES)
012152  112326      3700            JMS     LCATL           LOOK UP THE RESIDENT PROGRAM IN THE LIBRARY CATALOG
012153  740040      3710            HLT                     FATAL ERROR IF CAN'T FIND IT
012154  440011      3720            INX     $CATX           MOVE THE POINTER TO THE DEVICE ADDRESS FOR THE PROGRAM
012155  113365      3730            JMS     CSRCOVR         SET UP THE HARDWARE ERROR RECOVERY
012156  200011      3740            LAC     $CATX           LOAD THE POINTER TO THE PROGRAM'S PARAMETERS
012157  117500      3750            JMS     $TAPIN          AND READ IN THE RESIDENT PROGRAM
                    3760    *
                    3770    *       NOW FILL IN THE RESIDENT CATALOG
012160  760043      3780            LAW     $CSWP-1
012161  040010      3790            DAC     10              SET A POINTER TO THE RESIDENT CATALOG
012162  214400      3800            LAC     ($SWP)
012163  112623      3810            JMS     FILL            ENTER THE SWAPPER IN THE RESIDENT CATALOG
012164  214402      3820            LAC     ($MP1)
012165  112623      3830            JMS     FILL            ENTER MEMORY PROTECTION #1 IN THE RESIDENT CATALOG
012166  214403      3840            LAC     ($MP2)
012167  112623      3850            JMS     FILL            ENTER MEMORY PROTECTION #2 IN THE RESIDENT CATALOG
012170  214404      3860            LAC     ($SPL)
012171  112623      3870            JMS     FILL            ENTER THE SPECIAL IOT HANDLER IN THE RESIDENT CATALOG
                    3880    *
                    3890    *
                    3900    *       NOW INITIALIZE THE JOB TABLES IN SCRATCH STORAGE ON THE SYSTEM DISK.
                    3910    *            $UCORE -- PHYSICAL DISK ADDRESS OF THE USER CORE IMAGE
                    3920    *            $UDISK -- PHYSICAL DISK ADDRESS OF THE 'USER DISK'
                    3930    *            $NUMBR -- USER NUMBER (= POINTER TO HIS TELETYPE BUFFER)
                    3940    *            $OVER  -- $MP1 = STANDARD USER OVERLAY
                    3950    *            $TYPE  -- 0 = USER TYPE PROGRAM
                    3960    *            $SYSNM -- PHANTOM PROGRAM'S OWN NAME
                    3970    *            $PURNAM -- 0 = NO PURE CODE
                    3980    *            $RSTRT -- <LAW BOUNDARY> ASSURES A LEGAL START ADDRESS WITH MEMORY PROTECTION ON.
                    3990    *
                    4000    *       FIRST ZERO A FULL BUFFER LENGTH OF CORE, PLUS A TABLE LENGTH,
                    4010    *       TO BE USED IN INTIALIZING THE DISK SCRATCH AREA.
                    4020    *
        012172      4030    INT55   ...
012172  767700      4040            LAW     -BMAX-$JTLEN
012173  053650      4050            DAC     TEMP1           SET THE LENGTH TO BE ZEROED
012174  761677      4060    INT56   LAW     $JTSTRT-1
012175  040010      4070            DAC     10              SET A POINTER TO THE TABLE
012176  160010      4080    INT60   DZM     10,X            ZERO THE NEXT LOCATION
012177  453650      4090            ISZ     TEMP1           COUNT THE AMOUNT ZEROED
012200  612176      4100            JMP     INT60           LOOP
                    4110    *
                    4120    *
                    4130    *       ZERO THE SCRATCH DISK AREA
                    4140    *
012201  214405      4150            LAC     (-SCRSTR)
```

```
                I                       MAIN PROGRAM

012202  053651  4160            DAC     TEMP2           SET THE AMOUNT OF DISK TO ZERO
012203  214406  4170            LAC     (SCRSTR)
012204  707024  4180            DLAL                    SET THE START OF THE DISK AREA TO ZERO
    012205      4190    INT70   ...
012205  213651  4200            LAC     TEMP2           LOAD THE REMAINING LENGTH TO BE ZEROED
012206  741200  4210            SNA                     SKIP IF THERE IS ANY
012207  612231  4220            JMP     INT80           ELSE CONTINUE THE INITIALIZATION PROCESS
012210  352172  4230            TAD     INT55           SUBTRACT THE AMOUNT OF ZEROED CORE
012211  740100  4240            SMA                     SKIP IF LESS THAN A FULL COPY REMAINS
012212  612220  4250            JMP     INT72           ELSE DO THE FULL COPY
012213  777777  4260            LAW     -1              PREPARE TO NEGATE THE LENGTH TO COPY
012214  353651  4270            TAD     TEMP2           LOAD THE REMAINING LENGTH
012215  153651  4280            DZM     TEMP2           FLAG THE ZERO-COPYING DONE
012216  740001  4290            CMA                     NEGATE IT
012217  612222  4300            JMP     INT74
                4310
    012220      4320    INT72   ...
012220  053651  4330            DAC     TEMP2           SET THE REMAINING AMOUNT TO ZERO
012221  212172  4340            LAC     INT55           LOAD THE BUFFER LENGTH
012222  040036  4350    INT74   DAC     $DKWC           SET THE LENGTH OF THE COPY
012223  212174  4360            LAC     INT56
012224  040037  4370            DAC     $DKCA           SET THE CORE ADDRESS OF THE COPY
012225  772205  4380            LAW     INT70
012226  040654  4390            DAC     $DO             SET THE RESTART
012227  214377  4400            LAC     ($DKWRT)        LOAD THE DISK WRITE COMMAND
012230  600672  4410            JMP     $DKOVR          DO THE COPY
                4420    *
                4430    *       NEXT FILL IN THE PHANTOM JOB TABLE, SINCE IT IS THE SAME FOR ALL USERS
                4440    *
012231  200045  4450    INT80   LAC     $CMP1
012232  041773  4460            DAC     $OVER
012233  762000  4470            LAW     BOUNDARY
012234  041776  4480            DAC     $RSTRT
                4490    *
                4500    *       MACRO TO FINISH SETTING UP A USER'S JOB TABLE AND TO INITIALIZE
                4510    *       HIS DISK STORAGE AREA
                4520    *
                4530    INTT    .DEFIN
                4540            .PMC    SAVE,ON
                4550            LAC     UC#1+1
                4560            DAC     $UCORE
                4570            LAC     (UDK#1          )
                4580            DAC     $UDISK
                4590            LAC     ($US#1          )
                4600            DAC     $NUMBR
                4610            DAC     $NAME
                4620            LAW     TAB#1
                4630            JMS     UINIT
                4640            .PMC    RESTORE
                4650            .ENDM
                4660    *
                4670    *       SET UP THE JOB TABLE FOR USER #1 AND INITIALIZE HIS DISK SPACE
```

I MAIN PROGRAM

```
                    4680    *
    012235          4690            INTT    1
012235  213675                      LAC     UC1+1
012236  041766                      DAC     $UCORE
012237  214407                      LAC     (UDK1)
012240  041767                      DAC     $UDISK
012241  214410                      LAC     ($US1)
012242  041771                      DAC     $NUMBR
012243  041772                      DAC     $NAME
012244  773704                      LAW     TAB1
012245  112404                      JMS     UINIT
                    4700    *
                    4710    *       SET UP THE JOB TABLE FOR USER #2 AND INITIALIZE HIS DISK SPACE
                    4720    *
    012246          4730            INTT    2
012246  213711                      LAC     UC2+1
012247  041766                      DAC     $UCORE
012250  214411                      LAC     (UDK2)
012251  041767                      DAC     $UDISK
012252  214412                      LAC     ($US2)
012253  041771                      DAC     $NUMBR
012254  041772                      DAC     $NAME
012255  773720                      LAW     TAB2
012256  112404                      JMS     UINIT
                    4740    *
                    4750    *       SET UP THE JOB TABLE FOR USER #0 AND INITIALIZE HIS DISK SPACE
                    4760    *
    012257          4770            INTT    0
012257  213661                      LAC     UC0+1
012260  041766                      DAC     $UCORE
012261  214406                      LAC     (UDK0)
012262  041767                      DAC     $UDISK
012263  214413                      LAC     ($US0)
012264  041771                      DAC     $NUMBR
012265  041772                      DAC     $NAME
012266  773670                      LAW     TAB0
012267  112404                      JMS     UINIT
                    4780
012270  140266          4790            DZM     $DKLOK          FLAG THE DISK FREE
                    4800    *
                    4810    *
                    4820    *       SYSTEM INITIALIZATION HAS BEEN SUCCESSFUL, SO TELL THE OPERATOR
                    4830    *
    012271          4840            MESS    <SYSTEM INITIALIZATION COMPLETED[]>,33.
012271  700312                      KRB
    012272                          MESSR   <SYSTEM INITIALIZATION COMPLETED[]>,33.
012272  777735                      LAW     -33,-2
012273  113156                      JMS     T$SIXOT
                    4850    *
                    4860    *       NOW DO LAST MINUTE HARDWARE TIDYING UP
                    4870    *
012310  700312          4880            KRB                     KILL ANY MISCELLANEOUS TELETYPE INPUT
```

I MAIN PROGRAM

```
012311  704112  4890            KRBLT1
012312  704132  4900            KRBLT2
012313  701702  4910            MPCV
012314  777730  4920            LAW     -CLKMAX
012315  040007  4930            DAC     7               SET THE CLOCK
012316  700046  4940            IONICLON                AND TURN IT ON
                4950    *
                4960    *       SET UP THE MONITOR FOR ALL TELETYPES
                4970    *
012317  214413  4980            LAC     (SCTBIN-2)
012320  040076  4990            DAC     SCTBIN-2        SET UP THE CONSOLE TELETYPE
012321  214410  5000            LAC     (SL1BIN-2)
012322  040125  5010            DAC     SL1BIN-2        SET UP LT#1
012323  214412  5020            LAC     (SL2BIN-2)
012324  040154  5030            DAC     SL2BIN-2        SET Up LT#2
                5040    *
                5050    *       FAKE A MONITOR CALL BY THE CONSOLE TELETYPE
                5060    *
012325  601002  5070            JMP     $SWMTR          GET MONITOR FOR THE CONSOLE TELETYPE
```

I MISCELLANEOUS SUBROUTINES

```
                    5090
                    5100    *
                    5110    *
                    5120    *       LCATL OPERATES ON THE LIBRARY DEVICE CATALOG (LOCATED AT LCAT)
                    5130    *       IN ALL OTHER RESPECTS IT IS IDENTICAL TO CSCATL
                    5140    *
      012326        5150    LCATL   ENTER
012326  740040                      XX
012327  052634    5160              DAC     TSWORDB         PASS THE FILENAME TO CSCATL
012330  212326    5170              LAC     LCATL
012331  053440    5180              DAC     CSCATL          PASS THE RETURN TO CSCATL
012332  773777    5190              LAW     LCAT+3
012333  040011    5200              DAC     SCATX           PASS  A POINTER TO THE FIRST FCB IN THE LIBRARY CATALOG
012334  213776    5210              LAC     LCAT+2
012335  613445    5220              JMP     CSCATLL-1       NOW CSCATL IS SET TO OPERATE ON THE CORRECT  CATALOG
                    5230    *
                    5240    *
                    5250    *       UNSAVE UNSAVES FROM THE MAIN CATALOG THE FILES INDEXED BY AUTO-INDEX
                    5260    *       REGISTER 10. IT TERMINATES UPON INDEXING TO A ZERO FILENAME
                    5270    *
      012336        5280    UNSAVE  ENTER
012336  740040                      XX
012337  220010    5290      UNS1    LAC     10,X            LOAD THE NEXT FILENAME
012340  741200    5300              SNA
012341  632336    5310              RET     UNSAVE,X        ZERO FILENAME -- EXIT
012342  113440    5320              JMS     CSCATL          ELSE LOCATE THE FILE ENTRY IN THE CATALOG
012343  612337    5330              JMP     UNS1            FILE NOT SAVED -- DO THE NEXT ONE
012344  200011    5340              LAC     SCATX
012345  053650    5350              DAC     TEMP1           SET A POINTER TO THE FILENAME
012346  173650    5360              DZM     TEMP1,X         ZERO THE FILENAME TO UNSAVE THE FILE
012347  612337    5370              JMP     UNS1            LOOP
                    5380    *
                    5390    *
                    5400    *       SAVE SAVES INTO THE MAIN CATALOG THE FILES INDEXED BY AUTO-INDEX
                    5410    *       REGISTER 10. IT TERMINATES UPON INDEXING TO A ZERO FILENAME
                    5420    *
      012350        5430    SAVE    ENTER
012350  740040                      XX
012351  220010    5440      SAV1    LAC     10,X            LOAD THE NEXT FILENAME
012352  741200    5450              SNA
012353  632350    5460              RET     SAVE,X          ZERO FILENAME -- EXIT
012354  112326    5470              JMS     LCATL           LOCATE IT IN THE LIBRARY CATALOG
012355  740040    5480              HLT                     FATAL ERROR IF THE FILE CANNOT BE FOUND IN THE LIBRARY CATALOG
012356  200011    5490              LAC     SCATX
012357  040012    5500              DAC     SCMDX           SET A POINTER TO THE FILE IN THE LIBRARY CATALOG
012360  220012    5510              LAC     SCMDX,X
012361  052421    5520              DAC     INDA            SET THE INPUT DEVICE ADDRESS
      012362        5530            WORD1                   RECOVER THE FILENAME
012362  212634                      LAC     TSWORDB
012363  113555    5540              JMS     CSSAVE          SAVE THE FILE, IF POSSIBLE
012364  740040    5550              HLT                     FATAL ERROR IF THE FILE IS ALREADY SAVED -- SHOULD BE IMPOSSIBLE
012365  200011    5560              LAC     SCATX
```

```
          I                                MISCELLANEOUS SUBROUTINES

012366 040013  5570          DAC     13              SAVE A POINTER TO THE FILE'S SYSTEM DISK ADDRESS
012367 440011  5580          INX     $CATX           BYPASS THE DEVICE ADDRESS FOR NOW
012370 220012  5590          LAC     $CMDX,X
012371 060011  5600          DAC     $CATX,X         TRANSFER THE FILE'S CORE ADDRESS
012372 220012  5610          LAC     $CMDX,X
012373 060011  5620          DAC     $CATX,X         TRANSFER THE FILE'S LENGTH
012374 052423  5630          DAC     LEN             SET THE LENGTH FOR THE COPY ROUTINE
012375 113604  5640          JMS     CSALC           ALLOCATE SPACE FOR THE FILE, IF POSSIBLE
012376 060013  5650          DAC     13,X            SET THE SYSTEM DEVICE ADDRESS
012377 052422  5660          DAC     OUTDA           SET THE OUTPUT DEVICE ADDRESS FOR THE COPY
012400 220012  5670          LAC     $CMDX,X
012401 060011  5680          DAC     $CATX,X         COPY THE FILE'S TRANSFER CARD
012402 112424  5690          JMS     COPY            COPY THE FILE FROM THE LIBRARY DEVICE TO THE SYSTEM DEVICE
012403 612351  5700          JMP     SAV1            LOOP
               5710  *
               5720  *
               5730  *       UINIT IS A SUBROUTINE TO COPY THE USER'S INITIALIZED JOB TABLE OUT
               5740  *
   012404      5750  UINIT   ENTER
012404 740040                XX
012405 040010  5760          DAC     10              SET THE PARAMETER POINTER
012406 220010  5770          LAC     10,X
012407 053650  5780          DAC     TEMP1           SET THE PHYSICAL DISK LOCATION
012410 220010  5790          LAC     10,X
012411 053651  5800          DAC     TEMP1+1         SET THE CORE ADDRESS -1
012412 220010  5810          LAC     10,X
012413 053652  5820          DAC     TEMP1+2         SET THE TWO'S COMPLEMENT LENGTH
012414 214377  5830          LAC     (SDKWRT)
012415 053653  5840          DAC     TEMP1+3
012416 773647  5850          LAW     TEMP1-1
012417 100654  5860          JMS     $DO             DO THE WRITE
012420 632404  5870          RET     UINIT,X
               5880  *
               5890  *
               5900  *       ALSO USE UINIT TO ZERO ON THE DISK
               5910  *            USER CORE STORAGE
               5920  *            PHANTOM CORE STORAGE
               5930  *            USER "PHYSICAL DISK" STORAGE
               5940  *
               5950  *
```

I MISCELLANEOUS SUBROUTINES

```
                    5970    *
                    5980    *       COPY SUBROUTINE
                    5990    *
                    6000    *       COPIES FROM DEVICE INDA TO DEVICE OUTDA FOR LEN WORDS
                    6010    *
    000100          6020    BUF     .EQU    100             START OF COPY BUFFER
    010000          6030    BMAX    .EQU    10000           ALLOW A 4K COPY BUFFER
                    6040
012421  000000      6050    INDA    .DSA
012422  000000      6060    OUTDA   .DSA
012423  000000      6070    LEN     .DSA
                    6080
012424  740040      6090    COPY    XX
012425  212423      6100    COPL    LAC     LEN             GET LENGTH REMAINING
012426  741200      6110            SNA
012427  632424      6120            RET     COPY,X          RETURN IF DONE
012430  354414      6130            TAD     (-BMAX)         SUBTRACT AMOUNT WE CAN COPY IN (1) OPERATION
012431  741100      6140            SPA
012432  612436      6150            JMP     COPL2
012433  052423      6160            DAC     LEN             RESTORE LENGTH REMAINING
012434  214415      6170            LAC     (BMAX)          GET AMOUNT FOR CURRENT COPY
012435  612440      6180            JMP     COPL4           SKIP THE OTHER BRANCH
012436  212423      6190    COPL2   LAC     LEN             GET LENGTH FOR COPY
012437  152423      6200            DZM     LEN             NONE REMAINING
012440  053656      6210    COPL4   DAC     TPARAM+2        SAVE NEW LENGTH TO COPY
012441  212421      6220            LAC     INDA            GET INPUT DA
012442  053654      6230            DAC     TPARAM          SAVE IT
012443  552422      6240            SAD     OUTDA           CHECK FOR NOTHINGISH COPIES
012444  632424      6250            RET     COPY,X
012445  354416      6260            TAD     (BMAX/SBLKLEN)  COMPUTE AMOUNT TO COPY IN BLOCKS
012446  052421      6270            DAC     INDA            RESTORE FOR NEXT COPY
012447  113365      6280            JMS     CSRCOVR         SET UP THE ERROR RECOVERY
012450  773654      6290            LAW     TPARAM          GET PARAMETERS FOR READ
012451  117500      6300            JMS     STAPIN          COPY IN
012452  212422      6310            LAC     OUTDA           GET OUTPUT DA
012453  741200      6320            SNA
012454  632424      6330            RET     COPY,X          RETURN IF INPUT ONLY
012455  053654      6340            DAC     TPARAM          SAVE IT
012456  354416      6350            TAD     (BMAX/SBLKLEN)
012457  052422      6360            DAC     OUTDA           SET THE UPDATED OUTPUT DEVICE ADDRESS FOR NEXT TIME
012460  113365      6370            JMS     CSRCOVR         SET UP THE HARDWARE ERROR RECOVERY
012461  773654      6380            LAW     TPARAM          GET PARAMETERS
012462  117502      6390            JMS     STAPOT          OUTPUT IT
012463  612425      6400            JMP     COPL            LOOP
```

I MISCELLANEOUS SUBROUTINES

```
                    6420    *
                    6430    *
                    6440    *       CORCPY COPIES THE FILE ENTRIES FOR THE FILES INDEXED BY AUTO-INDEX
                    6450    *       REGISTER 10 FROM THE MAIN CATALOG TO THE INITIALIZATION CATALOG ($NTCAT),
                    6460    *       INDEXED BY AUTO-INDEX REGISTER 12, AT THE SAME TIME IT CONVERTS LOGICAL
                    6470    *       DISK BLOCK ADDRESSES TO PHYSICAL DISK ADDRESSES.
                    6480    *       IT TERMINATES UPON INDEXING TO A ZERO FILENAME.
                    6490    *
    012464          6500    CORCPY  ENTER
012464  740040                      XX
012465  220010      6510    COR2    LAC     10,X            LOAD THE NEXT FILENAME
012466  741200      6520            SNA
012467  632464      6530            RET     CORCPY,X        ZERO FILENAME -- EXIT
012470  112472      6540            JMS     CRCP            COPY THE CATALOG ENTRY, ADJUSTING DISK ADDRESS AND CORE ADDRESS FORMATS
012471  612465      6550            JMP     COR2            LOOP
                    6560    *
                    6570    *       SUBROUTINE TO COPY A FILES CATALOG ENTRY FROM THE MAIN CATALOG TO
                    6580    *       INTCAT, CONVERTING THE FORMAT OF CORE ADDRESSES AND DISK ADDRESSES.
                    6590    *
    012472          6600    CRCP    ENTER
012472  740040                      XX
012473  113440      6610            JMS     CSCATL          FIND THE FILE IN THE SYSTEM DEVICE CATALOG
012474  740040      6620            HLT                     FATAL ERROR IF THE FILE IS NOT SAVED
    012475          6630            WORD1                   RECOVER THE NAME
012475  212634                      LAC     TSWORDB
012476  060012      6640            DAC     12,X            SET THE FILENAME
012477  220011      6650            LAC     SCATX,X
012500  614417      6660            AND     (SBLKMSK)
012501  354420      6670            TAD     (SSYSBAS)
012502  660710      6680            ALSS    8.
012503  060012      6690            DAC     12,X            SET THE PHYSICAL DISK ADDRESS
012504  777777      6700            LAW     -1
012505  360011      6710            TAD     SCATX,X
012506  060012      6720            DAC     12,X            SET THE CORE ADDRESS -1
012507  777777      6730            LAW     -1
012510  360011      6740            TAD     SCATX,X
012511  740001      6750            CMA
012512  060012      6760            DAC     12,X            SET THE (TWO'S COMPLEMENT) LENGTH
012513  632472      6770            RET     CRCP,X
                    6780    *
                    6790    *
                    6800    *       PHCRCP DOES THE SAME THING FOR PHANTOM PROGRAMS THAT CORCPY DOES
                    6810    *       FOR OTHER PROGRAMS. IN ADDITION, PHCRCP ALSO SETS UP A CATALOG ENTRY FOR
                    6820    *       THE PURE CODE PORTION OF THE PHANTOM.
                    6830    *
    012514          6840    PHCRCP  ENTER
012514  740040                      XX
012515  220010      6850    PHCRC1  LAC     10,X            LOAD THE NEXT FILENAME
012516  741200      6860            SNA
012517  632514      6870            RET     PHCRCP,X        RETURN WHEN DONE
012520  112472      6880            JMS     CRCP            SET UP THE PROGRAM'S CATALOG ENTRY
                    6890    *
```

I MISCELLANEOUS SUBROUTINES

```
                    6900    *       SET UP THE ENTRIES FOR THE PURE-CODE SECTION OF THE PHANTOM PROGRAM
                    6910    *
012521  777775      6920            LAW     -3
012522  340011      6930            TAD     $CATX           BACK UP THE POINTER SO IT WILL SCAN THE SAME CATALOG ENTRY AGAIN
012523  040011      6940            DAC     $CATX
    012524          6950    WORD1                           RECOVER THE FILENAME
012524  212634                      LAC     T$WORD8
012525  514421      6960            AND     (7777)          GET RID OF THE FIRST CHARACTER
012526  354422      6970            TAD     (600000)        MAKE THE FIRST CHARACTER A 'P' (FOR 'PURE')
012527  060012      6980            DAC     12,X            SET THE PURE CODE FILENAME
012530  220011      6990            LAC     $CATX,X
012531  514417      7000            AND     ($BLKMSK)
012532  354420      7010            TAD     ($SYSBAS)
012533  660710      7020            ALSS    8.              AC NO INDICATES THE PHYSICAL DISK ADDRESS OF THE ENTIRE PHANTOM
012534  354423      7030            TAD     ($PURSTR-$IMPSTR)  MOVE IT TO THE START OF PURE CODE
012535  060012      7040            DAC     12,X            SET THE PHYSICAL DISK ADDRESS OF THE START OF PURE CODE
012536  763677      7050            LAW     $PURSTR-1       LOAD THE CORE ADDRESS OF THE PURE CODE
012537  440011      7060            INX     $CATX           MOVE THE POINTER OVER THE OTHER CORE ADDRESS
012540  060012      7070            DAC     12,X            SET THE CORE ADDRESS -1 OF THE PURE CODE
012541  777777      7080            LAW     -1
012542  360011      7090            TAD     $CATX,X
012543  740001      7100            CMA                     AC IS (TWO'S COMPLEMENT) LENGTH OF THE ENTIRE PHANTOM
012544  354423      7110            TAD     ($PURSTR-$IMPSTR)
012545  060012      7120            DAC     12,X            SET THE (TWO'S COMPLEMENT) LENGTH OF THE PURE CODE
012546  612515      7130            JMP     PHCRC1          LOOP
                    7140    *
                    7150    *
                    7160    *       OVCRCP DOES THE SAME THING FOR OVERLAY PROGRAMS THAT CORCPY DOES
                    7170    *       FOR USER PROGRAMS, THEN IT GOES THROUGH THE OVERLAY PROGRAMS AND
                    7180    *       SUBSTITUTES FOR THEIR NAME THEIR PHYSICAL DISK ADDRESS, WHICH
                    7190    *       BECOMES THEIR NAME AS FAR AS THE EXECUTIVE IS CONCERNED.
                    7200    *
    012547          7210    OVCRCP  ENTER
012547  740040                      XX
012550  200012      7220            LAC     12              LOAD THE INTCAT POINTER
012551  040015      7230            DAC     15              SAVE IT FOR LATER OPERATIONS
012552  040016      7240            DAC     16
012553  440016      7250            INX     16              GET A POINTER TO THE DISK ADDRESSES
012554  112464      7260            JMS     CORCPY          SET UP THE OVERLAY PROGRAMS IN THE STANDARD MANNER
                    7270
012555  777774      7280            LAW     OFILES-EFILES+1  LOAD A COUNT OF THE NUMBER OF OVERLAY FILES
012556  040017      7290            DAC     17              SET IT
012557  220016      7300    OV1     LAC     16,X            LOAD THE NEXT DISK ADDRESS
012560  060015      7310            DAC     15,X            AND SET IT AS THE NEW OVERLAY NAME
012561  440017      7320            ISZ     17              COUNT THE FILE
012562  741000      7330            SKP                     NOT YET DONE
012563  632547      7340            RET     OVCRCP,X        DONE -- EXIT
012564  200016      7350            LAC     16
012565  354424      7360            TAD     (3)
012566  040016      7370            DAC     16              UPDATE THE ADDRESS POINTER
012567  200015      7380            LAC     15
012570  354424      7390            TAD     (3)
```

```
                I                       MISCELLANEOUS SUBROUTINES

012571  040015  7400            DAC     15              UPDATE THE NAMES POINTER
012572  012557  7410            JMP     OV1             DO THE NEXT FILE
                7420    *
                7430    *
                7440    *       READ READS INTO CORE THE FILE WHOSE NAME IS PASSED IN THE AC
                7450    *
    012573      7460    READ    ENTER
012573  740040                  XX
012574  113440  7470            JMS     CSCATL          LOOK UP THE FILE
012575  740040  7480            HLT                     FATAL ERROR IF THE FILE CANNOT BE FOUND
012576  440011  7490            INX     SCATX           MOVE THE POINTER TO THE DEVICE ADDRESS
012577  113365  7500            JMS     CSRCOVR         SET UP HARDWARE ERROR RECOVERY
012600  200011  7510            LAC     SCATX           LOAD THE POINTER TO THE PARAMETERS
012601  117500  7520            JMS     STAPIN          READ THE FILE
012602  632573  7530            RET     READ,X
                7540    *
                7550    *
                7560    *       WRITE WRITES FROM CORE THE FILE WHOSE NAME IS PASSED IN THE AC
                7570    *
    012603      7580    WRITE   ENTER
012603  740040                  XX
012604  113440  7590            JMS     CSCATL          LOOK UP THE FILE
012605  740040  7600            HLT                     FATAL ERROR IF THE FILE CANNOT BE FOUND
012606  440011  7610            INX     SCATX           MOVE THE POINTER TO THE DEVICE ADDRESS
012607  113365  7620            JMS     CSRCOVR         SET UP THE HARDWARE ERROR RECOVERY
012610  200011  7630            LAC     SCATX           LOAD THE POINTER TO THE PARAMETERS
012611  117502  7640            JMS     STAPOT          WRITE THE FILE
012612  632603  7650            RET     WRITE,X
                7660    *
                7670    *
                7680    *       SETUP COPIES THREE SCATLOG ENTRIES INDEXED BY AUTO-INDEX REGISTER 12
                7690    *       INTO THE LOCATIONS REFERENCED BY AUTO-INDEX REGISTER 10.
                7700    *
    012613      7710    SETUP   ENTER
012613  740040                  XX
012614  220012  7720            LAC     12,X
012615  060010  7730            DAC     10,X            COPY THE FILE'S DEVICE ADDRESS
012616  220012  7740            LAC     12,X
012617  060010  7750            DAC     10,X            COPY THE FILE'S CORE ADDRESS
012620  220012  7760            LAC     12,X
012621  060010  7770            DAC     10,X            COPY THE FILE'S LENGTH
012622  632613  7780            RET     SETUP,X
                7790    *
                7800    *
                7810    *       FIND THE PHYSICAL DISK ADDRESS OF THE FILE WHOSE NAME WAS PASSED
                7820    *       IN THE AC AND ENTER IT IN THE RESIDENT CATALOG.
                7830    *
    012623      7840    FILL    ENTER
012623  740040                  XX
012624  113440  7850            JMS     CSCATL          LOOK UP THE SWAPPER PROGRAM IN THE SYSTEM DISK CATALOG
012625  740040  7860            HLT                     FATAL ERROR IF THE PROGRAM CAN'T BE FOUND
012626  220011  7870            LAC     SCATX,X         LOAD THE SYSTEM DISK LOGICAL BLOCK NUMBER
```

I MISCELLANEOUS SUBROUTINES

```
012627  514417  7880          AND     ($BLKMSK)       RECOVER JUST THE BLOCK NUMBER
012630  354420  7890          TAD     ($SYSEAS)       CONVERT TO A PHYSICAL BLOCK NUMBER
012631  660710  7900          ALSS    8.              MAKE INTO A PHYSICAL DISK ADDRESS
012632  060010  7910          DAC     10,X            AND ENTER IT IN THE RESIDENT CATALOG
012633  632623  7920          RET     FILL,>
                7930
                7940          .INSRT  :DLIBFARY:PDP9LIB:TTYNON
```

T MTSS-PDP9 NON-INTERRUPTING TELETYPE HANDLER

```
140                 .HEAD   T
150       *
160       *
170       *         PROGRAMMED BY ROBERT W. BLEAN
180       *
190       *
200       *         LAST REVISED 24 MARCH 1972
210       *
220       *
230       *         THIS HANDLER PERMITS NON-INTERRUPT DRIVEN INPUT FROM AND OUTPUT
240       *         TO THE CONSOLE TELETYPE ON THE PDP-9 COMPUTER.
250       *
260       *         THIS HANDLER ALTERS THE  AC, AND MQ, IT DOES NOT ALTER ANY CORE
270       *         MEMORY OUTSIDE OF ITSELF. IN PARTICULAR IT DOES NOT ALTER ANY AUTO-INDEX REGISTER.
280       *
290       *         DATA FORMATS:
300       *
310       *              1) OCTAL
320       *
330       *              2) SIXBIT -- SIXBIT IS 8-BIT ASCII MINUS 240. THIS MAPS THE PRINTING
340       *                 CHARACTERS ONTO THE SET 0-77. ASCII VALUE 333 ([) IS USED FOR
350       *                 CARRIAGE RETURN AND 335 (]) IS USED FOR LINEFEED. NOTE THAT NEITHER
360       *                 333, 335, NOR ANY CONTROL CHARACTERS CAN BE RECOGNIZED IN SIXBIT.
370       *
380       *              3) ASCII -- ONE ASCII CHARACTER IS STORED PER WORD, LINE INPUT
390       *                 IS STORED IN THIS FORMAT, SINCE THERE IS ONLY ONE LINE-BUFFER
400       *                 THE EXTRA BUFFER LENGTH WASTES LESS SPACE THAN WOULD THE HANDLING
410       *                 ROUTINES NECESSARY FOR OTHER FORMS OF PACKING CHARACTERS.
```

T (MTSS TELETYPE HANDLER) STORAGE AREA

```
                    450
                    460
     012634         470     WORDB   ,BLOCK  2              ROOM TO ACCUMULATE TWO VALID WORDS
     000120         480     STD     ,EQU    80.            STANDARD IS AN 80-CHARACTER LINE BUFFER
     012636         490     BUFFR   ,BLOCK  STD
                    500     *
                    510     *
                    520     *       VARIABLES
                    530     *
012756  012755      540     BEND    ,-1                    END OF THE CHARACTER BUFFER
012757  000000      550     BPTR    ,DSA                   POINTER TO CURRENTLY ACTIVE WORD IN LINE BUFFER
012760  000000      560     T1      ,DSA                   TEMPORARY VARIABLE
012761  000000      570     T2      ,DSA                   TEMPORARY VARIABLE
012762  000000      580     CHAR    ,DSA                   STORES LATEST CHARACTER FROM FGET
012763  000000      590     DLMTR   ,DSA                   STORES LATEST DELIMITER THROUGH CHRID
012764  000000      600     COUNT   ,DSA
```

T (MTSS TELETYPE HANDLER) LINE BUFFER INPUT

```
                    640
                    650
                    660     *
                    670     *       THE PROGRAM IS PROTECTED AGAINST OVERFLOW OR UNDERFLOW OF THE LINE
                    680     *       BUFFER. UNDERFLOW (EXCESS DELETIONS) IS IGNORED, AND OVERFLOW CHARACTERS
                    690     *       ARE LOST, EXCEPT FOR THE LAST CHARACTER TYPED.
                    700     *
                    710
     012765         720             ENTER   INLIN          SUBROUTINE TO READ IN AND BUFFER A LINE FROM THE TELETYPE
012765  740040              INLIN   XX
012766  700312      730             KRB                     ONCE, ON ENTRANCE, CLEAN UP ANY PRIOR INPUT
012767  214425      740     INL     LAC     (BUFFR-1)       LOAD A POINTER TO START OF THE BUFFER MINUS ONE
012770  052757      750             DAC     BPTR            INITIALIZE THE BUFFER POINTER
012771  152764      760             DZM     COUNT           INITIALIZE THE WORD FETCHED COUNT
012772  152763      770             DZM     DLMTR           INITIALIZE THE LAST DELIMITER STORAGE
012773  700313      780     IN1     KSF!KRB                 GET THE NEXT INPUT CHARACTER
012774  612773      790             JMP     .-1
012775  554426      800             SAD     ($BKARR)
012776  613020      810             JMP     1CHAR           DELETE ONE CHARACTER IF IT WAS A BACKARROW
012777  554427      820             SAD     ($CONTX)
013000  613016      830             JMP     1LINE           DELETE THE ENTIRE LINE IF IT WAS A CONTROL X
013001  652000      840     IN4     LMQ                     SAVE THE CHARACTER
013002  212757      850             LAC     BPTR            LOAD THE CURRENT BUFFER POINTER
013003  552756      860             SAD     BEND            SKIP IF NO OVERFLOW
013004  741000      870             SKP                     AVOID DAMAGE DUE TO OVERFLOW
013005  452757      880             ISZ     BPTR            ADVANCE THE POINTER -- IT IS STILL WITHIN THE BUFFER
013006  641002      890             LACQ                    RELOAD THE CHARACTER
013007  072757      900             DAC     BPTR,X          AND PUT IT IN THE BUFFER
013010  554430      910             SAD     ($CR)
013011  741000      920             SKP                     EXIT WHEN A CARRIAGE RETURN IS FOUND
013012  612773      930             JMP     IN1             ELSE GET THE NEXT CHARACTER
013013  772635      940             LAW     BUFFR-1
013014  052757      950             DAC     BPTR            RESET THE BUFFER POINTER AT THE END OF THE LINE
013015  632765      960             JMP     INLIN,X         AND RETURN TO THE CALLER
                    970
013016  113270      980     1LINE   JMS     CRLF            PRINT THE RESPONSE TO A LINE-DELETE
013017  612767      990             JMP     INL             REREAD THE LINE
013020  212757     1000     1CHAR   LAC     BPTR            LOAD THE BUFFER POINTER
013021  552767     1010             SAD     INL             SKIP IF NO UNDERFLOW
013022  612773     1020             JMP     IN1             ELSE IGNORE THE COMMAND
013023  354431     1030             TAD     (-1)            DECREMENT THE BUFFER POINTER
013024  052757     1040             DAC     BPTR            AND SAVE IT
013025  612773     1050             JMP     IN1             GET THE NEXT CHARACTER
```

T (MTSS TELETYPE HANDLER) OCTAL WORD INPUT/OUTPUT

```
                    1070
                    1080    *
                    1090    *       OPERATION RETURN L    AC      MQ    MEANING
                    1100    *       --------- ------ - ------ ------ -------------------------------
                    1110    *       INPUT       +1   0   X      X     FORMAT ERROR DISCOVERED
                    1120    *                   +1   1 DELIM    X     FIRST NON-BLANK CHARACTER IS A DELIMITER
                    1130    *                   +2   1 OCTAL  DELIM   SUCCESSFUL READ OF AN OCTAL NUMBER
                    1140    *       OUTPUT      +1   X   X      X     SUCCESSFUL WRITE OF AN OCTAL NUMBER
                    1150    *
                    1160
    013026          1170            ENTER   NUMIN
013026  740040              NUMIN   XX
013027  152761  1180              DZM     T2              INITIALIZE THE DECIMAL-DIGIT-RECEIVED FLAG
013030  113211  1190              JMS     INTIN           INITIALIZE THE INPUT STRING, ETC
013031  633026  1200              JMP     NUMIN,X         RETURN +1 FOR DELIMITER IS FIRST NON-BLANK CHARACTER
013032  113204  1210      NUM20   JMS     FGET            GET THE NEXT CHARACTER
013033  113230  1220              JMS     CHRID           IDENTIFY IT
013034  613056  1230              JMP     NUM26           IT IS A DELIMITER, SO EXIT
013035  633026  1240              JMP     NUMIN,X         IT IS A LETTER, SO EXIT +1 FOR A FORMAT ERROR
013036  741400  1250              SZL                     SKIP IF THE CHARACTER IS AN OCTAL DIGIT
013037  452761  1260              ISZ     T2              ELSE BE SURE THE DECIMAL-DIGIT-RECEIVED FLAG IS SET
013040  614432  1270              AND     (17)            RETAIN JUST THE DIGIT
013041  052760  1280              DAC     T1              AND SAVE IT FOR DECIMAL ACCUMULATION
                    1290
013042  640503  1300              LRS     3               SAVE THE "OCTAL DIGIT"
013043  212634  1310              LAC     WORDB           LOAD THE PREVIOUSLY GATHERED "OCTAL NUMBER"
013044  640603  1320              LLS     3               CONCATENATE THE "OCTAL DIGITS"
013045  052634  1330              DAC     WORDB           AND SAVE THE RESULT
                    1340
013046  212635  1350              LAC     WORDB-1         LOAD THE PREVIOUSLY GATHERED "DECIMAL NUMBER"
013047  744000  1360              CLL                     SET THE LINK FOR THE MULTIPLY
013050  653122  1370              MUL                     MULTIPLY THE PREVIOUS "DECIMAL VALUE"
013051  000012  1380              10.                     BY 10 FOR DECIMAL
013052  641002  1390              LACQ                    LOAD THE RESULT
013053  352760  1400              TAD     T1              ADD THE CURRENT "DECIMAL DIGIT"
013054  052635  1410              DAC     WORDB-1         AND SAVE THE TOTAL "DECIMAL NUMBER"
                    1420
013055  613032  1430              JMP     NUM20           LOOP
                    1440
                    1450
013056  554433  1460      NUM26   SAD     (SPOINT)        CHECK FOR A PERIOD
013057  613065  1470              JMP     NUM27           IF SO, PICK UP THE DECIMAL VALUE
013060  212761  1480              LAC     T2              ELSE LOAD THE DECIMAL-DIGITS-RECEIVED FLAG
013061  744200  1490              SZA!CLL                 AND SKIP IF THERE WERE NONE
013062  633026  1500              JMP     NUMIN,X         RETURN +1, LK=0 FOR A FORMAT ERROR; DECIMAL DIGITS, BUT NO PERIOD
013063  212634  1510              LAC     WORDB           LOAD THE OCTAL VALUE
013064  613074  1520              JMP     NUM29
013065  113204  1530      NUM27   JMS     FGET            GET THE NEXT CHARACTER
013066  113230  1540              JMS     CHRID           AND IDENTIFY IT
013067  613073  1550              JMP     NUM28           A DELIMITER IS LEGAL, SO EXIT
013070  633026  1560              JMP     NUMIN,X         A LETTER -- EXIT +1 FOR A FORMAT ERROR
013071  744000  1570              CLL                     A NUMBER -- CLEAR THE LINK FOR A FORMAT ERROR
```

T (MTSS TELETYPE HANDLER) OCTAL WORD INPUT/OUTPUT

```
013072  633026  1580            JMP     NUMIN,X         AND EXIT +1
013073  212635  1590    NUM28   LAC     WORDB+1         LOAD THE DECIMAL VALUE
013074  052634  1600    NUM29   DAC     WORDB           SAVE THE CORRECT VALUE
013075  453026  1610            ISZ     NUMIN           BUMP TO A RETURN +2 FOR SUCCESSFUL
013076  633026  1620            JMP     NUMIN,X
                1630
                1640
                1650
    013077      1660            ENTER   OCTOT
013077  740040          OCTOT   XX
013100  652000  1670    OCT42   LMQ                     SET THE VALUE TO BE OUTPUT
013101  741400  1680            SZL                     SKIP IF NO LEADING ZEROES ARE TO BE SUPPRESSED
013102  750201  1690            SZA!CLC                 SET A FLAG TO PRINT ONE CHARACTER, ANYWAY; IF THE AC IS ZERO
013103  777772  1700            LAW     -6              ELSE SET THE COUNT FOR THE STANDARD SIX CHARACTERS
013104  052760  1710            DAC     T1              SET THE NUMBER OF CHARACTERS TO BE OUTPUT
013105  641002  1720            LACQ                    RELOAD THE USER'S VALUE
013106  741200  1730            SNA                     SKIP FOR A NON-ZERO VALUE
013107  744000  1740            CLL                     ELSE FORCE A SINGLE ZERO TO PRINT
013110  641603  1750    OCT44   LLSC    3.              GET THE NEXT OCTAL DIGIT

013111  740200  1760            SZA                     IF IT IS ZERO, DON'T CHANGE PRINT-SUPPRESSION STATE
013112  744000  1770            CLL                     ELSE CLEAR THE PRINT INHIBIT AT THE FIRST NON-ZERO FOUND
013113  354434  1780            TAD     (260)           MAKE ASCII IN ANY CASE
013114  740400  1790            SNL                     BUT SKIP IF PRINT IS INHIBITED
013115  113262  1800            JMS     TTYOT           ELSE PRINT THE DIGIT
013116  452760  1810            ISZ     T1              DONE???
013117  613110  1820            JMP     OCT44           NO -- LOOP
013120  700401  1830            TSF
013121  613120  1840            JMP     .-1             WAIT FOR THE TELETYPE TO SETTLE
013122  633077  1850            JMP     OCTOT,X         YES -- EXIT
```

T (MTSS TELETYPE HANDLER) SIXBIT WORD INPUT & SIXBIT BUFFER OUTPUT

```
                    1870
                    1880
                    1890    *
                    1900    *       OPERATION RETURN L    AC      MQ    MEANING
                    1910    *       --------- ------ - ------ ------ ------------------------------
                    1920    *       INPUT       +1   1 DELIM    X     FIRST NON-BLANK CHARACTER IS A DELIMITER
                    1930    *                   +2   1 SIXBIT DELIM   SUCCESSFUL READ OF A SIXBIT WORD
                    1940    *       OUTPUT      +1   X   X      X     SUCCESSFUL WRITE OF A SIXBIT BUFFER
                    1950    *
                    1960
      013123        1970            ENTER   SIXIN
013123  740040              SIXIN   XX
013124  772634      1980            LAW     WORDB
013125  052760      1990            DAC     T1              INITIALIZE THE SIXBIT BUFFER POINTER
013126  113211      2000            JMS     INTIN           INITIALIZE THE INPUT
013127  633123      2010            JMP     SIXIN,X         RETURN +1 FOR DELIMITER IS FIRST NON-BLANK CHARACTER
013130  453123      2020            ISZ     SIXIN           ELSE BUMP TO A GOOD RETURN
013131  113147      2030    SIX2    JMS     SIX5            GET THE FIRST GOOD CHARACTER
013132  660714      2040            ALSS    12.             AND PUT IT IN THE FIRST CHARACTER POSITION
013133  072760      2050            DAC     T1,X            AND SAVE IT
013134  113147      2060            JMS     SIX5            GET THE SECOND CHARACTER
013135  660706      2070            ALSS    6.              PUT IT IN THE SECOND CHARACTER POSITION
013136  272760      2080            XOR     T1,X            CONCATENATE THE CHARACTERS
013137  072760      2090            DAC     T1,X            AND SAVE THE RESULT
013140  113147      2100            JMS     SIX5            GET THE THIRD CHARACTER
013141  272760      2110            XOR     T1,X            CONCATENATE THE CHARACTERS
013142  072760      2120            DAC     T1,X            AND SAVE THE RESULT
013143  452760      2130            ISZ     T1              BUMP THE STORAGE BUFFER POINTER
013144  613131      2140            JMP     SIX2            LOOP
                    2150
013145  212634      2160    SIX9    LAC     WORDB           LOAD THE FIRST SIXBIT WORD
013146  633123      2170            JMP     SIXIN,X         EXIT
                    2180
      013147        2190            ENTER   SIX5            SUBROUTINE TO GET THE NEXT CHARACTER, MAKE IT SIXBIT, EXIT IF A DELIMITER
013147  740040              SIX5    XX
013150  113204      2200            JMS     FGET            GET THE NEXT CHARACTER
013151  113230      2210            JMS     CHRID           IDENTIFY IT
013152  613145      2220            JMP     SIX9            EXIT IF IT IS A DELIMITER
013153  740000      2230            NOP                     PERMIT LETTERS
013154  354435      2240            TAD     (-240)          MAKE SIXBIT
013155  633147      2250            JMP     SIX5,X
                    2260
                    2270
      013156        2280            ENTER   SIXOT
013156  740040              SIXOT   XX
013157  052760      2290            DAC     T1              SET THE NEGATIVE CHARACTER COUNT
013160  233156      2300    SIX24   LAC     SIXOT,X         LOAD THE NEXT WORD OF OUTPUT
013161  652000      2310            LMQ                     SAVE IT FOR PRINTING
013162  453156      2320            ISZ     SIXOT           BUMP THE POINTER
013163  113167      2330            JMS     SIX26           OUTPUT THE FIRST CHARACTER
013164  113167      2340            JMS     SIX26           OUTPUT THE SECOND CHARACTER
013165  113167      2350            JMS     SIX26           OUTPUT THE THIRD CHARACTER
```

T (MTSS TELETYPE HANDLER) SIXBIT WORD INPUT & SIXBIT BUFFER OUTPUT

```
013166  613160  2360            JMP     SIX24           LOOP
                2370
    013167      2380            ENTER   SIX26
013167  740040          SIX26   XX
013170  641606  2390            LLSC    6.              GET THE NEXT SIXBIT CHARACTER
013171  354436  2400            TAD     (240)           MAKE IT ASCII
013172  554437  2410            SAD     (333)           CHECK FOR CARRIAGE RETURN MAPPING
013173  760215  2420            LAW     $CR
013174  554440  2430            SAD     (335)           CHECK FOR LINE FEED MAPPING
013175  760212  2440            LAW     $LF
013176  113262  2450            JMS     TTYOT           PRINT THE CHARACTER
013177  452760  2460            ISZ     T1              ALL CHARACTERS PRINTED?
013200  633167  2470            JMP     SIX26,X         NO -- LOOP
013201  700401  2480            TSF
013202  613201  2490            JMP     .-1             WAIT FOR THE TELETYPE TO SETTLE
013203  633156  2500            JMP     SIXOT,X         YES -- EXIT
                2510    *
                2520    *
```

T (MTSS TELETYPE HANDLER) MISCELLANEOUS LINE BUFFER ROUTINES

```
                    2540
                    2550
                    2560
                    2570
    013204          2580            ENTER   FGET            SUBROUTINE TO GET THE FIRST REMAINING CHARACTER FROM THE LINE BUFFER
013204  740040              FGET    XX
013205  452757      2590            ISZ     BPTR            NO -- BUMP THE POINTER
013206  232757      2600            LAC     BPTR,X          LOAD THE NEXT CHARACTER
013207  052762      2610            DAC     CHAR            AND SAVE IT
013210  633204      2620    FGET9   JMP     FGET,X
                    2630
    013211          2640            ENTER   INTIN           INITIALIZE INPUT WORD-GETTING
013211  740040              INTIN   XX
013212  452764      2650            ISZ     COUNT           COUNT THE WORD, SUCCESSFUL OR NOT
013213  152634      2660            DZM     WORDB           INITIALIZE THE TWO FIRST WORDS OF THE INPUT BUFFER
013214  152635      2670            DZM     WORDB+1
013215  113204      2680            JMS     FGET            GET THE NEXT CHARACTER
013216  554436      2690            SAD     (SSPACE)        CHECK IT FOR A SPACE
013217  613215      2700            JMP     .-2             THROW AWAY SPACES
013220  113230      2710            JMS     CHRID           IDENTIFY THE NON-SPACE
013221  633211      2720            JMP     INTIN,X         RETURN +1 FOR A DELIMITER
013222  740000      2730            NOP
013223  453211      2740            ISZ     INTIN           ELSE BUMP THE RETURN FOR A NUMBER OR A LETTER
013224  750001      2750            CLC
013225  352757      2760            TAD     BPTR            BACK UP THE POINTER TO POINT TO THE FIRST GOOD CHARACTER
013226  052757      2770            DAC     BPTR
013227  633211      2780            JMP     INTIN,X
```

T (MTSS TELETYPE HANDLER) MISCELLANEOUS CHARACTER-HANDLING SUBROUTINES

```
                    2800    *
                    2810    *
                    2820    *       CHRID -- SUBROUTINE TO CLASSIFY EIGHT-BIT ASCII CHARACTERS.
                    2830    *       ENTER WITH THE CHARACTER IN THE AC; LEAVE WITH THE EIGHT-BIT CHARACTER
                    2840    *       IN AC(0-17) AND THE LINK AS FOLLOWS:
                    2850    *
                    2860    *       RETURN LINK MEANING
                    2870    *       ------ ---- ------------------------------------------------
                    2880    *         +1    1   THE CHARACTER IS A DELIMITER (I.E. NEITHER A DIGIT NOR A LETTER
                    2890    *         +2    0   THE CHARACTER IS EITHER AN UPPER CASE OR A LOWER CASE LETTER
                    2900    *         +3    0   THE CHARACTER IS AN OCTAL DIGIT
                    2910    *         +3    1   THE CHARACTER IS A DECIMAL DIGIT (8 OR 9)
                    2920    *
    013230          2930            ENTER   CHRID
013230  740040              CHRID   XX
013231  614441  2940              AND     (377)
013232  053262  2950              DAC     TTYOT           SAVE THE EIGHT-BIT ASCII CHARACTER
013233  354442  2960              TAD     (-260)          AC < 0 FOR DELIMITERS
013234  745102  2970              SPA!STL
013235  613253  2980              JMP     DLMR            CHARACTER IS A DELIMITER
013236  354443  2990              TAD     (-10)           AC < 0 FOR OCTAL DIGITS
013237  745100  3000              SPA!CLL
013240  613256  3010              JMP     DIGIT           CHARACTER IS AN OCTAL DIGIT
013241  354444  3020              TAD     (-2)            AC < 0 FOR DECIMAL DIGITS
013242  745102  3030              SPA!STL
013243  613256  3040              JMP     DIGIT           CHARACTER IS A DECIMAL DIGIT
013244  354445  3050              TAD     (-6)            AC <= 0 FOR DELIMITERS
013245  745302  3060              SNA!SPA!STL
013246  613253  3070              JMP     DLMR            CHARACTER IS A DELIMITER
013247  614446  3080              AND     (777737)        MAP LOWER CASE INTO UPPER CASE
013250  354447  3090              TAD     (-33)           AC < 0 FOR LETTERS -- L=1 FOR LETTERS; L=0 FOR DELIMITERS
013251  741102  3100              SPA!CML
013252  613257  3110              JMP     LETTR           THE CHARACTER IS A LETTER
                    3120
013253  213262  3130      DLMR    LAC     TTYOT           LOAD THE DELIMITER
013254  052763  3140              DAC     DLMTR           SAVE IT
013255  633230  3150              JMP     CHRID,X
                    3160
013256  453230  3170      DIGIT   ISZ     CHRID
013257  453230  3180      LETTR   ISZ     CHRID
013260  213262  3190              LAC     TTYOT           RELOAD THE CHARACTER
013261  633230  3200              JMP     CHRID,X
                    3210
                    3220
                    3230
    013262          3240            ENTER   TTYOT
013262  740040              TTYOT   XX
013263  700401  3250              TSF
013264  613263  3260              JMP     .-1             WAIT FOR THE TELEPRINTER TO BE FREE
013265  700301  3270              KSF                     KILL-THE-OUTPUT FEATURE
013266  700406  3280              TLS                     PRINT THE CHARACTER IN THE AC
013267  633262  3290              JMP     TTYOT,X
```

T (MTSS TELETYPE HANDLER) MISCELLANEOUS CHARACTER-HANDLING SUBROUTINES

```
                    3300
                    3310
          013270    3320            ENTER   CRLF
013270    740040            CRLF    XX
013271    760215    3330            LAW     215
013272    113262    3340            JMS     TTYOT
013273    760215    3350            LAW     215
013274    113262    3360            JMS     TTYOT
013275    760212    3370            LAW     212
013276    113262    3380            JMS     TTYOT
013277    700401    3390            TSF
013300    613277    3400            JMP     .-1             WAIT FOR THE TTY TO SETTLE
013301    633270    3410            JMP     CRLF,X
                    3420
                    3430
                    3440            ,HEAD                   TURN OFF THE INSERT'S HEAD SYMBOL
                    3460            ,END
                    7950            ,INSRT  :DLIBRARY:PDP9LIB:GROCAT
```

C DESCRIPTION OF THE GROWTH SYSTEM CATALOG STRUCTURE

```
                .HEAD   C
*
*       MAJOR REVISION -- JAN 21, 1972 BY ROBERT W. BLEAN
*
*       A GROWTH CATALOG FOR A FILE-ORIENTED DEVICE IS LOCATED IN THE 400 WORDS
*       OF LOGICAL BLOCK 1 OF THE LOGICAL DEVICE. THIS PERMITS DISK AND DECTAPE
*       TO BE USED INTERCHANGEABLY BY THE GROWTH SYSTEM PROGRAMS.
*
*       THE DEVICE ADDRESS OF A HANDLER IS THE HANDLER NUMBER IN BITS 0-2
*       AND THE TYPE (DISK (1) OR DECTAPE (0)) IN BIT 3.
*
*       THE DEVICE ADDRESS OF A FILE IS THE DEVICE ADDRESS OF THE HANDLER IT
*       IS ON PLUS IN BITS 8-17 ITS STARTING BLOCK NUMBER.
*
*       ALL DEVICE ADDRESSES IN A DECTAPE CATALOG ARE CORRECT FOR THE HANDLER
*       THE TAPE WAS MOUNTED ON THE LAST TIME IT WAS ALTERED.
*
*       THE FIRST FOUR WORDS OF THE CATALOG BLOCK ARE A HEADER:
*            1) THE DEVICE ADDRESS OF THE FIRST FREE BLOCK ON THE DEVICE
*            2) UNUSED
*            3) TWOS COMPLEMENT COUNT OF THE NUMBER OF FILES CATALOGED
*            4) TWOS COMPLEMENT MAXIMUM BLOCK NUMBER ON THE DEVICE
*
*       THE REMAINDER OF THE CATALOG CONSISTS OF A SERIES OF FIVE WORD FILE-
*       CONTROL BLOCKS. THE FIRST FILE CONTROL BLOCK IS FOR THE CATALOG ITSELF.
*       THEN THERE IS ONE FILE CONTROL BLOCK FOR EACH FILE ON THE DEVICE.
*
*       FORMAT OF THE FILE CONTROL BLOCKS:
*            1) THE FIRST WORD IS THE SIXBIT ASCII (EIGHTBIT ASCII - 240)
*               FILENAME. THIS MEANS THE FILENAME IS RESTRICTED TO THREE
*               CHARACTERS, WITH NO EXTENSION OR PASSWORD.
*            2) THE DEVICE ADDRESS OF THE FILE.
*            3) THE FILE'S CORE ADDRESS
*            4) THE FILE'S LENGTH (IN WORDS)
*            5) THE PROGRAM START
*
*       THIS LEAVES TWO WORDS OF THE CATALOG BLOCK UNUSED. IT IS SUGGESTED THAT
*       THE SECOND OF THESE CONTAIN THE BLOCK NUMBER OF A CONTINUATION OF THE
*       CATALOG, SHOULD THIS EVER BE NECESSARY; IT WOULD BE ZERO IF THERE
*       IS NO CONTINUED CATALOG BLOCK.
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
                    570
013302  000000      580     CTEM1   .DSA                    CATALOG ROUTINE'S PRIVATE TEMP
013303  017000      590             CATLOG                  CATALOG CORE ADDRESS
013304  000400      600             CATLEN                  CATALOG LENGTH
013305  000000      610     CATALT  .DSA                    CATALOG ALTERED FLAG
                    620
                    630
                    640     *
                    650     *       RCAT -- THE BASIC CATALOG ROUTINE, IT READS IN CATALOGS AND UPDATES THEM
                    660     *       FOR THE CURRENT DEVICE AND (POSSIBLY NON-STANDARD) CATALOG LOCATION.
                    670     *
                    680     *
                    690     *       A HANDLER DEVICE ADDRESS IS PASSED IN THE AC (POSSIBLY ALONG WITH OTHER
                    700     *       GARBAGE). IF THAT HANDLER'S CATALOG IS ALREADY IN CORE, RCAT EXITS
                    710     *       IMMEDIATELY, OTHERWISE THE CURRENT CATALOG IS READ OUT IF IT HAS BEEN
                    720     *       ALTERED SINCE IT WAS READ IN, THEN THE REQUESTED CATALOG IS READ
                    730     *       IN AND ALL OF THE DEVICE ADDRESSES ARE UPDATED. THE CATALOG ALTERS
                    740     *       FLAG IS CLEARED IF A CATALOG IS READ IN, UNTOUCHED OTHERWISE.
                    750     *
                    760     *
                    770     *       AS A RESULT, THE CATALOG IN CORE ALWAYS HAS THE PROPER DEVICE ADDRESSES
                    780     *       FOR THE DEVICE IT WAS READ FROM.
                    790     *
                    800     *       RETURN IS +1 WHEN THE DESIRED CATALOG IS IN CORE.
                    810     *
                    820     *       IN THE EVENT OF UNRECOVERABLE ERROR, EXIT IS TO AN ERROR ROUTINE.
                    830     *
    013306          840             .USE
013306  740040      850     RCAT    XX
013307  053302      860             DAC     CTEM1           SAVE THE DEVICE ADDRESS OF THE DEVICE WHOSE CATALOG IS BEING REQUESTED
013310  257000      870             XOR     CATLOG          COMPARE THE REQUESTED DEVICE ADDRESS WITH CURRENT CATALOG'S DEVICE ADDRESS
013311  514450      880             AND     (DVCMSK)        EXTRACT JUST THE DEVICE ADDRESS PORTION
013312  741200      890             SNA                     SKIP IF A DIFFERENT CATALOG IS BEING REQUESTED
013313  633306      900             JMP     RCAT,X          ELSE EXIT DIRECTLY
013314  113354      910             JMS     FORCE           FORCE THE OLD CATALOG BEFORE READING A NEW ONE
                    920
    013315          930     RCAT1   ...
013315  213302      940             LAC     CTEM1
013316  514450      950             AND     (DVCMSK)        GET THE NEW HANDLER'S DEVICE ADDRESS
013317  254376      960             XOR     (CATBLK)        ADD IN THE CATALOG BLOCK NUMBER
013320  053302      970             DAC     CTEM1           SAVE THE NEW CATALOG'S DEVICE ADDRESS
013321  113365      980             JMS     CSRCOVR         SET UP THE ERROR RECOVERY
013322  773302      990             LAW     CTEM1           GET A POINTER TO THE CATALOG PARAMETERS
013323  117500     1000             JMS     $TAPIN          READ THE NEW CATALOG
                   1010     *
                   1020     *       NOW UPDATE THE DEVICE ADDRESSES
                   1030     *
013324  213302     1040             LAC     CTEM1
013325  514450     1050             AND     (DVCMSK)
013326  053302     1060             DAC     CTEM1           SET THE CURRENT DEVICE ADDRESS
                   1070
013327  217000     1080             LAC     CATLOG
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
013330  514417  1090          AND   (BLKMSK)
013331  253302  1100          XOR   CTEM1
013332  057000  1110          DAC   CATLOG          UPDATE THE OLD DEVICE ADDRESS OF THE FIRST FREE BLOCK
                1120
013333  777005  1130          LAW   CATLOG+5
013334  053354  1140          DAC   FORCE
013335  053440  1150          DAC   CATL            SET POINTERS TO THE FIRST OLD DEVICE ADDRESS
013336  217002  1160          LAC   CATLOG+2
013337  053365  1170          DAC   RCOVR           SET THE COUNT OF FCB'S
                1180
013340  233354  1190  RCAT4   LAC   FORCE,X         LOAD THE NEXT OLD DEVICE ADDRESS
013341  514417  1200          AND   (BLKMSK)        RECOVER THE BLOCK NUMBER
013342  253302  1210          XOR   CTEM1           ADD IN THE CURRENT HANDLER DEVICE ADDRESS
013343  073440  1220          DAC   CATL,X         SAVE THE UPDATED FILE DEVICE ADDRESS
                1230
013344  453365  1240          ISZ   RCOVR           COUNT THE FILES DONE
013345  741000  1250          SKP
013346  633306  1260          JMP   RCAT,X          ALL DONE
                1270
013347  213354  1280          LAC   FORCE           LOAD THE FCB POINTER
013350  354451  1290          TAD   (FCBLEN)        ADVANCE IT TO THE NEXT FCB
013351  053354  1300          DAC   FORCE
013352  053440  1310          DAC   CATL            SAVE THE NEW POINTER
013353  613340  1320          JMP   RCAT4           LOOP
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
                    1340    *
                    1350    *       FORCE WRITES THE CURRENT CATALOG TO ITS DEVICE IF IT HAS BEEN ALTERED
                    1360    *
013354  740040      1370    FORCE   XX
013355  213305      1380            LAC     CATALT          IS THE CATALOG ALTERED
013356  741200      1390            SNA                     SKIP IF THE CATALOG HAS BEEN ALTERED
013357  633354      1400            JMP     FORCE,X         ELSE DON'T BOTHER TO WRITE IT
013360  113365      1410            JMS     RCOVR           INITIALIZE THE ERROR RECOVERY
013361  777005      1420            LAW     CPARAM          POINT TO CATALOG
013362  117502      1430            JMS     STAPOT          FORCE IT
013363  153305      1440            DZM     CATALT          CLEAR CATALOG ALTER FLAG
013364  633354      1450            JMP     FORCE,X         RETURN
                    1460
                    1470
                    1480
013365  740040      1490    RCOVR   XX                      SUBROUTINE TO SET UP RECOVERY FROM HARDWARE ERRORS
013366  777776      1500            LAW     -2              SET FOR TWO RETRIES BEFORE GIVING UP
013367  053435      1510            DAC     ERCNT
013370  214452      1520            LAC     (JMP RCVR4)
013371  057505      1530            DAC     $RECOV          SET UP THE ERROR JUMP TO THE ERROR MESSAGE
013372  633365      1540            JMP     RCOVR,X
                    1550
    013373          1560    RCVR4   MESS    <DEVICE ERROR>,12.
013373  700312                      KRB
    013374                          MESSR   <DEVICE ERROR>,12.
013374  777762                      LAW     -12.-2
013375  113156                      JMS     TSSIXOT
013403  453435      1570            ISZ     ERCNT           COUNT THE ERROR
013404  633365      1580            JMP     RCOVR,X
    013405          1590    RCVR5   MESS    <TYPE 'IGNORE' OR 'CONTINUE'! >,29.
013405  700312                      KRB
    013406                          MESSR   <TYPE 'IGNORE' OR 'CONTINUE'! >,29.
013406  777741                      LAW     -29.-2
013407  113156                      JMS     TSSIXOT
    013423          1600            LINE                    GET THE USER'S ANSWER TO WHAT HE WANTS TO DO ABOUT IT
013423  112765                      JMS     TSINLIN
    013424          1610            WORD                    READ HIS ANSWER
013424  113123                      JMS     TSSIXIN
013425  613405      1620            JMP     RCVR5           NO INPUT IS ILLEGAL
013426  553436      1630            SAD     IGN
013427  613433      1640            JMP     RCVR6           IGNORE THE LAST COMMAND
013430  553437      1650            SAD     CON
013431  613366      1660            JMP     RCOVR+1         SET UP TO TRY AGAIN
013432  613405      1670            JMP     RCVR5           ANY OTHER ANSWER IS ILLEGAL
                    1680
013433  153305      1690    RCVR6   DZM     CATALT          FORGET THE CATALOG WAS ALTERED
013434  612000      1700            JMP     $NEXTL          GET THE NEXT COMMAND LINE
                    1710
013435  000000      1720    ERCNT   .DSA
013436  514756      1730    IGN     .ACI6   *IGN*
013437  435756      1740    CON     .ACI6   *CON*
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
                    1760    *
                    1770    *       CATL
                    1780    *
                    1790    *       CATL SEARCHES THE CATALOG IN CORE FOR THE FILENAME
                    1800    *       PASSED IN THE AC
                    1810    *
                    1820    *       RETURN +2 WITH CATX POINTING TO THE FILE NAME IF SUCCESSFUL
                    1830    *
                    1840    *       RETURN +1 WITH CATX POINTING TO THE FIRST FREE SPACE -1 IN THE
                    1850    *       CATALOG IF THE FILE NAME IS NOT FOUND
                    1860    *
013440  740040      1870    CATL    XX
013441  052634      1880            DAC     TSWORDB         SAVE CATALOG NAME
013442  777003      1890            LAW     CATLOG+3
013443  040011      1900            DAC     SCATX           SET A POINTER TO THE FIRST FCB IN THE CATALOG AUTO-INDEX REGISTER
013444  217002      1910            LAC     CATLOG+2        GET CATALOG COUNT
013445  053302      1920            DAC     CTEM1           SAVE IT
013446              1930    CATLL   WORD1                   RESTORE NAME TO SEARCH FOR
013446  212634                      LAC     TSWORDB
013447  560011      1940            SAD     SCATX,X         CHECK IT
013450  613457      1950            JMP     CATL9           FOUND IT
013451  200011      1960            LAC     SCATX
013452  354377      1970            TAD     (FCBLEN-1)      FAILED -- MOVE THE POINTER TO THE NEXT FILE CONTROL BLOCK
013453  040011      1980            DAC     SCATX
013454  453302      1990            ISZ     CTEM1           COUNT
013455  613446      2000            JMP     CATLL           LOOP
013456  633440      2010            JMP     CATL,X          EXHAUSTED, NO FILE FOUND -- BAD RETURN
013457  453440      2020    CATL9   ISZ     CATL            GOOD RETURN
013460  633440      2030            JMP     CATL,X
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
                    2050    *
                    2060    *       GNAME
                    2070    *
                    2080    *       GNAME GETS A FILE NAME FROM THE TTY BUFFER
                    2090    *       AND READS IN THE CATALOG IF NECESSARY
                    2100    *
                    2110    *       RETURN IS +1 FOR PAPER TAPE DESIRED
                    2120    *       RETURN IS +2 FOR SUCCESS ON DISK OR DECTAPE
                    2130    *       OTHERWISE EXIT IS TO FORMAT ERROR
                    2140    *
                    2150    *       THE FILE NAME IS RETURNED IN TSWORDB AND IN THE AC.
                    2160    *
013461  740040      2170    GNAME   XX
    013462          2180            WORD                    GET A WORD OF SIX BIT ASCII
013462  113123                      JMS     TSSIXIN
013463  740000      2190            NOP
    013464          2200            DELIM                   GET THE DELIMITER
013464  212763                      LAC     TSDLMTR
013465  554453      2210            SAD     ($COLON)        CHECK FOR COLON
013466  613472      2220            JMP     GNAM2
013467  113543      2230            JMS     PAPER           CHECK FOR PAPER TAPE
013470  633461      2240            JMP     GNAME,X         YES -- PAPER TAPE
013471  613502      2250            JMP     GNAM5           NO -- SO USE CURRENT CATALOG
013472  773476      2260    GNAM2   LAW     GNAM3
013473  053516      2270            DAC     DEVCV
    013474          2280            WORD1                   RELOAD THE CATALOG NAME
013474  212634                      LAC     TSWORDB
013475  613523      2290            JMP     DEVC3           CONVERT IT TO A DEVICE ADDRESS
013476  633461      2300    GNAM3   JMP     GNAME,X
013477  113306      2310            JMS     RCAT            READ IN THE CATALOG
    013500          2320            WORD                    GET ANOTHER WORD
013500  113123                      JMS     TSSIXIN
013501  740000      2330            NOP
    013502          2340    GNAM5   DELIM                   GET THE DELIMITER
013502  212763                      LAC     TSDLMTR
013503  554454      2350            SAD     ($SLASH)        CHECK FOR SLASH
013504  613511      2360            JMP     GNAME6          LOOK FOR OCTAL
    013505          2370            WORD1                   ELSE RECOVER THE SIXBIT NAME
013505  212634                      LAC     TSWORDB
013506  741200      2380            SNA                     CHECK FOR ALL SPACES
    013507          2390            FORMAT                  FORMAT ERROR -- ALL SPACES IS AN ILLEGAL NAME
013507  612000                      JMP     FORMAT
013510  613514      2400            JMP     GNAM8
    013511          2410    GNAME6  NUM                     GET THE NUMBER
013511  113026                      JMS     TSNUMIN
    013512          2420            FORMAT
013512  612000                      JMP     FORMAT
013513  052634      2430            DAC     TSWORDB         TO BE COMPATABLE WITH SIXBIT INPUT
013514  453461      2440    GNAM8   ISZ     GNAME           GOOD RETURN
013515  633461      2450            JMP     GNAME,X
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
                    2470    *
                    2480    *       DEVCV -- READS THE NEXT WORD,
                    2490    *       RETURN IS +1 WITH THE NAME IN THE AC IF IT IS A PAPER TAPE CALL
                    2500    *
                    2510    *       OTHERWISE IT ATTEMPTS TO CONVERT THE NAME TO DEVICE ADDRESS FORMAT,
                    2520    *       IF SUCCESSFUL, IT RETURNS +2 WITH THE HANDLER NUMBER IN AC(0-2) AND
                    2530    *       THE DEVICE TYPE (DISK (1) OR DECTAPE (0)) IN AC(3). REMAINING BITS
                    2540    *       ARE ZEROED.
                    2550    *
                    2560    *       EXIT IS TO THE FORMAT ERROR MESSAGE IF THE DEVICE IS NEITHER PAPER TAPE
                    2570    *       NOR DISK NOR DECTAPE.
                    2580    *
013516  740040      2590    DEVCV   XX
    013517          2600            WORD                    GET THE DEVICE NAME
013517  113123                      JMS     T$SIXIN
    013520          2610            FORMAT
013520  612000                      JMP     FORMAT
013521  113543      2620            JMS     PAPER           CHECK FOR PAPER TAPE
013522  633516      2630            JMP     DEVCV,X         YES -- PAPER TAPE
013523  514455      2640    DEVC3   AND     (777700)        REMOVE DEVICE NUMBER
013524  554456      2650            SAD     ($TP.)          CHECK FOR DECTAPE
013525  613535      2660            JMP     DEVC1           YES
013526  554457      2670            SAD     ($DT.)          CHECK FOR DECTAPE
013527  613535      2680            JMP     DEVC1
013530  554460      2690            SAD     ($DK.)          CHECK FOR DISK
013531  741000      2700            SKP
    013532          2710            FORMAT                  NO OTHERS -- FORMAT ERROR
013532  612000                      JMP     FORMAT
013533  650004      2720            CLQ!CMQ                 FOR DISK PUT THE SIGN BIT ON IN THE MQ
013534  741000      2730            SKP
013535  650000      2740    DEVC1   CLQ                     CLEAR Q FOR TAPE
    013536          2750            WORD1                   RESTORE NAME
013536  212634                      LAC     T$WORDB
013537  640617      2760            LLS     18.-3           SHIFT TO POSITION
013540  514450      2770            AND     (DVCMSK)        CONVERT TO HANDLER DEVICE ADDRESS FORMAT
013541  453516      2780            ISZ     DEVCV           INCREMENT RETURN
013542  633516      2790            JMP     DEVCV,X         AND NOW RETURN
                    2800    *
                    2810    *       PAPER CHECKS THE AC FOR A PAPER TAPE MNEMONIC, IT RETURNS +1 IF IT
                    2820    *       FINDS ONE, ELSE RETURNS +2. THE AC IS UNCHANGED.
                    2830    *
    013543          2840    PAPER   ENTER
013543  740040                      XX
    013544          2850            WORD1                   RECOVER THE WORD
013544  212634                      LAC     T$WORDB
013545  554461      2860            SAD     ($PPT)
013546  633543      2870            JMP     PAPER,X
013547  554462      2880            SAD     ($PTR)
013550  633543      2890            JMP     PAPER,X
013551  554463      2900            SAD     ($PTP)
013552  633543      2910            JMP     PAPER,X
013553  453543      2920            ISZ     PAPER           NO PAPER TAPE MNEMONIC
```

```
                C                       GROWTH SYSTEM STANDARD CATALOG ROUTINES

    013554  633543  2930                JMP     PAPER,X
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
                    2980    *
                    2990    *       SAVE CHECKS THE CATALOG FOR THE NAME FOUND IN THE AC
                    3000    *
                    3010    *       RETURN IS +1 IF THE FILE IS ALREADY SAVED
                    3020    *       A CATALOG ENTRY IS CREATED FOR THE NAME AND RETURN IS +2 OTHERWISE
                    3030    *       EXITS TO AN ERROR MESSAGE IF THE CATALOG IS FULL
                    3040    *
                    3050    *       ON RETURN CATX POINTS TO THE FILE NAME IN THE CATALOG
                    3060    *
013555  740040      3070    SAVE    XX
013556  113440      3080            JMS     CATL            LOOK UP NAME
013557  741000      3090            SKP
013560  633555      3100            JMP     SAVE,X          DON'T ALLOW DUPLICATES
013561  217002      3110            LAC     CATLOG+2        LOAD THE FCB COUNT
013562  554464      3120            SAD     (CATMAX)        CHECK FOR CATALOG ALREADY FULL
013563  613573      3130            JMP     CFULL           YES -- EXIT TO AN ERROR MESSAGE
013564  354431      3140            TAD     (-1)            COUNT THE NEW FILE
013565  057002      3150            DAC     CATLOG+2        UPDSOATE THE FCB COUNT
    013566          3160            WORD1                   RECOVER THE FILE NAME
013566  212634                      LAC     T$WORDB
013567  060011      3170            DAC     $CATX,X         SAVE IT
013570  453305      3180            ISZ     CATALT          FLAG THE CATALOG HAS BEEN ALTERED
013571  453555      3190            ISZ     SAVE
013572  633555      3200            JMP     SAVE,X
                    3210
    013573          3220    CFULL   MESS    <CATALOG FULL>,12.
013573  700312                      KRB
    013574                          MESSR   <CATALOG FULL>,12.
013574  777762                      LAW     -12.,-2
013575  113156                      JMS     T$SIXOT
013603  612000      3230            JMP     $NEXTL
                    3240    *
                    3250    *       ALC RECEIVES A WORD COUNT IN THE AC AND CALCULATES THE LEAST INTEGER
                    3260    *       NUMBER OF BLOCKS THAT CAN HOLD THAT LENGTH. IT THEN ALLOCATES THE STORAGE
                    3270    *       IN THE CORE CATALOG HEADER AND RETURNS WITH THE DEVICE ADDRESS OF THE
                    3280    *       FIRST FREE BLOCK IN THE AC.
                    3290    *
                    3300    *       EXIT IS TO AN ERROR MESSAGE IF THIS ALLOCATION WOULD RESULT IN
                    3310    *       OVERFLOWING THE DEVICE. IN THIS CASE THE CATALOG IS UNALTERED.
                    3320    *
013604  740040      3330    ALC     XX
013605  354441      3340            TAD     (377)           ROUND UP TO A BLOCK
013606  660510      3350            LRSS    8.              AC = MINIMUM INTEGER NUMBER OF BLOCKS REQUIRED
013607  053302      3360            DAC     CTEM1           SAVE IN A GOOD RANDOM PLACE
013610  217000      3370            LAC     CATLOG          GET THE POINTER TO THE FIRST FREE BLOCK
013611  652000      3380            LMQ                     SAVE IT
013612  353302      3390            TAD     CTEM1           ADD THE REQUESTED NUMBER OF BLOCKS TO FORM A NEW POINTER
013613  053302      3400            DAC     CTEM1           SAVE THE NEW POINTER
013614  514417      3410            AND     (1777)          EXTRACT BLOCK NUMBER
013615  357003      3420            TAD     CATLOG+3        SEE IF WE OVERFLOWED THE DEVICE
013616  740100      3430            SMA                     NO IF SKP
013617  613624      3440            JMP     DFULL           FULL -- HELP*?!@
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
013620  213302  3450            LAC     CTEM1
013621  057000  3460            DAC     CATLOG          SET THE FREE FCB POINTER NOW WE KNOW IT WILL BE OK
013622  641002  3470            LACQ                    RESTORE THE DEVICE ADDRESS OF THE FIRST FREE BLOCK
013623  633604  3480            JMP     ALC,X
                3490
    013624      3500    DFULL   MESS    <DEVICE FULL>,11.
013624  700312                  KRB
    013625                      MESSR   <DEVICE FULL>,11.
013625  777763                  LAW     -11,-2
013626  113156                  JMS     T$SIXCT
013634  612000  3510            JMP     $NEXTL
    013635      3520    MON9    ...
                3530            .HEAD
                3550            .END
```

I CONSTANTS, TEMPORARY STORAGE, ETC

```
                    7970            ,HEAD   I               RESTORE THE HEAD SYMBOL AFTER THE INSERTS
                    7980
                    7990    *
                    8000    *
                    8010    *       LIST OF SWAPPABLE SYSTEM FILES ('INT' AND 'RES' ARE NOT SWAPPABLE
                    8020    *       AND SO LIVE ONLY ON THE LIBRARY DECTAPE)
                    8030    *
    013634          8040    UFILES  ,EQU    .-1             START OF USER-TYPE SYSTEM FILENAMES
013635  422027      8050            $DDT
013636  422030      8060            $BAS
013637  000000      8070    PFILES  0                       START OF PHANTOM-TYPE SYSTEM FILENAMES
013640  422025      8080            $MTR
013641  422026      8090            $LDR
013642  000000      8100    OFILES  0                       AUTO-INDEX POINTER TO THE LIST OF OVERLAY FILENAMES
013643  422022      8110            $SWP                    SWP MUST BE FIRST OVERLAY FILE, SINCE IT IS THE ONLY ONE ENTERED IN SWPCAT
013644  422023      8120            $MP1
013645  422024      8130            $MP2
013646  422122      8140            $SPL
013647  000000      8150    EFILES  0                       END OF THE FILES LIST
013650  000000      8160    TEMP1   ,DSA
013651  000000      8170    TEMP2   ,DSA
013652  000000      8180    TEMP3   ,DSA
013653  000000      8190    TEMP4   ,DSA
013654  000000      8200    TPARAM  ,DATA   0,BUF,0
013655  000100
013656  000000
013657  000000      8210    SYSDVC  ,DSA
```

I CONSTANTS, TEMPORARY STORAGE, ETC

```
                    8230
                    8240    *
                    8250    *
                    8260    *       CONSTANTS TO SET UP THE INITIALIZATION CATALOG
                    8270    *
001677              8280    USTRT   .EQU    BOUNDARY-TABLEN-1  USER CORE ADDRESS -1
640000              8290    SPTR    .EQU    $SCRSTR            POINTER TO SCRATCH STORAGE
                    8300    *
                    8310    *       MACRO TO SET UP A USER'S SCRATCH FILES IN INTCAT
                    8320    *
                    8330    INTUS   .DEFIN                     MACRO TO SET UP A SET OF SCRATCH FILE DATA FOR EACH USER
                    8340            .PMC    SAVE,ON            PRINT AT LEAST THIS MACRO!
                    8350
                    8360    UDK#1   .EQU    SPTR
                    8370    SPTR    .EQU    SPTR+$DKLEN
                    8380
                    8390    UC#1    $US#1                      USER NAME
                    8400            SPTR                       PHYSICAL DISK LOCATION ABOVE THE BASE OF THE SCRATCH AREA
                    8410            BOUNDARY-1                 CORE ADDRESS -1
                    8420            -$USLEN                    TWO'S COMPLEMENT LENGTH
                    8430    SPTR    .EQU    SPTR+$USLEN
                    8440
                    8450            $PH#1
                    8460    UP#1    SPTR
                    8470            BOUNDARY-1
                    8480            -$PHLEN
                    8490    SPTR    .EQU    SPTR+$PHLEN
                    8500
                    8510    TAB#1   $UT#1
                    8520            SPTR
                    8530            USTRT
                    8540            -$TABLEN
                    8550    SPTR    .EQU    SPTR+$TABLEN
                    8560
                    8570            .PMC    RESTORE
                    8580            .ENDM
                    8590    *
                    8600    *
                    8610    *       INITIALIZATION CATALOG
                    8620    *
013660              8630    INTCAT  ...
                    8640    *
                    8650    *       SET UP THE CATALOG ENTRIES FOR EACH USER
013660              8660            INTUS   0                  SET UP USER #0

640000                      UDK0    .EQU    SPTR
656000                      SPTR    .EQU    SPTR+$DKLEN

013660  000076              UC0     $US0                       USER NAME
013661  656000                      SPTR                       PHYSICAL DISK LOCATION ABOVE THE BASE OF THE SCRATCH AREA
013662  001777                      BOUNDARY-1                 CORE ADDRESS -1
013663  764000                      -$USLEN                    TWO'S COMPLEMENT LENGTH
```

```
                I                       CONSTANTS, TEMPORARY STORAGE, ETC

    672000              SPTR    .EQU    SPTR+$USLEN

013664  000077                  $PH0
013665  672000          UP0     SPTR
013666  001777                  BOUNDARY-1
013667  776100                  -$PHLEN
    673700              SPTR    .EQU    SPTR+$PHLEN

013670  000075          TAB0    $UT0
013671  673700                  SPTR
013672  001677                  USTRT
013673  777700                  -$TABLEN
    674000              SPTR    .EQU    SPTR+$TABLEN

    013674      8670            INTUS   1               SET UP USER #1

    674000              UDK1    .EQU    SPTR
    712000              SPTR    .EQU    SPTR+$DKLEN

013674  000125          UC1     $US1                    USER NAME
013675  712000                  $PTR                    PHYSICAL DISK LOCATION ABOVE THE BASE OF THE SCRATCH AREA
013676  001777                  BOUNDARY-1              CORE ADDRESS -1
013677  764000                  -$USLEN                 TWO'S COMPLEMENT LENGTH
    726000              SPTR    .EQU    SPTR+$USLEN

013700  000126                  $PH1
013701  726000          UP1     SPTR
013702  001777                  BOUNDARY-1
013703  776100                  -$PHLEN
    727700              SPTR    .EQU    SPTR+$PHLEN

013704  000124          TAB1    $UT1
013705  727700                  $PTR
013706  001677                  USTRT
013707  777700                  -$TABLEN
    730000              SPTR    .EQU    SPTR+$TABLEN

    013710      8680            INTUS   2               SET UP USER #2

    730000              UDK2    .EQU    SPTR
    746000              SPTR    .EQU    SPTR+$DKLEN

013710  000154          UC2     $US2                    USER NAME
013711  746000                  $PTR                    PHYSICAL DISK LOCATION ABOVE THE BASE OF THE SCRATCH AREA
013712  001777                  BOUNDARY-1              CORE ADDRESS -1
013713  764000                  -$USLEN                 TWO'S COMPLEMENT LENGTH
    762000              SPTR    .EQU    SPTR+$USLEN

013714  000155                  $PH2
013715  762000          UP2     SPTR
013716  001777                  BOUNDARY-1
013717  776100                  -$PHLEN
```

I CONSTANTS, TEMPORARY STORAGE, ETC

```
    763700              SPTR    .EQU    SPTR+$PHLEN

013720  000153          TAB2    $UT2
013721  763700                  SPTR
013722  001677                  USTRT
013723  777700                  -$TABLEN
    764000              SPTR    .EQU    SPTR+$TABLEN

                 8690
                 8700   *
                 8710   *       ALLOCATE CATALOG ROOM FOR ALL SYSTEM FILES
                 8720   *
    013723       8730   UCAT    .EQU    .-1             START OF THE CATALOG OF USER-TYPE SYSTEM FILES
    013724       8740           .BLOCK  PFILES-UFILES-1*4
    013733       8750   PCAT    .EQU    .-1
    013734       8760           .BLOCK  OFILES-PFILES-1*4*2  TWO FILE ENTRIES PER PHANTOM: <NAM> & <PNAM>
    000100       8770   CLEN    .EQU    .-INTCAT*4      LENGTH OF INTCAT
    013753       8780   OCAT    .EQU    .-1             START OF THE CATALOG OF OVERLAY FILES
    013754       8790           .BLOCK  EFILES-OFILES-1*4
                 8800
    013774       8810   LCAT    .BLOCK  CATLEN          LIBRARY DEVICE CATALOG ROOM
014374  002000   8820           .END    START
014375  445320
014376  000001
014377  000004
014400  422022
014401  422021
014402  422023
014403  422024
014404  422122
014405  140000
014406  640000
014407  674000
014410  000125
014411  730000
014412  000154
014413  000076
014414  770000
014415  010000
014416  000020
014417  001777
014420  001300
014421  007777
014422  600000
014423  000510
014424  000003
014425  012635
014426  000337
014427  000230
014430  000215
014431  777777
014432  000017
```

I CONSTANTS, TEMPORARY STORAGE, ETC

```
014433  000256
014434  000260
014435  777540
014436  000240
014437  000333
014440  000335
014441  000377
014442  777520
014443  777770
014444  777776
014445  777772
014446  777737
014447  777745
014450  740000
014451  000005
014452  613373
014453  000272
014454  000257
014455  777700
014456  646000
014457  446400
014460  445300
014461  606064
014462  606462
014463  606460
014464  777716
014465  000000

TRANSFER ADDRESS  612000
```

I CROSS REFERENCE TABLE

1713	.0	4510	4520										
26	.310	3400											
27	.311	3410											
4453	.DK	770											
4464	.DT	750											
6460	.TP	760											
2023	10SAVE	1870	1880										
2024	11SAVE	1880	1920										
5	3AC	3370											
305	3REST	3870	3880										
51	3TEM0	3530	3540										
52	3TEM1	3540	3550										
53	3TEM2	3550	3560										
54	3TEM3	3560	3570										
55	3TEM4	3570	3580										
56	3TEM5	3580	3590										
57	3TEM6	3590	3600										
50	3TM20	3520	3530										
2	3TM21	3350											
3	3TM22	3360											
14000	7K	1030											
16000	8K	1020	910	1010	2640	2650							
414263	ABS	850											
1753	AC	4520	4530										
1756	ACS	4550	4560										
2015	ACSAVE	1810	1820										
2022	ACSW	1860	1870										
17777	ADRSS	430	4110										
246	AMPRSN	290											
300	AT	2950											
300	ATSGN	2900											
422030	BAS	420	8060										
2151	BCNTRL	2330	2340										
425156	BIN	860											
634	BIT0	3970	3980										
641	BIT17	4020	4030										
635	BIT36	3980	3990										
636	BIT5	3990	4000										
637	BIT6	4000	4010										
640	BIT7	4010	4020										
337	BKARR	240	800										
642	BL7	4030	4040										
643	BL8	4040	4050										
400	BLKLEN	630	6260	6350									
1777	BLKMSK	610	6660	7000	7880	1090	1200						
17740	BOOT	520	530										
2000	BOUNDA	440	960	980	990	1000	1630	5040	5080	2260	4470	8280	8410	8470
			8660	8660	8670	8670	8680	8680					
377	BRK	5550											
334	BSLASH	390											
2170	BUFFER	2490	2550										
1000	BUFLEN	2500	2550										

I

CROSS REFERENCE TABLE

I CROSS REFERENCE TABLE

272	COLON	370	2210										
2053	COMFLG	2200	2210										
254	COMMA	320											
2150	COMSTO	2270	2280										
230	CONTX	230	820										
435762	COR	880											
12464	CORCPY	6500	3430	6530	7260								
16000	CORMAX	910	980										
13543	CPAPER	2840	2230	2620	2870	2890	2910	2920	2930				
17005	CPARAM	590	2790	1420									
215	CR	220	910	2420									
13315	CRCAT1	930											
13340	CRCAT4	1190	1320										
13365	CRCOVR	1490	3730	6280	6370	7500	7620	980	1170	1240	1410	1540	1580	1660
13373	CRCVR4	1560	1520										
13405	CRCVR5	1590	1620	1670									
13433	CRCVR6	1690	1640										
47	CSPL	3510	3520										
44	CSWP	3480	3490	3780									
60	CTBFR	3600	3630	3640									
100	CTBIN	3640	3650	3670	4250	4980	4990						
2000	CTEMP0	1630											
2001	CTEMP1	1640											
2002	CTEMP2	1650											
2003	CTEMP3	1660											
2004	CTEMP4	1670											
2005	CTEMP5	1680											
2006	CTEMP6	1690											
2007	CTEMP7	1700											
2010	CTEMP8	1710											
2011	CTEMP9	1720											
102	CTFLG	3650	3660										
436454	CTL	810											
104	CTNAM	3660											
2043	D PC	2120	2130										
2154	D BCA	2370	2380										
2153	D BDA	2360	2370										
2163	D FDA	2440	2450										
2042	D LOC	2110	2120										
2022	D ACSW	1860											
2156	D BALT	2390	2400										
2155	D BLEN	2380	2390										
2161	D BMAX	2420	2430										
2157	D BMIN	2400	2410										
2162	D BPTR	2430	2440										
2167	D FMAX	2480	2490										
2165	D FMIN	2460	2470										
2046	D MASK	2150	2160										
2164	D MFDA	2450	2460										
2036	DADRSW	2070	2080										
1762	DAP0	4590	4600										
1763	DAP1	4600	4610										

I CROSS REFERENCE TABLE

653	DBK	4120	4130											
24	DBKNUM	2220	2270											
2054	DBKTAB	2210	2270											
2035	DBSTOR	2050	2060											
422027	DDT	410	8050											
12000	DDTST	5000												
2037	DDUMSW	2080	2090											
1	DEBUG	2020	100	120	100	120	100	120	100	120				
1	DEFINS	2010	100	100	120	120	580	580	2680	2680	3210	3210	5330	5330
1761	DFLAG	4580	4590											
1764	DFN	4610	4620											
2151	DFTYPE	2340	2350											
2045	DHICOR	2140	2150											
2050	DINDIR	2170	2180											
445300	DK.	740	2690											
445320	DK0	780	2130											
127	DK1	4310												
156	DK2	4350												
37	DKCA	2750	4370											
675	DKDON	4170	4180											
16000	DKLEN	2650	2660	8660	8670	8680								
34	DKLENB	2660												
266	DKLOK	3830	3840	4790										
777601	DKMAX	650												
672	DKOVR	4160	4170	4410										
2	DKRD	2760												
36	DKWC	2740	4350											
4	DKWRT	2770	4400	5830										
2041	DLIMIT	2100	2110											
2044	DLOCOR	2130	2140											
2160	DMBMIN	2410	2420											
2166	DMFMIN	2470	2480											
654	DO	4130	4140	4390	5860									
662	DO2	4140	4150											
663	DO3	4150	4160											
2152	DOFTYP	2350	2360											
244	DOLLAR	280												
2032	DPACSW	1980												
2040	DPATSW	2090	2100											
2051	DPCMSK	2180	2190											
2052	DREGBR	2190	2200											
2035	DREGSW	2060	2070											
2047	DRELOC	2160	2170											
1765	DSTAT	4620	4630											
446400	DT.	720	2670											
2000	DTEMP0	1630												
2001	DTEMP1	1640												
2002	DTEMP2	1650												
2003	DTEMP3	1660												
2004	DTEMP4	1670												
2005	DTEMP5	1680												
2006	DTEMP6	1690												

I CROSS REFERENCE TABLE

2007	DTEMP7	1700											
2010	DTEMP8	1710											
2011	DTEMP9	1720											
776701	DTMAX	640											
740000	DVCMSK	600	880	950	1050	2770							
13647	EFILES	8150	7280	8790									
275	EQUAL	2910											
241	EXCLAM	260											
5	FCBLEN	570	3190	1290	1970								
602	FGET	3950	3960										
12000	FORMAT	2040	2390	2420	2610	2710							
1701	FRCA	4410	4420										
1700	FRDA	4400	4410										
1702	FRLEN	4420	4430										
1703	FRSTA	4430	4440										
2	FUDGE	3190	3200										
13511	GNAME6	2410	2360										
276	GREAT	2930											
476257	GRO	870											
4	HDRLEN	580											
100	I BUF	6020	8200										
12423	I LEN	6070	3070	3130	5630	6100	6160	6190	6200				
12557	I OV1	7300	7410										
13660	I UC0	8660	4770										
13674	I UC1	8670	4690										
13710	I UC2	8680	4730										
13665	I UP0	8660											
13701	I UP1	8670											
13715	I UP2	8680											
10000	I BMAX	6030	4040	6130	6170	6260	6350						
100	I CLEN	8770	3550										
12425	I COPL	6100	6400										
12424	I COPY	6090	3170	5690	6120	6250	6330						
12465	I COR2	6510	6550										
12472	I CRCP	6600	6540	6770	6880								
12623	I FILL	7840	3810	3830	3850	3870	7920						
12421	I INDA	6050	2720	3030	5520	6220	6270						
13774	I LCAT	8810	2150	2370	5190	5210							
13753	I OCAT	8780											
13733	I PCAT	8750											
12060	I PURL	2940	3180	3220									
12111	I PURZ	3190	2990										
12573	I READ	7460	3500	7530									
12351	I SAV1	5440	5700										
12350	I SAVE	5430	3290	3300	3310	5460							
764000	I SPTR	8680	8400	8460	8520	8660	8660	8660	8660	8660	8660	8660	8670
			8670	8670	8670	8670	8670	8670	8670	8680	8680	8680	8680	8680
			8680	8680	8680								
13670	I TAB0	8660	4770										
13704	I TAB1	8670	4690										
13720	I TAB2	8680	4730										
13723	I UCAT	8730	3410										

I CROSS REFERENCE TABLE

```
640000  I UDK0      8660  4770
674000  I UDK1      8670  4690
730000  I UDK2      8680  4730
 12337  I UNS1      5290  5330  5370
 12274  I.000,      4840
 12436  ICOPL2      6190  6150
 12440  ICOPL4      6210  6180
 12016  IINT00      2410  2440
 12027  IINT01      2530  2480
 12115  IINT03      3260  2960
 12143  IINT05      3570  3600
 12124  IINT10      3390
 12151  IINT50      3680
 12172  IINT55      4030  4230  4340
 12174  IINT56      4060  4360
 12176  IINT60      4080  4100
 12205  IINT70      4190  4380
 12220  IINT72      4320  4250
 12222  IINT74      4350  4300
 12231  IINT80      4450  4220
 12326  ILCATL      5150  3700  5170  5470
  1700  IMPLEN       990
  3170  IMPSTR      2550  7030  7110
    10  INDEX        490
422020  INT          320
 13660  INTCAT      8630  3530  8770
   513  IO.IN       3910  3920
   525  IO.OT       3920  3930
300000  IOBLK       2830
  1760  IORS        4570  4580
  1002  IOTQ        4900  4910
 12422  IOUTDA      6060  2730  3100  5660  6240  6310  6360
 12613  ISETUP      7710  7780
 12000  ISTART      2220  2030  2040  8820
 13650  ITEMP1      8160  2750  2940  3560  3590  4050  4090  5350  5360  5780  5800  5820  5840
                          5850
 13651  ITEMP2      8170  2400  2430  3050  3110  4160  4200  4270  4280  4330
 13652  ITEMP3      8180
 13653  ITEMP4      8190
 12404  IUINIT      5750  4690  4730  4770  5870
  1677  IUSTRT      8280  8530  8660  8670  8680
 12603  IWRITE      7580  3620  7650
   652  JMP         4110  4120
   100  JTLEN        960  4040
  1700  JTSTRT       950   940   960  1000  4400  4060
    16  KBLEN       3610  3630  3640  3680  3690  3730  3740
    10  KBNUM       3620  3670  3720
    76  LOLOK       3630
   107  L1BFR       3670  3680  3690
   127  L1BIN       3690  3700  3720  4290  5000  5010
   131  L1FLG       3700  3710
   125  L1LOK       3680
```

I CROSS REFERENCE TABLE

133	L1NAM	3710					
136	L2BFR	3720	3730	3740			
156	L2BIN	3740	3750	4330	5020	5030	
160	L2FLG	3750	3760				
154	L2LOK	3730					
162	L2NAM	3760	3770				
422026	LDR	390	8090				
2000	LDRST	5040					
274	LESS	2920					
212	LF	210	2440				
13774	LIBDEV	2150					
2022	M ACSW	1860					
10	MINBUF	3200	3610				
255	MINUS	330					
422023	MP1	350	3820	8120			
422024	MP2	360	3840	8130			
2032	MPACSW	1980					
1004	MPOPR	4920					
1000	MPST	4880	4890				
1754	MQ	4530	4540				
2016	MQSAVE	1820	1830				
2000	MTEMP0	1630					
2001	MTEMP1	1640					
2002	MTEMP2	1650					
2003	MTEMP3	1660					
2004	MTEMP4	1670					
2005	MTEMP5	1680					
2006	MTEMP6	1690					
2007	MTEMP7	1700					
2010	MTEMP8	1710					
2011	MTEMP9	1720					
422025	MTR	370	8080				
2000	MTRST	5080					
1772	NAME	4670	4680	4690	4730	4770	
540	NEWBR	3930	3940				
12000	NEXTL	2030	1700	3230	3510		
1771	NUMBR	4660	4670	4690	4730	4770	
243	NUM$GN	270					
623	NXPTR	3960	3970				
702	OC0	4180	4190				
703	OC1	4190	4200				
704	OC2	4200	4210				
705	OC3	4210					
574646	OFF	2730					
13642	OFILES	8100	7280	8760	8790		
575600	ON	2720					
12547	OVCRCP	7210	3450	7340			
1773	OVER	4680	4690	4460			
700	OVLEN	940					
1000	OVSTRT	930	920	940	4750	4880	4960
2033	P10SAV	1990	2000				
2034	P11SAV	2000	2050				

I CROSS REFERENCE TABLE

2025	PACSAV	1930	1940						
2032	PACSW	1980	1990						
241	PBFLAG	3810	3820						
2017	PCSAVE	1830	1840						
256	PERIOD	340	350						
13637	PFILES	8070	8740	8760					
227	PFLAG	3770	3780						
77	PH0	4260	4270	8660					
126	PH1	4300	4310	8670					
155	PH2	4340	4350	8680					
1	PHANTO	2780							
12515	PHCRC1	6850	7130						
12514	PHCRCP	6840	3440	6870					
2150	PHFLAG	2280	2330						
1700	PHLEN	2640	8480	8660	8660	8670	8670	8680	8680
2025	PHSTOR	1920	1930						
274	PIDN2	3850	3860						
270	PIDON	3840	3850						
1001	PINT	4890	4900						
303	PIOUT	3860	3870						
602026	PLDR	400							
253	PLUS	310							
2026	PMQSAV	1940	1950						
602025	PMTR	380							
256	POINT	350	1460						
2027	PPCSAV	1950	1960						
606064	PPT	690	2860						
2031	PSCSAV	1970	1980						
2030	PSTSAV	1960	1970						
606460	PTP	710	2900						
606462	PTR	700	2880						
12100	PURLEN	1010							
1775	PURNM	4700	4710						
3700	PURSTR	2560	990	1010	2560	7030	7050	7110	
546	PUTIN	3940	3950						
34	RACS	3440							
6	RCNT	3390							
35	RCORE	3450							
1003	RDBLK	4910	4920						
32	RDT0	3420							
33	RDT1	3430							
17505	RECOV	470	1530						
422021	RES	330	3690	2950	2950				
40	RESCAT	3470	3480						
1000	RESLEN	920							
234	RFLAG	3790	3800						
230	RPTP	3780	3790						
235	RPTR	3800	3810						
242	RSCO	3820	3830						
1776	RSTRT	4710	4480						
1755	SC	4540	4550						
273	SCOLON	380							

I CROSS REFERENCE TABLE

640000	SCRSTR	2670	4150	4170	8290								
2021	SCSAVE	1850	1860										
243	SHARP	2890											
257	SLASH	360	2350										
240	SPACE	250	2690										
377	SPCOD	5410											
422122	SPL	430	3860	8140									
1000	SPLST	4960											
777400	SPMSK	5390											
252	STAR	300											
2020	STSAVE	1840	1850										
335	SWAP	3880	3890										
336	SWAP1	3890	3900										
340	SWAP3	3900	3910										
1000	SWCAT	4750	4760	3510									
1003	SWCLK	4780	4790										
1004	SWERR	4790	4800										
1007	SWMP1	4820	4830										
1010	SWMP2	4830	4840										
1002	SWMTR	4770	4780	5070									
1011	SWOPR	4840											
422022	SWP	340	3490	3610	3800	8110							
1001	SWPPR	4760	4770										
40	SWP$	3460	3470										
1005	SWSPL	4800	4810										
1006	SXSPL	4810	4820										
1300	SYSBAS	2800	2810	6670	7010	7890							
41300	SYSDA	2810											
17735	SYSDEV	530											
445320	SYSDSK	2130	2500										
13657	SYSDVC	8210	2540	2710									
1777	SYSMAX	2820											
12760	T T1	560	1280	1400	1710	1810	1990	2050	2080	2090	2110	2120	2130	2290
			2460											
12761	T T2	570	1180	1260	1480									
12773	T IN1	780	930	1020	1050									
13001	T IN4	840												
12767	T INL	740	990	1010										
120	T STD	480	490											
12756	T BEND	540	860											
12757	T BPTR	550	750	850	880	900	950	1000	1040	2590	2600	2760	2770	
12762	T CHAR	580	2610											
13270	T CRLF	3320	980	3410										
13253	T DLMR	3130	2980	3070										
13204	T FGET	2580	1210	1530	2200	2620	2680							
13131	T SIX2	2030	2140											
13147	T SIX5	2190	2030	2060	2100	2250								
13145	T SIX9	2160	2220											
13020	T1CHAR	1000	810											
13016	T1LINE	980	830											
100	TABLEN	2630	2640	8280	8540	8660	8660	8670	8670	8680	8680			
17500	TAPIN	450	3750	6300	7520	1000								

I CROSS REFERENCE TABLE

17502	TAPOT	460	6390	7640	1430					
12636	TBUFFR	490	740	940						
13230	TCHRID	2930	1220	1540	2210	2710	3150	3170	3180	3200
12764	TCOUNT	600	760	2650						
13256	TDIGIT	3170	3010	3040						
12763	TDLMTR	590	770	3140	2200	2340				
2000	TEMP0	1630	1640							
2001	TEMP1	1640	1650							
2012	TEMP10	1730	1740							
2013	TEMP11	1740	1750							
2014	TEMP12	1750	1800							
2002	TEMP2	1650	1660							
2003	TEMP3	1660	1670							
2004	TEMP4	1670	1680							
2005	TEMP5	1680	1690							
2006	TEMP6	1690	1700							
2007	TEMP7	1700	1710							
2010	TEMP8	1710	1720							
2011	TEMP9	1720	1730							
13210	TFGET9	2620								
12765	TINLIN	720	960	1600						
13211	TINTIN	2640	1190	2000	2720	2740	2780			
13257	TLETTR	3180	3110							
13032	TNUM20	1210	1430							
13056	TNUM26	1460	1230							
13065	TNUM27	1530	1470							
13073	TNUM28	1590	1550							
13074	TNUM29	1600	1520							
13026	TNUMIN	1170	1200	1240	1500	1560	1580	1610	1620	2410
13100	TOCT42	1670								
13110	TOCT44	1750	1820							
13077	TOCTOT	1660	1850							
646000	TP.	730	2650							
13654	TPARAM	8200	6210	6230	6290	6340	6380			
376	TRCOFF	5540								
375	TRCON	5530								
13160	TSIX24	2300	2360							
13167	TSIX26	2380	2330	2340	2350	2470				
13123	TSIXIN	1970	2010	2020	2170	1610	2180	2320	2690	
13156	TSIXOT	2280	4840	2300	2320	2500	1560	1590	3220	3500
2000	TTEMP0	1630								
2001	TTEMP1	1640								
2002	TTEMP2	1650								
2003	TTEMP3	1660								
2004	TTEMP4	1670								
2005	TTEMP5	1680								
2006	TTEMP6	1690								
2007	TTEMP7	1700								
2010	TTEMP8	1710								
2011	TTEMP9	1720								
13262	TTTYOT	3240	1800	2450	2950	3130	3190	3290	3340	3360	3380
6	TTYCLK	3170	3180							

I CROSS REFERENCE TABLE

3	TTYNUM	3140												
10	TTYSPD	3150	3170											
12634	TWORDB	470	2510	5160	5530	6630	6950	1310	1330	1350	1410	1510	1590	1600
			1980	2160	2660	2670	1880	1930	2280	2370	2430	2750	2850	3160
1774	TYPE	4690	4700											
1766	UCORE	4630	4640	4690	4730	4770								
1767	UDISK	4640	4650	4690	4730	4770								
13634	UFILES	8040	2600	3270	3390	8740								
12336	UNSAVE	5280	2620	2630	2640	5310								
336	UPARR	2940												
76	US0	4250	4260	4280	4770	8660								
125	US1	4290	4300	4320	4690	8670								
154	US2	4330	4340	4360	4730	8680								
0	USER	2790												
3	USERS	2850	3200											
14000	USLEN	980	2640	8420	8660	8660	8670	8670	8680	8680				
2015	USTORE	1800	1810											
75	UT0	4280	8660											
124	UT1	4320	8670											
153	UT2	4360	8680											
1704	UTEM0	4440	4450											
1705	UTEM1	4450	4460											
1706	UTEM2	4460	4470											
1707	UTEM3	4470	4480											
1710	UTEM4	4480	4490											
1711	UTEM5	4490	4500											
1712	UTEM6	4500	4510											
1770	VALID	4650	4660											
17777	VFLAG	480												

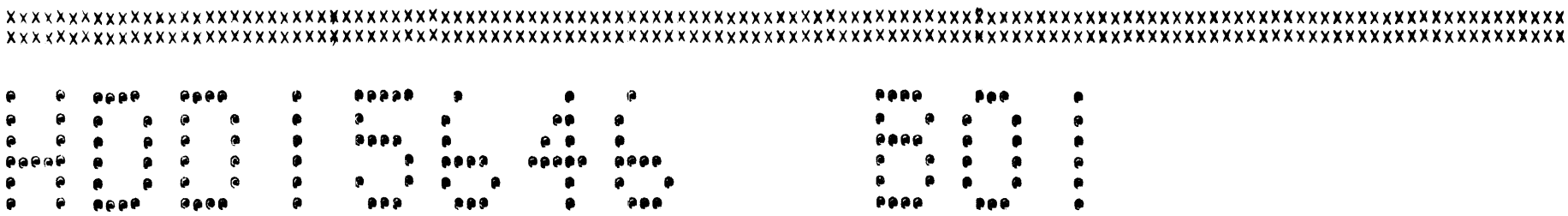
I UNDEFINED SYMBOLS

#1	5630			
#2	5640			
#3	5650			
#4	5660			
#5	5680			
LINE	1210			
MESS	1190	1200		
OCTZ	1790			
PH#1	8450			
PURCOD	5140	5270	430	610
US#1	8390			
UT#1	8510			

I

MACRO CROSS REFERENCE TABLE

CHAR	1380												
CHROT	1340												
COUNT	1460												
CRLF	1300												
DELIM	1420	2200	2340										
EMESS	1750												
ENTER	920	5150	5280	5430	5750	6500	6600	6840	7210	7460	7580	7710	7840
		720	1170	1660	1970	2190	2280	2380	2580	2640	2930	3240	3320
		2840											
FORMAT	1060	2390	2420	2610	2710								
INTT	4530	4690	4730	4770									
INTUS	8330	8660	8670	8680									
LINE	1100	1600											
LOOP	960												
MESS	1620	4840	1560	1590	3220	3500							
MESSR	1520	4840	1560	1590	3220	3500							
MPOFF	5430												
NEG	1010												
NMESS	1670												
NUM	1260	2410											
OCT	1890												
OCTZ	1840												
START	1100												
SWAP	5610												
WORD	1140	1610	2180	2320	2600								
WORD1	1180	5530	6630	6950	1930	2280	2370	2750	2850	3160			
WORD2	1220												



05/31/72 01:49:17

```
**********************************
**********************************
**                              **
**  PDP-9 MINI TIME-SHARING SYSTEM  **
**        RESIDENT PROGRAM      **
**            DTSS:RES          **
**            MTSS:B01          **
**                              **
**********************************
**********************************
```

```
100             .TITLE  PDP-9 MINI TIME-SHARING SYSTEM RESIDENT EXECUTIVE PROGRAM
110             .NAME   RES--B01
120             .ABS
130             .PMC    ON
140             .INSRT  DEFINS
100             .IFUND  DEFINS
```

```
5720            .LIST   ON
5730            .END
150
160
170
180
190
200     *
210     *       THE PDP-9 MINI TIME-SHARING SYSTEM RESIDENT PROGRAM CONTAINS A NUMBER OF
220     *       SOMEWHAT DISJOINT ITEMS. MAINLY IT CONTAINS:
230     *
240     *            1) RESIDENT STORAGE:
250     *                 A) TEMPORARY STORAGE USED BY THE RESIDENT PROGRAM ITSELF
260     *                 B) A SET OF RESIDENT PARAMETERS FOR EACH TELETYPE
270     *                 C) A TELETYPE I/O BUFFER FOR EACH TELETYPE
280     *                 D) STORAGE TO RECORD ALLOCATION OF VARIOUS RESOURCES
290     *                 E) FLAGS GIVING INFORMATION ON THE STATE OF THE SYSTEM
300     *                 F) SOFTWARE-IMAGE FLAGS FOR CERTAIN HARDWARE DEVICES (E.G. PAPER TAPE PUNCH)
310     *                 G) CATALOG INFORMATION TO RETRIEVE THE SWAPPER OVERLAY
320     *
330     *            2) ROUTINES TO HANDLE TELETYPE INPUT AND OUTPUT
340     *
350     *            3) ROUTINES TO SERVICE OTHER PROGRAM INTERRUPTS, INCLUDING THE CLOCK AND UNWANTED INTERRUPTS
360     *
370     *            4) A PHYSICAL DISK HANDLER
380     *
390     *            5) A ROUTINE TO SERVICE CAL INSTRUCTION ERRORS AND TO PREVENT CAL
400     *               (INDIRECT) FROM CRASHING THE SYSTEM.
410     *
420     *       ALSO IN THIS LISTING IS A COMPLETE LISTING OF THE CONTENTS OF USER JOB TABLES
430     *       AND PHANTOM JOB TABLES.
440     *
```

INITIALIZE LOCATIONS 0-37

```
                    450             .STITL  INITIALIZE LOCATIONS 0-37
                    460             .HEAD   R
                    470     *
                    480     *       LOCATIONS 0-37 CONTAIN <JMP .> UNTIL SOMETHING ELSE IS PUT THERE.
                    490     *       THIS IS PURELY AS A DEBUGGING AID TO TRAP MISCELLANEOUS TRANSFERS.
                    500     *
          000000    510             .LOC    0
                    520             .DET    SAVE,CFF
                    530             .DUP    1,40
    000000  600000  540             JMP     .
                    550             .DET    RESTORE
                    560     *
                    570     *       NORMALLY WHEN A USER'S PROGRAM IS RUNNING, LOCATIONS 0 & 10-17 CONTAIN
                    580     *       WHAT HE THINKS THEY DO. THIS IS TO PERMIT HIS INDIRECT REFERENCES
                    590     *       THROUGH HIS AUTO-INDEX REGISTERS
                    600     *       AND LOCATION 0 TO WORK PROPERLY.
                    610     *
                    620     *       LOCATIONS 1-7 & 20-37 ARE THEREFORE AVAILABLE FOR SYSTEM USE.
                    630     *
                    640     *
                    650     *       A PROGRAM INTERRUPT HAS OCCURRED -- GO SERVICE IT
                    660     *
          000001    670             .LOC    1
    000001  600165  680             JMP     PISVC
```

R RESOURCE ALLOCATION AND TEMPORARY STORAGE

```
                    690             .STITL  RESOURCE ALLOCATION AND TEMPORARY STORAGE
                    700     *
                    710     *       TEMPORARY VARIABLES STORAGE AND CONTROL LINE ALLOCATION RECORD
                    720     *
000002  000000      730     3TM21   .DSA
000003  000000      740     3TM22   .DSA
000004  000000      750     310TM   .DSA
000005  000000      760     3AC     .DSA
000006  000000      770     CNTRL   .DSA
                    780
                    790     *
                    800     *       CURRENTLY CAL'S ARE NOT PERMITTED AT ALL, THERE IS A SPECIAL IOT INSTRUCTION
                    810     *       PROVISION FOR SYSTEM SERVICES, ALL CAL'S AUTOMATIC ERROR MESSAGE PRINTOUT
                    820     *       (PROVIDED BY THE USER'S MEMORY PROTECT OVERLAY).
                    830     *
                    840     *       EVENTUALLY USER CAL'S SHOULD BE HANDLED MUCH LIKE USR PI --
                    850     *       BASICALLY DO AN XCT OF USER LOCATION 21, SYSTEM CAL'S ARE
                    860     *       AN ERROR ANYWAY -- THE SYSTEM CURRENTLY DOES NOT USE ANY CAL'S.
                    870     *
    000020          880             .LOC    20
000020  000020      890             20                      LOCATION 20 MUST ALWAYS CONTAIN AN ADDRESS FIELD OF 20 AS CAL,X PROTECTION
000021  200020      900             LAC     20              LOAD THE USER'S PC
000022  040000      910             DAC     0               AND SAVE IT FOR THE ERROR MESSAGE ROUTINE
000023  760020      920             LAW     20              RELOAD THE CAL,X PROTECTION
000024  040020      930             DAC     20              AND SET IT
000025  600342      940             JMP     ERRCAL          GO PRINT THE ERROR MESSAGE
                    950     *
                    960     *       TEMPORARY VARIABLE STORAGE
                    970     *
000026  000000      980     .310    .DSA
000027  000000      990     .311    .DSA
                    1000    *
                    1010    *       LOCATIONS 30-37 ARE THE DATA CHANNEL CELLS (30-31 FOR DECTAPE AND
                    1020    *       36-37 FOR THE DISK -- 32-35 ARE CURRENTLY UNUSED). CONSEQUENTLY
                    1030    *       30-31 & 36-37 MUST BE KEPT FREE, BUT 32-35 ARE AVAILABLE FOR
                    1040    *       SYSTEM USE.
                    1050    *
    000030          1060            .LOC    30
000030  776031      1070            -2000+.+1               DECTAPE WORD COUNT SO AS NOT TO DISTURB A DECTAPE READ-IN
                    1080    *
                    1090    *       TABLE OF AVAILABLE RESOURCES WHICH CAN ONLY BE ASSIGNED TO ONE
                    1100    *       USER AT A TIME, EACH RESOURCE'S ENTRY IS ZERO IF THE RESOURCE IS
                    1110    *       CURRENTLY UNASSIGNED, AND CONTAINS A POINTER TO THE USER'S RESIDENT
                    1120    *       PARAMETERS IF IT IS ASSIGNED.
                    1130    *
    000032          1140            .LOC    32
000032  000000      1150    RDT0    .DSA                    DECTAPE HANDLER
000033  000000      1160    RDT1    .DSA                    OTHER DECTAPE HANDLER
000034  000000      1170    RACS    .DSA                    ACCUMULATOR SWITCHES
                    1180    *
                    1190    *       RCORE CONTAINS THE STATUS OF NON-PROTECTED CORE. IT CONTAINS
                    1200    *       THE USER NUMBER OF THE CURRENTLY ACTIVE JOB, IF THERE IS ONE.
```

R RESOURCE ALLOCATION AND TEMPORARY STORAGE

```
                    1210    *       ELSE IT IS ZERO, IF RCORE IS NON-ZERO, IT IS ASSUMED THERE
                    1220    *       IS IN CORE A MEMORY PROTECTION OVERLAY TO GO WITH THE ACTIVE
                    1230    *       JOB. IN THIS CASE THE EXEC WILL FEEL FREE TO JUMP TO THE OVERLAY
                    1240    *       AREA TO ACCESS ROUTINES (E.G. TO CHECK THE NEED TO GENERATE A
                    1250    *       USER PROGRAM INTERRUPT.)
                    1260    *
000035  000000      1270    RCORE   .DSA                    NON-PROTECTED CORE STATUS -- 0= NO ACTIVE USER; ELSE CONTAINS USER NUMBER
                    1280
                    1290    *
                    1300    *       RESIDENT CATALOG -- ALL OVERALY FILES ARE CATALOGED HERE TO MINIMIZE SWAPPER USAGE.
                    1310    *
000040              1320            .LOC    40
000040  000000      1330    SWPS    .DSA                    PHYSICAL DISK ADDRESS OF THE OVERLAY
000041  000777      1340            $OVSTRT-1               OVERLAY FILE CORE ADDRESS -1
000042  777100      1350            -$OVLEN                 OVERLAY FILE (TWO'S COMPLEMENT) WORD COUNT
000043  000002      1360            $DKRD                   DISK READ COMMAND
                    1370
000044  000000      1380    CSWP    .DSA                    SWAPPER PHYSICAL DISK ADDRESS
000045  000000      1390    CMP1    .DSA                    MEMORY PROTECTION #1 PHYSICAL DISK ADDRESS
000046  000000      1400    CMP2    .DSA                    MEMORY PROTECTION #2 PHYSICAL DISK ADDRESS
000047  000000      1410    CSPL    .DSA                    SPECIAL IOT HANDLER PHYSICAL DISK ADDRESS
                    1420    *
                    1430    *
                    1440    *       TEMPORARY VARIABLE STORAGE
                    1450    *
000050  000000      1460    3TM20   .DSA
000051  000000      1470    3TEM0   .DSA
000052  000000      1480    3TEM1   .DSA
000053  000000      1490    3TEM2   .DSA
000054  000000      1500    3TEM3   .DSA
000055  000000      1510    3TEM4   .DSA
000056  000000      1520    3TEM5   .DSA
000057  000000      1530    3TEM6   .DSA
```

R TELETYPE BUFFERS AND CONSTANTS

```
                    1540            ,STITL  TELETYPE BUFFERS AND CONSTANTS
                    1550    *
                    1560    *       KEYBOARD BUFFERS MUST BE OF A CERTAIN MINIMUM SIZE (SEE DEFINITIONS).
                    1570    *       HOWEVER, THE LARGER THEY CAN BE, THE BETTER. TELEPRINTER I/O ROADBLOCK
                    1580    *       OCCURS WHEN THE BUFFER IS FULL OF OUTPUT AND IS NOT RELIEVED UNTIL THERE ARE
                    1590    *       ONLY ENOUGH CHARACTERS REMAINING IN THE BUFFER TO COVER THE WORST CASE
                    1600    *       REMAINING TIME UNTIL THE PROGRAM COULD GET BACK INTO CORE TO PUT MORE
                    1610    *       OUTPUT IN THE BUFFER. THUS THE TELEPRINTER IS KEPT CONTINUALLY BUSY PRINTING
                    1620    *       AS LONG AS THE PROGRAM HAS OUTPUT TO PRINT.
                    1630    *
                    1640    *       *****WARNING*****: CTNAM (& L1NAM & L2NAM) SERVE MANY FUNCTIONS.
                    1650    *       THEY ARE OFTEN REFERRED TO AS $3TEM4. THEY SERVE AS THE NAME
                    1660    *       OF THE USER'S CORE-IMAGE FILE ON THE DISK, WITH A (1) ADDED TO
                    1670    *       THEM THEY SERVE THE SAME PURPOSE FOR THEIR OWN PHANTOM PROGRAMS.
                    1680    *       THEY ARE POINTERS NOT ONLY TO THE END OF THEIR TELETYPE BUFFERS
                    1690    *       +1, BUT ALSO TO THE START OF THE RESIDENT PARAMETER LIST-1.
                    1700    *       WITH ONE OF THEM IN THE AC A <JMS $IO,IN> INSTRUCTION WILL SET
                    1710    *       UP THAT USER'S I/O PARAMETERS IN THE TEMPORARY VARIABLES
                    1720    *       USED FOR EXECUTIVE RE-ENTRANCE.
                    1730    *
                    1740    *       I WOULDN'T BE SURPRISED IF THERE ARE ALSO OTHER FUNCTIONS I AM FORGETTING.
                    1750    *
    000060          1760    CTBFR   ,BLOCK  SKBLEN
000076  000000      1770    LQLOK   ,DSA                    CONSOLE TELETYPE MONITOR REQUEST
000077  777770      1780            -SKBNUM                 NUMBER OF PARAMETERS FOLLOWING
000100  000060      1790    CTBIN   CTBFR                   BIT 0 = COUNT ALREADY IN; BITS 5-17 = ACTIVE ADDRESS
000101  000060      1800            CTBFR                   BIT 0 = COUNT ALREADY OUT; BITS 5-17 = ACTIVE ADDRESS
000102  000000      1810    CTFLG   ,DSA                    SOFTWARE TELETYPE I/O FLAG. BIT:
                    1820                                        0: OUTPUT-IN-PROGRESS FLAG
                    1830                                        1: TELEPRINTER I/O ROADBLOCKED FLAG
                    1840                                        2: KEYBOARD I/O ROADBLOCKED FLAG
                    1850                                        3: KEYBOARD FLAG
                    1860                                        4: TELEPRINTER FLAG
                    1870                                        5: PI INTERRUPT PENDING
                    1880                                        6: KEYBOARD CHARACTER ECHOED FLAG (0=YES  1=NEEDS AN ECHO)
                    1890                                        7: I/O BUFFER TYPE ( 0=INPUT  1=OUTPUT)
                    1900                                        10-17) KEYBOARD BUFFER
000103  000060      1910            CTBFR                   (CONSTANT) START OF CONTROL TELETYPE BUFFER
000104  000076      1920    CTNAM   CTBIN-2                 (CONSTANT) END OF CONSOLE TELETYPE BUFFER +1 -- ALSO SERVES AS USER IDENTITY
000105  700406      1930            TLS                     (CONSTANT) PRINT INSTRUCTION FOR CONSOLE TELETYPE
000106  741000      1940            SKP                     CONSOLE TELETYPE WILL NOT NEED TO ECHO ANY CHARACTERS BY SOFTWARE
                    1950
    000107          1960    L1BFR   ,BLOCK  SKBLEN
000125  000000      1970    L1LOK   ,DSA
000126  777770      1980            -SKBNUM
000127  000107      1990    L1BIN   L1BFR                   PARAMETER LIST SAME AS FOR CT ABOVE
000130  000107      2000            L1BFR
000131  100000      2010    L1FLG   100000                  START IN KEYBOARD I/O ROADBLOCK CONDITION
000132  000107      2020            L1BFR
000133  000125      2030    L1NAM   L1BIN-2
000134  704006      2040            TLSLT1
000135  704006      2050            TLSLT1
```

R TELETYPE BUFFERS AND CONSTANTS

```
                    2060
    000136          2070    L2BFR   .BLOCK  $KBLEN
000154  000000      2080    L2LOK   .DSA
000155  777770      2090            -$KBNUM
000156  000136      2100    L2BIN   L2BFR
000157  000136      2110            L2BFR
000160  100000      2120    L2FLG   100000
000161  000136      2130            L2BFR
000162  000154      2140    L2NAM   L2BIN-2
000163  704026      2150            TLSLT2
000164  704026      2160            TLSLT2
```

R PROGRAM INTERRUPT SYSTEM ENTRANCE ROUTINE

```
                    2170            .STITL  PROGRAM INTERRUPT SYSTEM ENTRANCE ROUTINE
                    2180    *
                    2190    *
                    2200    *       WHEN A PROGRAM INTERRUPT OCCURS, CONTROL IS ALWAYS TRANSFERRED TO HERE.
                    2210    *       SAVE REGISTERS AC, 10, AND 11 ON INTERRUPTS, ROUTINES USING MQ OR SC MUST SAVE THEIR OWN.
                    2220    *
    000165          2230    PISVC   ...
000165  040005      2240            DAC     3AC
000166  200010      2250            LAC     10
000167  040026      2260            DAC     .310
000170  200011      2270            LAC     11
000171  040027      2280            DAC     .311
                    2290    *
                    2300    *       WHEN EXIT FROM SERVICING THIS PROGRAM INTERRUPT FINALLY OCCURS,
                    2310    *       IT WILL BE WITH THE SEQUENCE:
                    2320    *            DBK
                    2330    *            JMP <USER CORE>
                    2340    *       UNLESS THE INTERRUPT WAS CAUSED BY THE USER TRYING TO DO A DBR
                    2350    *       INSTRUCTION, IF HE HAS DONE A LEGAL DBR INSTRUCTION THE EXIT
                    2360    *       SEQUENCE WILL HAVE THE DBK REPLACED BY A DBR, SO THAT THE RESTORE
                    2370    *       EFFECT WILL OCCUR WHEN THE USER EXPECTS IT TO.
                    2380    *       IN EITHER CASE THE STATE OF THE MACHINE WILL BE RESORED BY
                    2390    *       EXECUTIVE SOFTWARE IMMEDIATELY PRIOR TO EXIT.
                    2400    *
                    2410    *       IF THE USER TRIED TO DO A DBR INSTRUCTION, THE MEMORY PROTECT
                    2420    *       ROUTINES WILL GIVE AN ERROR MESSAGE IF IT CAN HURT THE SYSTEM.
                    2430    *       (CURRENTLY THEY REQUIRE A DBR TO BE FOLLOWED BY A JMP (INDIRECT)
                    2440    *       THROUGH A WORD WITH THE MEMORY PROTECT BIT ON.)
                    2450    *
                    2460    *       THE EFFECT OF ALL OF THIS IS TO PERMIT A REASONABLE AMOUNT OF
                    2470    *       TRANSPARENCY TO BE PRESERVED FOR USER PROGRAMS RUNNING WITH
                    2480    *       THE PROGRAM INTERRUPT SYSTEM ON, WHILE NOT ALLOWING AN UNWANTED
                    2490    *       RESTORE FUNCTION TO BE LEFT HANGING AROUND, IF ONE WERE LEFT,
                    2500    *       A NOT-VERY-CLEVER USER COULD CRASH THE SYSTEM.
                    2510    *
000172  200665      2520            LAC     DBK
000173  040303      2530            DAC     PIOUT           SET THE STANDARD EXIT -- PROGRAM WILL RESTORE THE LINK
                    2540    *
                    2550    *       NOW FIND OUT WHAT CAUSED THE INTERRUPT
                    2560    *       IF THE INTERRUPT WAS CAUSED BY A HARDWARE INTERRUPT FLAG SETTING,
                    2570    *       CLEAR IT AND TRANSFER TO A ROUTINE TO SET THE CORRESPONDING SOFTWARE
                    2580    *       FLAG AND TEST FOR A USER PROGRAM INTERRUPT.
                    2590    *
000174  701713      2600            MPSKIMPCV+10
000175  741000      2610            SKP
000176  601000      2620            JMP     SMPST           MEMORY PROTECTION VIOLATION
000177  700001      2630            CLSF
000200  741000      2640            SKP
000201  600317      2650            JMP     CLK             SERVICE A CLOCK INTERRUPT
000202  700403      2660            TSF!TCF
000203  741000      2670            SKP
000204  600347      2680            JMP     CTOUT           CONSOLE TELEPRINTER
```

R PROGRAM INTERRUPT SYSTEM ENTRANCE ROUTINE

```
000205  700313  2690            KSF!KRB
000206  741000  2700            SKP
000207  600346  2710            JMP     CTKBD           CONSOLE KEYBOARD
000210  704113  2720            KSFLT1!KRBLT1
000211  741000  2730            SKP
000212  600351  2740            JMP     L1KBD           LT#1 KEYBOARD
000213  704133  2750            KSFLT2!KRBLT2
000214  741000  2760            SKP
000215  600354  2770            JMP     L2KBD           LT#2 KEYBOARD
000216  704003  2780            TSFLT1!TCFLT1
000217  741000  2790            SKP
000220  600352  2800            JMP     L1OUT           LT#1 TELEPRINTER
000221  704023  2810            TSFLT2!TCFLT2
000222  741000  2820            SKP
000223  600355  2830            JMP     L2OUT           LT#2 TELEPRINTER
000224  700203  2840            PSF!PCF
000225  600231  2850            JMP     .+4
000226  100246  2860            JMS     FLAG            SERVICE THE PAPER TAPE PUNCH INTERRUPT
000227  000000  2870    PFLAG   ,DSA                    SOFTWARE PAPER TAPE PUNCH FLAG
000230  000000  2880    PPTP    ,DSA                    PAPER TAPE PUNCH ALLOCATION WORD
000231  700103  2890            RSF!RCF
000232  600236  2900            JMP     .+4
000233  100246  2910            JMS     FLAG            SERVICE THE PAPER TAPE READER INTERRUPT
000234  000000  2920    RFLAG   ,DSA                    SOFTWARE PAPER TAPE READER FLAG
000235  000000  2930    RPTR    ,DSA                    PAPER TAPE READER ALLOCATION WORD
000236  704405  2940            SPB!CPB
000237  600243  2950            JMP     .+4
000240  100246  2960            JMS     FLAG            SERVICE GRAPHICS II PUSHBUTTONS INTERRUPT
000241  000000  2970    PBFLG   ,DSA                    SOFTWARE PUSHBUTTONS FLAG
000242  000000  2980    RSCO    ,DSA                    GRAPHICS II ALLOCATION WORD
000243  707001  2990            DSSF
000244  600267  3000            JMP     PISV2           IT IS AN UNKNOWN INTERRUPT
000245  600253  3010            JMP     DKSVC           SERVICE A DISK INTERRUPT
                3020    *
                3030    *       RECORD THE HARDWARE FLAG IN THE CORRESPONDING SOFTWARE FLAG.
                3040    *       THE SKIP CHAIN WILL HAVE  ALREADY CLEARED THE HARDWARE FLAG.
                3050    *       THEN TEST FOR A USER PROGRAM INTERRUPT.
                3060    *
    000246      3070            ENTER   FLAG            SAVE THE TYPE OF INTERRUPT HERE
                                ,PMC    SAVE,ON
000246  740040                FLAG    XX
                                ,PMC    RESTORE
000247  460246  3080            INX     FLAG,X          SET THE SOFTWARE FLAG
000250  440246  3090            INX     FLAG            BUMP THE TABLE POINTER
000251  220246  3100            LAC     FLAG,X          LOAD THE ALLOCATION WORD
000252  600411  3110            JMP     PITST           SEE WHETHER OR NOT TO GENERATE A USER PROGRAM INTERRUPT
```

R PROGRAM INTERRUPT SYSTEM ENTRANCE ROUTINE

```
                    3120            .EJECT
                    3130    *
                    3140    *       AN INTERRUPT FROM A USER DISK OPERATION HAS BEEN RECEIVED.
                    3150    *       SAVE ALL OF HIS INFORMATION, AND CLEAR THE DISK FOR POSSIBLE SYSTEM
                    3160    *       DISK USE.
                    3170    *
000253  140266      3180    DKSVC   DZM     DKLOK           CLEAR THE USER-USING-DISK FLAG
000254  707272      3190            DSRS+10
000255  041765      3200            DAC     DSTAT           SAVE THE USER'S DISK STATUS REGISTER
000256  441761      3210            INX     DFLAG           SET THE USER'S DISK FLAG
000257  707023      3220            DSCCIDRAL               DISABLE POSSIBLE DISK FREEZE THROWN IN ON GENERAL PRINCIPLES
000260  041762      3230            DAC     DAP0            SAVE THE USER'S DISK REGISTER AP0
000261  707242      3240            DSCD                    CLEAR THE STATUS REGISTER AND DISK FLAG
                    3250    *
                    3260    *       WE HAVE HAD THE DISK LOCKED OUT FOR A WHILE, NOW. SO SEE IF A CLOCK
                    3270    *       INTERRUPT OCCURRED IN THE MEANTIME.
                    3280    *
000262  200316      3290            LAC     CLKLOK          LOAD THE CLOCK INTERRUPT RECORD
000263  741200      3300            SNA                     SKIP IF THERE HAS BEEN ONE
000264  600270      3310            RET     PIDON           IF NONE, EXIT NORMALLY
000265  600327      3320            JMP     CLKST           IF THERE HAS BEEN ONE, DELIVER IT NOW
                    3330
000266  000000      3340    DKLOK   .DSA                    MINUS (SYSTEM-USING-DISK); PLUS (USER-USING-DISK); OR ZERO (DISK FREE)
                    3350    *
                    3360    *
                    3370    *       AN UNKNOWN INTERRUPT OCCURRED -- CLEAR REMAINING FLAGS AND IGNORE IT
                    3380    *
    000267          3390    PISV2   ...
000267  703302      3400            CAF                     GET A CLEAN START
                    3410    *
                    3420    *
                    3430    *       WE ARE DONE SERVICING THE LATEST PROGRAM INTERRUPT, NOW FIX THINGS UP
                    3440    *       AND RETURN TO THE USER.
                    3450    *
                    3460    *       RESTORE THE REGISTERS ON EXIT
                    3470    *
000270  200026      3480    PIDON   LAC     .310
000271  040010      3490            DAC     10
000272  200027      3500            LAC     .311
000273  040011      3510            DAC     11
                    3520    *
                    3530    *       SET UP THE ADDRESS, LINK AND MEMORY PROTECT FOR THE RETURN
                    3540    *
000274  200000      3550    PIDN2   LAC     0
000275  500663      3560            AND     ADRSS
000276  340664      3570            TAD     JMP
000277  040304      3580            DAC     PIGO            SET UP RETURN (DIRECT)
000300  100305      3590            JMS     3REST           RESTORE LINK, USER LOCATION 0, AC, AND MEMORY PROTECT
000301  701742      3600            MPEV                    RETURN HERE IF MEMORY PROTECT BIT (BIT 2) WAS ON
000302  700042      3610            ION                     RETURN HERE IF MEMORY PROTECT BIT (BIT 2) WAS OFF
000303  740040      3620    PIOUT   XX                      DBK (UNLESS MEM PRO DBR INTERRUPT -- THEN DBR)
000304  740040      3630    PIGO    XX                      RETURN (DIRECT)
```

R PROGRAM INTERRUPT SYSTEM ENTRANCE ROUTINE

```
                    3640    *
                    3650    *       SUBROUTINE TO RESTORE THE USER'S LINK, AC, AND LOCATION 0.
                    3660    *       THE SUBROUTINE RETURNS TO THE ENABLE USER MODE (MPEU) INSTRUCTION
                    3670    *       ONLY IF USER MODE WAS ENABLED WHEN THE INTERRUPT OCCURRED.
                    3680    *       OTHERWISE THAT INSTRUCTION IS SKIPPED, THIS SUBROUTINE IS
                    3690    *       NECESSARY TO CORRECTLY ACCOMPLISH THE EXIT BY THE SEQUENCE:
                    3700    *            DBK
                    3710    *            JMP <USER CORE>
                    3720    *
    000305          3730    3REST   ENTER
                                    .PMC    SAVE,ON
000305  740040                      XX
                                    .PMC    RESTORE
000306  200000      3740            LAC     0               LOAD THE STATE OF THE MACHINE WHEN INTERRUPTED
000307  742010      3750            RTL                     MOVE THE MEMORY PROTECT BIT TO AC(0)
000310  740120      3760            SMA!RAR                 RESTORE THE CORRECT LINK FOR RETURN
000311  440305      3770            !NX     3REST           SET THE RETURN TO +2 FOR NO MEMORY PROTECTION ON
000312  201713      3780            LAC     .0              LOAD THE USER'S LOCATION 0
000313  040000      3790            DAC     0               RESTORE THE USER'S LOCATION ZERO BEFORE RETURN TO HIM
000314  200005      3800            LAC     3AC             RESTORE THE USER'S AC
000315  620305      3810            JMP     3REST,X
```

R DEVICE INTERRUPT SERVICE ROUTINES

```
                    3820            .STITL  DEVICE INTERRUPT SERVICE ROUTINES
                    3830
                    3840    *
                    3850    *
                    3860    *       CLOCK SERVICE -- WHEN THE CLOCK RUNS OUT:
                    3870    *            1) SET CLKLOK TO RECORD THE FACT WE HAVE RECEIVED AN INTERRUPT
                    3880    *            2) RESET THE CLOCK, THIS GUARANTEES ANOTHER CLOCK INTERRUPT, NO MATTER WHAT.
                    3890    *            3) CHECK TO SEE IF THE DISK IS AVAILABLE. IF NOT, EXIT.
                    3900    *            4) IF THE DISK IS AVAILABLE, FIND THE NEXT USER WHO IS FREE TO RUN
                    3910    *               (I.E. IS NOT I/O ROADBLOCKED).
                    3920    *            5) IF THIS IS THE CURRENT USER, EXIT NORMALLY.
                    3930    *            6) OTHERWISE CALL IN THE SWAPPER TO SWAP OUT THE CURRENT USER AND
                    3940    *               TO SWAP IN THE NEXT USER DUE TO RUN.
                    3950    *
000316  000000      3960    CLKLOK  .DSA
000317  440316      3970    CLK     INX     CLKLOK          RECORD WE HAVE RECEIVED ANOTHER CLOCK INTERRUPT
000320  740000      3980            NOP                     GENERAL PRINCIPLES -- IN CASE IT EVER DOES SKIP
000321  700044      3990            CLON                    NEED THIS TO CLEAR THE FLAG, EVEN THOUGH THE CLOCK IS NEVER TURNED OFF
000322  777730      4000            LAW     -CLKMAX
000323  040007      4010            DAC     7               IN ANY CASE, RESET THE CLOCK
000324  200266      4020            LAC     DKLOK
000325  740200      4030            SZA                     SKIP IF THE SOFTWARE FLAG SAYS THE DISK IS FREE
000326  600270      4040            RET     PIDON           ELSE EXIT -- CAN'T DO ANYTHING ELSE HERE UNTIL THE DISK IS FREE
                    4050
000327  121003      4060    CLKST   JMS     $RDBLK,X        SEE WHO IS THE NEXT NON-ROADBLOCKED USER
000330  200055      4070            LAC     3TEM4           LOAD HIS USER NUMBER
000331  341774      4080            TAD     TYPE            CONVERT TO A FILENAME
000332  541772      4090            SAD     NAME            SKIP IF HE IS NOT ALREADY RUNNING
000333  600270      4100            RET     PIDON           SAME USER, SO JUST RESTART HIM
000334  761003      4110            LAW     $SWCLK          THERE IS ANOTHER FREE USER, SO REQUEST A CLOCK SWAP
                    4120    *
                    4130    *       GET THE SWAPPER AND ENTER IT AT THE ADDRESS PASSED IN THE AC
                    4140    *
000335  040666      4150    SWAP    DAC     DQ              SET THE RETURN ADDRESS
000336  200044      4160    SWAP1   LAC     CSWP
000337  040040      4170            DAC     SWPS            SET THE SWAPPER'S PHYSICAL DISK ADDRESS
000340  760037      4180    SWAP3   LAW     SWPS-1          SET A POINTER TO THE CATALOG DATA FOR THE SWAPPER
000341  600667      4190            JMP     DQ+1            GET THE SWAPPER
                    4200    *
                    4210    *       CAL MUST BE A RESIDENT ERROR MESSAGE
                    4220    *
000342  760014      4230    ERRCAL  LAW     12.
000343  041706      4240            DAC     UTEM2           PASS THE ERROR MESSAGE NUMBER TO THE ERROR ROUTINE
000344  761004      4250            LAW     $SWERR
000345  600335      4260            JMP     SWAP            CALL THE ERROR MESSAGE PROGRAM
```

R KEYBOARD INPUT ROUTINES

```
                    4270            .STITL  KEYBOARD INPUT ROUTINES
                    4280
                    4290    *
                    4300    *       THE FOLLOWING THREE LINES OF CODE FOR EACH TELETYPE
                    4310    *       WHICH ARE UNIQUE TO THEIR RESPECTIVE TELETYPES. THESE INSTRUCTIONS
                    4320    *       PERMIT ALL INPUT AND OUTPUT TO BE CARRIED ON BY COMMON ROUTINES.
                    4330    *
                    4340    *       ON INPUT, A JMS KBDIN PROVIDES THE INPUT ROUTINE WITH A POINTER TO
                    4350    *       THE TELETYPE'S OWN RESIDENT STORAGE, KBDIN RETRIEVES THIS BY AN
                    4360    *       XCT KBDIN,X. KBDIN DOES NOT RETURN TO HERE -- THE JMS IS MERELY
                    4370    *       TO PROVIDE A POINTER TO THE CALLING TELETYPE.
                    4380    *
                    4390    *       ON OUTPUT KBDOT IS ENTERED WITH THE TELETYPE'S UNIQUE POINTER TO
                    4400    *       ITS RESIDENT STORAGE ALREADY IN THE ACCUMULATOR.
                    4410    *
                    4420    *       NOTE THAT <--KBD> IS THE ENTRANCE FOR AN INTERRUPT FROM A TELETYPE
                    4430    *       KEYBOARD, WHILE <--OUT> IS THE ENTRANCE FOR AN INTERRUPT FROM A
                    4440    *       TELETYPE TELEPRINTER.
                    4450    *
                    4460
                    4470    *
                    4480    *       CONSOLE TELETYPE
                    4490    *
    000346          4500    CTKBD   ...                     SERVICE INTERRUPTS FROM CONSOLE KEYBOARD
000346  100357      4510            JMS     KBDIN           PROCESS THE INPUT
    000347          4520    CTOUT   ...                     SERVICE INTERRUPTS FROM CONSOLE TELEPRINTER
000347  760076      4530            LAW     CTBIN-2         IDENTIFY YOURSELF
000350  600441      4540            JMP     KBDOT           PRINT THE NEXT CHARACTER, IF ANY
                    4550    *
                    4560    *       TELETYPE #1
                    4570    *
    000351          4580    L1KBD   ...
000351  100357      4590            JMS     KBDIN           PROCESS THE INPUT
000352  760125      4600    L1OUT   LAW     L1BIN-2         USED AS SYSTEM JOB NUMBER
000353  600441      4610            JMP     KBDOT
                    4620    *
                    4630    *       TELETYPE #2
                    4640    *
    000354          4650    L2KBD   ...
000354  100357      4660            JMS     KBDIN
000355  760154      4670    L2OUT   LAW     L2BIN-2
000356  600441      4680            JMP     KBDOT
                    4690
```

```
R                       ROTARY BUFFER CHARACTER INPUT/OUTPUT ROUTINES

                4700            .STITL  ROTARY BUFFER CHARACTER INPUT/OUTPUT ROUTINES
                4710    *
                4720    *
                4730    *       KBDIN HANDLES INPUT FROM ANY KEYBOARD.
                4740    *
    000357      4750    KBDIN   ENTER
                                .PMC    SAVE,ON
000357  740040                  XX
                                .PMC    RESTORE
                4760    *
                4770    *       INITIALIZE THE KEYBOARD INPUT ROUTINE
                4780    *
000360  040002  4790            DAC     3TM21           SAVE THE CHARACTER
000361  420357  4800            XCT     KBDIN,X         LOAD THE CALLER'S IDENTITY
000362  100526  4810            JMS     IO.IN           SET UP THE REENTRANT TEMPS
000363  200002  4820            LAC     3TM21           RESTORE THE CHARACTER
                4830    *
                4840    *       CHECK FOR SPECIAL CHARACTERS
                4850    *
                4860    *       A NULL (BREAK OR CONTROL SHIFT 'P') WILL STOP THE USER'S CURRENT JOB
                4870    *       CANCEL ANY INCOMPLETE I/O, AND RESTART THE MONITOR.
                4880    *
                4890    *       IF THE DELETE OPTION IS ON, A CONTROL 'X' WILL KILL THE REST OF THE
                4900    *       CURRENT OUTPUT BUFFER WITHOUT AFFECTING THE INPUT STATUS.
                4910    *
000364  500654  4920            AND     BL7             KEEP ONLY THE LAST 7 BITS -- NECESSARY TO CHECK FOR A NULL
000365  741200  4930            SNA                     CHECK FOR NULL (= CONTROL SHIFT P) ON INPUT
000366  600425  4940            JMP     KBD0            IF NULL, CANCEL JOB AND RESTART MONITOR
                4950    *
                4960    *       BEGIN TO PROCESS NORMAL INPUT, CLEAR THE FOLLOWING BITS:
                4970    *            BIT2: KEYBOARD I/O ROADBLOCK FLAG
                4980    *            BIT3: SOFTWARE KEYBOARD FLAG
                4990    *            BIT6: CHARACTER ECHO FLAG ( 0 = CHARACTER HAS BEEN ECHOED )
                5000    *            BITS 10-17: ONE CHARACTER KEYBOARD BUFFER
                5010    *
                5020    *       THEN SET THE FOLLOWING BITS:
                5030    *            BIT3: SOFTWARE KEYBOARD FLAG
                5040    *            BIT6: CHARACTER ECHO FLAG ( 1 = CHARACTER HAS NOT YET BEEN ECHOED )
                5050    *            BITS 10-17: SAVE THE CHARACTER JUST TYPED
                5060    *
000367  200053  5070            LAC     3TEM2           LOAD THE TELETYPE SOFTWARE FLAGS
000370  500720  5080            AND     (633400)        CLEAR  KEYBOARD & KEYBOARD I/O ROADBLOCK FLAGS, & KEYBOARD SOFTWARE BUFFER
000371  240647  5090            XOR     BIT36           SET THE KEYBOARD AND CHARACTER-NOT-ECHOED FLAGS
000372  240002  5100            XOR     3TM21           PUT THE LATEST CHARACTER IN THE SOFTWARE KEYBOARD BUFFER
000373  040053  5110            DAC     3TEM2           SAVE THE SOFTWARE FLAGS
                5120    *
                5130    *       CHECK FOR AN INPUT BUFFER, IF NOT, THE CHARACTER CANNOT YET BE PACKED
                5140    *
000374  500652  5150            AND     BIT7            RECOVER THE BUFFER TYPE
000375  740200  5160            SZA                     SKIP IF IT IS AN INPUT BUFFER
000376  600401  5170            JMP     KBD05           ELSE GO DIRECTLY TO THE OUTPUT-IN-PROGRESS TEST
000377  100561  5180            JMS     PUTIN           INPUT BUFFER -- TRY TO PACK THE CHARACTER
```

```
                R                       ROTARY BUFFER CHARACTER INPUT/OUTPUT ROUTINES

000400  400056  5190            XCT     3TEM5           ECHO A GARBAGE CHARACTER IF IT IS FULL
                5200    *
                5210    *       CHECK FOR OUTPUT-IN-PROGRESS. IF SO, THE INPUT CHARACTER CANNOT BE
                5220    *       ECHOED UNTIL THE NEXT TELEPRINTER INTERRUPT.
                5230    *
    000401      5240    KBD05   ...
000401  200053  5250            LAC     3TEM2           RELOAD THE SOFTWARE FLAGS
000402  741100  5260            SPA                     SKIP IF NO OUTPUT IS IN PROGRESS
000403  600417  5270            JMP     KBD4            ELSE DON'T ECHO IT
                5280    *
                5290    *       OUTPUT IS NOT IN PROGRESS.
                5300    *       TURN ON BIT 0; THE OUTPUT-IN-PROGRESS FLAG (IF THE TERMINAL REQUIRES A SOFTWARE ECHO)
                5310    *       TURN OFF BIT 6; THE CHARACTER-NOT-ECHOED FLAG ( 0=OFF )
                5320    *
                5330    *       ECHO THE CHARACTER IF THIS IS A TERMINAL REQUIRING A SOFTWARE ECHO
                5340    *
000404  240651  5350            XOR     BIT6            TURN OFF THE CHARACTER-NOT-ECHOED FLAG
000405  400057  5360            XCT     3TEM6           ECHO THE CHARACTER IF SOFTWARE ECHO TERMINAL; ELSE SKIP
000406  240646  5370            XOR     BIT0            SET THE OUTPUT-IN-PROGRESS FLAG (FOR SOFTWARE ECHO TERMINALS)
    000407      5380    KBD2    ...
000407  040053  5390            DAC     3TEM2           SAVE THE UPDATED TELETYPE FLAGS
                5400    *
                5410    *       RESIDENT EXECUTIVE ROUTINE TO GENERATE A SIMULATED PROGRAM
                5420    *       INTERRUPT FOR THE USER IF APPROPRIATE. IF THE USER IS CURRENTLY
                5430    *       RUNNING, TRANSFER TO HIS MEMORY PROTECTION OVERLAY TO SEE ABOUT
                5440    *       GIVING HIM THE SIMULATED INTERRUPT. OTHERWISE JUST SET THE
                5450    *       PROGRAM INTERRUPT BIT ON IN HIS I/O FLAGS WORD, AND THE SWAPPER
                5460    *       WILL SEE ABOUT GENERATING THE SIMULATED INTERRUPT (IF NECESSARY)
                5470    *       THE NEXT TIME HE IS SWAPPED IN.
                5480    *
000410  200055  5490    KBD9    LAC     3TEM4           LOAD THE USER NUMBER OF THE INTERRUPTING USER
000411  541771  5500    PITST   SAD     NUMBR           SKIP IF THE INTERRUPT DID NOT BELONG TO THE LAST CORE USER
000412  600421  5510            JMP     KBD7            IT DID -- SEE IF THE USER IS STILL RUNNING
000413  200053  5520    KBD5    LAC     3TEM2           LOAD THE SOFTWARE FLAGS
000414  500660  5530            AND     CB5
000415  240650  5540            XOR     BIT5            SET THE PI INTERRUPT REQUEST (SWAPPER WILL CHECK IT)
000416  040053  5550    KBD41   DAC     3TEM2
000417  100540  5560    KBD4    JMS     IO,OT           UPDATE THE FLAGS
000420  600270  5570            RET     PIDON
                5580
000421  200035  5590    KBD7    LAC     RCORE
000422  741200  5600            SNA                     SKIP IF THERE IS A USER RUNNING
000423  600413  5610            JMP     KBD5            NO
000424  601001  5620            JMP     SPINT           YES
```

R ROTARY BUFFER CHARACTER INPUT/OUTPUT ROUTINES

```
                    5630            ,EJECT
                    5640    *
                    5650    *       THE USER TYPED A NULL, STOP HIS CURRENT JOB, KILL ALL INCOMPLETE
                    5660    *       I/O, AND SET UP A MONITOR REQUEST FOR HIM.
                    5670    *
000425  100553      5680    KBD0    JMS     NEWBR           NULL KILLS ALL OLD I/O
000426  100540      5690            JMS     IO.OT           CLEAN UP THIS USER
000427  200055      5700            LAC     3TEM4           LOAD IDENTITY OF ONE WHO PRINTED A NULL
000430  060055      5710            DAC     3TEM4,X         AND SET IT AS A MONITOR REQUEST
000431  541771      5720            SAD     NUMBR           SEE IF HE IS THE CURRENT USER
000432  741000      5730            SKP
000433  600270      5740            RET     PIDON           NO -- EXIT
000434  200266      5750            LAC     DKLOK
000435  740200      5760            SZA                     SEE IF THE DISK IS AVAILABLE TO GET THE SWAPPER
000436  600270      5770            RET     PIDON           NO -- EXIT
000437  761002      5780    MTR1    LAW     $SWMTR
000440  600335      5790            JMP     SWAP            YES -- GET THE MONITOR
```

R ROTARY BUFFER CHARACTER INPUT/OUTPUT ROUTINES

```
                    5800            .EJECT
                    5810    *
                    5820    *       KBDOT HANDLES OUTPUT TO ANY TELETYPE FROM ITS ROTARY LINE BUFFER.
                    5830    *       SUFFICIENT OUTPUT CLEARS THE TELEPRINTER I/O ROADBLOCKED CONDITION, EMPTYING THE
                    5840    *       BUFFER CLEARS THE OUTPUT-IN-PROGRESS CONDITION.
                    5850    *       THE OUTPUT BUFFER IS FILLED BY THE MEMORY PROTECTION ROUTINES, THEY ALSO
                    5860    *       SET THE OUTPUT-IN-PROGRESS FLAG.
                    5870    *
                    5880    *       INITIALIZE....CHECK FOR OUTPUT IN PROGRESS....IF NONE, ASSUME THE INTERRUPT
                    5890    *       WAS GENERATED BY THE LAST CHARACTER OF THE PRECEEDING MESSAGE AND EXIT IMMEDIATELY.
                    5900    *
    000441          5910    KBDOT   ...;
000441  100526      5920            JMS     IO.IN
000442  200053      5930            LAC     3TEM2           LOAD THE TELETYPE SOFTWARE FLAGS
000443  740100      5940            SMA                     SKIP IF OUTPUT IS IN PROGRESS
000444  600410      5950            JMP     KBD9            ELSE EXIT THROUGH THE SIMULATED PI ROUTINE
                    5960    *
                    5970    *       SEE WHETHER OR NOT TO ECHO A CHARACTER
                    5980    *       ECHO IT IF NECESSARY, AND EXIT, ELSE CONTINUE.
                    5990    *
000445  500651      6000            AND     BIT6            RECOVER THE CHARACTER-NOT-ECHOED FLAG
000446  741200      6010            SNA                     SKIP IF THERE IS UN-ECHOED INPUT
000447  600454      6020            JMP     KBD01           NO ECHO NEEDED -- PROCEED TO NORMAL OUTPUT
000450  240053      6030            XOR     3TEM2           TURN OFF THE CHARACTER-NOT-ECHOED FLAG
000451  400057      6040            XCT     3TEM6           ECHO THE CHARACTER FOR SOFTWARE ECHO TERMINALS; ELSE SKIP
000452  600416      6050            JMP     KBD41           SOFTWARE ECHO TERMINALS EXIT HERE
000453  040053      6060            DAC     3TEM2           HARDWARE ECHO TERMINALS GET HERE
                    6070    *
                    6080    *       CHECK TO SEE WHETHER THE BUFFER IS AN INPUT BUFFER OR AN OUTPUT
                    6090    *       BUFFER CURRENTLY, IF IT IS AN OUTPUT BUFFER, CONTINUE WITH
                    6100    *       OUTPUT, IF IT IS AN INPUT BUFFER, BRANCH TO THE OUTPUT DONE ROUTINE.
    000454          6110    KBD01   ...;
000454  200053      6120            LAC     3TEM2           LOAD THE SOFTWARE FLAGS
000455  500652      6130            AND     BIT7            RECOVER THE BUFFER TYPE
000456  740200      6140            SZA                     SKIP IF THE I/O BUFFER IS AN INPUT BUFFER
000457  600463      6150            JMP     KBD02           ELSE CONTINUE WITH NORMAL OUTPUT
000460  200053      6160            LAC     3TEM2           LOAD THE SOFTWARE FLAGS
000461  500656      6170            AND     CB0             CLEAR THE OUTPUT IN PROGRESS FLAG
000462  600416      6180            JMP     KBD41           EXIT
                    6190    *
                    6200    *       CHECK FOR I/O ROADBLOCK REMOVAL -- REMOVE WHEN BARELY ENOUGH
                    6210    *       CHARACTERS ARE LEFT TO COVER OTHER USERS' MAXIMUM CPU TIMES
                    6220    *
000463  777777      6230    KBD02   LAW     -1
000464  340052      6240            TAD     3TEM1
000465  500663      6250            AND     ADRSS
000466  740001      6260            CMA
000467  040002      6270            DAC     3TM21           SET THE (TWO'S COMPLEMENT) START-OF-OUTPUT ADDRESS
000470  200051      6280            LAC     3TEM0           LOAD THE END-OF-OUTPUT ADDRESS
000471  500663      6290            AND     ADRSS
000472  340002      6300            TAD     3TM21           SUBTRACT THE START
000473  741100      6310            SPA                     SKIP IF O.K. (AC = AMOUNT OF OUTPUT STILL TO GO)
```

R ROTARY BUFFER CHARACTER INPUT/OUTPUT ROUTINES

```
000474  340721  6320            TAD     ($KBLEN)        ELSE MAKE AC = AMOUNT OF OUTPUT TO GO
000475  340722  6330            TAD     (-MINBUFF)      SUBTRACT THE MINIMUM NEEDED TO SUSTAIN THE I/O ROADBLOCK
000476  740300  6340            SMA!SZA                 SKIP IF THE I/O ROADBLOCK NEEDS CLEARING
000477  600503  6350            JMP     KBD8            ELSE CARRY ON
                6360    *
                6370    *       REMOVE THE TELEPRINTER I/O ROADBLOCK
                6380    *
000500  200053  6390            LAC     3TEM2           LOAD THE SOFTWARE FLAGS
000501  500657  6400            AND     CB1             CLEAR THE TELEPRINTER I/O ROADBLOCK FLAG
000502  040053  6410            DAC     3TEM2           SAVE THE UPDATED SOFTWARE FLAGS
                6420    *
                6430    *       NOW PRINT THE CHARACTER IF THERE IS ONE
                6440    *
000503  100614  6450    KBD8    JMS     FGET            OUTPUT ONGOING -- GET THE NEXT CHARACTER
000504  600507  6460            JMP     KBD6            OUTPUT-IN-PROGRESS IS DONE WHEN AN EMPTY BUFFER IS FOUND.
000505  400056  6470            XCT     3TEM5           PRINT THE CHARACTER
000506  600410  6480            JMP     KBD9            EXIT THROUGH THE PI SIMULATION ROUTINE
                6490    *
                6500    *       THERE IS NO CHARACTER....CLEAR THE OUTPUT FLAGS....LOAD THE LAST
                6510    *       INPUT CHARACTER IF THE KEYBOARD FLAG IS SET.
                6520    *
000507  200053  6530    KBD6    LAC     3TEM2           LOAD THE TELETYPE SOFTWARE FLAGS
000510  500723  6540            AND     (375777)        CLEAR BITS 0 (OUTPUT-IN-PROGRESS) AND 7 (0=INPUT BUFFER)
000511  040002  6550            DAC     3TM21           SAVE THE LAST INPUT CHARACTER
000512  040002  6560            DAC     3TM21           PASS TO THE PACKING ROUTINE
000513  040053  6570            DAC     3TEM2           SAVE THE FLAGS
000514  500652  6580            AND     BIT7            RECOVER THE BUFFER TYPE
000515  741200  6590            SNA                     SKIP IF IT WAS AN OUTPUT BUFFER
000516  600410  6600            JMP     KBD9            ELSE EXIT
000517  240053  6610            XOR     3TEM2           MAKE IT AN INPUT BUFFER
000520  040053  6620            DAC     3TEM2           SAVE THE CORRECTED SOFTWARE FLAGS
000521  500647  6630            AND     BIT36           GET THE SOFTWARE KEYBOARD FLAG (BIT 6 IS ALREADY CLEARED)
000522  741200  6640            SNA                     SKIP IF IT IS SET
000523  600410  6650            JMP     KBD9            ELSE DONE NOW -- EXIT THROUGH THE SIMULATED PI ROUTINE
000524  200053  6660            LAC     3TEM2           IF SET, LOAD THE INPUT CHARACTER
000525  600407  6670            JMP     KBD2            GO PACK THE CHARACTER AND SIMULATE A USER PROGRAM INTERRUPT
```

R ROTARY BUFFER CHARACTER INPUT/OUTPUT ROUTINES

```
                    6680            .EJECT
                    6690    *
                    6700    *
                    6710    *       DUE TO THE LACK OF INDEX REGISTER CAPABILITY ON THE PDP-9, IT IS NECESSARY TO
                    6720    *       COPY A TELETYPE'S PARAMETERS INTO A SET OF TEMPORARY VARIABLES WHENEVER
                    6730    *       WE GET AN INTERRUPT THT WILL REQUIRE US TO USE THEM. WHEN WE ARE DONE
                    6740    *       SERVICING THAT INTERRUPT IT IS NECESSARY TO COPY THEM BACK OUT TO PRESERVE
                    6750    *       ANY CHANGES WE MAY HAVE MADE.
                    6760    *
                    6770    *       ROUTINE TO PRESERVE RE-ENTRANT VARIABLES ON ENTRANCE
                    6780    *
    000526          6790    IO.IN   ENTER                   TRANSFER PARAMETERS TO TEMPS ON ENTRANCE
                                    .PMC    SAVE,ON
000526  740040                      XX
                                    .PMC    RESTORE
000527  040010  6800               DAC     10              AC = PTR TO 1 BEFORE FIRST PARAMETER
000530  040004  6810               DAC     3TM10
000531  760047  6820               LAW     3TM20-1
000532  040011  6830               DAC     11
000533  220010  6840    IO.1       LAC     10,X
000534  060011  6850               DAC     11,X            TRANSFER THE PARAMETER
000535  440050  6860               ISZ     3TM20
000536  600533  6870               JMP     .-3             DO THE NEXT ONE
000537  620526  6880               RET     IO.IN,X         DONE, EXIT
                    6890    *
                    6900    *       ROUTINE TO PRESERVE RE-ENTRANT VARIABLES ON EXIT
                    6910    *
    000540          6920    IO.OT   ENTER                   TRANSFER TEMPS TO PARAMETERS ON EXIT
                                    .PMC    SAVE,ON
000540  740040                      XX
                                    .PMC    RESTORE
000541  200004  6930               LAC     3TM10
000542  040011  6940               DAC     11
000543  220011  6950               LAC     11,X
000544  040050  6960               DAC     3TM20           2'S COMPLEMENT OF PARAMETER COUNT
000545  440050  6970               INX     3TM20           CORRECT THE TRANSFER COUNT
000546  760050  6980               LAW     3TEM0-1
000547  040010  6990               DAC     10
000550  200540  7000               LAC     IO.OT
000551  040526  7010               DAC     IO.IN
000552  600533  7020               JMP     IO.1
```

R I/O BUFFER HANDLING ROUTINES

```
                    7030            ,STITL  I/O BUFFER HANDLING ROUTINES
                    7040    *
                    7050    *       THESE ROUTINES ARE TO HANDLE EIGHT-BIT ASCII INPUT TO AND OUTPUT FROM
                    7060    *       ANY TELETYPE'S ROTARY INPUT/OUTPUT BUFFER. BUFFER FORMAT IS TWO EIGHT-
                    7070    *       BIT CHARACTERS PER WORD, CHARACTER #1 IN BITS 2-9 AND CHARACTER #2 IN
                    7080    *       BITS 10-17.
                    7090    *            3TEM3 HOLDS THE (CONSTANT) ADDRESS OF THE START OF THE BUFFER.
                    7100    *            3TEM4 HOLDS THE (CONSTANT) MAXIMUM+1 ADDRESS OF THE ROTARY BUFFER.
                    7110    *            3TEM0 IS THE BUFFER INPUT POINTER, BIT 0 = 0 IF THE NEXT CHARACTER
                    7120    *       IS TO BE THE FIRST CHARACTER STORED IN THAT WORD, BIT 0 = 1 IF THE NEXT
                    7130    *       CHARACTER IS TO BE THE SECOND CHARACTER STORED IN THAT WORD.
                    7140    *            3TEM1 IS THE BUFFER OUTPUT POINTER, BIT 0 = 0 IF THE NEXT CHARACTER
                    7150    *       IS TO BE THE FIRST CHARACTER SUPPLIED FROM THAT WORD, BIT 0 = 1 IF THE
                    7160    *       CHARACTER IS TO BE THE SECOND CHARACTER SUPPLIED FROM THAT WORD.
                    7170    *
                    7180    *       NEWBR IS A SUBROUTINE TO RE-INITIALIZE THE BUFFER AND TO ZERO ALL SOFTWARE TELETYPE FLAGS.
                    7190    *
                    7200    *
    000553          7210    NEWBR   ENTER                   ROUTINE TO CLEAR THE I/O BUFFER
                                    .PMC    SAVE,ON
000553  740040                      XX
                                    .PMC    RESTORE
000554  200054  7220              LAC     3TEM3           LOAD THE START OF THE BUFFER
000555  040051  7230              DAC     3TEM0           RESET THE IN-POINTER
000556  040052  7240              DAC     3TEM1           RESET THE OUT-POINTER
000557  140053  7250              DZM     3TEM2           KILL ALL SOFTWARE TELETYPE FLAGS
000560  620553  7260              RET     NEWBR,X
                    7270    *
                    7280    *       PUTIN PACKS THE EIGHT-BIT ASCII CHARACTER FROM 3TM21 INTO THE PROPER ROTARY LINE BUFFER.
                    7290    *       IT RETURNS +2 IF SUCCESSFUL, +1 ON OVERFLOW.
                    7300    *
    000561          7310    PUTIN   ENTER
                                    .PMC    SAVE,ON
000561  740040                      XX
                                    .PMC    RESTORE
000562  200002  7320              LAC     3TM21           LOAD THE ALLEGED CHARACTER
000563  500655  7330              AND     BL8             MASK TO JUST ASCII TO PROTECT OURSELVES
000564  040002  7340              DAC     3TM21           RESTORE THE CHARACTER
000565  200051  7350              LAC     3TEM0           LOAD THE ACTIVE ADDRESS
000566  100635  7360              JMS     NXPTR           ADVANCE THE POINTER TO THE NEXT CHARACTER LOCATION
000567  540052  7370              SAD     3TEM1           SKIP IF NO OVERFLOW
000570  620561  7380              RET     PUTIN,X         RETURN +1 FOR OVERFLOW
000571  440561  7390              INX     PUTIN           RETURN +2 FOR SUCCESSFUL
000572  040051  7400              DAC     3TEM0           SAVE THE UPDATED POINTER
000573  740100  7410              SMA                     SKIP IF IT IS THE FIRST CHARACTER IN THIS WORD
000574  600610  7420              JMP     PUT2
                    7430
    000575          7440    PUT1    ...                     PUT FIRST CHARACTER IN BITS 2-9
000575  220003  7450              LAC     3TM22,X         LOAD THE BUFFER WORD
000576  500655  7460              AND     BL8             CLEAR ROOM FOR THIS CHARACTER
000577  060003  7470              DAC     3TM22,X
000600  200002  7480              LAC     3TM21           LOAD THE CHARACTER -- BIT 0 MUST BE ZERO
```

R I/O BUFFER HANDLING ROUTINES

```
000601  742010  7490            RTL                     POSITION THE CHARACTER
000602  742010  7500            RTL
000603  742010  7510            RTL
000604  742010  7520            RTL
000605  260003  7530            XOR     3TM22,X         PACK THE CHARACTER
000606  060003  7540    PUT4    DAC     3TM22,X         STORE THE CHARACTER
000607  620561  7550            RET     PUTIN,X
                7560
    000610      7570    PUT2    ...                     PUT SECOND CHARACTER INTO BITS 11-17
000610  220003  7580            LAC     3TM22,X         LOAD THE STORAGE WORD
000611  500662  7590            AND     CBL8            CLEAR ROOM FOR THIS CHARACTER
000612  240002  7600            XOR     3TM21           AND INSERT THE NEW CHARACTER
000613  600606  7610            JMP     PUT4            EXIT
                7620    *
                7630    *       FGET REMOVES THE FIRST REMAINING CHARACTER FROM THE I/O BUFFER.
                7640    *       IT RETURNS IT AS 7-BIT ASCII IN THE AC.
                7650
    000614      7660    FGET    ENTER
                        ,PMC    SAVE,ON
000614  740040                  XX
                        ,PMC    RESTORE
000615  200052  7670            LAC     3TEM1           LOAD THE ACTIVE ADDRESS
000616  540051  7680            SAD     3TEM0
000617  620614  7690            RET     FGET,X          RETURN +1 FOR NO CHARACTER
000620  440614  7700            INX     FGET            RETURN +2 FOR SUCCESSFUL
000621  100635  7710            JMS     NXPTR           ADVANCE THE POINTER TO THE NEXT CHARACTER LOCATION
000622  040052  7720            DAC     3TEM1           AND SAVE THE NEW POINTER
000623  740010  7730            RAL
000624  220003  7740            LAC     3TM22,X         LOAD THE CHARACTER
000625  744400  7750            SNL!CLL
000626  600633  7760            JMP     FGET2
000627  742020  7770            RTR
000630  742020  7780            RTR
000631  742020  7790            RTR
000632  742020  7800            RTR
000633  500655  7810    FGET2   AND     BL8             MASK TO EIGHT-BIT ASCII
000634  620614  7820            RET     FGET,X
                7830    *
                7840    *       NXPTR INCREMENTS THE POINTER PASSED IN THE AC, AND RETURNS THE RESULT IN
                7850    *       THE AC. THE LINK WILL ALWAYS BE FLIPPED, THE ADDRESS INCREMENTED OR WRAPPED
                7860    *       ONLY AS APPROPRIATE.
                7870    *
    000635      7880    NXPTR   ENTER
                        ,PMC    SAVE,ON
000635  740040                  XX
                        ,PMC    RESTORE
000636  040003  7890            DAC     3TM22           SAVE THE OLD POINTER -- THE ROUTINES STILL NEED IT
000637  240646  7900            XOR     BIT0            FLIP BIT 0
000640  741100  7910            SPA
000641  620635  7920            RET     NXPTR,X         NO NEED TO INCREMENT THE ADDRESS
000642  340653  7930            TAD     BIT17           ELSE ADVANCE THE POINTER
000643  540055  7940            SAD     3TEM4           END OF THE BUFFER??
```

R I/O BUFFER HANDLING ROUTINES

```
000644  200054  7950            LAC     3TEM3           YES, SO WRAP THE POINTER
000645  620635  7960            RET     NXPTR,X         EXIT
                7970    *
                7980    *       RESIDENT CONSTANTS ARE USED INSTEAD OF LITERALS TO PERMIT THEM TO BE  ACCESSED BY OVERLAYS.
                7990    *
000646  400000  8000    BIT0    400000
000647  044000  8010    BIT36   044000
000650  010000  8020    BIT5    010000
000651  004000  8030    BIT6    004000
000652  002000  8040    BIT7    002000
000653  000001  8050    BIT17   000001
000654  000177  8060    BL7     177                     LAST SEVEN BITS
000655  000377  8070    BL8     000377                  LAST EIGHT BITS
000656  377777  8080    CB0     377777                  MASK TO CLEAR BIT 0
000657  577777  8090    CB1     577777                  MASK TO CLEAR BIT 1
000660  767777  8100    CB5     767777                  MASK TO CLEAR BIT 5
000661  775777  8110    CB7     775777
000662  777400  8120    CBL8    777400                  MASK TO CLEAR LAST EIGHT BITS
000663  017777  8130    ADRSS   17777                   MASK TO RETAIN JUST THE ADDRESS BITS
000664  600000  8140    JMP     JMP                     CONSTANT
000665  703304  8150    DBK     DBK                     CONSTANT
```

R RESIDENT DISK ROUTINES

```
                    8160            ,STITL  RESIDENT DISK ROUTINES
                    8170    *
                    8180    *       THE RESIDENT DISK ROUTINE IS STRICTLY A MINIMUM SIZE PHYSICAL
                    8190    *       DISK HANDLER.
                    8200    *
                    8210    *        CALLING SEQUENCE: LAW PNTR
                    8220    *                          JMS DO
                    8230    *
                    8240    *       PNTR+1: PHYSICAL DISK ADDRESS ( = BLOCK NUMBER * 400 )
                    8250    *       PNTR+2: CORE ADDRESS -1
                    8260    *       PNTR+3: TWO'S COMPLEMENT WORD COUNT
                    8270    *       PNTR+4: DISK READ (3) OR DISK WRITE (5)
                    8280    *
    000666          8290    DO      ENTER
                                    ,PMC    SAVE,ON
000666  740040                      XX
                                    ,PMC    RESTORE
000667  040010      8300            DAC     10              SET THE PARAMETER POINTER
000670  707074      8310            DLAH+10                 BE SURE WE ARE SET FOR DISK ZERO
000671  707212      8320            DLOK+10                 FIND OUT WHETHER OR NOT THE DISK IS IN USE
000672  751101      8330            SPA!CLA!CMA             SKIP IF IT IS FREE
000673  600671      8340            JMP     .-2             ELSE WAIT FOR IT
000674  040266      8350    DO2     DAC     DKLOK           FLAG THE SYSTEM IS USING THE DISK
000675  220010      8360    DO3     LAC     10,X            GET THE ADDRESS
000676  707024      8370            DLAL                    PLACE IT INTO THE ADDRESS REGISTER
000677  220010      8380            LAC     10,X            LOAD THE STARTING ADDRESS
000700  040037      8390            DAC     $DKCA           SET THE CORE ADDRESS POINTER
000701  220010      8400            LAC     10,X
000702  040036      8410            DAC     $DKWC           SET THE DISK WORD COUNT
000703  220010      8420            LAC     10,X
000704  707047      8430    DKOVR   DSCF!DSFX!DSCN          ISSUE THE READ COMMAND
000705  707001      8440            DSSF                    WAIT FOR THE OPERATION TO COMPLETE
000706  600705      8450            JMP     .-1
                    8460    *
                    8470    *        CHECK THE OPERATION AND RETURN TO THE APPROPRIATE PLACE
                    8480    *
000707  707272      8490    DKDON   DSRS+10                 CLEAR THE AC AND GET THE STATUS OF THE OPERATION
000710  707242      8500            DSCD                    CLEAR THE FLAGS
000711  741100      8510            SPA                     SEE IF OK
000712  740040      8520            HLT
000713  620666      8530            RET     DO,X
                    8540    *
                    8550    *
                    8560    *       OVERLAY COMMON CONTROL AND COMMUNICATION WORDS
                    8570    *
000714  000000      8580    OC0     ,DSA
000715  000000      8590    OC1     ,DSA
000716  000000      8600    OC2     ,DSA
000717  000000      8610    OC3     ,DSA
000720  633400      8620            ,LIT                    COLLECT LITERALS TEMPORARILY BEFORE USER TABLE
000721  000016
000722  777770
```

R RESIDENT DISK ROUTINES

000723 375777

R FORMAT OF THE USER JOB TABLE

```
                    8630            .STITL  FORMAT OF THE USER JOB TABLE
        001700      8640            .LOC    $UTSTRT
001700  000000      8650    FRDA    .DSA                    DEVICE ADDRESS OF THE USER'S CURRENTLY OPEN FILE
001701  000000      8660    FRCA    .DSA                    CORE ADDRESS OF THE USER'S CURRENTLYU OPEN FILE
001702  000000      8670    FRLEN   .DSA                    LENGTH OF THE USER'S CURRENTLY OPEN FILE
001703  000000      8680    FRSTA   .DSA                    START ADDRESS OF THE USER'S CURRENTLY OPEN FILE
                    8690
001704  000000      8700    UTEM0   .DSA                    USED FOR PASSING PARAMETERS TO OR BETWEEN EXECUTIVE PROGRAMS
001705  000000      8710    UTEM1   .DSA
001706  000000      8720    UTEM2   .DSA
001707  000000      8730    UTEM3   .DSA
001710  000000      8740    UTEM4   .DSA                    USED FOR PASSING PARAMETERS TO OR BETWEEN USER OR PHANTOM PROGRAMS
001711  000000      8750    UTEM5   .DSA
001712  000000      8760    UTEM6   .DSA
                    8770
        001713      8780    .0      .BLOCK  40              USER IMAGE OF FIRST 40 LOCATIONS
                    8790
001753  000000      8800    AC      .DSA                    USER AC SAVED
001754  000000      8810    MQ      .DSA                    USER MQ SAVED
001755  000000      8820    SC      .DSA                    USER STEP COUNTER SAVED
001756  000000      8830    ACS     .DSA                    SOFTWARE ACCUMULATOR SWITCHES REGISTER
                    8840
001757  000000      8850    CLOCK   .DSA                    STORE USER'S CLOCK HERE
001760  000000      8860    IORS    .DSA                    STORE USER'S PI STATUS
001761  000000      8870    DFLAG   .DSA                    SOFTWARE IMAGE OF THE DISK FLAG
001762  000000      8880    DAP0    .DSA                    SOFTWARE IMAGE OF DISK REGISTER AP0
001763  000000      8890    DAP1    .DSA                    SOFTWARE IMAGE OF DISK REGISTER AP1
001764  000000      8900    DFN     .DSA                    SOFTWARE IMAGE OF DISK FUNCTION REGISTER
001765  000000      8910    DSTAT   .DSA                    SOFTWARE IMAGE OF DISK STATUS REGISTER
                    8920
001766  000000      8930    UCORE   .DSA                    PHYSICAL DEVICE ADDRESS OF THE USER PROGRAM CORE IMAGE
001767  000000      8940    UDISK   .DSA                    PHYSICAL DEVICE ADDRESS OF THE ;PHYSICAL USER PHYSICAL DEVICE; FILE
001770  000000      8950    VALID   .DSA                    STORE THE USER'S VALIDATION BITS HERE
001771  000000      8960    NUMBR   .DSA                    TELETYPE NUMBER == POINTER TO RESIDENT TELETYPE PARAMETERS (US0, US1 OR US2)
                    8970
001772  000000      8980    NAME    .DSA                    NUMBER*TYPE
001773  000000      8990    OVER    .DSA                    NAME OF USER'S CURRENT OVERLAY PROGRAM
001774  000000      9000    TYPE    .DSA                    USER PROGRAM = 0; PHANTOM PROGRAM = -1
001775  000000      9010    PURNM   .DSA                    NAME FOR PURE CODE PORTION OF USER'S CURRENT PHANTOM PROGRAM (= 0 IF NONE)
001776  000000      9020    RSTRT   .DSA                    RESTART PC (CONTROLS ADDRESS,LINK,MEMORY PROTECT)
                    9030
        001777      9040            .END    1

TRANSFER ADDRESS  600001
```

R CROSS REFERENCE TABLE

1713	.0	4510	4520						
26	.310	3400							
27	.311	3410							
4464	.DT	570							
6460	.TP	550							
2023	10SAVE	1870	1880						
2024	11SAVE	1880	1920						
5	3AC	3370							
305	3REST	3870	3880						
51	3TEM0	3530	3540						
52	3TEM1	3540	3550						
53	3TEM2	3550	3560						
54	3TEM3	3560	3570						
55	3TEM4	3570	3580						
56	3TEM5	3580	3590						
57	3TEM6	3590	3600						
50	3TM20	3520	3530						
2	3TM21	3350							
3	3TM22	3360							
14000	7K	1030							
16000	8K	1020	910	1010	2640	2650			
1753	AC	4520	4530						
1756	ACS	4550	4560						
2015	ACSAVE	1810	1820						
2022	ACSW	1860	1870						
651	ADRSS	4100	4110						
300	AT	2950							
300	ATSGN	2900							
422030	BAS	420							
2151	BCNTRL	2330	2340						
634	BIT0	3970	3980						
641	BIT17	4020	4030						
635	BIT36	3980	3990						
636	BIT5	3990	4000						
637	BIT6	4000	4010						
640	BIT7	4010	4020						
642	BL7	4030	4040						
643	BL8	4040	4050						
2000	BOUNDA	970	960	980	990	1000	1630	5040	5080
377	BRK	5550							
2170	BUFFER	2490	2550						
1000	BUFLEN	2500	2550						
644	CB0	4050	4060						
645	CB1	4060	4070						
646	CB5	4070	4080						
647	CB7	4080	4090						
650	CBL8	4090	4100						
6	CHRMAX	3180	3200						
2	CHRPAK	3130	3200						
316	CLKLOK	3960	3290	3970					
50	CLKMAX	2840	3180	4000					
60	CLKSPD	3160	3170						

R CROSS REFERENCE TABLE

1757	CLOCK	4560	4570		
45	CMP1	3490	3500		
46	CMP2	3500	3510		
6	CNTRL	3380	3390		
2053	COMFLG	2200	2210		
2150	COMSTO	2270	2280		
16000	CORMAX	910	980		
47	CSPL	3510	3520		
44	CSWP	3480	3490		
60	CTBFR	3600	3630	3640	
100	CTBIN	3640	3650	3670	4250
2000	CTEMP0	1630			
2001	CTEMP1	1640			
2002	CTEMP2	1650			
2003	CTEMP3	1660			
2004	CTEMP4	1670			
2005	CTEMP5	1680			
2006	CTEMP6	1690			
2007	CTEMP7	1700			
2010	CTEMP8	1710			
2011	CTEMP9	1720			
102	CTFLG	3650	3660		
104	CTNAM	3660			
2043	D PC	2120	2130		
2154	D BCA	2370	2380		
2153	D BDA	2360	2370		
2163	D FDA	2440	2450		
2042	D LOC	2110	2120		
2022	D ACSW	1860			
2156	D BALT	2390	2400		
2155	D BLEN	2380	2390		
2161	D BMAX	2420	2430		
2157	D BMIN	2400	2410		
2162	D BPTR	2430	2440		
2167	D FMAX	2480	2490		
2165	D FMIN	2460	2470		
2046	D MASK	2150	2160		
2164	D MFDA	2450	2460		
2036	DADRSW	2070	2080		
1762	DAP0	4590	4600		
1763	DAP1	4600	4610		
653	DBK	4120	4130		
24	DBKNUM	2220	2270		
2054	DBKTAB	2210	2270		
2035	DBSTOR	2050	2060		
422027	DDT	410			
12000	DDT$T	5000			
2037	DDUMSW	2080	2090		
1761	DFLAG	4580	4590		
1764	DFN	4610	4620		
2151	DFTYPE	2340	2350		
2045	DHICOR	2140	2150		

R CROSS REFERENCE TABLE

2050	DINDIR	2170	2180
100	DK0	4270	
127	DK1	4310	
156	DK2	4350	
37	DKCA	2750	8390
675	DKDON	4170	4180
16000	DKLEN	2650	2660
34	DKLENB	2660	
266	DKLOK	3830	3840
672	DKOVR	4160	4170
2	DKRD	2760	1360
36	DKWC	2740	8410
4	DKWRT	2770	
2041	DLIMIT	2100	2110
2044	DLOCOR	2130	2140
2160	DMBMIN	2410	2420
2166	DMFMIN	2470	2480
654	DO	4130	4140
662	DO2	4140	4150
663	DO3	4150	4160
2152	DOFTYP	2350	2360
2032	DPACSW	1980	
2040	DPATSW	2090	2100
2051	DPCMSK	2180	2190
2052	DREGBR	2190	2200
2035	DREGSW	2060	2070
2047	DRELOC	2160	2170
1765	DSTAT	4620	4630
446400	DT.	560	
2000	DTEMP0	1630	
2001	DTEMP1	1640	
2002	DTEMP2	1650	
2003	DTEMP3	1660	
2004	DTEMP4	1670	
2005	DTEMP5	1680	
2006	DTEMP6	1690	
2007	DTEMP7	1700	
2010	DTEMP8	1710	
2011	DTEMP9	1720	
275	EQUAL	2910	
382	ERRCAL	4230	940
602	FGET	3950	3960
1701	FRCA	4410	4420
1700	FRDA	4400	4410
1702	FRLEN	4420	4430
1703	FRSTA	4430	4440
2	FUDGE	3190	3200
276	GREAT	2930	
1700	IMPLEN	990	
3170	IMPSTR	2550	
422020	INT	320	
513	IO.IN	3910	3920

R CROSS REFERENCE TABLE

525	IO.DT	3920	3930									
300000	IOBLK	2830										
1760	IORS	4570	4580									
1002	IOTO	4900	4910									
652	JMP	4110	4120									
100	JTLEN	960										
1700	JTSTRT	950	940	960	1000	4400	8640					
16	KBLEN	3610	3630	3640	3680	3690	3730	3740	1760	1960	2070	6320
10	KBNUM	3620	3670	3720	1780	1980	2090					
76	L0LOK	3630										
107	L1BFR	3670	3680	3690								
127	L1BIN	3690	3700	3720	4290							
131	L1FLG	3700	3710									
125	L1LOK	3680										
133	L1NAM	3710										
136	L2BFR	3720	3730	3740								
156	L2BIN	3740	3750	4330								
160	L2FLG	3750	3760									
154	L2LOK	3730										
162	L2NAM	3760	3770									
422026	LDR	390										
2000	LDRST	5040										
274	LESS	2920										
2022	M ACSW	1860										
10	MINBUF	3200	3610	6330								
422023	MP1	350										
422024	MP2	360										
2032	MPACSW	1980										
1004	MPOPR	4920										
1000	MPST	4880	4890	2620								
1754	MQ	4530	4540									
2016	MQSAVE	1820	1830									
2000	MTEMP0	1630										
2001	MTEMP1	1640										
2002	MTEMP2	1650										
2003	MTEMP3	1660										
2004	MTEMP4	1670										
2005	MTEMP5	1680										
2006	MTEMP6	1690										
2007	MTEMP7	1700										
2010	MTEMP8	1710										
2011	MTEMP9	1720										
422025	MTR	370										
2000	MTRST	5080										
1772	NAME	4670	4680									
540	NEWBR	3930	3940									
1771	NUMBR	4660	4670									
623	NXPTR	3960	3970									
702	OC0	4180	4190									
703	OC1	4190	4200									
704	OC2	4200	4210									
705	OC3	4210										

R CROSS REFERENCE TABLE

574646	OFF	2730						
575600	ON	2720						
1773	OVER	4680	4690					
700	OVLEN	940	1350					
1000	OVSTRT	930	920	940	4750	4880	4960	1340
2033	P10SAV	1990	2000					
2034	P11SAV	2000	2050					
2025	PACSAV	1930	1940					
2032	PACSW	1980	1990					
241	PBFLAG	3810	3820					
2017	PCSAVE	1830	1840					
227	PFLAG	3770	3780					
77	PH0	4260	4270					
126	PH1	4300	4310					
155	PH2	4340	4350					
1	PHANTO	2780						
2150	PHFLAG	2280	2330					
1700	PHLEN	2640						
2025	PHSTOR	1920	1930					
274	PIDN2	3850	3860					
270	PIDON	3840	3850					
1001	PINT	4890	4900	5620				
303	PIOUT	3860	3870					
602026	PLDR	400						
2026	PMQSAV	1940	1950					
602025	PMTR	380						
2027	PPCSAV	1950	1960					
606064	PPT	520						
2031	PSCSAV	1970	1980					
2030	PSTSAV	1960	1970					
606460	PTP	510						
606462	PTR	500						
12100	PURLEN	1010						
1775	PURNM	4700	4710					
3700	PURSTR	2560	990	1010	2560			
546	PUTIN	3940	3950					
1713	R .0	8780	3780					
1753	R AC	8800						
666	R DQ	8290	4150	4190	8530			
1754	R MQ	8810						
1755	R SC	8820						
5	R 3AC	760	2240	3800				
1756	R ACS	8830						
654	R BL7	8060	4920					
655	R BL8	8070	7330	7460	7810			
656	R CB0	8080	6170					
657	R CB1	8090	6400					
660	R CB5	8100	5530					
661	R CB7	8110						
317	R CLK	3970	2650					
665	R DBK	8150	2520					
1764	R DFN	8900						

R CROSS REFERENCE TABLE

674	R DO2	8350						
675	R DO3	8360						
664	R JMP	8140	3570					
714	R OC0	8580						
715	R OC1	8590						
716	R OC2	8600						
717	R OC3	8610						
26	R .310	980	2260	3480				
27	R .311	990	2280	3500				
646	R BIT0	8000	5370	7900				
650	R BIT5	8020	5540					
651	R BIT6	8030	5350	6000				
652	R BIT7	8040	5150	6130	6580			
662	R CBL8	8120	7590					
45	R CMP1	1390						
46	R CMP2	1400						
47	R CSPL	1410						
44	R CSWP	1380	4160					
1762	R DAP0	8880	3230					
1763	R DAP1	8890						
614	R FGET	7660	6450	7690	7700	7820		
246	R FLAG	3070	2860	2910	2960	3080	3090	3100
1701	R FRCA	8660						
1700	R FRDA	8650						
533	R IO.1	6840	7020					
1760	R IORS	8860						
425	R KBD0	5680	4940					
407	R KBD2	5380	6670					
417	R KBD4	5560	5270					
413	R KBD5	5520	5610					
507	R KBD6	6530	6460					
421	R KBD7	5590	5510					
503	R KBD8	6450	6350					
410	R KBD9	5490	5950	6480	6600	6650		
437	R MTR1	5780						
1772	R NAME	8980	4090					
1773	R OVER	8990						
304	R PIGO	3630	3580					
575	R PUT1	7440						
610	R PUT2	7570	7420					
606	R PUT4	7540	7610					
34	R RACS	1170						
32	R RDT0	1150						
33	R RDT1	1160						
230	R RPTP	2880						
235	R RPTR	2930						
242	R RSCO	2980						
335	R SWAP	4150	4260	5790				
40	R SWPS	1330	4170	4180				
1774	R TYPE	9000	4080					
4	R310TM	750	6810	6930				
305	R3REST	3730	3590	3770	3810			

R CROSS REFERENCE TABLE

R CROSS REFERENCE TABLE

127	RL1BIN	1990	2030	4600				
131	RL1FLG	2010						
351	RL1KBD	4580	2740					
125	RL1LOK	1970						
133	RL1NAM	2030						
352	RL1OUT	4600	2800					
136	RL2BFR	2070	2100	2110	2130			
156	RL2BIN	2100	2140	4670				
160	RL2FLG	2120						
354	RL2KBD	4650	2770					
154	RL2LOK	2080						
162	RL2NAM	2140						
355	RL2OUT	4670	2830					
553	RNEWBR	7210	5680	7260				
1771	RNUMBR	8960	5500	5720				
635	RNXPTR	7880	7360	7710	7920	7960		
241	RPBFLG	2970						
227	RPFLAG	2870						
274	RPIDN2	3550						
270	RPIDON	3480	3310	4040	4100	5570	5740	5770
303	RPIOUT	3620	2530					
267	RPI$V2	3390	3000					
165	RPI$VC	2230	680					
411	RPITST	5500	3110					
230	RPTP	3780	3790					
235	RPTR	3800	3810					
1775	RPURNM	9010						
561	RPUTIN	7310	5180	7380	7390	7550		
35	RRCORE	1270	5590					
234	RRFLAG	2920						
1776	RRSTRT	9020						
242	RSCO	3820	3830					
1776	RSTRT	4710						
336	RSWAP1	4160						
340	RSWAP3	4180						
1766	RUCORE	8930						
1767	RUDISK	8940						
1704	RUTEM0	8700						
1705	RUTEM1	8710						
1706	RUTEM2	8720	4240					
1707	RUTEM3	8730						
1710	RUTEM4	8740						
1711	RUTEM5	8750						
1712	RUTEM6	8760						
1770	RVALID	8950						
1755	SC	4540	4550					
640000	SCRSTR	2670						
2021	SCSAVE	1850	1860					
243	SHARP	2890						
377	SPCOD	5410						
422122	SPL	430						
1090	SPLST	4960						

R CROSS REFERENCE TABLE

777400	SPMSK	5390		
2020	STSAVE	1840	1850	
335	SWAP	3880	3890	
336	SWAP1	3890	3900	
340	SWAP3	3900	3910	
1000	SWCAT	4750	4760	
1003	SWCLK	4780	4790	4110
1004	SWERR	4790	4800	4250
1007	SWMP1	4820	4830	
1010	SWMP2	4830	4840	
1002	SWMTR	4770	4780	5780
1011	SWOPR	4840		
422022	SWP	340		
1001	SWPRR	4760	4770	
40	SWPS	3460	3470	
1005	SWSPL	4800	4810	
1006	SXSPL	4810	4820	
1300	SYSBAS	2800	2810	
41300	SYSDA	2810		
1777	SYSMAX	2820		
100	TABLEN	2630	2640	
2000	TEMP0	1630	1640	
2001	TEMP1	1640	1650	
2012	TEMP10	1730	1740	
2013	TEMP11	1740	1750	
2014	TEMP12	1750	1800	
2002	TEMP2	1650	1660	
2003	TEMP3	1660	1670	
2004	TEMP4	1670	1680	
2005	TEMP5	1680	1690	
2006	TEMP6	1690	1700	
2007	TEMP7	1700	1710	
2010	TEMP8	1710	1720	
2011	TEMP9	1720	1730	
646000	TP.	540		
376	TRCOFF	5540		
375	TRCON	5530		
2000	TTEMP0	1630		
2001	TTEMP1	1640		
2002	TTEMP2	1650		
2003	TTEMP3	1660		
2004	TTEMP4	1670		
2005	TTEMP5	1680		
2006	TTEMP6	1690		
2007	TTEMP7	1700		
2010	TTEMP8	1710		
2011	TTEMP9	1720		
6	TTYCLK	3170	3180	
3	TTYNUM	3140		
10	TTYSPD	3150	3170	
1774	TYPE	4690	4700	
1766	UCORE	4630	4640	

R CROSS REFERENCE TABLE

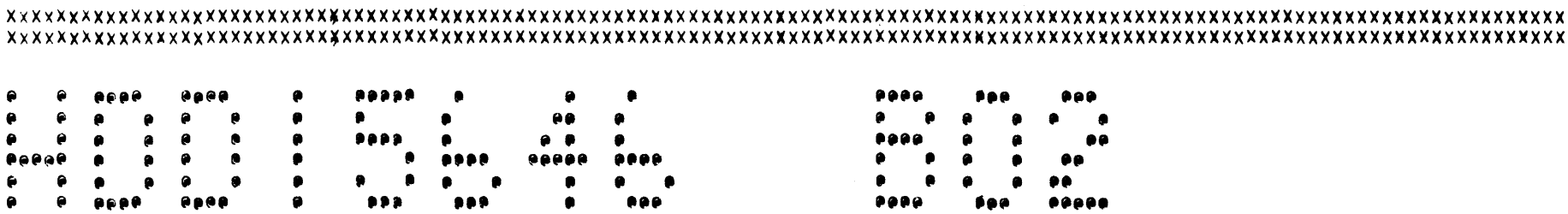
1767	UDISK	4640	4650	
336	UPARR	2940		
76	US0	4250	4260	4280
125	US1	4290	4300	4320
174	US2	4330	4340	4360
0	USER	2790		
3	USERS	2850	3200	
14000	USLEN	980	2640	
2015	USTORE	1800	1810	
75	UT0	4280		
124	UT1	4320		
173	UT2	4360		
1704	UTEM0	4440	4450	
1705	UTEM1	4450	4460	
1706	UTEM2	4460	4470	
1707	UTEM3	4470	4480	
1710	UTEM4	4480	4490	
1711	UTEM5	4490	4500	
1712	UTEM6	4500	4510	
1770	VALID	4650	4660	

R UNDEFINED SYMBOLS

#1	5630											
#2	5640											
#3	5650											
#4	5660											
#5	5680											
DEFINS	100	100	120	120	580	580	2680	2680	3210	3210	5330	5330
PURCOD	5140	5270										

R MACRO CROSS REFERENCE TABLE

ENTER	5280	3070	3730	4750	6790	6920	7210	7310	7660	7880	8290
MPOFF	5430										
SWAP	5610										



05/31/72 01:44:27

```
**********************************
**********************************
**                              **
**  PDP-9 MINI TIME-SHARING SYSTEM  **
**        SWAPPER OVERLAY       **
**            DTSS:SWP          **
**             DK0:B02          **
**             TPN:B02          **
**                              **
**********************************
**********************************
```

```
100             .TITLE  SWAPPING OVERLAY
110             .NAME   SWP--B02
120
130     *
140     *
150     *       A SWAPPER CALL CAN BE INITIATED BY EITHER THE RESIDENT PROGRAM OR A
160     *       RUNNING PROGRAM. IN EITHER CASE, THE ACTUAL SWAPPER FETCH IS ACCOMPLISHED BY
170     *       USING THE RESIDENT DISK
180     *       HANDLER TO READ IN THE SWAPPER AND TRANSFER TO IT. THIS MEANS THAT
190     *       NO MATTER WHO INITIATES THE SWAPPER CALL, IT IS ENTERED WITH THE
200     *       SYSTEM-USING-THE-DISK FLAG (DKLOK) SET. DKLOK BEING SET PREVENTS
210     *       FURTHER INTERRUPTS FROM TRYING TO CALL THE SWAPPER PREMATURELY. THEIR
220     *       OCCURENCE IS MERELY NOTED.
230     *
240     *
250     *       IF THE RESIDENT PROGRAM INITIATES A SWAPPER CALL, IT IS THE RESULT
260     *       OF A PROGRAM INTERRUPT:
270     *
280     *            1) A USER HAS TYPED A NULL (CONTROL SHIFT P), THIS KILLS HIS OUTPUT AND
290     *               RUNNING PROGRAM AND SETS A FLAG THAT HE NEEDS THE MONITOR.
300     *               IF NO OTHER USER IS CURRENTLY RUNNING, IT ALSO CAUSES THE SWAPPER
310     *               TO BE CALLED TO LOAD THE MONITOR FOR HIM.
320     *
330     *            2) THE CLOCK HAS RUN OUT AND THERE IS ANOTHER USER WHO IS READY
340     *               TO RUN (I.E. NOT I/O ROADBLOCKED). THIS CAUSES THE SWAPPER
350     *               TO BE CALLED TO SWAP THE USERS.
360     *
370     *
380     *       A RUNNING PROGRAM MAY INITIATE A SWAPPER CALL AS THE RESULT OF A PROGRAMMING
390     *       ERROR, A SPECIAL IOT INSTRUCTION, OR A DECISION.
400     *
410     *            1) WHEN A MEMORY PROTECT ROUTINE DETECTS A PROGRAMMING ERROR IT CALLS THE
420     *               SWAPPER TO TRANSFER THE USER TO THE MONITOR/SYSTEM MESSAGES PROGRAM.
430     *
440     *            2) WHEN THE USER PROGRAM EXECUTES A SPECIAL IOT INSTRUCTION A MEMORY PROTECT
450     *               ROUTINE DISCOVERS IT AND CALLS THE SWAPPER TO GET THE SPECIAL
460     *               IOT INSTRUCTION HANDLER OVERLAY AND TRANSFER TO IT.
470     *
480     *            3) A RUNNING PROGRAM MAY DECIDE TO CALL THE SWAPPER ITSELF (E.G.
490     *               MONITOR WISHES TO CALL DDT).
500     *
510     *
520     *       EACH OF THESE TYPES OF SWAPPER CALLS, EXCEPT THE LAST, HAS ITS
530     *       OWN SPECIAL ENTRY TO THE SWAPPER (E.G. SWMTR & SWSPL) TO LET THE
540     *       SWAPPER SET UP THE SWAPPING PARAMETERS AND TO MINIMIZE THE AMOUNT
550     *       OF REQUIRED RESIDENT OR OVERLAY CODE. THIS ALSO KEEPS THE SETTING OF
560     *       THE SWAPPER CONTROL WORD WITHIN THE SWAPPER, FACILITATING FUTURE CHANGES IN THE
570     *       SWAPPER.
580     *
590     *       THE INTERRUPT SYSTEM MUST BE TURNED ON FOR AS MUCH OF THE SWAPPER
600     *       OPERATING TIME AS POSSIBLE, OR ELSE TELETYPE OUTPUT CANNOT BE GUARANTEED
610     *       TO BE CONTINUOUS. BEFORE THE INTERRUPT SYSTEM CAN BE TURNED ON, THE
```

```
620      *       SAVED USER REGISTERS MUST BE COPIED TO PREVENT THEIR DESTRUCTION.
630      *       ALSO THE CURRENT CORE USER STATUS MUST BE SAVED SO THAT $RCORE CAN
640      *       BE SET TO ZERO TO SIGNAL THE RESIDENT PROGRAM THAT THERE IS NO MEMORY
650      *       PROTECT PROGRAM IN CORE, OTHERWISE THE RESIDENT PROGRAM WILL ATTEMPT
660      *       TO TRANSFER TO NON-EXISTANT ROUTINES.
670      *
680      *       THE SWAPPING OVERLAY IS CONTROLLED BY FOUR PARAMETERS PASSED TO IT
690      *       ($UTEM0-$UTEM3). IT WILL PASS TO THE CALLED PROGRAM THE CALLING
700      *       PROGRAM'S NAME (IN $UTEM4), THE CALLING PROGRAM'S OVERLAY (IN $UTEM5),
710      *       AND THE PASSED PARAMETERS ($UTEM6 - $UTEM7)
720      *       THIS IS TO ALLOW SOME INTER-PROGRAM COMMUNICATIONS.
730      *
740      *            $UTEM0: SWAPPER PARAMETER #1 -- BIT CODED WORD TO CONTROL THE SWAPPER OPERATION
750      *                                            SWAPPER PASSES CALLING PROGRAM'S NAME
760      *            $UTEM1: SWAPPER PARAMETER #2 -- NAME OF PHANTOM OR S-USER PROGRAM BEING SWAPPED
770      *                                            SWAPPER PASSES CALLING PROGRAM'S EXTENDED PC
780      *            $UTEM2: SWAPPER PARAMETER #3 -- RESTART ADDRESS OVERRIDE
790      *                                            (INTERNALLY) DIRECT RESTART
800      *
810      *            $UTEM4: PASSED PARAMETER #1
820      *            $UTEM5: PASSED PARAMETER #2
830      *            $UTEM6: PASSED PARAMETER #3
840      *
850      *       THE "CURRENT CORE USER" (CCU) IS THE JOB WHOSE PROGRAM NAME IS IN $RCORE.
860      *       INITIALLY THIS IS THE JOB WHICH THE SWAPPER INTERRUPTED.
870      *
880      *       THE "NEXT CORE USER" (NCU) IS THE ONE WHOSE USER NUMBER IS PASSED IN
890      *       $3TEM4.
900      *
910      *
920      *       THE GENERAL SWAPPER PROCEDURE IS:
930      *            1) SET UP ANY OF THE PARAMETERS THAT STILL NEED TO BE SET UP.
940      *               (SPECIAL ENTRANCES ONLY).
950      *            2) SAVE EVERYTHING THE INTERRUPT SYSTEM WOULD CLOBBER IF ALLOWED TO.
960      *            3) SAVE THE SWAPPER CONTROL PARAMETERS, THE PASSED PARAMETERS, THE CCU
970      *               AND NCU STATUS; FLAG THE SYSTEM IS USING THE DISK AND THERE
980      *               IS NO MEMORY PROTECT OVERLAY IN CORE CURRENTLY.
990      *            4) INITIALIZE THE EXIT TO BE THROUGH $PIDON AND TURN ON THE INTERRUPTS.
1000     *            5) SET UP A COPY OF THE MONITOR FOR EACH USER WHO HAS REQUESTED IT.
1010     *            6) RESTORE THE SWAPPER CONTROL PARAMETERS.
1020     *            7) DO THE REQUESTED SWAPPER ACTIVITY.
1030     *
1040             .INSRT  DEFINS
100              .IFUND  DEFINS
```

```
5720            ,LIST   ON
5730            ,END
```

SWAPPER CONSTANTS, ENTRANCES, ETC.

```
                    1050            .STITL  SWAPPER CONSTANTS, ENTRANCES, ETC.
                    1060            .HEAD   S
                    1070
                    1080    *
                    1090    *
                    1100    *       ENTRANCE VECTOR
                    1110    *
    001000          1120            .LOC    OVSTRT          START OF THE OVERLAY AREA
001000  001426      1130    SWCAT   CATLG-1
001001  601054      1140    SWPPR   JMP     SWAP0           GENERAL ENTRANCE TO THE SWAPPER
001002  601015      1150    SWMTR   JMP     SWAP3           ENTRANCE TO CALL THE MONITOR
001003  601012      1160    SWCLK   JMP     SWAP4           ENTRANCE TO EFFECT A CLOCK SWAP (SAVES RESIDENT CODE)
001004  601016      1170    SWERR   JMP     SWAP1           ENTRANCE TO OUTPUT A STANDARD SYSTEM ERROR MESSAGE
001005  601031      1180    SWSPL   JMP     SWP10           ENTRANCE WHEN MEMORY PROTECT ROUTINES DETECT A SPECIAL IOT
001006  601043      1190    SXSPL   JMP     SWP13           ENTRANCE FROM SPECIAL IOT HANDLING TO GET THE USER BACK TO NORMAL
001007  601035      1200    SWMP1   JMP     SWP11           GET MEMORY PROTECTION OVERLAY #1
001010  601037      1210    SWMP2   JMP     SWP12           GET MEMORY PROTECTION OVERLAY #2
001011  601047      1220    SWOPR   JMP     SWP14           MP#2 FOR AN OPERATE INSTRUCTION
                    1230    *
                    1240    *
                    1250    *       ENTRANCES FROM THE RESIDENT PROGRAM
                    1260    *
                    1270    *       CLOCK SWAP ENTRANCE
                    1280    *
    001012          1290    SWAP4   ...
001012  201577      1300            LAC     (662000)
001013  040702      1310            DAC     $OC0            SET THE SWAPPER CONTROL WORD
001014  601054      1320            JMP     SWAP0
                    1330    *
                    1340    *       MONITOR ENTRANCE
                    1350    *
    001015          1360    SWAP3   ...
001015  141706      1370            DZM     $UTEM2          CALL THE MONITOR INSTEAD OF AN ERROR MESSAGE
                    1380    *
                    1390    *
                    1400    *       STANDARD SYSTEM ERROR MESSAGE PRINTOUTS
                    1410    *       ENTER WITH THE MESSAGE NUMBER ALREADY SET IN $UTEM2
                    1420    *
    001016          1430    SWAP1   ...
001016  201600      1440            LAC     (652000)
001017  040702      1450            DAC     $OC0            SET THE SWAPPER CONTROL WORD
001020  201601      1460            LAC     ($MTR)
001021  040703      1470            DAC     $OC1            SET THE SYSTEM PROGRAM NAME: MONITOR/MESSAGES
001022  777777      1480    M1      LAW     -1
001023  340000      1490            TAD     0
001024  500651      1500            AND     $ADRSS
001025  041704      1510            DAC     $UTEM0          SET THE PROGRAM COUNTER TO PRINT
001026  221704      1520            LAC     $UTEM0,X
001027  041705      1530            DAC     $UTEM1          SAVE THE INSTRUCTION GENERATING THE ERROR
001030  601054      1540            JMP     SWAP0
```

S OVERLAY EXCHANGES

```
                    1550            .STITL  OVERLAY EXCHANGES
                    1560    *
                    1570    *
                    1580    *       ENTRANCES TO SWITCH OVERLAYS AROUND. MOST OF THEM ARE MORE
                    1590    *       ABRUPT THAN THE OTHER ENTRANCES. THEY DO NOT NECESSARILY FOLLOW
                    1600    *       THE NORMAL SEQUENCE OF SWAPPER OPERATIONS. THIS IS BECAUSE IT
                    1610    *       IS ASSUMED THAT THEY ARE IN-AND-OUT SO FAST. THIS APPROACH SIMPLIFIES
                    1620    *       CODING A GOOD DEAL AND ALSO LETS THE SYSTEM RUN MORE EFFICIENTLY.
                    1630    *
    001031          1640    SWP10   ...                     BRING IN THE SPECIAL IOT HANDLER OVERLAY
001031  200047      1650            LAC     $CSPL           LOAD ITS PHYSICAL ADDRESS FROM THE CORE CATALOG
001032  040040      1660            DAC     $SWPS           SET IT IN THE PARAMETERS LIST FOR THE RESIDENT DISK HANDLER
001033  761000      1670            LAW     $SPLST          GET A POINTER TO ITS ENTRANCE
001034  601052      1680            JMP     SWP19
                    1690    *
                    1700    *
                    1710    *       SWITCH FROM MEMORY PROTECTION OVERLAY #2 TO MEMORY PROTECTION OVERLAY #1
                    1720    *
    001035          1730    SWP11   ...
001035  200045      1740            LAC     $CMP1           LOAD ITS PHYSICAL BISK ADDRESS FROM THE CORE CATALOG
001036  741000      1750            SKP
                    1760    *
                    1770    *
                    1780    *       SWITCH FROM MEMORY PROTECTION OVERLAY #1 TO MEMORY PROTECTION OVERLAY #2
                    1790    *
    001037          1800    SWP12   ...
001037  200046      1810            LAC     $CMP2           LOAD ITS PHYSICAL DISK ADDRESS FROM THE CORE CATALOG
001040  040040      1820            DAC     $SWPS           SET IT IN THE PARAMETERS LIST FOR THE RESIDENT DISK HANDLER
001041  761002      1830            LAW     $IOT0           LOAD THE RESTART ADDRESS
001042  601052      1840            JMP     SWP19
                    1850    *
                    1860    *
                    1870    *       SWITCH FROM THE SPECIAL IOT HANDLER OVERLAY TO MEMORY PROTECTION #1 OVERLAY
                    1880    *
    001043          1890    SWP13   ...
001043  200045      1900            LAC     $CMP1           LOAD ITS PHYSICAL DISK ADDRESS FROM THE CORE CATALOG
001044  040040      1910            DAC     $SWPS           SET IT IN THE PARAMETERS LIST FOR THE RESIDENT DISK HANDLER
001045  760270      1920            LAW     $PIDON          LOAD THE RESTART ADDRESS
001046  601332      1930            JMP     SW121
                    1940    *
                    1950    *
                    1960    *       GET MEMORY PROTECTION OVERLAY #2 -- THE OPERATE INSTRUCTION ROUTINE
                    1970    *
    001047          1980    SWP14   ...
001047  200046      1990            LAC     $CMP2           LOAD ITS PHYSICAL DISK ADDRESS FROM THE CORE CATALOG
001050  040040      2000            DAC     $SWPS           SET IT IN THE PARAMETERS LIST FOR THE RESIDENT DISK HANDLER
001051  761004      2010            LAW     $MPOPR
                    2020    *
                    2030    *
    001052          2040    SWP19   ...
001052  040654      2050            DAC     $DO             SET THE RESTART ADDRESS
001053  601334      2060            JMP     SW120           SWAP IN THE CORRECT OVERLAY
```

S SETUP AND INITIALIZATION

```
                    2070            .STITL  SETUP AND INITIALIZATION
                    2080    *
                    2090    *
                    2100    *       ALL ENTRANCES MERGE HERE
                    2110    *       SAVE THE USER'S REGISTERS BEFORE THE INTERRUPT SYSTEM CLOBBERS THEM
                    2120    *
001054  200000      2130    SWAP0   LAC     0
001055  041776      2140            DAC     $RSTRT
001056  200005      2150            LAC     $3AC
001057  041753      2160            DAC     $AC
001060  200026      2170            LAC     $,310
001061  041723      2180            DAC     $,0+10
001062  200027      2190            LAC     $,311
001063  041724      2200            DAC     $,0+11
                    2210    *
                    2220    *       SAVE THE CURRENT CORE USER (CCU) AND NEXT CORE USER (NCU) NAMES
                    2230    *
001064  201772      2240            LAC     $NAME
001065  041575      2250            DAC     CCU
001066  140035      2260            DZM     $RCORE          SET NO CCU SO RESIDENT PROGRAM DOESN'T ASSUME MEMORY PROTECT OVERLAYS
001067  200055      2270            LAC     $3TEM4
001070  041576      2280            DAC     NCU
                    2290    *
                    2300    *       NOW THE SYSTEM IS SECURE, IT IS OK TO ALLOW INTERRUPTS AGAIN
                    2310    *
001071  700042      2320            ION
001072  760270      2330            LAW     $PIDON
001073  041573      2340            DAC     STRTWD          SET THE STANDARD EXIT, INITIALLY
                    2350    *
                    2360    *
                    2370    *       NOW SAVE THE CCU'S LOW CORE -- 12-17 (THE REST IS ALREADY SAVED)
                    2380    *
001074  760011      2390            LAW     11
001075  040010      2400            DAC     10              SET THE SAVE TO START AT LOCATION 12
001076  761724      2410            LAW     $,0+11
001077  040011      2420            DAC     11              SET THE STORE TO START AT THE IMAGE OF 12
001100  101337      2430            JMS     SW200           DO THE CORE SAVE
001101  200702      2440            LAC     $OC0            LOAD THE SWAPPER CONTROL WORD
```

S MAIN OPERATING ROUTINES

```
                    2450            .STITL  MAIN OPERATING ROUTINES
                    2460    *
                    2470    *       GENERAL SWAP ROUTINE, ACTIVATE SUBROUTINES CALLED FOR BY THE SWAPPER CONTROL WORD IN $$OC0
                    2480    *
  001102  741110    2490            SPA!RAL
  001103  601126    2500            JMP     SW00            SWAP OUT THE CURRENT USER'S CORE
  001104  741110    2510    SW09    SPA!RAL
  001105  601135    2520            JMP     SW10            SWAP OUT THE CURRENT USER'S JOB TABLE (IF NOT ALSO NCU)
  001106  741110    2530    SW19    SPA!RAL
  001107  601150    2540            JMP     SW20            SET NCU := CCU (NUMBER)
  001110  741110    2550    SW29    SPA!RAL
  001111  601155    2560            JMP     SW30            READ IN THE NCU'S JOB TABLE (IF NOT ALSO CCU)
  001112  601166    2570    SW39    JMP     SW40            NOW SEE IF WE NEED TO OVERRIDE THIS SWAP-IN WITH A MONITOR CALL
  001113  741110    2580    SW49    SPA!RAL
  001114  601203    2590            JMP     SW50            READ IN THE NCU'S CORE
  001115  741110    2600    SW59    SPA!RAL
  001116  601224    2610            JMP     SW60            READ IN THE NCU'S PHANTOM PROGRAM NAMED IN $OC1 OVER THE OLD PHANTOM CORE
  001117  741110    2620    SW69    SPA!RAL
  001120  601234    2630            JMP     SW70            READ IN THE NCU'S S-USER PROGRAM NAMED IN $OC1 OVER THE OLD USER CORE
  001121  741110    2640    SW79    SPA!RAL
  001122  601261    2650            JMP     SW80            RECORD THE NEW CCU
  001123  741110    2660    SW89    SPA!RAL
  001124  601270    2670            JMP     SW100           RESET THE RESTART ADDRESS TO THE ONE PASSED IN $OC2
  001125  601273    2680            JMP     SW110           RESTORE THE USER'S LOW CORE, PASSED PARAMETERS, OVERLAY, AND GO
                    2690    *
                    2700    *       SWAP OUT THE CURRENT CORE USER
                    2710    *
  001126  040702    2720    SW00    DAC     $OC0            FIRST SAVE THE SWAPPER CONTROL WORD
  001127  201575    2730            LAC     CCU             LOAD THE CURRENT CORE USER'S NAME
  001130  041361    2740            DAC     OUT1            SET IT FOR SWAPPING
  001131  761133    2750            LAW     .+2             SET THIS ROUTINE'S RESTART
  001132  601360    2760            JMP     OUT             SWAP OUT THE USER
  001133  200702    2770            LAC     $OC0            RELOAD THE SWAPPER CONTROL WORD
  001134  601104    2780            RET     SW09
                    2790    *
                    2800    *       SWAP OUT THE CURRENT CORE USER'S JOB TABLE (UNLESS ALSO NCU)
                    2810    *
  001135  040702    2820    SW10    DAC     $OC0            FIRST SAVE THE SWAPPER CONTROL WORD
  001136  201576    2830            LAC     NCU             LOAD THE NEXT CORE USER'S I.D.
  001137  541575    2840            SAD     CCU             CHECK FOR DIFFERENT FROM CCU
  001140  601146    2850            JMP     SW18            SAME -- DON'T BOTHER
  001141  777777    2860            LAW     -1
  001142  341771    2870            TAD     $NUMBR          FORM THE JOB TABLE FILENAME
  001143  041361    2880            DAC     OUT1            SET FOR THE WRITE
  001144  761146    2890            LAW     .+2             LOAD THE STANDARD RESTART
  001145  601360    2900            JMP     OUT             WRITE OUT THE JOB TABLE
  001146  200702    2910    SW18    LAC     $OC0            RELOAD THE SWAPPER CONTROL WORD
  001147  601106    2920            RET     SW19
                    2930    *
                    2940    *       SET NCU := CCU (NUMBER)
                    2950    *
  001150  040702    2960    SW20    DAC     $OC0            FIRST SET THE SWAPPER CONTROL WORD
```

S MAIN OPERATING ROUTINES

```
001151  201771  2970          LAC     $NUMBR          LOAD THE CURRENT CORE USER'S NUMBER
001152  041576  2980          DAC     NCU             SET IT ALSO AS THE NEXT CORE USER
001153  200702  2990          LAC     $OC0            RELOAD THE SWAPPER CONTROL WORD
001154  601110  3000          JMP     SW29
                3010  *
                3020  *
                3030  *       READ IN THE JOB TABLE OF THE USER WHOSE NUMBER IS PASSED IN NCU
                3040  *
001155  040702  3050  SW30    DAC     $OC0            FIRST SAVE THE SWAPPER CONTROL WORD
001156  201576  3060          LAC     NCU             LOAD THE NEXT CORE USER'S I.D.
001157  541575  3070          SAD     CCU             CHECK FOR DIFFERENT FROM CCU
001160  601165  3080          JMP     SW38            SAME -- DON'T BOTHER
001161  341602  3090          TAD     (-1)            FORM THE NAME OF THE NCU JOB TABLE
001162  041364  3100          DAC     IN1             SET FOR READ-IN
001163  761165  3110          LAW     .+2             LOAD THE STANDARD RESTART ADDRESS
001164  601363  3120          JMP     IN              READ IN THE JOB TABLE
001165  200702  3130  SW38    LAC     $OC0            RELOAD THE SWAPPER CONTROL WORD
                3140  *
                3150  *       SEE IF THERE IS A MONITOR CALL OUTSTANDING FOR THIS USER
                3160  *
    001166      3170  SW40    ...
001166  040702  3180          DAC     $OC0            FIRST SAVE THE SWAPPER CONTROL WORD
001167  221771  3190          LAC     $NUMBR,X        LOAD THE MONITOR CALL FLAG
001170  741200  3200          SNA                     SKIP IF THERE IS AN OUTSTANDING FLAG
001171  601201  3210          JMP     SW48            NO -- EXIT
001172  161771  3220          DZM     $NUMBR,X        CLEAR THE REQUEST
001173  201601  3230          LAC     ($MTR)
001174  040703  3240          DAC     $OC1            SET THE MONITOR PROGRAM NAME
001175  200702  3250          LAC     $OC0            LOAD THE GIVEN SWAPPER CONTROL WORD
001176  501603  3260          AND     (077777)        GET RID OF THE GIVEN LOAD CALL
001177  241604  3270          XOR     (200000)        SET A CALL TO LOAD A PHANTOM PROGRAM (MONITOR)
001200  040702  3280          DAC     $OC0            RESTORE THE CORRECTED SWAPPER CONTROL WORD
001201  200702  3290  SW48    LAC     $OC0            LOAD THE SWAPPER CONTROL WORD
001202  601113  3300          RET     SW49
                3310  *
                3320  *
                3330  *       READ IN THE NEXT CORE USER, ALLOWING FOR THE PURE CODE PORTION OF PHANTOM PROGRAMS
                3340  *
001203  040702  3350  SW50    DAC     $OC0            FIRST SAVE THE SWAPPER CONTROL WORD
001204  201576  3360          LAC     NCU             LOAD THE NCU'S NUMBER
001205  101347  3370          JMS     SW210           LOCATE HIM IN THE SWAPPER TABLE
001206  221423  3380          LAC     USER,X          LOAD THE NCU'S NAME
001207  041364  3390          DAC     IN1             AND SET IT FOR SWAPPING
001210  761212  3400          LAW     .+2
001211  601363  3410          JMP     IN              SWAP IN THE USER
001212  201775  3420          LAC     $PURNM          SEE IF THERE IS A PURE CODE PORTION TO LOAD
001213  741200  3430          SNA
001214  601222  3440          JMP     SW58            NO -- CLEAN UP AND EXIT
001215  543700  3450          SAD     $PURSTR         YES -- SEE IF THE PURE CODE IS ALREADY IN
001216  601222  3460          JMP     SW58            YES, SO DON'T RE-READ IT
001217  041364  3470          DAC     IN1             YES -- SET THE PURE CODE DATA FOR SWAP-IN
001220  761222  3480          LAW     .+2             LOAD THE STANDARD RESTART
```

```
                S                       MAIN OPERATING ROUTINES

001221  601363  3490            JMP     IN              AND READ IN THE REST OF THE PHANTOM
001222  200702  3500    SW58    LAC     SOC0            RESTORE THE SWAPPER CONTROL WORD
001223  601115  3510            RET     SW59
                3520    *
                3530    *
                3540    *       READ IN THE PHANTOM PROGRAM NAMED IN SOC1 OVER THE OLD PHANTOM COMMON
                3550    *
    001224      3560    SW60    ...
001224  740010  3570            RAL                     BYPASS THE S-USER OPTION
001225  040702  3580            DAC     SOC0            NOW SAVE THE SWAPPER CONTROL WORD
001226  763701  3590            LAW     SPURSTR+1

001227  041776  3600            DAC     SRSTRT          SET THE STANDARD PHANTOM PROGRAM START ADDRESS
001230  201605  3610            LAC     (PHANTOM)
001231  041774  3620            DAC     STYPE           SET THE TYPE TO BE A PHANTOM PROGRAM
001232  341576  3630            TAD     NCU
001233  601241  3640            JMP     SW72            READ THE NCU'S OLD PHANTOM CORE
                3650    *
                3660    *
                3670    *       READ IN THE S-USER PROGRAM NAMED IN SOC1 OVER THE OLD USER CORE
                3680    *
001234  040702  3690    SW70    DAC     SOC0            FIRST SAVE THE SWAPPER CONTROL WORD
001235  775777  3700            LAW     -8K-1           LOAD THE MAXIMUM ADDRESS
001236  041776  3710            DAC     SRSTRT          SET IT AS THE S-USER PROGRAM START
001237  141774  3720            DZM     STYPE           SET A USER-TYPE PROGRAM
001240  201576  3730            LAC     NCU             LOAD THE NCU'S TELETYPE NUMBER
001241  041772  3740    SW72    DAC     SNAME           AND SET IT ALSO AS THE SCRATCH FILE NAME
001242  041364  3750            DAC     IN1             SET IT FOR SWAP-IN
001243  761245  3760            LAW     .+2             LOAD THE STANDARD RESTART
001244  601363  3770            JMP     IN              READ THE OLD USER CORE
001245  200045  3780            LAC     SCMP1           LOAD A POINTER TO THE MEMORY PROTECTION OVERLAY #1 LOCATION
001246  041773  3790            DAC     SOVER           SET IT AS THE STANDARD SYSTEM PROGRAM OVERLAY
001247  200703  3800            LAC     SOC1
001250  041364  3810            DAC     IN1             SET THE FILENAME DESIRED FOR SWAP IN
001251  761253  3820            LAW     .+2             SET THE ROUTINE RESTART
001252  601363  3830            JMP     IN              DO THE SWAP
001253  201774  3840            LAC     STYPE           LOAD THE PROGRAM TYPE
001254  740200  3850            SZA                     SKIP FOR USER PROGRAMS
001255  203700  3860            LAC     SPURSTR         ELSE LOAD THE PURE-CODE PORTION'S NAME
001256  041775  3870            DAC     SPURNM          SET THE PURE-CODE PORTION NAME (IF ANY)
001257  200702  3880            LAC     SOC0            RESTORE THE SWAPPER CONTROL WORD
001260  601121  3890            RET     SW79
                3900    *
                3910    *
                3920    *       THE FILE HAS BEEN SWAPPED IN -- NOW SET ITS TABLE ENTRY
                3930    *
001261  040702  3940    SW80    DAC     SOC0            FIRST SAVE THE SWAPPER CONTROL WORD
001262  201771  3950            LAC     SNUMBR          LOAD THE USER NUMBER
001263  101347  3960            JMS     SW210           FIND HIS ENTRY IN THE SWAPPER TABLE
001264  201772  3970            LAC     SNAME           RELOAD HIS USER NUMBER
001265  061423  3980            DAC     USER,X          AND UPDATE THIS USER'S PROGRAM NAME IN SWAPPER'S TABLE
001266  200702  3990            LAC     SOC0            RESTORE THE SWAPPER CONTROL WORD
001267  601123  4000            RET     SW89            AND EXIT
```

S MAIN OPERATING ROUTINES

```
                    4010    *
                    4020    *
                    4030    *       OVERRIDE THE RESTART ADDRESS
                    4040    *
        001270      4050    SW100   ...                     FIRST SAVE THE SWAPPER CONTROL WORD
001270  200704      4060            LAC     $OC2            LOAD THE NEW RESTART ADDRESS
001271  740200      4070            SZA                     SKIP IF NONE
001272  041776      4080            DAC     $RSTRT          AND SET IT
                    4090    *
                    4100    *       RESTORE THE CURRENT CORE USER'S LOW CORE
                    4110    *
001273  760011      4120    SW110   LAW     11
001274  040011      4130            DAC     11              SET THE RESTORATION TO START AT LOCATION 12
001275  761724      4140            LAW     $,0+11
001276  040010      4150            DAC     10              SET THE LOAD TO START AT THE IMAGE OF LOCATION 12
001277  101337      4160            JMS     SW200           DO THE RESTORATION
                    4170    *
                    4180    *       RESTORE THE USER'S MQ AND SC
                    4190    *
001300  201755      4200            LAC     $SC             RELOAD THE OLD STEP COUNT
001301  741606      4210            XOR     (77)            COMPLEMENT THE STEP COUNT
001302  341607      4220            TAD     (640402)        DEVELOP A PSEUDO-NORMALIZE INSTRUCTION
001303  501610      4230            AND     (640477)        DELETE POSSIBLE STEP COUNT OVERFLOW
001304  041305      4240            DAC     .+1             PLACE THE NORMALIZE INSTRUCTION IN SEQUENCE
001305  740040      4250            XX                      STEP COUNT TO THE SC
001306  201754      4260            LAC     $MQ             RELOAD THE OLD MQ
001307  652000      4270            LMQ                     AND SET IT
                    4280    *
                    4290    *       READ IN THE OVERLAY AND GO
                    4300    *
001310  201773      4310            LAC     $OVER           LOAD THE OVERLAY NAME
001311  040040      4320            DAC     $SWPS           SET THE NAME OF THE OVERLAY TO READ
001312  200044      4330            LAC     $CSWP
001313  041361      4340            DAC     OUT1            SET TO COPY THE SWAPPER OUT TO UPDATE CURRENT FILENAMES
001314  761316      4350            LAW     .+2
001315  601360      4360            JMP     OUT             READ OUT THE SWAPPER
001316  700002      4370            IOF                     INHIBIT INTERRUPTS TO RE-ENTER THE RESIDENT ENVIRONMENT
001317  201771      4380            LAC     $NUMBR
001320  040035      4390            DAC     $RCORE
001321  201753      4400            LAC     $AC
001322  040005      4410            DAC     $3AC
001323  201723      4420            LAC     $,0+10
001324  040026      4430            DAC     $,310
001325  201724      4440            LAC     $,0+11
001326  040027      4450            DAC     $,311
001327  201776      4460            LAC     $RSTRT
001330  040000      4470            DAC     0
001331  201573      4480            LAC     STRTWD          SET THE ADDRESS AT WHICH TO RESTART
        001332      4490    SW121   ...
001332  040654      4500            DAC     $DO
001333  140266      4510            DZM     $DKLOK          FLAG THE SYSTEM IS DONE WITH THE DISK
        001334      4520    SW120   ...
```

S MAIN OPERATING ROUTINES

```
001334  760037  4530        LAW     $SWPS-1
001335  040010  4540        DAC     10
001336  600663  4550        JMP     $D03            READ IN A NEW OVERLAY
```

```
                S                       MISCELLANEOUS SUBROUTINES

                      4560              ,STITL  MISCELLANEOUS SUBROUTINES
                      4570
                      4580    *
                      4590    *         SUBROUTINE TO TRANSFER 12-17 TO OR FROM USER CORE IMAGE
                      4600    *
          001337      4610    SW200     ENTER
                                        ,PMC    SAVE,ON
          001337  740040                XX
          001340  777772      4620      LAW     -6
          001341  041347      4630      DAC     SW210           SET THE NUMBER OF LOCATIONS TO BE TRANSFERRED
          001342  220010      4640    SW203     LAC     10,X
          001343  060011      4650      DAC     11,X            TRANSFER ONE MORE LOCATION
          001344  441347      4660      ISZ     SW210           AND TEST FOR DONE
          001345  601342      4670      JMP     SW203           NOT DONE -- TRANSFER NEXT LOCATION
          001346  621337      4680      RET     SW200,X         YES -- RETURN
                      4690    *
                      4700    *
                      4710    *         SUBROUTINE TO LOCATE A USER'S ENTRY IN THE SWAPPER TABLE
                      4720    *
          001347      4730    SW210     ENTER
                                        ,PMC    SAVE,ON
          001347  740040                XX
          001350  541611      4740      SAD     ($US0)          CHECK FOR USER #0
          001351  761424      4750      LAW     UN0
          001352  541612      4760      SAD     ($US1)          CHECK FOR USER #1
          001353  761425      4770      LAW     UN1
          001354  541613      4780      SAD     ($US2)          CHECK FOR USER #2
          001355  761426      4790      LAW     UN2
          001356  041423      4800      DAC     USER            SET THE POINTER
          001357  621347      4810      RET     SW210,X
```

```
S                                DISK ROUTINES

                4820            ,STITL  DISK ROUTINES
                4830
                4840    *
                4850    *
                4860    *       ROUTINE TO SWAP A FILE OUT TO THE DISK
                4870    *
001360  101366  4880    OUT     JMS     CAT             CALL THE DISK ROUTINE
001361  000000  4890    OUT1    ,DSA                    TO SWAP THIS FILENAME OUT
001362  000004  4900            $DKWRT                  DISK WRITE COMMAND
                4910    *
                4920    *       ROUTINE TO SWAP A FILE IN FROM THE DISK
                4930    *
001363  101366  4940    IN      JMS     CAT             CALL THE DISK ROUTINE
001364  000000  4950    IN1     ,DSA                    TO SWAP THIS FILENAME IN
001365  000002  4960    IN2     $DKRD                   DISK READ COMMAND
                4970    *
                4980    *       SWAPPER CATALOG ROUTINE
                4990    *
                5000    *       CALLING FORMAT:
                5010    *            LAW <RETURN ADDRESS>
                5020    *            JMS CAT
                5030    *            <SIXBIT (AC!6) ASCII FILENAME>
                5040    *            <COMMAND: READ = 3; WRITE = 5>
                5050    *       ERROR MESSAGE WILL BE PRINTED IF THE FILE CANNOT BE FOUND
                5060    *
                5070    *       ROUTINE INITIALIZATION
                5080    *
    001366      5090    CAT     ENTER
                                ,PMC    SAVE,ON
001366  740040                  XX
001367  040654  5100            DAC     $DO             SET UP THE RESTART AFTER THE DISK OPERATION
001370  221366  5110            LAC     CAT,X           LOAD THE FILENAME
001371  741200  5120            SNA
001372  620654  5130            RET     $DO,X           IGNORE A NULL FILENAME
001373  441366  5140            INX     CAT
001374  041337  5150            DAC     SW200           SAVE THE FILENAME FOR THE SEARCH
001375  777747  5160            LAW     -SWPCAT
001376  041347  5170            DAC     SW210           SAVE THE NUMBER OF FILES IN THIS CATALOG
001377  761426  5180            LAW     CATLG-1
001400  040010  5190            DAC     10              SET THE POINTER TO THE CATALOG
                5200    *
                5210    *       INITIALIZATION DONE -- NOW FIND THE FILENAME
                5220    *
001401  220010  5230    CAT01   LAC     10,X            LOAD THE NEXT FILENAME FROM THE CATALOG
001402  541337  5240            SAD     SW200           IS IT THE ONE WE WANT?
001403  601413  5250            JMP     CAT09           YES -- CARRY ON
001404  441347  5260            ISZ     SW210           NO -- HAVE WE TRIED ALL OF THE POSSIBLE FILENAMES?
001405  741000  5270            SKP
001406  740040  5280            HLT                     YES, AND IT WAS NOT FOUND
001407  440010  5290            INX     10              NO, SO UPDATE THE FILENAME POINTER
001410  440010  5300            INX     10
001411  440010  5310            INX     10
```

S DISK ROUTINES

```
001412  601401  5320            JMP     CAT01           CHECK THE NEXT FILENAME
                5330    *
                5340    *       FILENAME MATCH HAS BEEN FOUND -- 10 POINTS TO IT
                5350    *
    001413      5360    CAT09   ...,
001413  220010  5370            LAC     10,X
001414  707024  5380            DLAL                    SET THE TRUE PHYSICAL DISK ADDRESS
001415  220010  5390            LAC     10,X
001416  040037  5400            DAC     $DKCA           SET THE CORE ADDRESS -1
001417  220010  5410            LAC     10,X
001420  040036  5420            DAC     $DKWC           SET THE TWO'S COMPLEMEMT WORD COUNT
001421  221366  5430            LAC     CAT,X           LOAD THE COMMAND
001422  600672  5440            JMP     $DKOVR          AND DO THE DISK OPERATION
```

S MISCELLANEOUS STORAGE

```
                    5450            ,STITL  MISCELLANEOUS STORAGE
                    5460    *
                    5470    *
                    5480    *       CONSTANTS AND POINTERS
                    5490    *
001423  000000      5500    USER    ,DSA                    POINTER TO THE TABLE ENTRY ASSOCIATED WITH THE CURRENT USER
001424  000076      5510    UN0     $US0                    CURRENT FILENAME FOR USER #0
001425  000125      5520    UN1     $US1                    CURRENT FILENAME FOR USER #1
001426  000154      5530    UN2     $US2                    CURRENT FILENAME FOR USER #2
                    5540    *
                    5550    *
                    5560    *       SWAPPER CATALOG FORMAT:
                    5570    *            SIXBIT (ACI6) ASCII NAME
                    5580    *            PHYSICAL DISK ADDRESS
                    5590    *            CORE ADDRESS - 1
                    5600    *            TWO'S COMPLEMENT FILE LENGTH
                    5610    *            (TRANSFER CARD OMITTED)
                    5620    *
                    5630    *
    000031          5640    SWPCAT  ,EQU    3*3+2*2+2+1     SCRATCH + PHANTOM + S-USER + SWAPPER
    001427          5650    CATLG   ,BLOCK  SWPCAT*4
                    5660    *
                    5670    *
                    5680    *       TEMPORARY STORAGE: IN THE COURSE OF ITS OPERATIONS THE SWAPPER MAY
                    5690    *       OVER WRITE THE USER JOB TABLE TEMPORARY STORAGE LOCATIONS ($UTEM0-$UTEM6).
                    5700    *       THEREFORE SWAPPER FIRST COPIES THEM INTO $OC0-TEMP6.
                    5710    *
001573  000000      5720    STRTWD  ,DSA
001574  000000      5730    OVOLD   ,DSA
001575  000000      5740    CCU     ,DSA
001576  000000      5750    NCU     ,DSA
001577  662000      5760            ,END    OVSTRT
001600  652000
001601  422025
001602  777777
001603  077777
001604  200000
001605  000001
001606  000077
001607  640402
001610  640477
001611  000076
001612  000125
001613  000154

TRANSFER ADDRESS  601000
```

S CROSS REFERENCE TABLE

1713	.0	4510	4520	2180	2200	2410	4140	4420	4440
26	.310	3400	2170	4430					
27	.311	3410	2190	4450					
4464	.DT	570							
6460	.TP	550							
2023	10SAVE	1870	1880						
2024	11SAVE	1880	1920						
5	3AC	3370	2150	4410					
305	3RE$T	3870	3880						
51	3TEM0	3530	3540						
52	3TEM1	3540	3550						
53	3TEM2	3550	3560						
54	3TEM3	3560	3570						
55	3TEM4	3570	3580	2270					
56	3TEM5	3580	3590						
57	3TEM6	3590	3600						
50	3TM20	3520	3530						
2	3TM21	3350							
3	3TM22	3360							
14000	7K	1030							
16000	8K	1020	910	1010	2640	2650	3780		
1753	AC	4520	4530	2160	4400				
1756	ACS	4550	4560						
2015	ACSAVE	1810	1820						
2022	ACSW	1860	1870						
651	ADR$S	4100	4110	1500					
390	AT	2950							
300	ATSGN	2900							
422030	BAS	420							
2151	BCNTRL	2330	2340						
634	BIT0	3970	3980						
641	BIT17	4020	4030						
635	BIT36	3980	3990						
636	BIT5	3990	4000						
637	BIT6	4000	4010						
640	BIT7	4010	4020						
642	BL7	4030	4040						
643	BL8	4040	4050						
2000	BOUNDA	970	960	980	990	1000	1630	5040	5080
377	BRK	5550							
2170	BUFFER	2490	2550						
1000	BUFLEN	2500	2550						
644	CB0	4050	4060						
645	CB1	4060	4070						
646	CB5	4070	4080						
647	CB7	4080	4090						
650	CBL8	4090	4100						
6	CHRMAX	3180	3200						
2	CHRPAK	3130	3200						
50	CLKMAX	2840	3180						
60	CLKSPD	3160	3170						
1757	CLOCK	4560	4570						

S CROSS REFERENCE TABLE

45	CMP1	3490	3500	1740	1900	3780
46	CMP2	3500	3510	1810	1990	
6	CNTRL	3380	3390			
2053	COMFLG	2200	2210			
2150	COMSTO	2270	2280			
16000	CORMAX	910	980			
47	CSPL	3510	3520	1650		
44	CSWP	3480	3490	4330		
60	CTBFR	3600	3630	3640		
100	CTBIN	3640	3650	3670	4250	
2000	CTEMP0	1630				
2001	CTEMP1	1640				
2002	CTEMP2	1650				
2003	CTEMP3	1660				
2004	CTEMP4	1670				
2005	CTEMP5	1680				
2006	CTEMP6	1690				
2007	CTEMP7	1700				
2010	CTEMP8	1710				
2011	CTEMP9	1720				
102	CTFLG	3650	3660			
104	CTNAM	3660				
2043	D PC	2120	2130			
2154	D BCA	2370	2380			
2153	D BDA	2360	2370			
2163	D FDA	2440	2450			
2042	D LOC	2110	2120			
2022	D ACSW	1860				
2156	D BALT	2390	2400			
2155	D BLEN	2380	2390			
2161	D BMAX	2420	2430			
2157	D BMIN	2400	2410			
2162	D BPTR	2430	2440			
2167	D FMAX	2480	2490			
2165	D FMIN	2460	2470			
2046	D MASK	2150	2160			
2164	D MFDA	2450	2460			
2036	DADRSW	2070	2080			
1762	DAP0	4590	4600			
1763	DAP1	4600	4610			
653	DBK	4120	4130			
24	DBKNUM	2220	2270			
2054	DBKTAB	2210	2270			
2035	DBSTOR	2050	2060			
422027	DDT	410				
12000	DDTST	5000				
2037	DDUMSW	2080	2090			
1761	DFLAG	4580	4590			
1764	DFN	4610	4620			
2151	DFTYPE	2340	2350			
2045	DHICOR	2140	2150			
2050	DINDIR	2170	2180			

S CROSS REFERENCE TABLE

100	DK0	4270					
127	DK1	4310					
156	DK2	4350					
37	DKCA	2750	5400				
675	DKDON	4170	4180				
16000	DKLEN	2650	2660				
34	DKLENB	2660					
266	DKLOK	3830	3840	4510			
672	DKOVR	4160	4170	5440			
2	DKRD	2760	4960				
36	DKWC	2740	5420				
4	DKWRT	2770	4900				
2041	DLIMIT	2100	2110				
2044	DLOCOR	2130	2140				
2160	DMBMIN	2410	2420				
2166	DMFMIN	2470	2480				
654	DQ	4130	4140	2050	4500	5100	5130
662	DQ2	4140	4150				
663	DQ3	4150	4160	4550			
2152	DQFTYP	2350	2360				
2032	DPACSW	1980					
2040	DPATSW	2090	2100				
2051	DPCMSK	2180	2190				
2052	DREGBR	2190	2200				
2035	DREGSW	2060	2070				
2047	DRELOC	2160	2170				
1765	DSTAT	4620	4630				
446400	DT.	560					
2000	DTEMP0	1630					
2001	DTEMP1	1640					
2002	DTEMP2	1650					
2003	DTEMP3	1660					
2004	DTEMP4	1670					
2005	DTEMP5	1680					
2006	DTEMP6	1690					
2007	DTEMP7	1700					
2010	DTEMP8	1710					
2011	DTEMP9	1720					
275	EQUAL	2910					
602	FGET	3950	3960				
1701	FRCA	4410	4420				
1700	FRDA	4400	4410				
1702	FRLEN	4420	4430				
1703	FRSTA	4430	4440				
2	FUDGE	3190	3200				
276	GREAT	2930					
1700	IMPLEN	990					
3170	IMPSTR	2550					
422020	INT	320					
513	IO.IN	3910	3920				
525	IO.OT	3920	3930				
300000	IOBLK	2830					

S CROSS REFERENCE TABLE

1760	IORS	4570	4580											
1002	IOTO	4900	4910	1830										
652	JMP	4110	4120											
100	JTLEN	960												
1700	JTSTRT	950	940	960	1000	4400								
16	KBLEN	3610	3630	3640	3680	3690	3730	3740						
10	KBNUM	3620	3670	3720										
76	LOLOK	3630												
107	L1BFR	3670	3680	3690										
127	L1BIN	3690	3700	3720	4290									
131	L1FLG	3700	3710											
125	L1LOK	3680												
133	L1NAM	3710												
136	L2BFR	3720	3730	3740										
156	L2BIN	3740	3750	4330										
160	L2FLG	3750	3760											
154	L2LOK	3730												
162	L2NAM	3760	3770											
422026	LOR	390												
2000	LORST	5040												
274	LSS	2920												
2022	M ACSW	1860												
10	MINBUF	3200	3610											
422023	MP1	350												
422024	MP2	360												
2032	MPACSW	1980												
1004	MPOPR	4920	2010											
1000	MPST	4880	4890											
1754	MQ	4530	4540	4260										
2016	MQSAVE	1820	1830											
2000	MTEMP0	1630												
2001	MTEMP1	1640												
2002	MTEMP2	1650												
2003	MTEMP3	1660												
2004	MTEMP4	1670												
2005	MTEMP5	1680												
2006	MTEMP6	1690												
2007	MTEMP7	1700												
2010	MTEMP8	1710												
2011	MTEMP9	1720												
422025	MTR	370	1460	3230										
2000	MTRST	5080												
1772	NAME	4670	4680	2240	3740	3970								
540	NEWBR	3930	3940											
1771	NUMBR	4660	4670	2870	2970	3190	3220	3950	4380					
623	NXPTR	3960	3970											
702	OC0	4180	4190	1310	1450	2440	2720	2770	2820	2910	2960	2990	3050	3130
			3180	3250	3280	3290	3350	3500	3580	3690	3880	3940	3990	
703	OC1	4190	4200	1470	3240	3800								
704	OC2	4200	4210	4060										
705	OC3	4210												
574686	OFF	2730												

S CROSS REFERENCE TABLE

575600	ON	2720							
1773	OVER	4680	4690	3790	4310				
790	OVLEN	940							
1000	OVSTRT	930	920	940	4750	4880	4960	1120	5760
2043	P10$AV	1990	2000						
2034	P11$AV	2000	2050						
2025	PAC$AV	1930	1940						
2032	PAC$W	1980	1990						
241	PBFLAG	3810	3820						
2037	PCSAVE	1830	1840						
227	PFLAG	3770	3780						
77	PH0	4260	4270						
126	PH1	4300	4310						
155	PH2	4340	4350						
1	PHANTQ	2780	3610						
2150	PHFLAG	2280	2330						
1700	PHLEN	2640							
2025	PHSTOR	1920	1930						
274	PIDN2	3850	3860						
270	PIDON	3840	3850	1920	2330				
1091	PINT	4890	4900						
303	PIOUT	3860	3870						
602026	PLDR	400							
2026	PMQ$AV	1940	1950						
602025	PMTR	380							
2027	PPC$AV	1950	1960						
606064	PPT	520							
2031	PSC$AV	1970	1980						
2030	PST$AV	1960	1970						
606460	PTP	510							
606462	PTR	500							
12100	PURLEN	1010							
1775	PURNM	4700	4710	3420	3870				
3700	PUR$TR	2560	990	1010	2560	3450	3590	3860	
546	PUTIN	3940	3950						
34	RAC$	3440							
6	RCNT	3390							
35	RCORE	3450	2260	4390					
1003	RDBLK	4910	4920						
32	RDT0	3420							
33	RDT1	3430							
422021	RES	330							
40	RESCAT	3470	3480						
1000	RESLEN	920							
234	RFLAG	3790	3800						
230	RPTP	3780	3790						
235	RPTR	3800	3810						
242	RSCD	3820	3830						
1776	RSTRT	4710	2140	3600	3710	4080	4460		
1363	S IN	4940	3120	3410	3490	3770	3830		
1022	S M1	1480							
1366	S CAT	5090	4880	4940	5110	5140	5430		

S CROSS REFERENCE TABLE

1575	S CCU	5740	2250	2730	2840	3070			
1364	S IN1	4950	3100	3390	3470	3750	3810		
1365	S IN2	4960							
1576	S NCU	5750	2280	2830	2980	3060	3360	3630	3730
1360	S OUT	4880	2760	2900	4360				
1424	S UN0	5510	4750						
1425	S UN1	5520	4770						
1426	S UN2	5530	4790						
1361	S OUT1	4890	2740	2880	4340				
1126	S SW00	2720	2500						
1104	S SW09	2510	2780						
1135	S SW10	2820	2520						
1146	S SW18	2910	2850						
1106	S SW19	2530	2920						
1150	S SW20	2960	2540						
1110	S SW29	2550	3000						
1155	S SW30	3050	2560						
1165	S SW38	3130	3080						
1112	S SW39	2570							
1166	S SW40	3170	2570						
1201	S SW48	3290	3210						
1113	S SW49	2580	3300						
1203	S SW50	3350	2590						
1222	S SW58	3500	3440	3460					
1115	S SW59	2600	3510						
1224	S SW60	3560	2610						
1117	S SW69	2620							
1234	S SW70	3690	2630						
1241	S SW72	3740	3640						
1121	S SW79	2640	3890						
1261	S SW80	3940	2650						
1123	S SW89	2660	4000						
1423	S USER	5500	3380	3980	4800				
1755	SC	4540	4550	4200					
1401	SCAT01	5230	5320						
1433	SCAT09	5360	5250						
1427	SCATLG	5650	1130	5180					
640000	SCRSTR	2670							
2021	SCSAVE	1850	1860						
283	SHARP	2890							
1574	SOVOLD	5730							
377	SPCOD	5410							
422122	SPL	430							
1000	SPLST	4960	1670						
777400	SPMSK	5390							
1270	SSW100	4050	2670						
1273	SSW110	4120	2680						
1334	SSW120	4520	2060						
1332	SSW121	4490	1930						
1337	SSW200	4610	2430	4160	4680	5150	5240		
1342	SSW203	4640	4670						
1347	SSW210	4730	3370	3960	4630	4660	4810	5170	5260

S CROSS REFERENCE TABLE

1054	SSWAP0	2130	1140	1320	1540				
1016	SSWAP1	1430	1170						
1015	SSWAP3	1360	1150						
1012	SSWAP4	1290	1160						
1000	SSWCAT	1130							
1003	SSWCLK	1160							
1004	SSWERR	1170							
1007	SSWMP1	1200							
1010	SSWMP2	1210							
1002	SSWMTR	1150							
1011	SSWOPR	1220							
1031	SSWP10	1640	1180						
1035	SSWP11	1730	1200						
1037	SSWP12	1800	1210						
1043	SSWP13	1890	1190						
1047	SSWP14	1980	1220						
1052	SSWP19	2040	1680	1840					
1001	SSWPPR	1140							
1005	SSWSPL	1180							
1006	SSXSPL	1190							
1573	STRTWD	5720	2340	4480					
2020	STSAVE	1840	1850						
335	SWAP	3880	3890						
336	SWAP1	3890	3900						
340	SWAP3	3900	3910						
1000	SWCAT	4750	4760						
1003	SWCLK	4780	4790						
1004	SWERR	4790	4800						
1007	SWMP1	4820	4830						
1010	SWMP2	4830	4840						
1002	SWMTR	4770	4780						
1011	SWOPR	4840							
422022	SWP	340							
31	SWPCAT	5640	5160	5650					
1001	SWPPR	4760	4770						
40	SWPS	3460	3470	1660	1820	1910	2000	4320	4530
1005	SWSPL	4800	4810						
1006	SXSPL	4810	4820						
1390	SYSBAS	2800	2810						
41390	SYSDA	2810							
1777	SYSMAX	2820							
180	TABLEN	2630	2640						
2000	TEMP0	1630	1640						
2001	TEMP1	1640	1650						
2012	TEMP10	1730	1740						
2013	TEMP11	1740	1750						
2014	TEMP12	1750	1800						
2002	TEMP2	1650	1660						
2003	TEMP3	1660	1670						
2004	TEMP4	1670	1680						
2005	TEMP5	1680	1690						
2006	TEMP6	1690	1700						

S CROSS REFERENCE TABLE

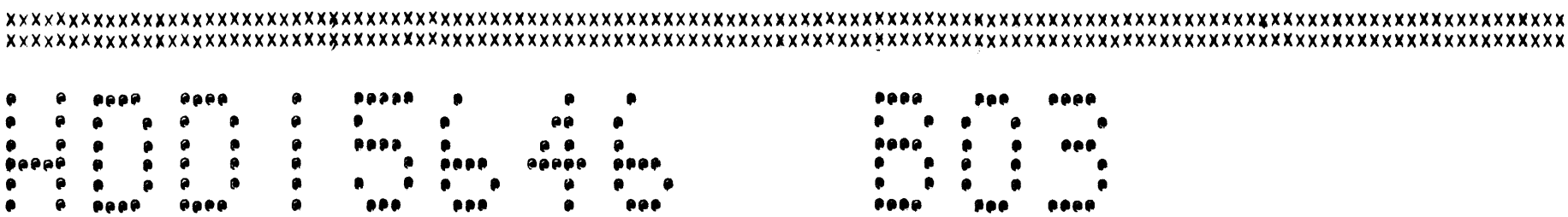
2007	TEMP7	1700	1710			
2010	TEMP8	1710	1720			
2011	TEMP9	1720	1730			
646000	TP.	540				
376	TRCOFF	5540				
375	TRCON	5530				
2000	TTEMP0	1630				
2001	TTEMP1	1640				
2002	TTEMP2	1650				
2003	TTEMP3	1660				
2004	TTEMP4	1670				
2005	TTEMP5	1680				
2006	TTEMP6	1690				
2007	TTEMP7	1700				
2010	TTEMP8	1710				
2011	TTEMP9	1720				
6	TTYCLK	3170	3180			
3	TTYNUM	3140				
10	TTYSPD	3150	3170			
1774	TYPE	4690	4700	3620	3720	3840
1766	UCORE	4630	4640			
1767	UDISK	4640	4650			
336	UPARR	2940				
76	US0	4250	4260	4280	4740	5510
125	US1	4290	4300	4320	4760	5520
154	US2	4330	4340	4360	4780	5530
0	USER	2790				
3	USERS	2850	3200			
14000	USLEN	980	2640			
2015	USTORE	1800	1810			
75	UT0	4280				
124	UT1	4320				
153	UT2	4360				
1704	UTEM0	4440	4450	1510	1520	
1705	UTEM1	4450	4460	1530		
1706	UTEM2	4460	4470	1370		
1707	UTEM3	4470	4480			
1710	UTEM4	4480	4490			
1711	UTEM5	4490	4500			
1712	UTEM6	4500	4510			
1770	VALID	4650	4660			

S UNDEFINED SYMBOLS

#1	5630											
#2	5640											
#3	5650											
#4	5660											
#5	5680											
DEFINS	100	100	120	120	580	580	2680	2680	3210	3210	5330	5330
PURCOD	5140	5270										

S MACRO CROSS REFERENCE TABLE

ENTER	5280	4610	4730	5090
MPOFF	5430			
SWAP	5610			



05/31/72 01:48:01

```
**********************************
**********************************
**                              **
**  PDP-9 MINI TIME-SHARING SYSTEM  **
**   MEMORY PROTECTION OVERLAY #1   **
**            DTSS:MP1              **
**             DK0:B03              **
**             TPN:B03              **
**                              **
**********************************
**********************************
```

```
100             .NAME   MP1--B03
110             .INSRT  MP0
100             .TITLE  ROUTINES TO SERVICE MEMORY PROTECT VIOLATIONS
110             .HEAD   M
120             .INSRT  DEFINS
100             .IFUND  DEFINS
```

```
5720            .LIST   ON
5730            .END
130             .HEAD   M
```

```
                M                       MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS

                                140             .STITL  MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS
                                150
                                160
                                170     TABLE   .DEFIN                  IOT TABLE ENTRY
                                180             .PMC    SAVE.OFF
                                190             #1/16.+740000
                                200             .PMC    RESTORE
                                210             .ENDM
                    001000      220             .LOC    OVSTRT
                    000702      230     INSTR   .EQU    $OC0
                    000702      240     TEMP2   .EQU    $OC0
                    000703      250     OPCOD   .EQU    $OC1
                    000703      260     TEMP3   .EQU    $OC1
                    000704      270     TEMP4   .EQU    $OC2
                    000705      280     TEMP5   .EQU    $OC3
001000  601027      290     MPSTRT  JMP     MPST            OVERLAY ENTRANCE VECTOR
001001  601006      300             JMP     PINT-1          ENTRANCE FOR PROGRAM INTERRUPT SIMULATION
001002  601214      310             JMP     IOT0
001003  001265      320             RDBLK                   POINTER TO THE ROADBLOCK ROUTINE
001004  601631      330             JMP     .OPR.           POINTER TO OPERATE HANDLING
001005  000000      340     PINST   .DSA                    PERMANENT INSTRUCTION SAVE IN CASE IT IS A GRAPHICS II INSTRUCTION
                    350     *
                    360     *       AN IOT INTERRUPT HAS OCCURRED -- GENERATE A USER PI INTERRUPT IF THE PI SYSTEM IS ON
                    370     *       TREAT AS IF LOCATION 1 CONTAINED AN XCT OF THE USER'S LOCATION 1 -- ONE XCT IS STILL LEGAL
                    380     *
001006  101162      390             JMS     REGSAVE
001007  100525      400     PINT    JMS     $IO,OT          SAVE USER TEMPS
001010  201760      410             LAC     $IORS           LOAD THE USER'S IORS WORD
001011  740110      420             SMA!RAL
001012  600270      430             RET     $PIDON          EXIT IF PI SYSTEM IS NOT ENABLED
001013  744020      440             CLL!RAR
001014  041760      450             DAC     $IORS           ELSE TURN OFF THE PI SYSTEM
001015  200000      460             LAC     0
001016  501633      470             AND     (677777)        REMOVE THE MEMORY PROTECT BIT
001017  041713      480             DAC     $,0             SET THE USER PC IN HIS LOCATION 0
001020  500634      490             AND     $BIT0           SAVE THE LINK STATUS
001021  341634      500             TAD     (100001)
001022  040000      510             DAC     0               FAKE THE NEXT INSTRUCTION CAME FROM LOCATION 1 WITH MEMORY PROTECT ON
001023  201714      520             LAC     $,0+1           LOAD THE USER'S LOCATION 1 INSTRUCTION
001024  140704      530             DZM     TEMP4           INITIALIZE THE COUNT OF MEMORY OVERLAY EXCHANGES
001025  140705      540             DZM     TEMP5           INITIALIZE THE XCT COUNT
001026  601042      550             JMP     MP111+1         AND CHECK THE INSTRUCTION
                    560
                    570
                    580
                    001027      590     MPST    ...
001027  101162      600             JMS     REGSAVE         SAVE THE MQ AND SC
001030  140704      610             DZM     TEMP4           INITIALIZE THE COUNT OF MEMORY OVERLAY EXCHANGES
001031  140705      620             DZM     TEMP5           INITIALIZE THE XCT COUNT (CHECK FOR XCT LOOPS)
                    630     *
                    640     *       CHECK TO SEE IF THE VIOLATION WAS CAUSED BY AN ATTEMPT TO TRANSFER TO PROTECTED MEMORY
                    650     *
```

M MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS

```
001032  200000  660           LAC     0               LOAD THE USER PC
001033  500651  670           AND     SADRSS          GET RID OF NON-ADDRESS BITS
001034  341635  680           TAD     (-BOUNDARY)
001035  755101  690           SPA!CLA!CMA!CLL         SKIP UNLESS A TRANSFER TO PROTECTED MEMORY WAS REQUESTED
001036  601240  700           JMP     ERR1            YES -- VIOLATION WAS A TRANSFER TO PROTECTED MEMORY -- ILLEGAL
                710     *
                720     *       THE VIOLATION WAS NOT CAUSED BY AN ATTEMPT TO TRANSFER TO PROTECTED MEMORY
                730     *       NOW GET THE VIOLATING INSTRUCTION AND SEPARATE OUT ITS OP CODE
                740     *
001037  340000  750           TAD     0               YIELDS LOCATION OF THE OFFENDING INSTRUCTION AND SETS THE LINK
001040  040702  760           DAC     INSTR
001041  220702  770   MP111   LAC     INSTR,X
001042  040702  780           DAC     INSTR           INSTR CONTAINS THE BAD INSTRUCTION
001043  041005  790           DAC     PINST           SAVE THE INSTRUCTION IN CASE IT IS A GRAPHICS II INSTRUCTION
001044  501636  800           AND     (NOP)
001045  741200  810           SNA
001046  601236  820           JMP     ERR3            CAL IS AN ILLEGAL INSTRUCTION
001047  040703  830           DAC     OPCOD           SAVE THE OP CODE OF THE VIOLATING INSTRUCTION
                840     *
                850     *       NOW CHECK FOR A MICRO-CODED INSTRUCTION; OPERATE OR IOT
                860     *
001050  541637  870           SAD     (EAE)
001051  601115  880           JMP     O,K.            EAE INSTRUCTIONS ARE INNOCENT
001052  541636  890           SAD     (OPR)
001053  601631  900   OPRST   JMP     .OPR.           OPERATE INSTRUCTION VIOLATION
001054  541640  910           SAD     (IOT)
001055  601202  920           JMP     .IOT.           SERVICE IOT VIOLATION
```

M MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS

```
                    930             .EJECT
                    940     *
                    950     *       NOT A MICROCODED INSTUCTION, SO ESTABLISH THE EFFECTIVE INSTRUCTION
                    960     *
001056  200702      970             LAC     INSTR
001057  501641      980             AND     (020000)
001060  741200      990             SNA                     SKIP IF THE VIOLATING INSTRUCTION HAD THE INDIRECT BIT ON
001061  601073     1000             JMP     MPBA1           ELSE CARRY ON NORMALLY
001062  200702     1010             LAC     INSTR           LOAD THE ORIGINAL INSTRUCTION
001063  501642     1020             AND     (17770)         RETAIN THE ADDRESS ONLY -- DELETING THE LOW ORDER THREE BITS
001064  541643     1030             SAD     (10)            SKIP IF ADDRESSED LOCATION WAS NOT AUTO-INDEX REGISTER
001065  460702     1040             INX     INSTR,X         ELSE INCREMENT THE LOCATION
001066  740000     1050             NOP                     IT JUST MIGHT SKIP
001067  220702     1060             LAC     INSTR,X
001070  500651     1070             AND     SADRSS          RECOVER THE EFFECTIVE ADDRESS
001071  240703     1080             XOR     OPCOD           AND ADD THE OP CODE BACK IN
001072  040702     1090             DAC     INSTR           SAVE THE (CONSTRUCTED) EFFECTIVE INSTRUCTION
                    1100    *
                    1110    *       VIOLATION WAS A MEMORY REFERENCE INSTRUCTION (OTHER THAN CAL, JMP, JMS, OR XCT)
                    1120    *       WHICH ATTEMPTED TO REFERENCE A LOCATION BELOW THE BOUNDARY.
                    1130    *       OR THE VIOLATION WAS AN XCT OR PI INTERRUPT AND THE REFERENCE MAY NOT BE TO PROTECTED MEMORY
                    1140    *       REFERENCES TO 0-7 & 21-27 MUST BE FAKED; REFERENCES TO 10-20 & 30-37 ARE CARRIED OUT LITERALLY
                    1150    *
001073  200702     1160     MPBA1   LAC     INSTR           LOAD THE OFFENDING INSTRUCTION
001074  500651     1170             AND     SADRSS          RETAIN JUST THE ADDRESS BITS
001075  341644     1180             TAD     (-10)
001076  741100     1190             SPA
001077  601112     1200             JMP     FAKIT           REFERENCE TO 0-7
001100  341644     1210             TAD     (-10)
001101  741100     1220             SPA
001102  601115     1230             JMP     O.K.            REFERENCE TO 10-17
001103  341645     1240             TAD     (-20)
001104  741100     1250             SPA
001105  601112     1260             JMP     FAKIT           REFERENCE TO 21-37
001106  341646     1270             TAD     (-BOUNDARY+40)
001107  751100     1280             SPA!CLA
001110  601241     1290             JMP     ERR2            THE REFERENCE IS TO 40-BOUNDARY
001111  601115     1300             JMP     O.K.            THE REFERENCE IS ABOVE THE BOUNDARY
                    1310    *
                    1320    *       CONVERT THE LEGAL PROTECTED MEMORY REFERENCE TO A REFERENCE TO THE USER TABLE
                    1330    *
001112  201647     1340     FAKIT   LAC     ($.0)
001113  340702     1350             TAD     INSTR           CONVERT REFERENCE TO POINT TO USER TABLE IMAGE
001114  040702     1360             DAC     INSTR
                    1370    *
                    1380    *       NOW DO THE USER INSTRUCTION
                    1390    *
001115              1400     O.K.    ...                     NOTE THE USER MQ, SC, 10, & 11 ARE STILL O.K. AT THIS POINT
                    1410                                     UNLESS ARRIVED AS A PI INTERRUPT, IN THAT CASE 10 & 11
                    1420                                     MAY BE OFF, BUT THE USER'S LOCATION 1 CAN'T LEGALLY BE A
                    1430                                     MEMORY REFERENCE INSTRUCTION, ANYWAY.
                    1440    *
```

M MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS

```
                    1450    *       SPECIAL CASE OP CODES
                    1460    *
001115  200703      1470    MP12    LAC     OPCOD           RELOAD THE OP CODE
001116  540634      1480            SAD     $BIT0
001117  601155      1490            JMP     .XCT.           VIOLATION WAS AN XCT INSTRUCTION
001120  541650      1500            SAD     (JMS)
001121  601142      1510            JMP     .JMS.           VIOLATION WAS CAUSED BY A JMS INSTRUCTION
001122  541651      1520            SAD     (JMP)
001123  601146      1530            JMP     .JMP.
001124  200000      1540            LAC     0
001125  740010      1550            RAL                     RESTORE THE USER'S LINK FOR THIS OPERATION
001126  200005      1560            LAC     $3AC            AND ALSO HIS AC
001127  400702      1570            XCT     INSTR           EXECUTE THE USER'S INSTRUCTION
001130  741000      1580            SKP                     AVOID INCREMENTING THE RETURN IF THE USER'S INSTRUCTION DID NOT SKIP
001131  440000      1590            INX     0               BUT BUMP THE RETURN IF THE USER'S INSTRUCTION DID SKIP
001132  040005      1600            DAC     $3AC            AND SAVE THE USER'S AC
001133  200000      1610            LAC     0               INITIATE SAVING THE CURRENT USER LINK
001134  500644      1620            AND     $CB0            CLEAR THE OLD LINK
001135  741400      1630            SZL                     IS THE LINK ON?
001136  240634      1640            XOR     $BIT0           YES, SO SAVE IT
001137  040000      1650    MP15    DAC     0               RESAVE THE CURRENT USER RETURN, WITH LINK
001140  101170      1660            JMS     REGRES          RESTORE HIS REGISTERS THAT WON'T GET OTHERWISE RESTORED
001141  600274      1670            RET     $PIDN2
                    1680    *
                    1690    *       SPECIAL MEMORY REFERENCE INSTRUCTIONS
                    1700    *
001142  200000      1710    .JMS.   LAC     0               LOAD THE USER'S PC
001143  501633      1720            AND     (677777)        TURN OFF THE MEMORY PROTECT BIT
001144  060702      1730            DAC     INSTR,X         SET THE USER PC AT THE START OF THE SUBROUTINE
001145  440702      1740            INX     INSTR           AND INCREMENT THE TRANSFER
                    1750
001146  200000      1760    .JMP.   LAC     0               LOAD THE USER'S PC
001147  501640      1770            AND     (700000)        KEEP THE HIGH ORDER BITS OF THE PC
001150  040000      1780            DAC     0
001151  200702      1790            LAC     INSTR
001152  500651      1800            AND     $ADRSS          GET THE NEW USER PC
001153  240000      1810            XOR     0               COMBINE IT WITH THE OLD HIGH-ORDER BITS
001154  601137      1820            JMP     MP15            EXIT
                    1830
001155  200705      1840    .XCT.   LAC     TEMP5           LOAD THE XCT COUNT
001156  750200      1850            SZA!CLA
001157  601235      1860            JMP     ERR4            CHAINED XCT'S NOT YET LEGAL
001160  440705      1870    XCT1    INX     TEMP5           NOW COUNT THE XCT
001161  601041      1880            JMP     MP111           AND ITERATE
                    1890
    001162          1900    REGSAVE ENTER                   SAVE THE REGISTERS THAT HAVEN'T ALREADY BEEN SAVED
                                    .PMC    SAVE,ON
001162  740040                      XX
001163  641002      1910            LACQ                    THE FOLLOWING LOCATIONS MAY ALTER IF THE VIOLATION WAS AN IOT INSTRUCTION
001164  041754      1920            DAC     $MQ
001165  641001      1930            LACS
001166  041755      1940            DAC     $SC
```

```
                M                               MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS

001167  621162  1950            RET     REGSAVE,(
                1960
    001170      1970    REGRES  ENTER                   RESTORE THE REGISTERS THAT REGSAVE SAVED
                        ,PMC    SAVE,ON
001170  740040                    XX
001171  201755  1980            LAC     $SC             RELOAD THE OLD STEP COUNT
001172  241652  1990            XOR     (77)            COMPLEMENT THE STEP COUNT
001173  341653  2000            TAD     (640402)        DEVELOP A PSEUDO-NORMALIZE INSTRUCTION
001174  501654  2010            AND     (640477)        DELETE POSSIBLE STEP COUNT OVERFLOW
001175  041176  2020            DAC     .+1             PLACE THE NORMALIZE INSTRUCTION IN SEQUENCE
001176  740040  2030            XX                      STEP COUNT TO THE SC
001177  201754  2040            LAC     $MQ             RELOAD THE OLD MQ
001200  652000  2050            LMQ                     AND SET IT
001201  621170  2060            RET     REGRES,X
```

M MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS

```
                    2070            .EJECT
                    2080    *
                    2090    *       VIOLATION WAS AN IOT INSTRUCTION -- SEPARATE THE MICROCODING FROM THE REST
                    2100    *
001202  200702      2110    .IOT,   LAC     INSTR           LOAD THE VIOLATING IOT INSTRUCTION
001203  501655      2120            AND     (SSPMSK)        RECOVER THE "SPECIAL" BITS
001204  541656      2130            SAD     (SSPECIAL)      AND CHECK THEM
001205  601246      2140            JMP     SWAP2           YES -- GET THE SPECIALS HANDLER
001206  200702      2150            LAC     INSTR           ELSE RELOAD THE VIOLATING IOT INSTRUCTION
001207  640504      2160            LRS     4               PUT THE MICROCODED BITS IN THE MQ
001210  040702      2170            DAC     INSTR           SAVE FILL + ALL BUT THE MICROCODED BITS
001211  641601      2180            EAECLA!LLS 1            RECOVER THE CLEAR AC BIT
001212  740200      2190            SZA
001213  140005      2200            DZM     S3AC            ZERO THE USER AC IF THAT BIT WAS SET
                    2210    *
                    2220    *       LOOK UP THE IOT AND BRANCH TO THE PROPER HANDLING ROUTINE
                    2230    *
   001214           2240    IOT0    ...
001214  140266      2250            DZM     SDKLOK          CLEAR THE DISK-USE FLAG
001215  761333      2260            LAW     IOTTT-1
001216  040010      2270            DAC     10              SET UP THE TABLE READ
001217  220010      2280    IOT1    LAC     10,X            READ THE NEXT TABLE ENTRY
001220  540702      2290            SAD     INSTR           CHECK AGAINST THE INSTRUCTION IN QUESTION
001221  601225      2300            JMP     IOT2            MATCHES--BRANCH TO THE HANDLING ROUTINE
001222  541346      2310            SAD     IOTT9           CHECK FOR THE END OF THE TABLE
001223  601366      2320            JMP     IOTSW           DONE, AND NO MATCH FOUND
001224  601217      2330            JMP     IOT1
                    2340
001225  220010      2350    IOT2    LAC     10,X
001226  040702      2360            DAC     TEMP2           SET THE TRANSFER
001227  641002      2370            LACQ
001230  040703      2380            DAC     TEMP3           SET THE MICROCODE
001231  742010      2390            RTL
001232  751100      2400            SPAICLA                 SKIP IF THERE IS NO IOPS EVENT TIME 1 EVENT
001233  440702      2410            INX     TEMP2           ELSE BUMP THE ENTRANCE
001234  601374      2420            JMP     IOT3
                    2430    *
                    2440    *       COMMON ERROR MESSAGES
                    2450    *
001235  340641      2460    ERR4    TAD     SBIT17          CHAINED XCT'S
001236  340641      2470    ERR3    TAD     SBIT17          ILLEGAL INSTRUCTION
001237  741000      2480            SKP
001240  440000      2490    ERR1    INX     0               ILLEGAL TRANSFER PC NEEDS TO BE FUDGED TO BE ONE TOO GREAT (LIKE ALL ELSE)
001241  341657      2500    ERR2    TAD     (2)             BAD ADDRESS
001242  041706      2510    ERR     DAC     SUTEM2          SET THE ERROR MESSAGE NUMBER
001243  101170      2520            JMS     REGRES          FIX UP THE USER REGISTERS BEFORE TRANSFERRING OUT OF THIS ROUTINE
001244  761004      2530    SWAP1   LAW     $SWERR
001245  600335      2540            JMP     $SWAP           GET THE SWAPPER -- ERROR MESSAGE ENTRY POINT
001246  101170      2550    SWAP2   JMS     REGRES          FIX UP THE USER REGISTERS BEFORE TRANSFERRRING OUT OF THIS ROUTINE
001247  761005      2560            LAW     $SWSPL
001250  600335      2570            JMP     $SWAP           GO READ IN THE MONITOR/MESSAGE PHANTOM PROGRAM
                    2580
```

```
                M                       MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS

                        2590
                        2600
        001251          2610    OAC     ENTER                   INCLUSIVE OR AC WITH USER AC FOR IOT'S
                                        ,PMC    SAVE,ON
    001251  740040                      XX
    001252  040002      2620            DAC     $3TM21
    001253  740001      2630            CMA
    001254  500005      2640            AND     $3AC
    001255  240002      2650            XOR     $3TM21
    001256  040005      2660            DAC     $3AC
    001257  621251      2670            RET     OAC,X
                        2680
                        2690
        001260          2700    TIM3    ENTER                   DETERMINE WHETHER OR NOT THERE IS AN IOPS EVENT TIME 3 EVENT REQUESTED
                                        ,PMC    SAVE,ON
    001260  740040                      XX
    001261  200703      2710            LAC     TEMP3
    001262  751100      2720            SPAICLA
    001263  621260      2730            RET     TIM3,X          YES -- GO DO IT
    001264  601614      2740            JMP     MEMD1           EXIT
                        2750
                        2760    *
                        2770    *       RDBLK LOOKS FIRST AT THE USER DUE TO RUN NEXT, IF HE IS I/O ROADBLOCKED, THE
                        2780    *       FOLLOWING USER IS EXAMINED, THIS PROCESS IS REPEATED UNTIL SOME USER IS FOUND
                        2790    *       WHO IS FREE TO RUN, NOTE THAT THE ROUTINE, ONCE ENTERED, WILL LOOP INDEFINITELY
                        2800    *       UNTIL A FREE USER IS FOUND.
                        2810    *
                        2820    *       WHEN A FREE USER IS FOUND, THE RETURN IS IMMEDIATE WITH HIS RE-ENTRANT
                        2830    *       TEMPORARY STORAGE SET UP.
                        2840    *
                        2850    *       RDBLK ENABLES THE INTERRUPT SYSTEM TO PERMIT TELETYPE I/O TO
                        2860    *       GO ON WHILE CHECKING FOR ROADBLOCKS, OTHERWISE A TELETYPE
                        2870    *       I/O ROADBLOCK COULD NEVER BE RELIEVED, THIS MEANS THAT BEFORE
                        2880    *       ENABLING INTERRUPTS, THE CALLER'S SAVED AC, 10, & 11 MUST BE
                        2890    *       COPIED, AND RECOPIED BEFORE EXIT,
                        2900    *
                        2910    *       THIS ROUTINE RUNS WITH THE CLOCK OFF TO PREVENT RE-ENTRANCE
                        2920    *       AT A TIME WHEN IT WOULD CRASH THE SYSTEM, ALSO NO ONE IS RUNNING
                        2930    *       AS LONG AS WE ARE HUNG IN THIS LOOP, SO NOTHING IS LOST,
                        2940    *
        001265          2950    RDBLK   ENTER
                                        ,PMC    SAVE,ON
    001265  740040                      XX
    001266  700004      2960            CLOF
    001267  200000      2970            LAC     0
    001270  040702      2980            DAC     TEMP2           SAVE THE RETURN
    001271  200005      2990            LAC     $3AC
    001272  040703      3000            DAC     TEMP3           THE SAVED AC
    001273  200026      3010            LAC     $,310
    001274  040010      3020            DAC     10              THE SAVED AUTO-INDEX REGISTER 10
    001275  200027      3030            LAC     $,311
    001276  040011      3040            DAC     11              THE SAVED AUTO-INDEX REGISTER 11
```

```
                M                       MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS

001277  700042  3050            ION
001300  201771  3060            LAC     $NUMBR          SEE WHO IS RUNNING; IF ANYONE
001301  540104  3070            SAD     $CTNAM
001302  601310  3080            JMP     RDBK2           #1 IS NOW RUNNING -- SO GIVE #2 A CHANCE
001303  540133  3090            SAD     $L1NAM
001304  601313  3100            JMP     RDBK3           #2 IS NOW RUNNING -- SO GIVE #3 A CHANCE
                3110
001305  200102  3120    RDBK1   LAC     $CTFLG          LOAD USER #0 I/O FLAGS
001306  101317  3130            JMS     RDBK5           SEE IF USER #0 IS I/O ROADBLOCKED
001307  760076  3140            LAW     $CTBIN-2        LOAD A POINTER TO USER #0 PARAMETERS
001310  200131  3150    RDBK2   LAC     $L1FLG
001311  101317  3160            JMS     RDBK5           SEE IF USER #1 IS I/O ROADBLOCKED
001312  760125  3170            LAW     $L1BIN-2
001313  200160  3180    RDBK3   LAC     $L2FLG
001314  101317  3190            JMS     RDBK5           SEE IF USER #2 IS I/O ROADBLOCKED
001315  760154  3200            LAW     $L2BIN-2
001316  601305  3210            JMP     RDBK1           LOOP
                3220
                3230    *       SEE IF THE SPECIFIED USER IS I/O ROADBLOCKED. IF SO, RETURN TO THE
                3240    *       ROADBLOCK ROUTINE FOR ANOTHER TRY, IF NOT, EXIT WITH HIS TEMPS SET UP.
                3250
    001317      3260    RDBK5   ENTER
                        .PMC    SAVE,ON
001317  740040                    XX
001320  742010  3270            RTL                     TELEPRINTER FLAG TO LINK; KEYBOARD FLAG TO AC(0)
001321  741500  3280            SZLISPA                 SKIP IF THERE IS NO I/O ROADBLOCK
001322  621317  3290            RET     RDBK5,X         ELSE TRY THE NEXT ONE
                3300    *
                3310    *       A NON-ROADBLOCKED USER HAS BEEN FOUND, TURN OFF THE INTERRUPT
                3320    *       SYSTEM, SET UP HIS RE-ENTRANT PARAMETERS AND EXIT.
                3330    *       NOTE THAT AUTO-INDEX REGISTERS 10 & 11 ARE ALREADY CORRECT
                3340    *       SO ONLY THE AC AND THE RESTART ADDRESS NEED TO BE RESTORED.
                3350    *
001323  700002  3360            IOF                     TURN OFF THE INTERRUPT SYSTEM
001324  200702  3370            LAC     TEMP2
001325  040000  3380            DAC     0               RESTORE THE SAVED RETURN
001326  200703  3390            LAC     TEMP3
001327  040005  3400            DAC     $3AC            AND THE SAVED AC
001330  421317  3410            XCT     RDBK5,X         LOAD THE POINTER TO HIS PARAMETERS
001331  100513  3420            JMS     $IO,IN          AND GO SET THEM UP
                3430    *
                3440    *       NOW WE ARE EVIDENTLY READY TO RUN SOMEONE AGAIN, SO TURN IT
                3450    *       BACK ON.
                3460    *
001332  700044  3470            CLON
001333  621265  3480            RET     RDBLK,X         DONE
                3490            .END
```

```
M                                   TELETYPE, DISK, AND DECTAPE ROUTINES (MAINLY)

                    120             .STITL  TELETYPE, DISK, AND DECTAPE ROUTINES (MAINLY)
                    130     *
                    140     *       IOT INSTRUCTION TRANSFER TABLE
                    150     *
    001334          160     IOTTT   ...                     IOT INSTRUCTION TRANSFER TABLE
001334  774020      170             774020                  700400
001335  601463      180             JMP     CTLP            TSF;TCF;TLS
001336  774014      190             774014                  700300
001337  601523      200             JMP     CKBD            KSF;KRB;;ORS
001340  774002      210             774002                  700040
001341  601576      220             JMP     .ON             ION;CLON
001342  774154      230             774154                  703300
001343  601553      240             JMP     BRK1            TTS;DBK;CAF
001344  774000      250             774000                  IOT NOP
001345  601614      260             JMP     MEMD1
001346  001000      270     IOTT9   1000                    FLAG THE END OF THE TABLE
                    280
                    290
                    300     *
                    310     *       SUBROUTINE TO CHECK FOR AN I/O ROADBLOCKED CONDITION.  THIS OCCURS IF AN IOT SKIP
                    320     *       INSTRUCTION FAILS TO SKIP AND THE FOLLOWING INSTRUCTION IS A <JMP .-1>.  THE ROUTINE
                    330     *       RETURNS +1 IF ALL IS NORMAL OR +2 IF AN I/O ROADBLOCK WAS DISCOVERED
                    340     *
    001347          350     IOBLK   ENTER
                                    .PMC    SAVE,ON
001347  740040                      XX
001350  751101      360             SPA!CLA!CMA             SKIP IF IOT FLAG IS NOT SET
001351  601360      370             JMP     IOBL8           ELSE EXIT, INCREMENTING THE USER'S PC
                    380     *
                    390     *       NOW CHECK FOR A TIGHT LOOP -- <IOT SKIP ON FLAG>, <JMP S-1>.
                    400     *
001352  340000      410             TAD     0               NOTE THIS ALSO COMPLEMENTS THE LINK
001353  500651      420             AND     SADRSS          ESTABLISH THE VALUE OF <.-1>
001354  240652      430             XOR     SJMP            FORM <JMP .-1> INSTRUCTION
001355  560000      440             SAD     0,X             SKIP IF NOT A TIGHT LOOP
001356  441347      450             INX     IOBLK           ELSE BUMP THE RETURN
001357  741002      460             SKP!CML                 CORRECT THE LINK
001360  440000      470     IOBL8   INX     0
001361  621347      480             RET     IOBLK,X
                    490
    001362          500     MFLG    ENTER
                                    .PMC    SAVE,ON
001362  740040                      XX
001363  241760      510             XOR     SIORS
001364  041760      520             DAC     SIORS
001365  621362      530             RET     MFLG,X
                    540     *
                    550     *
                    560     *       ROUTINE TO CALL MEMORY PROTECTION OVERLAY #2 IF IT HAS NOT ALREADY
                    570     *       HAD ITS CHANCE AT THE VIOLATION
                    580     *
    001366          590     IOTSW   ...
```

M TELETYPE, DISK, AND DECTAPE ROUTINES (MAINLY)

```
001366  200704  600          LAC     TEMP4           LOAD THE OVERLAY COUNT
001367  750200  610          SZAICLA                 SKIP UNLESS ALL OVERLAYS HAVE ALREADY HAD A CHANCE
001370  601236  620          JMP     ERR3            ELSE THIS WAS THE LAST CHANCE -- IT MUST HAVE BEEN AN ILLEGAL INSTRUCTION
001371  440704  630          INX     TEMP4           COUNT THIS OVERLAY
001372  761010  640          LAW     SSWMP2          LOAD A POINTER TO THE SWAPPER ENTRANCE TO GET THE NEXT OVERLAY
001373  600335  650          JMP     SSWAP           AND GET IT
```

M TELETYPE, DISK, AND DECTAPE ROUTINES (MAINLY)

```
                    660             .EJECT
          001374    670     IOT3    ...
001374    201771    680             LAC     $NUMBR          LOAD TTY NUMBER (= POINTER TO RESIDENT PARAMETERS)
001375    100513    690             JMS     $IO.IN          SET UP THE RESIDENT PARAMETERS
                    700     *
                    710     *       SET UP THE USER'S IORS WORD -- MOST ROUTINES IN THIS OVERLAY NEED IT
                    720     *
001376    201760    730             LAC     $IORS
001377    500634    740             AND     $BIT0
001400    700304    750             IORS-10                 INCLUSIVE OR THE STATUS -- SOME FLAGS KEPT IN HARDWARE
001401    501660    760             AND     (401400)        KEEP ONLY THE NO-TAPE FLAGS
001402    041760    770             DAC     $IORS
                    780     *
                    790     *       SET UP THE READER AND PUNCH FLAGS IF THIS READER HAS THE APPROPRIATE DEVICE
                    800     *
001403    200235    810     MTAPE   LAC     $RPTR
001404    540035    820             SAD     $RCORE
001405    601412    830             JMP     MPT1            THIS USER HAS THE READER, SO SET HIS FLAG
001406    201760    840             LAC     $IORS
001407    501661    850             AND     (776777)        ELSE REMOVE THE READER-OUT-OF-TAPE FLAG
001410    041760    860             DAC     $IORS
001411    601416    870             JMP     MPT2            AND CHECK ON THE PUNCH
001412    200234    880     MPT1    LAC     $RFLAG
001413    750200    890             SZA!CLA
001414    201662    900             LAC     (200000)
001415    101362    910             JMS     MFLG            SET THE READER FLAG IN THE IORS WORD
001416    200230    920     MPT2    LAC     $RPTP
001417    540035    930             SAD     $RCORE
001420    601425    940             JMP     MP3             THIS USER HAS THE PUNCH, SO SET HIS FLAG
001421    201760    950             LAC     $IORS
001422    501663    960             AND     (777377)        ELSE REMOVE THE PUNCH-OUT-OF-TAPE FLAG
001423    041760    970             DAC     $IORS
001424    601431    980             JMP     MKBD            AND CHECK ON THE TELETYPE KEYBOARD
001425    200227    990     MP3     LAC     $PFLAG
001426    750200    1000            SZA!CLA
001427    201650    1010            LAC     (100000)
001430    101362    1020            JMS     MFLG            SET THE PUNCH FLAG IN THE IORS WORD
                    1030    *
                    1040    *       THE KEYBOARD FLAG GETS SET IF EITHER:
                    1050    *            BOTH THE OUTPUT-IN-PROGRESS AND SOFTWARE KEYBOARD FLAGS ARE SET OR
                    1060    *            THE OUTPUT-IN-PROGRESS FLAG IS NOT SET AND THE ROTARY BUFFER IS NON-EMPTY.
                    1070    *
001431    200053    1080    MKBD    LAC     $3TEM2          LOAD THE SOFTWARE KEYBOARD FLAGS
001432    741100    1090            SPA                     SKIP IF OUTPUT IS NOT IN PROGRESS
001433    601440    1100            JMP     MK1             ELSE CHECK THE SOFTWARE KEYBOARD FLAG
001434    200052    1110            LAC     $3TEM1
001435    540051    1120            SAD     $3TEM0          SKIP IF THE ROTARY BUFFER IS NON-EMPTY
001436    601442    1130            JMP     MTLP            EMPTY BUFFER -- EXIT WITHOUT SETTING THE FLAG
001437    750001    1140            CLC                     NON-EMPTY BUFFER -- LOAD KEYBOARD FLAG (WITH OTHER GARBAGE)
001440    501664    1150    MK1     AND     (040000)        KEEP JUST THE KEYBOARD FLAG
001441    101362    1160            JMS     MFLG            SET THE KEYBOARD FLAG
                    1170    *
```

M TELETYPE, DISK, AND DECTAPE ROUTINES (MAINLY)

```
                    1180    *       THE TELEPRINTER FLAG GETS SET IF EITHER:
                    1190    *            THE OUTPUT-IN-PROGRESS FLAG IS SET AND THE ROTARY BUFFER IS NON-FULL OR
                    1200    *            THE OUTPUT-IN-PROGRESS FLAG IS NOT SET
                    1210    *
001442  200053      1220    MTLP    LAC     S3TEM2          LOAD THE TELETYPE SOFTWARE FLAGS
001443  740100      1230            SMA                     SKIP IF OUTPUT IS IN PROGRESS
001444  601454      1240            JMP     MT1             ELSE OUTPUT IS O.K.
001445  201641      1250            AND     (020000)        RECOVER THE TELEPRINTER FLAG
001446  741200      1260            SNA                     SKIP IF IT IS SET
001447  601462      1270            JMP     MT2             ELSE DON'T SET IT IN THE IORS WORD
001450  200051      1280            LAC     S3TEM0          LOAD THE INPUT POINTER
001451  100623      1290            JMS     SNXPTR          AND FIND OUT THE NEXT LOCATION
001452  540052      1300            SAD     S3TEM1          SKIP UNLESS THE BUFFER WOULD OVERFLOW
001453  601462      1310            JMP     MT2             IN WHICH CASE DO NOT SET THE FLAG IN THE IORS WORD
001454  201641      1320    MT1     LAC     (020000)        ALL SET -- LOAD THE TELEPRINTER FLAG
001455  101362      1330            JMS     MFLG            AND SET IT IN THE IORS WORD
                    1340    *
                    1350    *       CHECK THE DISK CONDITION
                    1360    *
001456  201761      1370    MDISK   LAC     SDFLAG          LOAD THE USER'S SOFTWARE DISK FLAG
001457  740200      1380            SZA
001460  201665      1390            LAC     (000020)
001461  101362      1400            JMS     MFLG            SET THE DISK FLAG IN THE IORS WORD
                    1410
                    1420
001462  620702      1430    MT2     JMP     TEMP2,X         GO TO THE CORRECT SERVICE ROUTINE
```

M TELETYPE, DISK, AND DECTAPE ROUTINES (MAINLY)

```
                    1440            ,EJECT
001463  601474      1450    CTLP    JMP     CTLP1           NO IOPS EVENT TIME 1
001464  201760      1460    .TSF    LAC     $IORS           LOAD THE IORS WORD
001465  640704      1470            ALS     4               GET THE FLAG
001466  101347      1480            JMS     IOBLK           CHECK THE FLAG AND FOR I/O ROADBLOCK
001467  601474      1490            JMP     CTLP1           NO ROADBLOCK -- CARRY ON NORMALLY
001470  200645      1500            LAC     $CB1
001471  500053      1510            AND     $3TEM2          CLEAR THE I/O ROADBLOCK FLAG
001472  241662      1520            XOR     (200000)        SET THE I/O ROADBLOCK FLAG
001473  040053      1530            DAC     $3TEM2          REPLACE THE SOFTWARE FLAGS
001474  740400      1540    CTLP1   SNL                     SKIP IF THERE IS AN EVENT TIME TWO ACTIVITY
001475  601501      1550            JMP     .T2             ELSE TRY FOR EVENT TIME THREE
001476  200053      1560    .TCF    LAC     $3TEM2          LOAD THE TELETYPE SOFTWARE FLAGS
001477  501666      1570            AND     (757777)        CLEAR THE TELEPRINTER FLAG
001500  040053      1580            DAC     $3TEM2          AND RESTORE THE UPDATED FLAGS
001501  101260      1590    .T2     JMS     TIM3            RETURN IF THERE IS AN IOPS TIME 3 EVENT
001502  200053      1600    .TLS    LAC     $3TEM2          LOAD THE TELETYPE SOFTWARE FLAGS
001503  741100      1610            SPA                     SKIP IF OUTPUT IS NOT INPROGRESS
001504  601511      1620            JMP     .TLS1           ELSE JUST PACK THE OUTPUT
001505  100540      1630            JMS     $NEWBR          CLEAR THE INPUT BUFFER
001506  200005      1640            LAC     $3AC            ELSE LOAD WHAT THE USER WANTS PRINTED
001507  400056      1650            XCT     $3TEM5          AND PRINT IT
001510  601516      1660            JMP     .TLS2
001511  200005      1670    .TLS1   LAC     $3AC
001512  500643      1680            AND     $BL8            THE BUFFER EXPECTS EIGHT-BIT ASCII
001513  040002      1690            DAC     $3TM21
001514  100546      1700            JMS     $PUTIN          PLACE THE CHARACTER IN THE BUFFER
001515  740000      1710            NOP                     DISCARD ANY OVERFLOW
001516  200053      1720    .TLS2   LAC     $3TEM2          LOAD THE TELETYPE SOFTWARE FLAGS
001517  501667      1730            AND     (355777)        CLEAR THE O-I-P, TELEPRINTER, AND BUFFER TYPE FLAGS
001520  241670      1740            XOR     (422000)        SET THE OUTPUT-IN-PROGRESS, TELEPRINTER, AND OUTPUT-BUFFER FLAGS
001521  040053      1750            DAC     $3TEM2          RESTORE THE UPDATED FLAGS
001522  601614      1760            JMP     MEMD1
                    1770
                    1780
                    1790
001523  601534      1800    CKBD    JMP     CKBD1           NO IOPS EVENT TIME 1
001524  201760      1810    .KSF    LAC     $IORS           LOAD THE USER'S STATUS WORD
001525  640703      1820            ALS     3               GET THE KEYBOARD FLAG
001526  101347      1830            JMS     IOBLK           CHECK THE FLAG AND FOR I/O ROADBLOCK
001527  601534      1840            JMP     CKBD1           NO ROADBLOCK -- CARRY ON NORMALLY
001530  201633      1850            LAC     (677777)
001531  500053      1860            AND     $3TEM2          CLEAR THE I/O ROAD BLOCK FLAG
001532  241650      1870            XOR     (100000)        SET THE I/O ROADBLOCK FLAG
001533  040053      1880            DAC     $3TEM2          REPLACE THE SOFTWARE FLAGS
001534  740400      1890    CKBD1   SNL
001535  601547      1900            JMP     CKBD2           NO IOPS EVENT TIME 2
                    1910    *
                    1920    *       TRY TO READ THE CHARACTER FROM THE ROTARY BUFFER IF THERE IS NO OUTPUT
                    1930    *       IN PROGRESS, READ THE CHARACTER FROM THE SOFTWARE KEYBOARD BUFFER IF THE
                    1940    *       ROTARY BUFFER TURNS OUT TO BE EMPTY, OR IF OUTPUT IS IN PROGRESS.
                    1950    *
```

M TELETYPE, DISK, AND DECTAPE ROUTINES (MAINLY)

```
                    1960    *       CLEAR THE SOFTWARE KEYBOARD FLAG IN ANY CASE.
                    1970    *
001536  200053      1980    .KRB    LAC     $3TEM2          LOAD THE TELETYPE SOFTWARE FLAGS
001537  501671      1990            AND     (737777)        CLEAR THE SOFTWARE KEYBOARD FLAG ON ANY KRB
001540  040053      2000            DAC     $3TEM2          SAVE THE UPDATED SOFTWARE FLAGS
001541  500640      2010            AND     $BIT7           RECOVER THE BUFFER TYPE
001542  741200      2020            SNA                     SKIP IF IT IS AN OUTPUT BUFFER
001543  100602      2030            JMS     $FGET           ELSE GET THE OLDEST CHARACTER IN THE BUFFER
001544  200053      2040            LAC     $3TEM2          LOAD THE SOFTWARE KEYBOARD BUFFER IF NO INPUT IN BUFFER
001545  500643      2050            AND     $BL8            RETAIN ONLY 8-BIT INPUT
001546  101251      2060    KRB2    JMS     OAC             AND PUT IT IN THE USERS AC
001547  101260      2070    CK8D2   JMS     TIM3            RETURN IF THERE IS AN IOPS TIME 3 EVENT
001550  201760      2080    .IORS,  LAC     $IORS           LOAD THE USER IORS WORD
001551  101251      2090            JMS     OAC
001552  601614      2100            JMP     MEMD1
                    2110
                    2120
                    2130
001553  741000      2140    BRK1    SKP
001554  440000      2150            INX     0               'TTS' -- SKIP IF NOT TYPE 28
001555  740400      2160    .CAF    SNL
001556  600270      2170            RET     $PIDON          NO IOPS EVENT TIME 2
001557  200230      2180            LAC     $RPTP
001560  541771      2190            SAD     $NUMBR          SKIP IF THE PUNCH IS NOT ASSIGNED TO THIS USER
001561  140227      2200            DZM     $PFLAG          PUNCH FLAG
001562  200235      2210            LAC     $RPTR
001563  541771      2220            SAD     $NUMBR          SKIP IF THE READER IS NOT ASSIGNED TO THIS USER
001564  140234      2230            DZM     $RFLAG          READER FLAG
001565  200053      2240            LAC     $3TEM2          LOAD THE SOFTWARE FLAGS
001566  501672      2250            AND     (420000)        KILL ALL EXCEPT THE OUTPUT-IN-PROGRESS FLAG AND EXEC'S TLP FLAG
001567  040053      2260            DAC     $3TEM2          RESTORE THE UPDATED FLAGS
001570  141760      2270            DZM     $IORS           CAF ALWAYS CLEARS ALL THINGS ON THE IORS WORD
001571  141762      2280            DZM     $DAP0
001572  141763      2290            DZM     $DAP1
001573  141764      2300            DZM     $DFN
001574  141761      2310            DZM     $DFLAG          DISK FLAG
001575  601614      2320    .DBK    JMP     MEMD1           DBK REQUIRES NO ACTION, EVEN IF PRESENT
                    2330
                    2340
                    2350
                    2360
001576  740400      2370    .ON     SNL                     SKIP ONLY IF THERE IS AN EVENT TIME 2 EVENT (ION)
001577  601236      2380            JMP     ERR3
001600  200703      2390    .ION    LAC     TEMP3
001601  751100      2400            SPA!CLA
001602  601236      2410            JMP     ERR3            THERE WAS AN ILLEGAL EVENT TIME 3 EVENT (CLON)
001603  201760      2420            LAC     $IORS           LOAD THE USER'S IORS WORD
001604  500644      2430            AND     $CB0
001605  240634      2440            XOR     $BIT0
001606  041760      2450            DAC     $IORS           RESTORE THE IORS WORD WITH THE PI ON
001607  501673      2460            AND     (375220)
001610  740200      2470            SZA                     SKIP IF THERE WERE NO FLAGS ON TO CAUSE INTERRUPTS
```

M TELETYPE, DISK, AND DECTAPE ROUTINES (MAINLY)

```
001611  601007  2480            JMP     PINT            GENERATE A USER PROGRAM INTERRUPT
001612  101170  2490            JMS     REGRES          RESTORE THE REGISTERS
001613  600270  2500            RET     $PIDON          EXIT
                2510
    001614      2520    MEMD1   ...                     EXIT MEMORY PROTECT SERVICE
001614  740000  2530            NOP                     ALLOW FOR IOT NOP
001615  101170  2540            JMS     REGRES
001616  100525  2550            JMS     $IO,OT
001617  200053  2560            LAC     $3TEM2
001620  501674  2570            AND     ($IOBLK)        GET JUST THE USER'S I/O ROADBLOCK FLAGS
001621  741200  2580            SNA                     SKIP IF A ROADBLOCK CONDITION EXISTS
001622  600270  2590            RET     $PIDON          ELSE RETURN
001623  101265  2600            JMS     RDBLK           SEE WHO IS NEXT USER NOT I/O ROADBLOCKED
001624  200055  2610            LAC     $3TEM4          LOAD HIS USER NUMBER
001625  540035  2620            SAD     $RCORE          SEE IF IT IS THE CURRENT USER
001626  600270  2630            RET     $PIDON          IF SO, EXIT (CONTINUE)
001627  761003  2640            LAW     $SWCLK
001630  600335  2650            JMP     $SWAP
                2660
    001631      2670    .OPR,   ...
001631  761011  2680            LAW     $SWOPR
001632  600335  2690            JMP     $SWAP           SWITCH OVERLAYS AND RESTART AT OPERATE INSTRUCTION HANDLING
001633  677777  2700            .END    OVSTRT
001634  100001
001635  776000
001636  740000
001637  640000
001640  700000
001641  020000
001642  017770
001643  000010
001644  777770
001645  777760
001646  776040
001647  001713
001650  100000
001651  600000
001652  000077
001653  640402
001654  640477
001655  777400
001656  705000
001657  000002
001660  401400
001661  776777
001662  200000
001663  777377
001664  040000
001665  000020
001666  757777
001667  355777
001670  422000
```

M TELETYPE, DISK, AND DECTAPE ROUTINES (MAINLY)

```
001671  737777
001672  420000
001673  375220
001674  300000

TRANSFER ADDRESS  601000
```

M CROSS REFERENCE TABLE

1713	.0	4510	4520	480	520	3340							
26	.310	3400	3010										
27	.311	3410	3030										
4464	.DT	570											
1590	.IORS;	2080											
6460	.TP	550											
2023	10SAVE	1870	1880										
2024	11SAVE	1880	1920										
5	3AC	3370	1560	1600	2200	2640	2660	2990	3400	1640	1670		
305	3REST	3870	3880										
51	3TEM0	3530	3540	1120	1280								
52	3TEM1	3540	3550	1110	1300								
53	3TEM2	3550	3560	1080	1220	1510	1530	1560	1580	1600	1720	1750	1860	1880
			1980	2000	2040	2240	2260	2560					
54	3TEM3	3560	3570										
55	3TEM4	3570	3580	2610									
56	3TEM5	3580	3590	1650									
57	3TEM6	3590	3600										
50	3TM20	3520	3530										
2	3TM21	3350	2620	2650	1690								
3	3TM22	3360											
14000	7K	1030											
16000	8K	1020	910	1010	2640	2650							
1753	AC	4520	4530										
1756	ACS	4550	4560										
2015	ACSAVE	1810	1820										
2022	ACSW	1860	1870										
651	ADRSS	4100	4110	670	1070	1170	1800	420					
300	AT	2950											
300	ATSGN	2900											
422030	BAS	420											
2151	BCNTRL	2330	2340										
634	BIT0	3970	3980	490	1480	1640	740	2440					
641	BIT17	4020	4030	2460	2470								
635	BIT36	3980	3990										
636	BIT5	3990	4000										
637	BIT6	4000	4010										
640	BIT7	4010	4020	2010									
642	BL7	4030	4040										
643	BL8	4040	4050	1680	2050								
2000	BOUNDA	970	960	980	990	1000	1630	5040	5080	680	1270		
377	BRK	5550											
2170	BUFFER	2490	2550										
1000	BUFLEN	2500	2550										
644	CB0	4050	4060	1620	2430								
645	CB1	4060	4070	1500									
646	CB5	4070	4080										
647	CB7	4080	4090										
650	CBL8	4090	4100										
6	CHRMAX	3180	3200										
2	CHRPAK	3130	3200										
50	CLKMAX	2840	3180										

M CROSS REFERENCE TABLE

60	CLK$PD	3160	3170			
1757	CLOCK	4560	4570			
45	CMP1	3490	3500			
46	CMP2	3500	3510			
6	CNTRL	3380	3390			
2053	COMFLG	2200	2210			
2150	COM$TO	2270	2280			
16000	CORMAX	910	980			
47	CSPL	3510	3520			
44	CSWP	3480	3490			
60	CTBFR	3600	3630	3640		
100	CTBIN	3640	3650	3670	4250	3140
2000	CTEMP0	1630				
2001	CTEMP1	1640				
2002	CTEMP2	1650				
2003	CTEMP3	1660				
2004	CTEMP4	1670				
2005	CTEMP5	1680				
2006	CTEMP6	1690				
2007	CTEMP7	1700				
2010	CTEMP8	1710				
2011	CTEMP9	1720				
102	CTFLG	3650	3660	3120		
104	CTNAM	3660	3070			
2043	D PC	2120	2130			
2154	D BCA	2370	2380			
2153	D BDA	2360	2370			
2163	D FDA	2440	2450			
2042	D LOC	2110	2120			
2022	D ACSW	1860				
2156	D BALT	2390	2400			
2155	D BLEN	2380	2390			
2161	D BMAX	2420	2430			
2157	D BMIN	2400	2410			
2162	D BPTR	2430	2440			
2167	D FMAX	2480	2490			
2165	D FMIN	2460	2470			
2046	D MASK	2150	2160			
2164	D MFDA	2450	2460			
2036	DADRSW	2070	2080			
1762	DAP0	4590	4600	2280		
1763	DAP1	4600	4610	2290		
653	DBK	4120	4130			
24	DBKNUM	2220	2270			
2054	DBKTAB	2210	2270			
2035	DBSTOR	2050	2060			
422027	DDT	410				
12000	DDTST	5000				
2037	DDUMSW	2080	2090			
1761	DFLAG	4580	4590	1370	2310	
1764	DFN	4610	4620	2300		
2151	DFTYPE	2340	2350			

M CROSS REFERENCE TABLE

2045	DHICOR	2140	2150		
2050	DINDIR	2170	2180		
100	DK0	4270			
127	DK1	4310			
156	DK2	4350			
37	DKCA	2750			
675	DKDON	4170	4180		
16080	DKLEN	2650	2660		
34	DKLENB	2660			
266	DKLOK	3830	3840	2250	
672	DKOVR	4160	4170		
2	DKRD	2760			
36	DKWC	2740			
4	DKWRT	2770			
2041	DLIMIT	2100	2110		
2044	DLOCOR	2130	2140		
2160	DMBMIN	2410	2420		
2166	DMFMIN	2470	2480		
654	DO	4130	4140		
662	DO2	4140	4150		
663	DO3	4150	4160		
2152	DOFTYP	2350	2360		
2032	DPACSW	1980			
2040	DPATSW	2090	2100		
2051	DPCMSK	2180	2190		
2052	DREGBR	2190	2200		
2035	DREGSW	2060	2070		
2047	DRELOC	2160	2170		
1765	DSTAT	4620	4630		
446400	DT.	560			
2000	DTEMP0	1630			
2001	DTEMP1	1640			
2002	DTEMP2	1650			
2003	DTEMP3	1660			
2004	DTEMP4	1670			
2005	DTEMP5	1680			
2006	DTEMP6	1690			
2007	DTEMP7	1700			
2010	DTEMP8	1710			
2011	DTEMP9	1720			
275	EQUAL	2910			
602	FGET	3950	3960	2030	
1701	FRCA	4410	4420		
1700	FRDA	4400	4410		
1702	FRLEN	4420	4430		
1703	FRSTA	4430	4440		
2	FUDGE	3190	3200		
276	GREAT	2930			
1700	IMPLEN	990			
3170	IMPSTR	2550			
422020	INT	320			
513	IO.IN	3910	3920	3420	690

M CROSS REFERENCE TABLE

```
   525  IO.OT      3920  3930   400  2550
300000  IOBLK      2830  2570
  1760  IORS       4570  4580   410   450   510   520   730   770   840   860   950   970  1460
                   1810  2080  2270  2420  2450
  1002  IOTO       4900  4910
   652  JMP        4110  4120   430
   100  JTLEN       960
  1700  JTSTRT      950   940   960  1000  4400
    16  KBLEN      3610  3630  3640  3680  3690  3730  3740
    10  KBNUM      3620  3670  3720
    76  LOLOK      3630
   107  L1BFR      3670  3680  3690
   127  L1BIN      3690  3700  3720  4290  3170
   131  L1FLG      3700  3710  3150
   125  L1LOK      3680
   133  L1NAM      3710  3090
   136  L2BFR      3720  3730  3740
   156  L2BIN      3740  3750  4330  3200
   160  L2FLG      3750  3760  3180
   154  L2LOK      3730
   162  L2NAM      3760  3770
422026  LDR         390
  2000  LDRST      5040
   274  LESS       2920
  1576  M  ,ON     2370   220
  1501  M  ,T2     1590  1550
  1242  M  ERR     2510
  1440  M  MK1     1150  1100
  1425  M  MP3      990   940
  1454  M  MT1     1320  1240
  1462  M  MT2     1430  1270  1310
  1251  M  OAC     2610  2670  2060  2090
  1555  M .CAF     2160
  1575  M .DBK     2320
  1600  M .ION     2390
  1536  M .KRB     1980
  1524  M .KSF     1810
  1476  M .TCF     1560
  1502  M .TLS     1600
  1464  M .TSF     1460
  2022  M ACSW     1860
  1553  M BRK1     2140   240
  1523  M CKBD     1800   200
  1463  M CTLP     1450   180
  1240  M ERR1     2490   700
  1241  M ERR2     2500  1290
  1236  M ERR3     2470   820   620  2380  2410
  1235  M ERR4     2460  1860
  1214  M IOTO     2240   310
  1217  M IOT1     2280  2330
  1225  M IOT2     2350  2300
  1374  M IOT3      670  2420
```

M CROSS REFERENCE TABLE

1546	M KRB2	2060											
1362	M MFLG	500	530	910	1020	1160	1330	1400					
1431	M MKBD	1080	980										
1115	M MP12	1470											
1137	M MP15	1650	1820										
1027	M MPST	590	290										
1412	M MPT1	880	830										
1416	M MPT2	920	870										
1442	M MTLP	1220	1130										
1115	M O,K,	1400	880	1230	1300								
1007	M PINT	400	300	2480									
1260	M TIM3	2700	2730	1590	2070								
1160	M XCT1	1870											
1202	M,IOT,	2110	920										
1146	M,JMP,	1760	1530										
1142	M,JMS,	1710	1510										
1631	M,OPR,	2670	330	900									
1511	M,TLS1	1670	1620										
1516	M,TLS2	1720	1660										
1155	M,XCT,	1840	1490										
1534	MCKBD1	1890	1800	1840									
1547	MCKBD2	2070	1900										
1474	MCTLP1	1540	1450	1490									
1112	MFAKIT	1340	1200	1260									
10	MINBUF	3200	3610										
702	MINSTR	230	760	770	780	970	1010	1040	1060	1090	1160	1350	1360	1570
			1730	1740	1790	2110	2150	2170	2290				
1360	MIOBLB	470	370										
1347	MIOBLK	350	450	480	1480	1830							
1366	MIOTSW	590	2320										
1346	MIOTT9	270	2310										
1334	MIOTTT	160	2260										
1456	MMDISK	1370											
1614	MMEMD1	2520	2740	260	1760	2100	2320						
1041	MMP111	770	550	1880									
1073	MMPBA1	1160	1000										
1403	MMTAPE	810											
703	MOPCOD	250	830	1080	1470								
1053	MOPRST	900											
422023	MP1	350											
422024	MP2	360											
2032	MPACSW	1980											
1005	MPINST	340	790										
1004	MPOPR	4920											
1000	MPST	4880	4890										
1000	MPSTRT	290											
1754	MQ	4530	4540	1920	2040								
2016	MQSAVE	1820	1830										
1305	MRDBK1	3120	3210										
1310	MRDBK2	3150	3080										
1313	MRDBK3	3180	3100										
1317	MRDBK5	3260	3130	3160	3190	3290	3410						

M

CROSS REFERENCE TABLE

1265	MRDBLK	2950	320	3480	2600				
1244	MSWAP1	2530							
1246	MSWAP2	2550	2140						
2000	MTEMP0	1630							
2001	MTEMP1	1640							
702	MTEMP2	240	2360	2410	2980	3370	1430		
703	MTEMP3	260	2380	2710	3000	3390	2390		
704	MTEMP4	270	530	610	600	630			
705	MTEMP5	280	540	620	1840	1870			
2006	MTEMP6	1690							
2007	MTEMP7	1700							
2010	MTEMP8	1710							
2011	MTEMP9	1720							
422025	MTR	370							
2000	MTRST	5080							
1772	NAME	4670	4680						
540	NEWBR	3930	3940	1630					
1771	NUMBR	4660	4670	3060	680	2190	2220		
623	NXPTR	3960	3970	1290					
702	OC0	4180	4190	230	240				
703	OC1	4190	4200	250	260				
704	OC2	4200	4210	270					
705	OC3	4210	280						
574646	OFF	2730							
575600	ON	2720							
1773	OVER	4680	4690						
700	OVLEN	940							
1000	OVSTRT	930	920	940	4750	4880	4960	220	2700
2033	P10SAV	1990	2000						
2034	P11SAV	2000	2050						
2025	PACSAV	1930	1940						
2032	PACSW	1980	1990						
241	PBFLAG	3810	3820						
2017	PCSAVE	1830	1840						
227	PFLAG	3770	3780	990	2200				
77	PH0	4260	4270						
126	PH1	4300	4310						
155	PH2	4340	4350						
1	PHANTO	2780							
2150	PHFLAG	2280	2330						
1700	PHLEN	2640							
2025	PHSTOR	1920	1930						
274	PIDN2	3850	3860	1670					
270	PIDON	3840	3850	430	2170	2500	2590	2630	
1001	PINT	4890	4900						
303	PIOUT	3860	3870						
602026	PLDR	400							
2026	PMQSAV	1940	1950						
602025	PMTR	380							
2027	PPCSAV	1950	1960						
606064	PPT	520							
2031	PSCSAV	1970	1980						

M CROSS REFERENCE TABLE

2030	PSTSAV	1960	1970					
606460	PTP	510						
606462	PTR	500						
12100	PURLEN	1010						
1775	PURNM	4700	4710					
3700	PURSTR	2560	990	1010	2560			
546	PUTIN	3940	3950	1700				
34	RACS	3440						
6	RCNT	3390						
35	RCORE	3450	820	930	2620			
1003	RDBLK	4910	4920					
32	RDT0	3420						
33	RDT1	3430						
1170	REGRES	1970	1660	2060	2520	2550	2490	2540
1162	REGSAV	1900	390	600	1950			
422021	RES	330						
40	RESCAT	3470	3480					
1000	RESLEN	920						
234	RFLAG	3790	3800	880	2230			
230	RPTP	3780	3790	920	2180			
235	RPTR	3800	3810	810	2210			
242	RSCO	3820	3830					
1776	RSTRT	4710						
1755	SC	4540	4550	1940	1980			
640000	SCRSTR	2670						
2021	SCSAVE	1850	1860					
243	SHARP	2890						
377	SPCOD	5410						
422122	SPL	430						
1000	SPLST	4960						
777400	SPMSK	5390	2120					
2020	STSAVE	1840	1850					
335	SWAP	3880	3890	2540	2570	650	2650	2690
336	SWAP1	3890	3900					
340	SWAP3	3900	3910					
1000	SWCAT	4750	4760					
1003	SWCLK	4780	4790	2640				
1004	SWERR	4790	4800	2530				
1007	SWMP1	4820	4830					
1010	SWMP2	4830	4840	640				
1002	SWMTR	4770	4780					
1011	SWOPR	4840	2680					
422022	SWP	340						
1001	SWPPR	4760	4770					
40	SWPS	3460	3470					
1005	SWSPL	4800	4810	2560				
1006	SXSPL	4810	4820					
1300	SYSBAS	2800	2810					
41300	SYSDA	2810						
1777	SYSMAX	2820						
100	TABLEN	2630	2640					
2000	TEMP0	1630	1640					

M CROSS REFERENCE TABLE

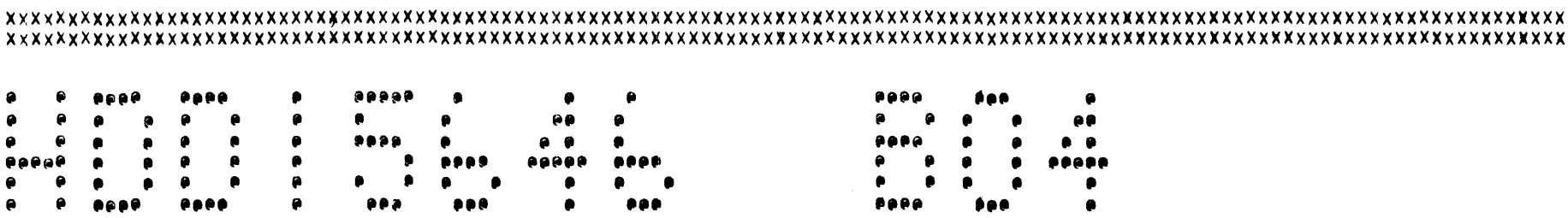
2001	TEMP1	1640	1650	
2012	TEMP10	1730	1740	
2013	TEMP11	1740	1750	
2014	TEMP12	1750	1800	
2002	TEMP2	1650	1660	
2003	TEMP3	1660	1670	
2004	TEMP4	1670	1680	
2005	TEMP5	1680	1690	
2006	TEMP6	1690	1700	
2007	TEMP7	1700	1710	
2010	TEMP8	1710	1720	
2011	TEMP9	1720	1730	
646000	TP.	540		
376	TRCOFF	5540		
375	TRCON	5530		
2000	TTEMP0	1630		
2001	TTEMP1	1640		
2002	TTEMP2	1650		
2003	TTEMP3	1660		
2004	TTEMP4	1670		
2005	TTEMP5	1680		
2006	TTEMP6	1690		
2007	TTEMP7	1700		
2010	TTEMP8	1710		
2011	TTEMP9	1720		
6	TTYCLK	3170	3180	
3	TTYNUM	3140		
10	TTYSPD	3150	3170	
1774	TYPE	4690	4700	
1766	UCORE	4630	4640	
1767	UDISK	4640	4650	
336	UPARR	2940		
76	US0	4250	4260	4280
125	US1	4290	4300	4320
154	US2	4330	4340	4360
0	USER	2790		
3	USERS	2850	3200	
14000	USLEN	980	2640	
2015	USTORE	1800	1810	
75	UT0	4280		
124	UT1	4320		
153	UT2	4360		
1704	UTEM0	4440	4450	
1705	UTEM1	4450	4460	
1706	UTEM2	4460	4470	2510
1707	UTEM3	4470	4480	
1710	UTEM4	4480	4490	
1711	UTEM5	4490	4500	
1712	UTEM6	4500	4510	
1770	VALID	4650	4660	

M UNDEFINED SYMBOLS

#1	5630											
#2	5640											
#3	5650											
#4	5660											
#5	5680											
DEFINS	100	100	120	120	580	580	2680	2680	3210	3210	5330	5330
M #1	190											
PURCOD	5140	5270										

M MACRO CROSS REFERENCE TABLE

ENTER	5280	1900	1970	2610	2700	2950	3260	350	500
MPOFF	5430								
SWAP	5610								
TABLE	170								



05/31/72 01:42:54

```
**********************************
**********************************
**                              **
**  PDP-9 MINI TIME-SHARING SYSTEM  **
**   MEMORY PROTECTION OVERLAY #2   **
**             DTSS:MP2             **
**              DK0:B04             **
**              TPN:B04             **
**                              **
**********************************
**********************************
```

```
100             .NAME   MP2--B04
110             .INSRT  MP0
100             .TITLE  ROUTINES TO SERVICE MEMORY PROTECT VIOLATIONS
110             .HEAD   M
120             .INSRT  DEFINS
100             .IFUND  DEFINS
```

```
5720            .LIST   ON
5730            .END
130             .HEAD   M
```

```
                M                       MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS

                    140                 .STITL  MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS
                    150
                    160
                    170     TABLE       .DEFIN                      IOT TABLE ENTRY
                    180                 .PMC    SAVE,OFF
                    190                 #1/16,+740000
                    200                 .PMC    RESTORE
                    210                 .ENDM
        001000      220                 .LOC    OVSTRT
        000702      230     INSTR       .EQU    $OC0
        000702      240     TEMP2       .EQU    $OC0
        000703      250     OPCOD       .EQU    $OC1
        000703      260     TEMP3       .EQU    $OC1
        000704      270     TEMP4       .EQU    $OC2
        000705      280     TEMP5       .EQU    $OC3
001000  601027      290     MPSTRT      JMP     MPST                OVERLAY ENTRANCE VECTOR
001001  601006      300                 JMP     PINT-1              ENTRANCE FOR PROGRAM INTERRUPT SIMULATION
001002  601214      310                 JMP     IOT0
001003  001265      320                 RDBLK                       POINTER TO THE ROADBLOCK ROUTINE
001004  601365      330                 JMP     .OPR,               POINTER TO OPERATE HANDLING
001005  000000      340     PINST       .DSA                        PERMANENT INSTRUCTION SAVE IN CASE IT IS A GRAPHICS II INSTRUCTION
                    350     *
                    360     *       AN IOT INTERRUPT HAS OCCURRED -- GENERATE A USER PI INTERRUPT IF THE PI SYSTEM IS ON
                    370     *       TREAT AS IF LOCATION 1 CONTAINED AN XCT OF THE USER'S LOCATION 1 -- ONE XCT IS STILL LEGAL
                    380     *
001006  101162      390                 JMS     REGSAVE
001007  100525      400     PINT        JMS     $IO,OT              SAVE USER TEMPS
001010  201760      410                 LAC     $IORS               LOAD THE USER'S IORS WORD
001011  740110      420                 SMA!RAL
001012  600270      430                 RET     $PIDON              EXIT IF PI SYSTEM IS NOT ENABLED
001013  744020      440                 CLL!RAR
001014  041760      450                 DAC     $IORS               ELSE TURN OFF THE PI SYSTEM
001015  200000      460                 LAC     0
001016  501621      470                 AND     (677777)            REMOVE THE MEMORY PROTECT BIT
001017  041713      480                 DAC     $,0                 SET THE USER PC IN HIS LOCATION 0
001020  500634      490                 AND     $BIT0               SAVE THE LINK STATUS
001021  341622      500                 TAD     (100001)
001022  040000      510                 DAC     0                   FAKE THE NEXT INSTRUCTION CAME FROM LOCATION 1 WITH MEMORY PROTECT ON
001023  201714      520                 LAC     $,0+1               LOAD THE USER'S LOCATION 1 INSTRUCTION
001024  140704      530                 DZM     TEMP4               INITIALIZE THE COUNT OF MEMORY OVERLAY EXCHANGES
001025  140705      540                 DZM     TEMP5               INITIALIZE THE XCT COUNT
001026  601042      550                 JMP     MP111+1             AND CHECK THE INSTRUCTION
                    560
                    570
                    580
        001027      590     MPST        ...,
001027  101162      600                 JMS     REGSAVE             SAVE THE MQ AND SC
001030  140704      610                 DZM     TEMP4               INITIALIZE THE COUNT OF MEMORY OVERLAY EXCHANGES
001031  140705      620                 DZM     TEMP5               INITIALIZE THE XCT COUNT (CHECK FOR XCT LOOPS)
                    630     *
                    640     *       CHECK TO SEE IF THE VIOLATION WAS CAUSED BY AN ATTEMPT TO TRANSFER TO PROTECTED MEMORY
                    650     *
```

```
                M                       MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS

001032  200000  660             LAC     0               LOAD THE USER PC
001033  500651  670             AND     $ADRSS          GET RID OF NON-ADDRESS BITS
001034  341623  680             TAD     (-BOUNDARY)
001035  755101  690             SPA!CLA!CMA!CLL         SKIP UNLESS A TRANSFER TO PROTECTED MEMORY WAS REQUESTED
001036  601240  700             JMP     ERR1            YES -- VIOLATION WAS A TRANSFER TO PROTECTED MEMORY -- ILLEGAL
                710     *
                720     *       THE VIOLATION WAS NOT CAUSED BY AN ATTEMPT TO TRANSFER TO PROTECTED MEMORY
                730     *       NOW GET THE VIOLATING INSTRUCTION AND SEPARATE OUT ITS OP CODE
                740     *
001037  340000  750             TAD     0               YIELDS LOCATION OF THE OFFENDING INSTRUCTION AND SETS THE LINK
001040  040702  760             DAC     INSTR
001041  220702  770     MP111   LAC     INSTR,X
001042  040702  780             DAC     INSTR           INSTR CONTAINS THE BAD INSTRUCTION
001043  041005  790             DAC     PINST           SAVE THE INSTRUCTION IN CASE IT IS A GRAPHICS II INSTRUCTION
001044  501624  800             AND     (NOP)
001045  741200  810             SNA
001046  601236  820             JMP     ERR3            CAL IS AN ILLEGAL INSTRUCTION
001047  040703  830             DAC     OPCOD           SAVE THE OP CODE OF THE VIOLATING INSTRUCTION
                840     *
                850     *       NOW CHECK FOR A MICRO-CODED INSTRUCTION: OPERATE OR IOT
                860     *
001050  541625  870             SAD     (EAE)
001051  601115  880             JMP     O.K.            EAE INSTRUCTIONS ARE INNOCENT
001052  541624  890             SAD     (OPR)
001053  601365  900     OPRST   JMP     .OPR.           OPERATE INSTRUCTION VIOLATION
001054  541626  910             SAD     (IOT)
001055  601202  920             JMP     .IOT.           SERVICE IOT VIOLATION
```

M MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS

```
                    930             .EJECT
                    940     *
                    950     *       NOT A MICROCODED INSTUCTION, SO ESTABLISH THE EFFECTIVE INSTRUCTION
                    960     *
 001056  200702     970             LAC     INSTR
 001057  501627     980             AND     (020000)
 001060  741200     990             SNA                     SKIP IF THE VIOLATING INSTRUCTION HAD THE INDIRECT BIT ON
 001061  601073    1000             JMP     MPBA1           ELSE CARRY ON NORMALLY
 001062  200702    1010             LAC     INSTR           LOAD THE ORIGINAL INSTRUCTION
 001063  501630    1020             AND     (17770)         RETAIN THE ADDRESS ONLY -- DELETING THE LOW ORDER THREE BITS
 001064  541631    1030             SAD     (10)            SKIP IF ADDRESSED LOCATION WAS NOT AUTO-INDEX REGISTER
 001065  460702    1040             INX     INSTR,X         ELSE INCREMENT THE LOCATION
 001066  740000    1050             NOP                     IT JUST MIGHT SKIP
 001067  220702    1060             LAC     INSTR,X
 001070  500651    1070             AND     $ADRSS          RECOVER THE EFFECTIVE ADDRESS
 001071  240703    1080             XOR     OPCOD           AND ADD THE OP CODE BACK IN
 001072  040702    1090             DAC     INSTR           SAVE THE (CONSTRUCTED) EFFECTIVE INSTRUCTION
                    1100    *
                    1110    *       VIOLATION WAS A MEMORY REFERENCE INSTRUCTION (OTHER THAN CAL, JMP, JMS, OR XCT)
                    1120    *       WHICH ATTEMPTED TO REFERENCE A LOCATION BELOW THE BOUNDARY,
                    1130    *       OR THE VIOLATION WAS AN XCT OR PI INTERRUPT AND THE REFERENCE MAY NOT BE TO PROTECTED MEMORY
                    1140    *       REFERENCES TO 0-7 & 21-27 MUST BE FAKED; REFERENCES TO 10-20 & 30-37 ARE CARRIED OUT LITERALLY
                    1150    *
 001073  200702    1160    MPBA1   LAC     INSTR           LOAD THE OFFENDING INSTRUCTION
 001074  500651    1170            AND     $ADRSS          RETAIN JUST THE ADDRESS BITS
 001075  341632    1180            TAD     (-10)
 001076  741100    1190            SPA
 001077  601112    1200            JMP     FAKIT           REFERENCE TO 0-7
 001100  341632    1210            TAD     (-10)
 001101  741100    1220            SPA
 001102  601115    1230            JMP     O.K.            REFERENCE TO 10-17
 001103  341633    1240            TAD     (-20)
 001104  741100    1250            SPA
 001105  601112    1260            JMP     FAKIT           REFERENCE TO 21-37
 001106  341634    1270            TAD     (-BOUNDARY+40)
 001107  751100    1280            SPA!CLA
 001110  601241    1290            JMP     ERR2            THE REFERENCE IS TO 40-BOUNDARY
 001111  601115    1300            JMP     O.K.            THE REFERENCE IS ABOVE THE BOUNDARY
                    1310    *
                    1320    *       CONVERT THE LEGAL PROTECTED MEMORY REFERENCE TO A REFERENCE TO THE USER TABLE
                    1330    *
 001112  201635    1340    FAKIT   LAC     ($.0)
 001113  340702    1350            TAD     INSTR           CONVERT REFERENCE TO POINT TO USER TABLE IMAGE
 001114  040702    1360            DAC     INSTR
                    1370    *
                    1380    *       NOW DO THE USER INSTRUCTION
                    1390    *
    001115          1400    O.K.    ...                     NOTE THE USER MQ, SC, 10, & 11 ARE STILL O.K. AT THIS POINT
                    1410                                    UNLESS ARRIVED AS A PI INTERRUPT. IN THAT CASE 10 & 11
                    1420                                    MAY BE OFF, BUT THE USER'S LOCATION 1 CAN'T LEGALLY BE A
                    1430                                    MEMORY REFERENCE INSTRUCTION, ANYWAY.
                    1440    *
```

M MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS

```
                    1450    *       SPECIAL CASE OP CODES
                    1460    *
001115  200703      1470    MP12    LAC     OPCOD           RELOAD THE OP CODE
001116  540634      1480            SAD     $BIT0
001117  601155      1490            JMP     .XCT.           VIOLATION WAS AN XCT INSTRUCTION
001120  541636      1500            SAD     (JMS)
001121  601142      1510            JMP     .JMS.           VIOLATION WAS CAUSED BY A JMS INSTRUCTION
001122  541637      1520            SAD     (JMP)
001123  601146      1530            JMP     .JMP.
001124  200000      1540            LAC     0
001125  740010      1550            RAL                     RESTORE THE USER'S LINK FOR THIS OPERATION
001126  200005      1560            LAC     $3AC            AND ALSO HIS AC
001127  400702      1570            XCT     INSTR           EXECUTE THE USER'S INSTRUCTION
001130  741000      1580            SKP                     AVOID INCREMENTING THE RETURN IF THE USER'S INSTRUCTION DID NOT SKIP
001131  440000      1590            INX     0               BUT BUMP THE RETURN IF THE USER'S INSTRUCTION DID SKIP
001132  040005      1600            DAC     $3AC            AND SAVE THE USER'S AC
001133  200000      1610            LAC     0               INITIATE SAVING THE CURRENT USER LINK
001134  500644      1620            AND     $CB0            CLEAR THE OLD LINK
001135  741400      1630            SZL                     IS THE LINK ON?
001136  240634      1640            XOR     $BIT0           YES, SO SAVE IT
001137  040000      1650    MP15    DAC     0               REBAVE THE CURRENT USER RETURN, WITH LINK
001140  101170      1660            JMS     REGRES          RESTORE HIS REGISTERS THAT WON'T GET OTHERWISE RESTORED
001141  600274      1670            RET     $PIDN2
                    1680    *
                    1690    *       SPECIAL MEMORY REFERENCE INSTRUCTIONS
                    1700    *
001142  200000      1710    .JMS,   LAC     0               LOAD THE USER'S PC
001143  501621      1720            AND     (677777)        TURN OFF THE MEMORY PROTECT BIT

001144  060702      1730            DAC     INSTR,X         SET THE USER PC AT THE START OF THE SUBROUTINE
001145  440702      1740            INX     INSTR           AND INCREMENT THE TRANSFER
                    1750
001146  200000      1760    .JMP,   LAC     0               LOAD THE USER'S PC
001147  501626      1770            AND     (700000)        KEEP THE HIGH ORDER BITS OF THE PC
001150  040000      1780            DAC     0
001151  200702      1790            LAC     INSTR
001152  500651      1800            AND     $ADRSS          GET THE NEW USER PC
001153  240000      1810            XOR     0               COMBINE IT WITH THE OLD HIGH-ORDER BITS
001154  601137      1820            JMP     MP15            EXIT
                    1830
001155  200705      1840    .XCT,   LAC     TEMP5           LOAD THE XCT COUNT
001156  750200      1850            SZAICLA
001157  601235      1860            JMP     ERR4            CHAINED XCT'S NOT YET LEGAL
001160  440705      1870    XCT1    INX     TEMP5           NOW COUNT THE XCT
001161  601041      1880            JMP     MP111           AND ITERATE
                    1890
    001162          1900    REGSAVE ENTER                   SAVE THE REGISTERS THAT HAVEN'T ALREADY BEEN SAVED
                                    .PMC    SAVE.ON
001162  740040                      XX
001163  641002      1910            LACQ                    THE FOLLOWING LOCATIONS MAY ALTER IF THE VIOLATION WAS AN IOT INSTRUCTION
001164  041754      1920            DAC     $MQ
001165  641001      1930            LACS
001166  041755      1940            DAC     $SC
```

```
                M                               MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS

001167  621162  1950            RET     REGSAVE,X
                1960
    001170      1970    REGRES  ENTER                   RESTORE THE REGISTERS THAT REGSAVE SAVED
                        .PMC    SAVE,ON
001170  740040                    XX
001171  201755  1980            LAC     $SC             RELOAD THE OLD STEP COUNT
001172  241640  1990            XOR     (77)            COMPLEMENT THE STEP COUNT
001173  341641  2000            TAD     (640402)        DEVELOP A PSEUDO-NORMALIZE INSTRUCTION
001174  501642  2010            AND     (640477)        DELETE POSSIBLE STEP COUNT OVERFLOW
001175  041176  2020            DAC     .+1             PLACE THE NORMALIZE INSTRUCTION IN SEQUENCE
001176  740040  2030            XX                      STEP COUNT TO THE SC
001177  201754  2040            LAC     $MQ             RELOAD THE OLD MQ
001200  652000  2050            LMQ                     AND SET IT
001201  621170  2060            RET     REGRES,X
```

M MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS

```
                    2070            .EJECT
                    2080    *
                    2090    *       VIOLATION WAS AN IOT INSTRUCTION -- SEPARATE THE MICROCODING FROM THE REST
                    2100    *
 001202  200702     2110    .IOT.   LAC     INSTR           LOAD THE VIOLATING IOT INSTRUCTION
 001203  501643     2120            AND     ($SPMSK)        RECOVER THE "SPECIAL" BITS
 001204  541644     2130            SAD     ($SPECIAL)      AND CHECK THEM
 001205  601246     2140            JMP     SWAP2           YES -- GET THE SPECIALS HANDLER
 001206  200702     2150            LAC     INSTR           ELSE RELOAD THE VIOLATING IOT INSTRUCTION
 001207  640504     2160            LRS     4               PUT THE MICROCODED BITS IN THE MQ
 001210  040702     2170            DAC     INSTR           SAVE FILL + ALL BUT THE MICROCODED BITS
 001211  641601     2180            EAECLA!LLS 1            RECOVER THE CLEAR AC BIT
 001212  740200     2190            SZA
 001213  140005     2200            DZM     $3AC            ZERO THE USER AC IF THAT BIT WAS SET
                    2210    *
                    2220    *       LOOK UP THE IOT AND BRANCH TO THE PROPER HANDLING ROUTINE
                    2230    *
    001214          2240    IOT0    ...
 001214  140266     2250            DZM     $DKLOK          CLEAR THE DISK-USE FLAG
 001215  761333     2260            LAW     IOTTT-1
 001216  040010     2270            DAC     10              SET UP THE TABLE READ
 001217  220010     2280    IOT1    LAC     10,X            READ THE NEXT TABLE ENTRY
 001220  540702     2290            SAD     INSTR           CHECK AGAINST THE INSTRUCTION IN QUESTION
 001221  601225     2300            JMP     IOT2            MATCHES--BRANCH TO THE HANDLING ROUTINE
 001222  541364     2310            SAD     IOTT9           CHECK FOR THE END OF THE TABLE
 001223  601416     2320            JMP     IOTSW           DONE, AND NO MATCH FOUND
 001224  601217     2330            JMP     IOT1
                    2340
 001225  220010     2350    IOT2    LAC     10,X
 001226  040702     2360            DAC     TEMP2           SET THE TRANSFER
 001227  641002     2370            LACQ
 001230  040703     2380            DAC     TEMP3           SET THE MICROCODE
 001231  742010     2390            RTL
 001232  751100     2400            SPA!CLA                 SKIP IF THERE IS NO IOPS EVENT TIME 1 EVENT
 001233  440702     2410            INX     TEMP2           ELSE BUMP THE ENTRANCE
 001234  601424     2420            JMP     IOT3
                    2430    *
                    2440    *       COMMON ERROR MESSAGES
                    2450    *
 001235  340641     2460    ERR4    TAD     $BIT17          CHAINED XCT'S
 001236  340641     2470    ERR3    TAD     $BIT17          ILLEGAL INSTRUCTION
 001237  741000     2480            SKP
 001240  440000     2490    ERR1    INX     0               ILLEGAL TRANSFER PC NEEDS TO BE FUDGED TO BE ONE TOO GREAT (LIKE ALL ELSE)
 001241  341645     2500    ERR2    TAD     (2)             BAD ADDRESS
 001242  041706     2510    ERR     DAC     $UTEM2          SET THE ERROR MESSAGE NUMBER
 001243  101170     2520            JMS     REGRES          FIX UP THE USER REGISTERS BEFORE TRANSFERRING OUT OF THIS ROUTINE
 001244  761004     2530    SWAP1   LAW     $SWERR
 001245  600335     2540            JMP     $SWAP           GET THE SWAPPER -- ERROR MESSAGE ENTRY POINT
 001246  101170     2550    SWAP2   JMS     REGRES          FIX UP THE USER REGISTERS BEFORE TRANSFERRRING OUT OF THIS ROUTINE
 001247  761005     2560            LAW     $SWSPL
 001250  600335     2570            JMP     $SWAP           GO READ IN THE MONITOR/MESSAGE PHANTOM PROGRAM
                    2580
```

M MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS

```
                    2590
                    2600
      001251        2610    OAC     ENTER                   INCLUSIVE OR AC WITH USER AC FOR IOT'S
                                    .PMC    SAVE,ON
  001251  740040                    XX
  001252  040002   2620             DAC     $3TM21
  001253  740001   2630             CMA
  001254  500005   2640             AND     $3AC
  001255  240002   2650             XOR     $3TM21
  001256  040005   2660             DAC     $3AC
  001257  621251   2670             RET     OAC,X
                    2680
                    2690
      001260        2700    TIM3    ENTER                   DETERMINE WHETHER OR NOT THERE IS AN IOPS EVENT TIME 3 EVENT REQUESTED
                                    .PMC    SAVE,ON
  001260  740040                    XX
  001261  200703   2710             LAC     TEMP3
  001262  751100   2720             SPAICLA
  001263  621260   2730             RET     TIM3,X          YES -- GO DO IT
  001264  601431   2740             JMP     MEMD1           EXIT
                    2750
                    2760    *
                    2770    *       RDBLK LOOKS FIRST AT THE USER DUE TO RUN NEXT. IF HE IS I/O ROADBLOCKED, THE
                    2780    *       FOLLOWING USER IS EXAMINED. THIS PROCESS IS REPEATED UNTIL SOME USER IS FOUND
                    2790    *       WHO IS FREE TO RUN. NOTE THAT THE ROUTINE, ONCE ENTERED, WILL LOOP INDEFINITELY
                    2800    *       UNTIL A FREE USER IS FOUND.
                    2810    *
                    2820    *       WHEN A FREE USER IS FOUND, THE RETURN IS IMMEDIATE WITH HIS RE-ENTRANT
                    2830    *       TEMPORARY STORAGE SET UP.
                    2840    *
                    2850    *       RDBLK ENABLES THE INTERRUPT SYSTEM TO PERMIT TELETYPE I/O TO
                    2860    *       GO ON WHILE CHECKING FOR ROADBLOCKS, OTHERWISE A TELETYPE
                    2870    *       I/O ROADBLOCK COULD NEVER BE RELIEVED. THIS MEANS THAT BEFORE
                    2880    *       ENABLING INTERRUPTS, THE CALLER'S SAVED AC, 10, & 11 MUST BE
                    2890    *       COPIED, AND RECOPIED BEFORE EXIT.
                    2900    *
                    2910    *       THIS ROUTINE RUNS WITH THE CLOCK OFF TO PREVENT RE-ENTRANCE
                    2920    *       AT A TIME WHEN IT WOULD CRASH THE SYSTEM, ALSO NO ONE IS RUNNING
                    2930    *       AS LONG AS WE ARE HUNG IN THIS LOOP, SO NOTHING IS LOST.
                    2940    *
      001265        2950    RDBLK   ENTER
                                    .PMC    SAVE,ON
  001265  740040                    XX
  001266  700004   2960             CLOF
  001267  200000   2970             LAC     0
  001270  040702   2980             DAC     TEMP2           SAVE THE RETURN
  001271  200005   2990             LAC     $3AC
  001272  040703   3000             DAC     TEMP3           THE SAVED AC
  001273  200026   3010             LAC     $,310
  001274  040010   3020             DAC     10              THE SAVED AUTO-INDEX REGISTER 10
  001275  200027   3030             LAC     $,311
  001276  040011   3040             DAC     11              THE SAVED AUTO-INDEX REGISTER 11
```

```
                M                       MEMORY PROTECT ROUTINES COMMON TO BOTH OVERLAYS

001277  700042  3050            ION
001300  201771  3060            LAC     $NUMBR          SEE WHO IS RUNNING, IF ANYONE
001301  540104  3070            SAD     $CTNAM
001302  601310  3080            JMP     RDBK2           #1 IS NOW RUNNING -- SO GIVE #2 A CHANCE
001303  540133  3090            SAD     $L1NAM
001304  601313  3100            JMP     RDBK3           #2 IS NOW RUNNING -- SO GIVE #3 A CHANCE
                3110
001305  200102  3120    RDBK1   LAC     $CTFLG          LOAD USER #0 I/O FLAGS
001306  101317  3130            JMS     RDBK5           SEE IF USER #0 IS I/O ROADBLOCKED
001307  760076  3140            LAW     $CTBIN-2        LOAD A POINTER TO USER #0 PARAMETERS
001310  200131  3150    RDBK2   LAC     $L1FLG
001311  101317  3160            JMS     RDBK5           SEE IF USER #1 IS I/O ROADBLOCKED
001312  760125  3170            LAW     $L1BIN-2
001313  200160  3180    RDBK3   LAC     $L2FLG
001314  101317  3190            JMS     RDBK5           SEE IF USER #2 IS I/O ROADBLOCKED
001315  760154  3200            LAW     $L2BIN-2
001316  601305  3210            JMP     RDBK1           LOOP
                3220
                3230    *       SEE IF THE SPECIFIED USER IS I/O ROADBLOCKED, IF SO, RETURN TO THE
                3240    *       ROADBLOCK ROUTINE FOR ANOTHER TRY, IF NOT, EXIT WITH HIS TEMPS SET UP.
                3250
    001317      3260    RDBK5   ENTER
                                ,PMC    SAVE,ON
001317  740040                    XX
001320  742010  3270            RTL                     TELEPRINTER FLAG TO LINK; KEYBOARD FLAG TO AC(0)
001321  741500  3280            SZL!SPA                 SKIP IF THERE IS NO I/O ROADBLOCK
001322  621317  3290            RET     RDBK5,X         ELSE TRY THE NEXT ONE
                3300    *
                3310    *       A NON-ROADBLOCKED USER HAS BEEN FOUND, TURN OFF THE INTERRUPT
                3320    *       SYSTEM, SET UP HIS RE-ENTRANT PARAMETERS AND EXIT.
                3330    *       NOTE THAT AUTO-INDEX REGISTERS 10 & 11 ARE ALREADY CORRECT
                3340    *       SO ONLY THE AC AND THE RESTART ADDRESS NEED TO BE RESTORED.
                3350    *
001323  700002  3360            IOF                     TURN OFF THE INTERRUPT SYSTEM
001324  200702  3370            LAC     TEMP2
001325  040000  3380            DAC     0               RESTORE THE SAVED RETURN
001326  200703  3390            LAC     TEMP3
001327  040005  3400            DAC     $3AC            AND THE SAVED AC
001330  421317  3410            XCT     RDBK5,X         LOAD THE POINTER TO HIS PARAMETERS
001331  100513  3420            JMS     $IO,IN          AND GO SET THEM UP
                3430    *
                3440    *       NOW WE ARE EVIDENTLY READY TO RUN SOMEONE AGAIN, SO TURN IT
                3450    *       BACK ON.
                3460    *
001332  700044  3470            CLON
001333  621265  3480            RET     RDBLK,X         DONE
                3490            .END
```

```
                M                       PROTECTION OVERLAY #2

                        120             .STITL  PROTECTION OVERLAY #2
                        130     *
                        140     *       IOT INSTRUCTION TRANSFER TABLE
                        150     *
    001334              160     IOTTT   ...,                    IOT INSTRUCTION TRANSFER TABLE
001334  774340          170             774340                  707000
001335  601425          180             JMP     .DSSF           DSSF
001336  774341          190             774341                  707020
001337  601433          200             JMP     DSK1            DSCC;DRAL;DLAL
001340  774342          210             774342                  707040
001341  601444          220             JMP     DSK2            DSCF;DSFX;DSCN
001342  774343          230             774343                  707060
001343  601431          240             RET     MEMD1           DRAH;DLAH -- NEITHER HAS ANY LEGAL EFFECT, SO IGNORE THEM
001344  774352          250             774352                  707240
001345  601504          260             JMP     .DSCD           DSCD
001346  774353          270             774353                  707260
001347  601511          280             JMP     .DSRS           DSRS
001350  774004          290             774004                  700100
001351  601534          300             JMP     PTR1            RSF;RCF;RSA;RRB
001352  774006          310             774006                  700140
001353  601552          320             JMP     .RSB            RSB
001354  774010          330             774010                  700200
001355  601563          340             JMP     PTP1            PSF;PCF;PSA
001356  774012          350             774012                  700240
001357  601600          360             JMP     .PSB            PSB
001360  774000          370             774000                  700000
001361  601516          380             JMP     .OFF            IOF;CLSF;CLOF
001362  774156          390             774156                  703340
001363  601526          400             JMP     BRK             SKP7;DBR
001364  001000          410     IOTT9   1000                    END FLAG
                        420     *
                        430     *       ILLEGAL OPERATE INSTRUCTIONS HAVE EITHER THE HALT BIT (BIT 12) OR THE
                        440     *       OAS BIT (BIT 15) ON IF THEY TRAPPED THEMSELVES. IF THE TRAP WAS AN XCT (OPR)
                        450     *       IT IS POSSIBLE NEITHER ONE IS ON.
                        460     *
    001365              470     .OPR,   ...
001365  140266          480             DZM     $DKLOK          CLEAR THE DISK USE FLAG
001366  200702          490             LAC     INSTR           LOAD THE ILLEGAL INSTRUCTION
001367  501646          500             AND     (40)
001370  750200          510             SZA;CLA                 SKIP IF THE HALT BIT IS NOT SET
001371  601617          520             JMP     ERR5            GIVE THE USER HIS HALT MESSAGE
001372  200702          530             LAC     INSTR           RELOAD THE INSTRUCTION
001373  501647          540             AND     (100004)        RECOVER THE CLA AND OAC BITS
001374  746010          550             CLL;RTL
001375  740010          560             RAL                     MOVE THE CLA BIT TO THE LINK
001376  741200          570             SNA                     SKIP IF THE OAC BIT WAS SET
001377  601115          580             JMP     O.K.            ELSE DO THE USER'S OPERATE INSTRUCTION
001400  741400          590             SZL                     SKIP UNLESS CLA BIT WAS SET
001401  140005          600             DZM     $3AC            IN WHICH CASE CLEAR THE ACCUMULATOR
001402  201771          610             LAC     $NUMBR          LOAD THIS USER NUMBER
001403  540034          620             SAD     $RACS           SEE IF THIS USER WAS ALLOCATED THE ACCUMULATOR SWITCHES
001404  741000          630             SKP                     YES
```

```
                M                               PROTECTION OVERLAY #2

 001405  601410  640             JMP     .OPR2           NO -- USE THE SOFTWARE VALUE
 001406  750004  650             LAS                     YES -- USE THE HARDWARE SWITCH VALUE
 001407  741000  660             SKP
 001410  201756  670     .OPR2   LAC     $ACS            LOAD THE SOFTWARE ACCUMULATOR SWITCHES VALUE
 001411  101251  680             JMS     OAC             OR WHICHEVER VALUE IT IS INTO THE USER'S AC
 001412  200702  690             LAC     INSTR           RELOAD THE INSTRUCTION
 001413  501650  700             AND     (377773)        REMOVE THE OAS AND CLA BITS, SINCE THEY ARE DONE
 001414  040702  710             DAC     INSTR           RESET THE INSTRUCTION
 001415  601115  720             JMP     O.K.            DO ANY REMAINING OPERATE INSTRUCTION
                 730     *
                 740     *
                 750     *       ROUTINE TO CALL THE NEXT MEMORY PROTECTION OVERLAY UNLESS IT HAS
                 760     *       ALREADY HAD ITS CHANCE AT THE VIOLATION
                 770     *
    001416       780     IOTSW   ...,
 001416  200704  790             LAC     TEMP4           LOAD THE OVERLAY COUNT
 001417  750200  800             SZA!CLA                 SKIP UNLESS ALL OVERLAYS HAVE ALREADY HAD A CHANCE
 001420  601236  810             JMP     ERR3            ELSE THIS WAS THE LAST CHANCE -- IT MUST HAVE BEEN AN ILLEGAL INSTRUCTION
 001421  440704  820             INX     TEMP4           COUNT THIS OVERLAY
 001422  761007  830             LAW     $SWMP1          LOAD A POINTER TO THE SWAPPER ENTRANCE FOR NEXT MEMORY PROTECTION OVERLAY
 001423  600335  840             JMP     $SWAP           AND GET IT
                 850     *
                 860     *
                 870     *
 001424  620702  880     IOT3    JMP     TEMP2,X         GOTO THE PROPER SERVICE ROUTINE
                 890     *
                 900     *       DISK IOT INSTRUCTIONS
                 910     *
 001425  601431  920     .DSSF   RET     MEMD1           THERE IS NO IOPS EVENT TIME 2 OR 3 INSTRUCTION
 001426  201761  930             LAC     $DFLAG
 001427  740200  940             SZA
 001430  440000  950             INX     0               BUMP THE RETURN IF THE SOFTWARE FLAG WAS SET
 001431  101170  960     MEMD1   JMS     REGRES          RESTORE THE REGISTERS NOT OTHERWISE RESTORED
 001432  600270  970             RET     $PIDON
                 980
 001433  740000  990     DSK1    NOP                     NO IOPS EVENT TIME 1
 001434  740400  1000    .DSCC   SNL                     DSCC IS DONE ANYWAY, SO IGNORE IT & PROCEDE
 001435  601440  1010            JMP     DSK12           NO IOPS EVENT TIME 2 EVENT
 001436  201762  1020    .DRAL   LAC     $DAP0

 001437  101251  1030            JMS     OAC
 001440  101260  1040    DSK12   JMS     TIM3            RETURN ONLY IF THERE IS AN IOPS EVENT TIME 3 EVENT
 001441  200005  1050    .DLAL   LAC     $3AC
 001442  041762  1060            DAC     $DAP0
 001443  601431  1070            RET     MEMD1
                 1080
 001444  741000  1090    DSK2    SKP                     NO IOPS EVENT TIME 1 EVENT
 001445  141764  1100    .DSCF   DZM     $DFN
 001446  740400  1110            SNL
 001447  601453  1120            JMP     .+4
 001450  200005  1130    .DSFX   LAC     $3AC
 001451  241764  1140            XOR     $DFN
 001452  041764  1150            DAC     $DFN
```

M PROTECTION OVERLAY #2

```
001453  101260  1160          JMS     TIM3
001454  201762  1170  .DSCN   LAC     $DAP0
001455  707024  1180          DLAL                    SET UP AP0 CORRECTLY FOR THE USER
001456  777777  1190          LAW     -1
001457  340651  1200          TAD     $ADRSS
001460  740001  1210          CMA                     YIELDS MINUS THE HIGHEST LEGAL CORE ADDRESS TO START THE TRANSFER
001461  341751  1220          TAD     $,0+$DKWC
001462  341752  1230          TAD     $,0+$DKCA
001463  750100  1240          SMA!CLA                 SKIP IF THE START ADDRESS IS LEGAL FOR THIS LENGTH TRANSFER
001464  601615  1250          JMP     ERR7            ELSE ANNOUNCE EXCESSIVE WORD COUNT
001465  776001  1260          LAW     -BOUNDARY+1
001466  341752  1270          TAD     $,0+$DKCA
001467  751100  1280          SPA!CLA                 SKIP IF NOT TRYING TO START THE TRANSFER BELOW THE BOUNDARY
001470  601614  1290          JMP     ERR8            ELSE ANNOUNCE BAD CORE ADDRESS
001471  140266  1300          DZM     $DKLOK
001472  440266  1310          INX     $DKLOK          FLAG USER USING DISK
001473  201751  1320          LAC     $,0+$DKWC
001474  040036  1330          DAC     $DKWC           SET UP THE REAL WORD COUNT
001475  201752  1340          LAC     $,0+$DKCA
001476  040037  1350          DAC     $DKCA           SET UP THE REAL CORE ADDRESS
001477  201764  1360          LAC     $DFN
001500  501651  1370          AND     (6)
001501  240641  1380          XOR     $BIT17          FORCE AN INTERRUPTING COMMAND
001502  707047  1390          DSCF!DSFX!DSCN
001503  601431  1400          RET     MEMD1
                1410
001504  740400  1420  .DSCD   SNL                     NO IOPS EVENT TIME 1 EVENT
001505  601431  1430          RET     MEMD1           EXIT UNLESS AN EVENT 2 IS PRESENT -- 1 AND 3 DON'T EXIST
001506  141761  1440          DZM     $DFLAG
001507  141765  1450          DZM     $DSTAT
001510  601431  1460          RET     MEMD1
                1470
001511  740400  1480  .DSRS   SNL                     NO IOPS EVENT TIME 1 EVENT
001512  601431  1490          RET     MEMD1           EXIT UNLESS EVENT 2 -- 1 AND 3 DON'T EXIST
001513  201765  1500          LAC     $DSTAT
001514  101251  1510          JMS     OAC
001515  601431  1520          RET     MEMD1
                1530
                1540
                1550
001516  740400  1560  .OFF    SNL                     SKIP ONLY IF THERE IS AN EVENT TIME 2 EVENT (IOF)
001517  601236  1570  .CLSF   JMP     ERR3            CLOCK NOT YET ENABLED
001520  201760  1580  .IOF    LAC     $IORS           LOAD USER IORS WORD
001521  740010  1590          RAL
001522  744020  1600          CLL!RAR
001523  041760  1610          DAC     $IORS           RESTORE IORS WORD WITH PI FLAG $OFF
001524  101260  1620  .OFF2   JMS     TIM3            RETURN IF THERE IS AN IOPS TIME 3 EVENT
001525  601236  1630  .CLOF   JMP     ERR3            CLOCK NOT YET ENABLED
                1640
                1650
                1660
001526  741000  1670  BRK     SKP
```

```
                M                                PROTECTION OVERLAY #2

001527  440000  1680            INX     0               'SKP7' -- SKIP IF NOT A PDP4
001530  101260  1690            JMS     TIM3            RETURN IF THERE IS AN IOPS EVENT TIME 3 EVENT
001531  201652  1700    .DBR    LAC     (DBR)
001532  040303  1710            DAC     $PIOUT
001533  601431  1720            RET     MEMD1
                1730
                1740
001534  601541  1750    PTR1    JMP     PTR11           NO IOPS EVENT TIME 1
001535  101555  1760    .RSF    JMS     RDRP            CHECK FOR READER PERMISSION
001536  200234  1770            LAC     $RFLAG
001537  740200  1780            SZA
001540  440000  1790            INX     0               BUMP THE RETURN IF THE FLAG WAS SET
001541  101555  1800    PTR11   JMS     RDRP            CHECK FOR READER PERMISSION
001542  740400  1810            SNL
001543  601547  1820            JMP     PTR12           NO IOPS EVENT TIME 2
001544  140234  1830    .RCF    DZM     $RFLAG
001545  700112  1840            RRB
001546  101251  1850            JMS     OAC             UPDATE THE USER AC
001547  101260  1860    PTR12   JMS     TIM3            RETURN IF THERE IS AN IOPS TIME 3 EVENT
001550  700104  1870    .RSA    RSA
001551  601431  1880            RET     MEMD1
                1890
001552  101555  1900    .RSB    JMS     RDRP            CHECK FOR READER PERMISSION
001553  700144  1910            RSB
001554  601431  1920            RET     MEMD1
                1930
001555          1940    RDRP    ENTER                   CHECK FOR PERFORATED TAPE READER PERMISSION
                                .PMC    SAVE,ON
001555  740040                  XX
001556  200235  1950            LAC     $RPTR
001557  541771  1960            SAD     $NUMBR
001560  621555  1970            RET     RDRP,X
001561  760011  1980            LAW     9.
001562  601242  1990            JMP     ERR
                2000
                2010
                2020
001563  601570  2030    PTP1    JMP     PTP11           NO IOPS EVENT TIME 1
001564  101605  2040    .PSF    JMS     PTPP            CHECK FOR PUNCH PERMISSION
001565  200227  2050            LAC     $PFLAG
001566  740200  2060            SZA
001567  440000  2070            INX     0               BUMP THE RETURN IF THE FLAG WAS SET
001570  101605  2080    PTP11   JMS     PTPP            CHECK FOR PUNCH PERMISSION
001571  741400  2090            SZL
001572  140227  2100    .PCF    DZM     $PFLAG
001573  101260  2110            JMS     TIM3            RETURN IF THERE IS AN IOPS TIME 3 EVENT
001574  200005  2120    .PSA    LAC     $3AC
001575  700204  2130            PSA
001576  140227  2140            DZM     $PFLAG          CLEAR THE SOFTWARE FLAG
001577  601431  2150            RET     MEMD1
                2160
001600  101605  2170    .PSB    JMS     PTPP            CHECK FOR PUNCH PERMISSION
```

```
            M                           PROTECTION OVERLAY #2

001601  200005  2180          LAC     $3AC
001602  700244  2190          PSB
001603  140227  2200          DZM     $PFLAG          CLEAR THE SOFTWARE FLAG
001604  601431  2210          RET     MEMD1
                2220
    001605      2230    PTPP  ENTER                   CHECK FOR PAPER TAPE PERMISSION
                            ,PMC    SAVE,ON
001605  740040              XX
001606  200230  2240          LAC     $RPTP
001607  541771  2250          SAD     $NUMBR
001610  621605  2260          RET     PTPP,X
001611  760011  2270          LAW     9.
001612  601242  2280          JMP     ERR
                2290
001613  340641  2300    ERR9  TAD     $BIT17          UNASSIGNED DEVICE REQUESTED
001614  340641  2310    ERR8  TAD     $BIT17          ATTEMPTED DISK/DECTAPE TRANSFER TO/FROM PROTECTED MEMORY
001615  340641  2320    ERR7  TAD     $BIT17          EXCESSIVE WORD COUNT FOR DISK/DECTAPE TRANSFER
001616  340641  2330    ERR6  TAD     $BIT17          DISK OVERFLOW ATTEMPTED
001617  340641  2340    ERR5  TAD     $BIT17          HALTED PROGRAM
001620  601235  2350          JMP     ERR4            CARRY ON
001621  677777  2360          ,END    OVSTRT
001622  100001
001623  776000
001624  740000
001625  640000
001626  700000
001627  020000
001630  017770
001631  000010
001632  777770
001633  777760
001634  776040
001635  001713
001636  100000
001637  600000
001640  000077
001641  640402
001642  640477
001643  777400
001644  705000
001645  000002
001646  000040
001647  100004
001650  377773
001651  000006
001652  703344

TRANSFER ADDRESS  601000
```

M CROSS REFERENCE TABLE

1713	.0	4510	4520	480	520	1340	1220	1230	1270	1320	1340		
26	.310	3400	3010										
27	.311	3410	3030										
4464	.DT	570											
6460	.TP	550											
2023	10SAVE	1870	1880										
2024	11SAVE	1880	1920										
5	3AC	3370	1560	1600	2200	2640	2660	2990	3400	600	1050	1130	2120	2180
305	3REST	3870	3880										
51	3TEM0	3530	3540										
52	3TEM1	3540	3550										
53	3TEM2	3550	3560										
54	3TEM3	3560	3570										
55	3TEM4	3570	3580										
56	3TEM5	3580	3590										
57	3TEM6	3590	3600										
50	3TM20	3520	3530										
2	3TM21	3350	2620	2650									
3	3TM22	3360											
14000	7K	1030											
16000	8K	1020	910	1010	2640	2650							
1753	AC	4520	4530										
1756	ACS	4550	4560	670									
2015	ACSAVE	1810	1820										
2022	ACSW	1860	1870										
651	ADRSS	4100	4110	670	1070	1170	1800	1200					
300	AT	2950											
300	ATSGN	2900											
422030	BAS	420											
2151	BCNTRL	2330	2340										
634	BIT0	3970	3980	490	1480	1640							
641	BIT17	4020	4030	2460	2470	1380	2300	2310	2320	2330	2340		
635	BIT36	3980	3990										
636	BIT5	3990	4000										
637	BIT6	4000	4010										
640	BIT7	4010	4020										
642	BL7	4030	4040										
643	BL8	4040	4050										
2000	BOUNDA	970	960	980	990	1000	1630	5040	5080	680	1270	1260	
377	BRK	5550											
2170	BUFFER	2490	2550										
1000	BUFLEN	2500	2550										
644	CB0	4050	4060	1620									
645	CB1	4060	4070										
646	CB5	4070	4080										
647	CB7	4080	4090										
650	CBL8	4090	4100										
6	CHRMAX	3180	3200										
2	CHRPAK	3130	3200										
50	CLKMAX	2840	3180										
60	CLKSPD	3160	3170										
1757	CLOCK	4560	4570										

M CROSS REFERENCE TABLE

45	CMP1	3490	3500				
46	CMP2	3500	3510				
6	CNTRL	3380	3390				
2053	COMFLG	2200	2210				
2150	COMSTO	2270	2280				
16000	CORMAX	910	980				
47	CSPL	3510	3520				
44	CSWP	3480	3490				
60	CTBFR	3600	3630	3640			
100	CTBIN	3640	3650	3670	4250	3140	
2000	CTEMP0	1630					
2001	CTEMP1	1640					
2002	CTEMP2	1650					
2003	CTEMP3	1660					
2004	CTEMP4	1670					
2005	CTEMP5	1680					
2006	CTEMP6	1690					
2007	CTEMP7	1700					
2010	CTEMP8	1710					
2011	CTEMP9	1720					
102	CTFLG	3650	3660	3120			
104	CTNAM	3660	3070				
2043	D PC	2120	2130				
2154	D BCA	2370	2380				
2153	D BDA	2360	2370				
2163	D FDA	2440	2450				
2042	D LOC	2110	2120				
2022	D ACSW	1860					
2156	D BALT	2390	2400				
2155	D BLEN	2380	2390				
2161	D BMAX	2420	2430				
2157	D BMIN	2400	2410				
2162	D BPTR	2430	2440				
2167	D FMAX	2480	2490				
2165	D FMIN	2460	2470				
2046	D MASK	2150	2160				
2164	D MFDA	2450	2460				
2036	DADRSW	2070	2080				
1762	DAP0	4590	4600	1020	1060	1170	
1763	DAP1	4600	4610				
653	DBK	4120	4130				
24	DBKNUM	2220	2270				
2054	DBKTAB	2210	2270				
2035	DBSTOR	2050	2060				
422027	DDT	410					
12000	DDTST	5000					
2037	DDUMSW	2080	2090				
1761	DFLAG	4580	4590	930	1440		
1764	DFN	4610	4620	1100	1140	1150	1360
2151	DFTYPE	2340	2350				
2045	DHICOR	2140	2150				
2050	DINDIR	2170	2180				

M CROSS REFERENCE TABLE

100	DK0	4270					
127	DK1	4310					
156	DK2	4350					
37	DKCA	2750	1230	1270	1340	1350	
675	DKDON	4170	4180				
160000	DKLEN	2650	2660				
34	DKLENB	2660					
266	DKLOK	3830	3840	2250	480	1300	1310
672	DKOVR	4160	4170				
2	DKRD	2760					
36	DKWC	2740	1220	1320	1330		
4	DKWRT	2770					
2041	DLIMIT	2100	2110				
2044	DLOCOR	2130	2140				
2160	DMBMIN	2410	2420				
2166	DMFMIN	2470	2480				
654	D0	4130	4140				
662	D02	4140	4150				
663	D03	4150	4160				
2152	DOFTYP	2350	2360				
2032	DPACSW	1980					
2040	DPATSW	2090	2100				
2051	DPCMSK	2180	2190				
2052	DREGBR	2190	2200				
2035	DREGSW	2060	2070				
2047	DRELOC	2160	2170				
1765	DSTAT	4620	4630	1450	1500		
446400	DT.	560					
2000	DTEMP0	1630					
2001	DTEMP1	1640					
2002	DTEMP2	1650					
2003	DTEMP3	1660					
2004	DTEMP4	1670					
2005	DTEMP5	1680					
2006	DTEMP6	1690					
2007	DTEMP7	1700					
2010	DTEMP8	1710					
2011	DTEMP9	1720					
275	EQUAL	2910					
602	FGET	3950	3960				
1701	FRCA	4410	4420				
1700	FRDA	4400	4410				
1702	FRLEN	4420	4430				
1703	FRSTA	4430	4440				
2	FUDGE	3190	3200				
276	GREAT	2930					
1700	IMPLEN	990					
3170	IMPSTR	2550					
422020	INT	320					
513	IO.IN	3910	3920	3420			
525	IO.OT	3920	3930	400			
300000	IOBLK	2830					

M CROSS REFERENCE TABLE

1760	IORS	4570	4580	410	450	1580	1610	
1002	IOTO	4900	4910					
652	JMP	4110	4120					
100	JTLEN	960						
1700	JTSTRT	950	940	960	1000	4400		
16	KBLEN	3610	3630	3640	3680	3690	3730	3740
10	KBNUM	3620	3670	3720				
76	LOLOK	3630						
107	L1BFR	3670	3680	3690				
127	L1BIN	3690	3700	3720	4290	3170		
131	L1FLG	3700	3710	3150				
125	L1LOK	3680						
133	L1NAM	3710	3090					
136	L2BFR	3720	3730	3740				
156	L2BIN	3740	3750	4330	3200			
160	L2FLG	3750	3760	3180				
154	L2LOK	3730						
162	L2NAM	3760	3770					
422026	LDR	390						
2000	LDRST	5040						
274	LESS	2920						
1526	M BRK	1670	400					
1282	M ERR	2510	1990	2280				
1251	M DAC	2610	2670	680	1030	1510	1850	
1531	M .DBR	1700						
1520	M .IOF	1580						
1516	M .OFF	1560	380					
1572	M .PCF	2100						
1574	M .PSA	2120						
1600	M .PSB	2170	360					
1564	M .PSF	2040						
1544	M .RCF	1830						
1550	M .RSA	1870						
1552	M .RSB	1900	320					
1535	M .RSF	1760						
2022	M ACSW	1860						
1433	M DSK1	990	200					
1444	M DSK2	1090	220					
1240	M ERR1	2490	700					
1241	M ERR2	2500	1290					
1236	M ERR3	2470	820	810	1570	1630		
1235	M ERR4	2460	1860	2350				
1617	M ERR5	2340	520					
1616	M ERR6	2330						
1615	M ERR7	2320	1250					
1614	M ERR8	2310	1290					
1613	M ERR9	2300						
1214	M IOTO	2240	310					
1217	M IOT1	2280	2330					
1225	M IOT2	2350	2300					
1424	M IOT3	880	2420					
1145	M MP12	1470						

M CROSS REFERENCE TABLE

M CROSS REFERENCE TABLE

1754	MQ	4530	4540	1920	2040				
2016	MQSAVE	1820	1830						
1305	MRDBK1	3120	3210						
1310	MRDBK2	3150	3080						
1313	MRDBK3	3180	3100						
1317	MRDBK5	3260	3130	3160	3190	3290	3410		
1265	MRDBLK	2950	320	3480					
1244	MSWAP1	2530							
1246	MSWAP2	2550	2140						
2000	MTEMP0	1630							
2001	MTEMP1	1640							
702	MTEMP2	240	2360	2410	2980	3370	880		
703	MTEMP3	260	2380	2710	3000	3390			
704	MTEMP4	270	530	610	790	820			
705	MTEMP5	280	540	620	1840	1870			
2006	MTEMP6	1690							
2007	MTEMP7	1700							
2010	MTEMP8	1710							
2011	MTEMP9	1720							
422025	MTR	370							
2000	MTRST	5080							
1772	NAME	4670	4680						
540	NEWBR	3930	3940						
1771	NUMBR	4660	4670	3060	610	1960	2250		
623	NXPTR	3960	3970						
702	OC0	4180	4190	230	240				
703	OC1	4190	4200	250	260				
704	OC2	4200	4210	270					
705	OC3	4210	280						
574646	OFF	2730							
575600	ON	2720							
1773	OVER	4680	4690						
700	OVLEN	940							
1000	OVSTRT	930	920	940	4750	4880	4960	220	2360
2033	P10SAV	1990	2000						
2034	P11SAV	2000	2050						
2025	PACSAV	1930	1940						
2032	PACSW	1980	1990						
241	PBFLAG	3810	3820						
2017	PCSAVE	1830	1840						
227	PFLAG	3770	3780	2050	2100	2140	2280		
77	PH0	4260	4270						
126	PH1	4300	4310						
155	PH2	4340	4350						
1	PHANTO	2780							
2150	PHFLAG	2280	2330						
1700	PHLEN	2640							
2025	PHSTOR	1920	1930						
274	PIDN2	3850	3860	1670					
270	PIDON	3840	3850	430	970				
1001	PINT	4890	4900						
303	PIOUT	3860	3870	1710					

M CROSS REFERENCE TABLE

```
602026  PLDR        400
  2026  PMQSAV     1940    1950
602025  PMTR        380
  2027  PPCSAV     1950    1960
606064  PPT         520
  2031  PSCSAV     1970    1980
  2030  PSTSAV     1960    1970
606460  PTP         510
606462  PTR         500
 12100  PURLEN     1010
  1775  PURNM      4700    4710
  3700  PURSTR     2560     990   1010    2560
   546  PUTIN      3940    3950
    34  RACS       3440     620
     6  RCNT       3390
    35  RCORE      3450
  1003  RDBLK      4910    4920
    32  RDT0       3420
    33  RDT1       3430
  1170  REGRES     1970    1660   2060    2520    2550     960
  1162  REGSAV     1900     390    600    1950
422021  RES         330
    40  RESCAT     3470    3480
  1000  RESLEN      920
   234  RFLAG      3790    3800   1770    1830
   230  RPTP       3780    3790   2240
   235  RPTR       3800    3810   1950
   242  RSCO       3820    3830
  1776  RSTRT      4710
  1755  SC         4540    4550   1940    1980
640000  SCRSTR     2670
  2021  SCSAVE     1850    1860
   243  SHARP      2890
   377  SPCOD      5410
422122  SPL         430
  1000  SPLST      4960
777400  SPMSK      5390    2120
  2020  STSAVE     1840    1850
   335  SWAP       3880    3890   2540    2570     840
   336  SWAP1      3890    3900
   340  SWAP3      3900    3910
  1000  SWCAT      4750    4760
  1003  SWCLK      4780    4790
  1004  SWERR      4790    4800   2530
  1007  SWMP1      4820    4830    830
  1010  SWMP2      4830    4840
  1002  SWMTR      4770    4780
  1011  SWOPR      4840
422022  SWP         340
  1001  SWPPR      4760    4770
    40  SWPS       3460    3470
  1005  SWSPL      4800    4810   2560
```

M CROSS REFERENCE TABLE

1006	SXSPL	4810	4820	
1300	SYSBAS	2800	2810	
41300	SYSDA	2810		
1777	SYSMAX	2820		
100	TABLEN	2630	2640	
2000	TEMP0	1630	1640	
2001	TEMP1	1640	1650	
2012	TEMP10	1730	1740	
2013	TEMP11	1740	1750	
2014	TEMP12	1750	1800	
2002	TEMP2	1650	1660	
2003	TEMP3	1660	1670	
2004	TEMP4	1670	1680	
2005	TEMP5	1680	1690	
2006	TEMP6	1690	1700	
2007	TEMP7	1700	1710	
2010	TEMP8	1710	1720	
2011	TEMP9	1720	1730	
646000	TP.	540		
376	TRCOFF	5540		
375	TRCON	5530		
2000	TTEMP0	1630		
2001	TTEMP1	1640		
2002	TTEMP2	1650		
2003	TTEMP3	1660		
2004	TTEMP4	1670		
2005	TTEMP5	1680		
2006	TTEMP6	1690		
2007	TTEMP7	1700		
2010	TTEMP8	1710		
2011	TTEMP9	1720		
6	TTYCLK	3170	3180	
3	TTYNUM	3140		
10	TTYSPD	3150	3170	
1774	TYPE	4690	4700	
1766	UCORE	4630	4640	
1767	UDISK	4640	4650	
336	UPARR	2940		
96	US0	4250	4260	4280
125	US1	4290	4300	4320
154	US2	4330	4340	4360
0	USER	2790		
3	USERS	2850	3200	
14000	USLEN	980	2640	
2015	USTORE	1800	1810	
75	UT0	4280		
124	UT1	4320		
153	UT2	4360		
1704	UTEM0	4440	4450	
1705	UTEM1	4450	4460	
1706	UTEM2	4460	4470	2510
1707	UTEM3	4470	4480	

M CROSS REFERENCE TABLE

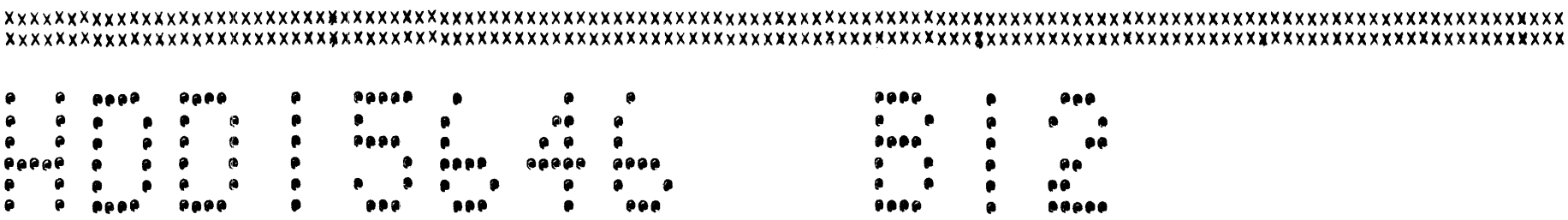
1710	UTEM4	4480	4490
1711	UTEM5	4490	4500
1712	UTEM6	4500	4510
1770	VALID	4650	4660

M UNDEFINED SYMBOLS

#1	5630											
#2	5640											
#3	5650											
#4	5660											
#5	5680											
DEFINS	100	100	120	120	580	580	2680	2680	3210	3210	5330	5330
M #1	190											
PURCOD	5140	5270										

M MACRO CROSS REFERENCE TABLE

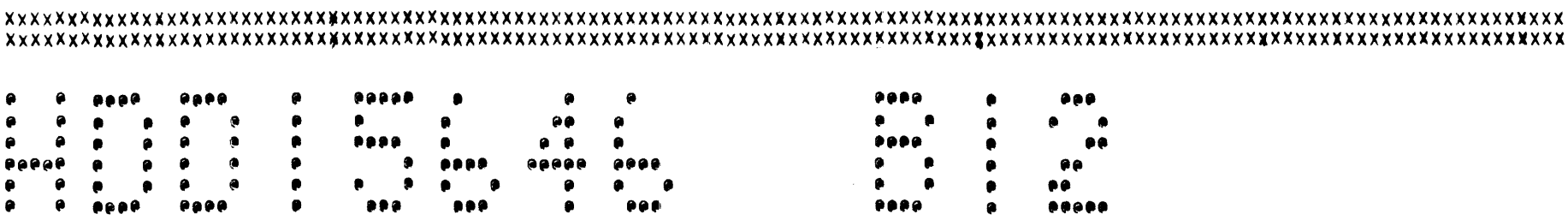
ENTER	5280	1900	1970	2610	2700	2950	3260	1940	2230
MPOFF	5430								
SWAP	5610								
TABLE	170								



05/31/72 01:50:55

```
**********************************
**********************************
**                              **
**  PDP-9 MINI TIME-SHARING SYSTEM  **
**  SPECIAL IOT HANDLER OVERLAY #1  **
**            DTSS:SPL          **
**            MTSS:B12          **
**                              **
**********************************
**********************************
```

```
100             .TITLE  SPECIAL IOT INSTRUCTION (EXECUTIVE SERVICE CALL) HANDLER
110             .NAME   SPL--B12
120             .INSRT  DEFINS
100             .IFUND  DEFINS
```



05/31/72 01:50:55

```
**********************************
**********************************
**                              **
**  PDP-9 MINI TIME-SHARING SYSTEM  **
**  SPECIAL IOT HANDLER OVERLAY #1  **
**            DTSS:SPL          **
**            MTSS:B12          **
**                              **
**********************************
**********************************
```

```
100             .TITLE  SPECIAL IOT INSTRUCTION (EXECUTIVE SERVICE CALL) HANDLER
110             .NAME   SPL--B12
120             .INSRT  DEFINS
100             .IFUND  DEFINS
```

```
5720            .LIST   ON
5730            .END
130             .HEAD   Q
140     *               1610
150     *
160     *       ENTRY TO THE SPECIAL IOT INSTRUCTION (EXECUTIVE CALL) HANDLER IS
170     *       FROM THE SWAPPER WITH:
180     *
190     *            1) THE PROGRAM INTERRUPT SYSTEM TURNED OFF
200     *            2) $DKLOK = 0
210     *            3) $RCORE = CURRENT CORE USER
220     *            4) AC, 10, 11 AND RESTART DATA NOT SAVED BOTH IN THE INTERRUPT SAVE LOCATIONS
230     *               AND IN THE USER JOB TABLE.
240     *            5) CLOCK IS ON
250     *
260     *       IN ORDER TO PERMIT TELETYPE INPUT/OUTPUT TO CONTINUE, THE SPECIAL
270     *       HANDLER RUNS WITH THE FOLLOWING SETTINGS:
280     *            1) $DKLOK IS NEGATIVE
290     *            2) $RCORE = 0
300     *            3) THE PROGRAM INTERRUPT SYSTEM TURNED ON
310     *            4) CLOCK IS ON
320     *            5) SAVE THE AC, ETC INTO THE USER JOB TABLE
330     *
340     *
350     *       IN ALL CASES, EXIT IS TO THE SWAPPER WITH THE FOLLOWING SETTINGS:
360     *            1) $DKLOK IS NEGATIVE
370     *            2) $RCORE IS RESET TO THE VALUE IT HAD ON ENTRANCE
380     *            3) AC, 10, 11, AND RESTART DATA ARE ALSO RESET
390     *            4) PROGRAM INTERRUPT SYSTEM TURNED OFF
400     *            5) CLOCK IS ON
```

```
                Q                       CORE LAYOUT FOR THIS OVERLAY

                    410                 .STITL  CORE LAYOUT FOR THIS OVERLAY
                    420         *
                    430         *
                    440         *       THE LAST 400(8) WORDS OF THE OVERLAY AREA ARE USED BY THIS OVERLAY
                    450         *       AS A BUFFER AREA TO ALLOW READING IN OF A FULL DEVICE CATALOG AT ONCE.
                    460         *
                    470         *       CORE IS ALLOCATED UNDER TWO USE COUNTERS: 'PERM' AND 'OVRLAY'.
                    480         *       'PERM' CONTAINS THE CODE WHICH MUST BE CORE-RESIDENT AT ALL TIMES,
                    490         *       'OVRLAY' CONTAINS THAT CODE WHICH WILL NOT BE NEEDED ANY MORE AFTER
                    500         *       THE CATALOG IS READ IN. IF 'PERM' GETS TOO LONG (I.E. IF IT STARTS
                    510         *       TO INTRUDE INTO THE OVERLAY AREA) AN ASSEMBLY-TIME ERROR MESSAGE
                    520         *       IS GENERATED.
                    530         *
                    540         *       ARRANGE THE USE COUNTERS IN ORDER:
    001000          550                 .LOC    OVSTRT
    001000          560                 .USE    PERM
001000  601447      570                 JMP     SPLST           START THE PROGRAM
    001300          580                 .USE    OVRLAY
                    590
    001300          600         BUFFER  .EQU    1300            START OF THE CATALOG BUFFER AREA
                    610                 .IFG    CHECK,BUFFER    PRINT THE ASSEMBLY ERROR MESSAGE IF 'PERM' IS TOO LONG
```

```
                Q                               MPOFF (705000)      TERMINATE (705001)

                     630                .STITL  MPOFF (705000)      TERMINATE (705001)
          001300     640                .USE    OVRLAY
                     650          *
                     660          *
                     670          *     SPECIAL IOT INSTRUCTIONS ARE THE MEANS BY WHICH A PROGRAM RUNNING
                     680          *     UNDER MTSS CAN CALL UPON THE EXECUTIVE FOR SYSTEM SERVICES. THE
                     690          *     FUNDAMENTAL SPECIAL IOT INSTRUCTION IS 705000. POSSIBLE SPECIALS
                     700          *     RANGE FROM SPECIAL+0 TO SPECIAL+377, ONLY A FEW OF THESE ARE
                     710          *     CURRENTLY ENABLED, LEAVING THIS AS ONE AREA FOR MAJOR FUTURE
                     720          *     SYSTEM EXPANSION.
                     730          *
                     740          *
                     750          *     MPOFF (705000) IS LEGAL ONLY FOR PHANTOM PROGRAMS.
                     760          *     CONTROL IS RETURNED TO THE USER AT THE NEXT INSTRUCTION AFTER THE
                     770          *     SPECIAL, WITH THE STATE OF THE MACHINE UNCHANGED EXCEPT THAT
                     780          *     USER MODE IS DISABLED.
                     790          *     THE PROGRAM ITSELF SHOULD RE-ENABLE USER MODE AS SOON AS POSSIBLE
                     800          *     BY ISSUING AN MPEU (701742) INSTRUCTION TO GUARD AGAINST ITS
                     810          *     OWN  BUGS CRASHING THE SYSTEM.
                     820          *
                     830          *     MPOFF SHOULD BE DISALLOWED AS SOON AS PRACTICABLE BY ADDING ENOUGH
                     840          *     EXECUTIVE SERVICES TO THE SYSTEM TO MAKE IT UNNECESSARY. THIS WILL
                     850          *     GREATLY ENHANCE SYSTEM RELIABILITY.
                     860          *
                     870          *
          001300     880          $P000  ...                        TURN OFF MEMORY PROTECT (MPOFF)
001300    101521     890                 JMS     PHCHK             ONLY PHANTOM PROGRAMS CAN BE ALLOWED TO TURN OFF MEMORY PROTECT.
001301    201776     900                 LAC     $RSTRT            GET THE RESTART
001302    501633     910                 AND     (677777)          REMOVE THE MEMORY PROTECTION BIT
001303    041776     920                 DAC     $RSTRT            REPLACE THE RESTART
001304    140703     930                 DZM     $OC1              NO NEW OVERLAY REQUESTED
001305    140704     940                 DZM     $OC2              NO RESTART OVERRIDE REQUESTED
001306    601077     950                 JMP     SPLDON            EXIT
                     960          *
                     970          *
                     980          *
                     990          *     TERMINATE (705001) IS LEGAL FOR ALL PROGRAMS.
                    1000          *     ITS EFFECT IS EXACTLY THE SAME AS IF A HLT INSTRUCTION WERE ENCOUNTERED
                    1010          *     IN THE RUNNING PROGRAM EXCEPT IT RETURNS CONTROL TO THE MONITOR
                    1020          *     WITHOUT THE ERROR MESSAGE "HALTED AT..." BEING PRINTED. IN EITHER
                    1030          *     CASE, IF THE MONITOR IS REQUESTED TO 'CONTINUE', PROGRAM EXECUTION
                    1040          *     WILL BE RESUMED AT THE NEXT INSTRUCTION WITH REGISTERS UNALTERED.
                    1050          *
                    1060          *
                    1070          *
          001307    1080          $P001  ...                        TERMINATE THE RUN AND RETURN TO THE MONITOR
001307    750000    1090                 CLA                       REQUEST THE MONITOR

001310    601074    1100                 JMP     ERR               WILL NOT PRINT AN ERROR MESSAGE
```

```
Q                       PHYSICAL DISK/DECTAPE READ (705002, 705003) & WRITE (705004, 705005)

                1110    .STITL  PHYSICAL DISK/DECTAPE READ (705002, 705003) & WRITE (705004, 705005)
001311          1120    .USE    OVRLAY
                1130    *
                1140    *
                1150    *       THE DISK AND DECTAPE SPECIALS MAKE USE OF A MODIFIED STANDARD GROWTH SYSTEM
                1160    *       DISK/DECTAPE HANDLER; TO SIMPLIFY MODIFYING STAND-ALONE PROGRAMS
                1170    *       TO RUN UNDER MTSS THE FORMAT USED BY THE SPECIALS IS THE SAME
                1180    *       ONE THAT THE HANDLER NORMALLY USES ANYWAY.
                1190    *
                1200    *       ALL DISK/DECTAPE SPECIALS ARE EXECUTED WITH THE AC CONTAINING
                1210    *       A POINTER TO A LIST OF PARAMETERS OF THE FOLLOWING FORM:
                1220    *            WORD1: BITS 0-2 ARE THE DECTAPE HANDLER NUMBER OR THE
                1230    *                   PHYSICAL DISK NUMBER, AS APPROPRIATE.
                1240    *                   BIT 3 = 0 FOR A DECTAPE OPERATION; = 1 FOR A DISK OPERATION.
                1250    *                   BITS 8-17 CONTAIN THE BLOCK NUMBER FOR THE START OF THE
                1260    *                   DATA TRANSFER.
                1270    *            WORD2: CORE ADDRESS FOR THE START OF THE DATA TRANSFER.
                1280    *            WORD3: WORD COUNT TO BE TRANSFERRED.
                1290    *
                1300    *       THE DISK/DECTAPE SPECIALS PERFORM THE FOLLOWING CHECKS:
                1310    *            1) AN ATTEMPT TO READ OR WRITE OFF THE END OF A DECTAPE OR DISK GENERATES
                1320    *               AN ERROR MESSAGE FOR THE USER.
                1330    *            2) AN ATTEMPT TO TRANSFER DATA TO OR FROM A CORE ADDRESS IN
                1340    *               EXCESS OF 8K GENERATES AN ERROR MESSAGE FOR THE USER.
                1350    *            3) A CORE ADDRESS BELOW THE MEMORY PROTECT BOUNDARY IS LEGAL
                1360    *               ONLY FOR PHANTOM PROGRAMS. IF A USER PROGRAM ATTEMPTS
                1370    *               A DATA TRANSFER TO OR FROM SUCH AN ADDRESS, AN ERROR
                1380    *               MESSAGE IS GENERATED FOR HIM.
                1390    *            4) AN ATTEMPT TO TRANSFER DATA TO/FROM A NON-EXISTANT DISK
                1400    *               GENERATES AN ERROR MESSAGE FOR THE USER.
                1410    *            5) AN ATTEMPT TO TRANSFER DATA TO/FROM A DECTAPE NOT ASSIGNED
                1420    *               TO THE USER GENERATES AN ERROR MESSAGE.
                1430    *            6) AN ATTEMPT BY A USER PROGRAM TO WRITE TO THE PHYSICAL
                1440    *               DISK GENERATES AN ERROR MESSAGE.
                1450    *
                1460    *       THE DISK/DECTAPE SPECIALS PROVIDE ALL PROGRAMS WITH THE CAPABILITIES OF:
                1470    *            1)READING OR WRITING IN A LOGICAL-BLOCK-ADDRESSED FORMAT
                1480    *               THE PROGRAM'S DECTAPES OR "USER "PHYSICAL DISK".
                1490    *            2) READING IN A LOGICAL-BLOCK-ADDRESSED FORMAT THE ACTUAL PHYSICAL DISK.
                1500    *
                1510    *       IN ADDITION, PHANTOM PROGRAMS CAN WRITE IN A LOGICAL-BLOCK-ADDRESSED
                1520    *       FORMAT THE ACTUAL PHYSICAL DISK.
                1530    *
                1540    *       THESE CAPABILITIES ALLOW DEVICE INDEPENDENT PROGRAMMING
                1550    *       WITH RESPECT TO DISK AND DECTAPE.
                1560    *
                1570    *        RETURN OF CONTROL TO THE USER:
                1580    *            1) IF THE DISK/DECTAPE TRANSFER IS SUCCESSFULLY COMPLETED
                1590    *               CONTROL IS RETURNED TO THE USER AT THE ADDRESS THE USER PASSED IN THE MQ.
                1600    *            2) IF A DEVICE ERROR WAS ENCOUNTERED CONTROL IS RETURNED TO
                1610    *               THE USER ONE LOCATION PAST THE SPECIAL.
                1620    *            3) IF A USER SOFTWARE ERROR IS ENCOUNTERED AN ERROR
```

```
Q                       PHYSICAL DISK/DECTAPE READ (705002, 705003) & WRITE (705004, 705005)

                1630    *               MESSAGE IS PRINTED ON HIS TELETYPE AND CONTROL
                1640    *               IS RETURNED TO MONITOR.
                1650    *
                1660    *       SOME POSSIBLE CAUSES OF A "DEVICE ERROR" ARE:
                1670    *           1) A DISK OR DECTAPE HARDWARE MALFUNCTION
                1680    *           2) A DECTAPE CALLED WHICH HAS NOT BEEN REMOTE-ENABLED.
                1690    *           3) A DECTAPE NOT WOUND FAR ENOUGH  ONTO THE SPOOL TO START.
                1700    *
                1710    *
                1720    *
                1730    *
                1740    *       READ (705002) AND WRITE (705004) ARE LEGAL FOR ALL PROGRAMS.
                1750    *       THESE SPECIALS USE THE STANDARD DISK/DECTAPE FORMAT (SEE ABOVE.)
                1760    *       THEY CAUSE THE OPERATION INDICATED BY THEIR PARAMETERS TO BE
                1770    *       ATTEMPTED TO/FROM THE DECTAPE OR "USER PHYSICAL DISK".
                1780    *           1) IF THE READ/WRITE IS TO/FROM DECTAPE, IT IS PASSED ALONG
                1790    *              UNALTERED.
                1800    *           2) IF THE READ/WRITE IS TO/FROM THE DISK, THE BLOCK NUMBER
                1810    *              IS UNDERSTOOD TO REFER TO THE BLOCK DESIRED ON THE USER'S
                1820    *              "PHYSICAL DISK".
                1830    *
                1840    *       PREAD (705003) AND PWRITE (705005) ARE IDENTICAL TO READ AND
                1850    *       WRITE EXCEPT THAT:
                1860    *           1) DISK REFERENCES ARE TO THE ACTUAL PHYSICAL DISK INSTEAD OF
                1870    *              TO THE "USER PHYSICAL DISK".
                1880    *           2) PWRITE IS ILLEGAL FOR USER PROGRAMS
                1890    *
                1900    *
                1910    *
                1920    *
    001311      1930            .USE    OVRLAY
    001311      1940    SP005   ...                     PWRITE
001311  101521  1950            JMS     PHCHK           PWRITE IS LEGAL ONLY FOR PHANTOM PROGRAMS
001312  761255  1960            LAW     WRITE           LOAD A POINTER TO THE WRITE COMMANDS
001313  741000  1970            SKP
    001314      1980    SP003   ...                     PREAD
001314  761257  1990            LAW     READ            LOAD A POINTER TO THE READ COMMANDS
001315  101522  2000            JMS     PARAM           SET UP THE PARAMETERS LIST; DO SOFTWARE ERROR CHECKS
001316  601323  2010            JMP     SP4
                2020
    001317      2030    SP002   ...                     READ
001317  761257  2040            LAW     READ            LOAD A POINTER TO THE READ COMMANDS
001320  741000  2050            SKP
    001321      2060    SP004   ...                     WRITE
001321  761255  2070            LAW     WRITE           LOAD A POINTER TO THE WRITE COMMANDS
001322  101526  2080            JMS     PARAM           SET UP THE PARAMETERS LIST; DO SOFTWARE ERROR CHECKS
                2090
    001323      2100    SP4     ...                     ALL DISK SPECIALS CONVERGE HERE
001323  761117  2110            LAW     TEMP0-1         LOAD A POINTER TO THE PARAMETERS LIST
001324  101130  2120            JMS     DO              DO THE OPERATION
001325  601077  2130            JMP     SPLDON          SOME SORT OF HARDWARE ERROR OCCURRED
001326  201124  2140            LAC     XFER            GOOD RETURN -- LOAD THE USER'S REQUESTED RETURN
```

Q PHYSICAL DISK/DECTAPE READ (705002, 705003) & WRITE (705004, 705005)

```
001327  500651  2150        AND     $ADRSS          MASK TO JUST THE ADDRESS BITS
001330  241264  2160        XOR     TPMSK           PUT THE MEMORY PROTECT BIT IN
001331  041776  2170        DAC     $RSTRT          SET IT FOR THE EXIT ROUTINE
001332  601077  2180        JMP     SPLDON          RETURN TO THE USER
```

```
          Q                                 OPEN (705018)

                    2190                    .STITL  OPEN (705018)
          001333    2200                    .USE    OVRLAY
                    2210    *
                    2220    *
                    2230    *       OPEN (705018) IS LEGAL FOR ALL PROGRAMS.
                    2240    *       THE DISK FILE WHOSE NAME IS PASSED IN THE AC IS LOCATED AND ITS
                    2250    *       PARAMETERS ARE STORED IN THE USER'S JOB TABLE.
                    2260    *
                    2270    *       ON ENTRANCE THE FOLLOWING PARAMETERS ARE PASSED:
                    2280    *               AC: FILENAME TO BE OPENED
                    2290    *            WORD1: OPEN
                    2300    *            WORD2: BITS 0-2 HANDLER NUMBER
                    2310    *                   BIT 3 IS 0 FOR DECTAPE; 1 FOR DISK
                    2320    *
                    2330    *       RETURN TO THE USER IS:
                    2340    *            +1 FOR A HARDWARE ERROR
                    2350    *               TO AN ERROR MESSAGE AND THE MONITOR FOR A SOFTWARE ERROR
                    2360    *            +2 FOR SUCCESS
                    2370    *
                    2380    *       IN ADDITION TO ANY APPLICABLE ERROR MESSAGE WHICH CAN BE CAUSED BY
                    2390    *       A DISK/DECTAPE SPECIAL, OPEN CAN ALSO CAUSE A "FILE NOT FOUND"
                    2400    *       MESSAGE.
                    2410    *
          001333    2420    $0006   ...                     OPEN THE FILE WHOSE NAME WAS PASSED IN THE AC
                    2430    *
                    2440    *       GET THE CATALOG FROM THE DEVICE SPECIFIED
                    2450    *
                    2460    *       FIRST GET THE DEVICE ADDRESS
                    2470    *
001333    760000    2480            LAW     0               LOAD A DEVICE NUMBER/TYPE MASK
001334    521776    2490            AND     SRSTRT,X        GET THE REQUESTED DEVICE NUMBER/TYPE
001335    441776    2500            INX     SRSTRT          CORRECT THE RETURN
001336    041120    2510            DAC     TEMP0           SAVE THE DEVICE NUMBER/TYPE FOR LATER OPERATIONS
001337    041115    2520            DAC     DA              SET THE DEVICE ADDRESS FOR THE CATALOG READ
001340    441115    2530            INX     DA              SET THE DEVICE ADDRESS TO BE THE CATALOG BLOCK
                    2540    *
                    2550    *       READ THE CATALOG
                    2560    *
001341    761114    2570            LAW     DA-1            GET A POINTER TO THE PARAMETERS LIST
001342    601001    2580            JMP     OPEN2
     001001         2590            .USE    PERM
001001    101130    2600    OPEN2   JMS     DO              READ THE CATALOG
001002    601077    2610            JMP     SPLDON          SOME SORT OF HARDWARE ERROR
001003    441776    2620            INX     SRSTRT          GOOD READ -- BUMP THE RETURN
                    2630    *
                    2640    *       NOW FIND THE REQUESTED FILE
                    2650    *
001004    761303    2660            LAW     BUFFER+3        LOAD A POINTER TO THE FIRST FILE CONTROL BLOCK
001005    040010    2670            DAC     10              AND SAVE IT FOR INDEXING
001006    201302    2680            LAC     BUFFER+2        GET THE COUNT OF SAVED FILES
001007    041121    2690            DAC     TEMP1           AND SAVE IT
001010    201753    2700    OPEN4   LAC     SAC             GET THE NAME TO SEARCH FOR
```

```
                Q                               OPEN (705018)

001011  560010  2710            SAD     10,X            CHECK IT AGAINST THE NEXT SAVED FILE'S NAME
001012  601021  2720            JMP     OPEN6           FOUND IT!!!
001013  200010  2730            LAC     10
001014  341634  2740            TAD     (FCBLEN-1)      FAILED -- MOVE THE POINTER TO THE NEXT FILE CONTROL BLOCK
001015  040010  2750            DAC     10
001016  441121  2760            ISZ     TEMP1           COUNT THE FILE JUST CHECKED
001017  601010  2770            JMP     OPEN4           TRY THE NEXT ONE
001020  601064  2780            JMP     ERR11           UTTER FAILURE -- THE FILE IS NOT SAVED
                2790    *
                2800    *       THE REQUESTED FILE HAS BEEN FOUND -- TRANSFER THE FILE CONTROL
                2810    *       BLOCK DATA TO THE JOB TABLE.
                2820    *
001021  220010  2830    OPEN6   LAC     10,X            LOAD THE FILE'S DEVICE ADDRESS
001022  501265  2840            AND     BMSK            RETAIN JUST THE BLOCK NUMBER
001023  241120  2850            XOR     TEMP0           FORM THE CURRENT DEVICE ADDRESS
001024  041700  2860            DAC     SFRDA           SET IT IN THE JOB TABLE -- REFERRING TO ITS CURRENT HANDLER
001025  220010  2870            LAC     10,X
001026  041701  2880            DAC     SFRCA           SET THE FILE'S CORE ADDRESS IN THE JOB TABLE
001027  220010  2890            LAC     10,X
001030  041702  2900            DAC     SFRLEN          SET THE FILE'S LENGTH IN THE JOB TABLE
001031  220010  2910            LAC     10,X
001032  041703  2920            DAC     SFRSTA          SET THE FILE'S TRANSFER ADDRESS IN THE JOB TABLE
001033  601077  2930            JMP     SPLDON          EXIT
```

```
Q                       COPY (705019)

                2940            ,STITL  COPY (705019)
        001343  2950            ,USE    OVRLAY
                2960    *
                2970    *
                2980    *       COPY (705019) IS LEGAL FOR ALL PROGRAMS.
                2990    *       IT PROVIDES CORE-TO-DEVICE AND DEVICE-TO-CORE COPIES TO OR
                3000    *       FROM FILES ON DECTAPE OR ON THE SYSTEM DISK.
                3010    *       ON ENTRANCE, THE PARAMETERS PASSED ARE:
                3020    *               AC: BIT 0 : = 0 FOR DEVICE-TO-CORE COPY
                3030    *                           = 1 FOR CORE-TO-DEVICE COPY
                3040    *                  MQ: BITS 5-17: USER'S DESIRED RESTART ADDRESS
                3050    *            WORD1: COPY
                3060    *            WORD2: BITS 5-17: STARTING CORE ADDRESS FOR THE COPY
                3070    *            WORD3: LENGTH OF THE COPY
                3080    *
                3090    *       DEFINITIONS USED IN THE COPY ROUTINES:
                3100    *            $FRDA: FILE'S DEVICE ADDRESS
                3110    *            $FRCA: FILE'S CORE ADDRESS
                3120    *           $FRLEN: FILE'S LENGTH IN WORDS
                3130    *              FEA: FILE'S END ADDRESS+1
                3140    *       LIKEWISE RCA, RLEN, AND REA ARE USED FOR THE VALUES REQUESTED
                3150    *       BY THE SPECIAL CALL AND CDA, CCA, CLEN, AND CEA ARE USED TO
                3160    *       DESIGNATE THE VALUES DECIDED ON BY THE COPY ROUTINES.
                3170    *
                3180    *
                3190    *       THE INTERSECTION OF THE SAVED FILE (WHICH MUST HAVE BEEN PREVIOUSLY
                3200    *       "OPEN"ED) WITH THE PORTION OF USER CORE INDICATED BY THE REQUESTED
                3210    *       CORE ADDRESS AND LENGTH WILL BE COPIED.
                3220    *
                3230    *       THE COPY VALUES ARE DECIDED AS FOLLOWS:
                3240    *            1) CCA = GREATER ($FRCA,RCA)
                3250    *            2) CEA = LESSER (FEA,REA)
                3260    *            3) CLEN = CEA-CCA (CLEN > 0 ELSE ERROR MESSAGE IS PRINTED)
                3270    *            4) STOFF = CCA - $FRCA IS START ADDRESS OFFSET
                3280    *            5) SOB = INTEGER (STOFF/400) IS STOFF IN BLOCKS
                3290    *            6) SOW = REMAINDER (STOFF/400) IS STOFF - SOB (0 <= SOW <=377)
                3300    *                     THIS IS THE NUMBER OF WORDS THE START IS PAST
                3310    *                     AN EVEN BLOCK BOUNDARY ON THE FILE'S DEVICE
                3320    *            7) CDA = $FRDA + SOB IS THE FIRST BLOCK BOUNDARY BEFORE THE
                3330    *                     DESIRED STARTING WORD
                3340    *
                3350    *       CONTROL IS RETURNED TO THE USER AFTER A SUCCESFUL COPY AT THE
                3360    *       USER-SPECIFIED RESTART ADDRESS. THIS ALLOWS A 100% OVERLAY.
                3370    *
                3380    *       AN ERROR MESSAGE IS PRINTED AND CONTROL IS RETURNED TO MONITOR
                3390    *       IF FOR ANY REASON THE COPY WAS UNSUCCESSFUL.
                3400    *       THIS IS BECAUSE THAT IS WHAT SHOULD HAPPEN FOR A SOFTWARE ERROR
                3410    *       ON THE PART OF THE USER. IF THE ERROR WAS A HARDWARE ERROR, IT IS
                3420    *       PROBABLY UNRECOVERABLE, ANYWAY.
                3430    *
                3440    *
                3450    *       THE DEVICE-TO-CORE COPY ALGORITHM IS:
```

Q COPY (705019)

```
                    3460    *
                    3470    *           1) IF SOW = 0 GOTO 5, SINCE THERE ARE NO ODD WORDS TO COPY
                    3480    *           2) COPY FROM CDA TO BUFFER FOR 400 WORDS (ONE BLOCK)
                    3490    *           3) CORE-COPY FROM (BUFFER+SOW) TO CCA FOR (400-SOW) WORDS
                    3500    *              CDA := CDA + 1
                    3510    *              CCA := CCA + 400 - SOW
                    3520    *             CLEN := CLEN - SOW
                    3530    *           5) IF CLEN <= 0 THEN DONE
                    3540    *           6) COPY FROM CDA TO CCA FOR CLEN WORDS
                    3550    *           7) DONE
                    3560    *
                    3570    *
                    3580    *       THE CORE-TO-DEVICE COPY IS NOT YET IMPLIMENTED
                    3590    *
                    3600    *
                    3610    *       INITIALIZE THE COPY
    001120          3620    SOB     ,EQU    TEMP0
    001120          3630    CDA     ,EQU    TEMP0
    001121          3640    RCA     ,EQU    TEMP1
    001121          3650    CCA     ,EQU    TEMP1
    001122          3660    RLEN    ,EQU    TEMP2
    001122          3670    CLEN    ,EQU    TEMP2
    001123          3680    STOFF   ,EQU    TEMP3
    001123          3690    SOW     ,EQU    TEMP3
    001343          3700    SP007   ,,,                     COPY
901343  141124      3710            DZM     XFER            INITIALIZE THE COPY DIRECTION FLAG
901344  777777      3720            LAW     -1
901345  341776      3730            TAD     SRSTRT          ADD THE SPECIAL'S ADDRESS
901346  040010      3740            DAC     10              SET A POINTER TO THE COPY PARAMETERS
901347  501635      3750            AND     (700000)        RETAIN JUST THE USER'S MACHINE STATE
901350  041776      3760            DAC     SRSTRT          AND SAVE IT FOR NOW
901351  201753      3770            LAC     SAC             LOAD THE USER'S DESIRED RESTART ADDRESS
901352  741100      3780            SPA                     SKIP IF A DEVICE-TO-CORE COPY IS REQUESTED
901353  441124      3790            INX     XFER            ELSE FLAG A CORE-TO-DEVICE COPY
901354  501636      3800            AND     (077777)        RETAIN JUST THE ADDRESS BITS
901355  241776      3810            XOR     SRSTRT          ADD IN THE PREVIOUS MACHINE STATE
901356  041776      3820            DAC     SRSTRT          SAVE THE CORRECTED USER RESTART DATA
                    3830    *
                    3840    *       SET UP THE REQUESTED CORE ADDRESS AND LENGTH
                    3850    *
901357  220010      3860            LAC     10,X
901360  041121      3870            DAC     RCA             SET THE REQUESTED CORE ADDRESS
901361  041123      3880            DAC     TEMP3           AND SAVE IT FOR LATER CHECKS
901362  220010      3890            LAC     10,X
901363  041122      3900            DAC     RLEN            SET THE REQUESTED LENGTH
                    3910    *
                    3920    *       THE COPY WILL ACTUALLY BE DONE FROM GREATER ($FRCA,RCA) TO
                    3930    *       LESSER (FEA,REA). AN ERROR MESSAGE WILL BE GENERATED IF THIS
                    3940    *       RESULTS IN A NEGATIVE OR ZERO LENGTH COPY.
                    3950    *
                    3960    *       SET UP THE ACTUAL COPY START
                    3970    *
```

```
          Q                              COPY (705019)

001364  777777  3980            LAW     -1
001365  341121  3990            TAD     RCA
001366  740001  4000            CMA                     AC = TWO'S COMPLEMENT OF THE RCA
001367  341701  4010            TAD     $FRCA
001370  741100  4020            SPA                     SKIP IF THE COPY START NEEDS TO BE OVER RIDDEN
001371  601374  4030            JMP     COPY2           ELSE GO ON TO THE NEXT CHECK
001372  201701  4040            LAC     $FRCA
001373  041121  4050            DAC     CCA             $FRCA > RCA, SO USE $FRCA FOR A STARTING CORE ADDRESS
                4060    *
                4070    *       SET UP THE ACTUAL COPY LENGTH
                4080    *
001374  777777  4090    COPY2   LAW     -1
001375  341123  4100            TAD     TEMP3           ADD THE REQUESTED START ADDRESS
001376  341122  4110            TAD     RLEN            ADD THE REQUESTED LENGTH TO GET THE REQUESTED END ADDRESS
001377  740001  4120            CMA                     AC = TWO'S COMPLEMENT OF THE REA
001400  341701  4130            TAD     $FRCA
001401  341702  4140            TAD     $FRLEN          ADD THE FEA
001402  740100  4150            SMA                     SKIP IF THE COPY LENGTH NEEDS TO BE CHANGED
001403  601414  4160            JMP     COPY4           ELSE GO ON TO FURTHER CHECKS
                4170    *
                4180    *       FEA < REA, SO COMPUTE CLEN, WHICH MAY BE LESS THAN EITHER
                4190    *       $FRLEN OR RLEN.
                4200    *
001404  777777  4210            LAW     -1
001405  341121  4220            TAD     CCA             ADD THE ACTUAL CORE ADDRESS OF THE COPY
001406  740001  4230            CMA                     AC = TWO'S COMPLEMENT OF CCA
001407  341701  4240            TAD     $FRCA
001410  341702  4250            TAD     $FRLEN          ADD FEA -- AC = CLEN
001411  741300  4260            SNA!SPA                 SKIP IF THE COPY LENGTH IS LEGAL
001412  601073  4270            JMP     ERR3            ELSE GO PRINT AN ERROR MESSAGE
001413  041122  4280            DAC     CLEN            SET THE LENGTH TO ACTUALLY BE COPIED
                4290    *
                4300    *       SET THE STARTING ADDRESS OFFSET CONSTANTS
                4310    *
001414  777777  4320    COPY4   LAW     -1
001415  341701  4330            TAD     $FRCA
001416  740001  4340            CMA                     AC = TWO'S COMPLEMENT OF $FRCA
001417  341121  4350            TAD     CCA             ADD THE ACTUAL CORE ADDRESS OF THE COPY
001420  041123  4360            DAC     STOFF           SAVE THE ENTIRE STARTING ADDRESS OFFSET
001421  744000  4370            CLL                     PREPARE FOR THE SHIFT
001422  640510  4380            LRS     8.              DIVIDE BY 400
001423  041120  4390            DAC     SOB             SAVE THE NUMBER OF WHOLE BLOCKS IN THE OFFSET
001424  201123  4400            LAC     STOFF
001425  501637  4410            AND     (377)           MODULO 400
001426  041123  4420            DAC     SOW             SAVE THE NUMNER OF WORDS OF OFFSET PAST A BLOCK BOUNDARY
                4430    *
                4440    *       CALCULATE THE STARTING DEVICE ADDRESS
                4450    *
001427  201700  4460            LAC     $FRDA           LOAD THE FILE'S ACTUAL DEVICE ADDRESS
001430  341120  4470            TAD     SOB             ADD THE NUMBER OF BLOCKS OF OFFSET
001431  041120  4480            DAC     CDA             SET THE DEVICE ADDRESS OF THE START OF THE COPY
001432  041115  4490            DAC     DA              SET THE CDA FOR A (1) BLOCK READ IN CASE IT IS NEEDED
```

Q COPY (705019)

```
                    4500    *
                    4510    *       NOW UPDATE ALL POINTERS TO WHAT THEY SHOULD BE AFTER THE FIRST BLOCK IS READ
                    4520    *
001433  441120      4530            INX     CDA             COUNT THE BLOCK JUST READ
001434  777777      4540            LAW     -1
001435  341123      4550            TAD     SOW
001436  740001      4560            CMA                     AC = TWO'S COMPLEMENT OF SOW
001437  341122      4570            TAD     CLEN
001440  041122      4580            DAC     CLEN            UPDATE THE LENGTH BY THE AMOUNT JUST COPIED
001441  101627      4590            JMS     PARAM2          CHECK THE PARAMETERS BEFORE TRYING THE TRANSFER
                    4600    *
                    4610    *       THE COPY IS LEGAL -- CHECK TO SEE WHETHER OR NOT IT STARTS FROM A BLOCK BOUNDARY
                    4620    *
001442  201123      4630            LAC     SOW             LOAD THE WORD OFFSET
001443  741200      4640            SNA                     SKIP IF THERE IS ONE
001444  601055      4650            JMP     COPY6           ELSE THE COPY STARTS FROM A BLOCK BOUNDARY
                    4660    *
                    4670    *       THE COPY DOES NOT START AT A BLOCK BOUNDARY, READ A BLOCK INTO
                    4680    *       OUR BUFFER SO THE JUNK CAN BE DELETED.
                    4690    *
001445  761115      4700            LAW     DA              LOAD THE POINTER TO THE READ PARAMETERS
001446  601034      4710            JMP     COPY7
    001034          4720            .USE    PERM
001034  101130      4730    COPY7   JMS     DO              AND READ THE FIRST BLOCK OF THE COPY
001035  601073      4740            JMP     ERR3            SOME SORT OF A HARDWARE ERROR
                    4750    *
                    4760    *       WE NOW HAVE IN OUR CORE THE ODD WORDS AND SOME GARBAGE, TOO.
                    4770    *       COPY THE GOOD WORDS INTO THE USER CORE
                    4780    *
001036  777777      4790            LAW     -1
001037  341116      4800            TAD     BUFADD          ADD THE START ADDRESS OF THE BUFFER
001040  341123      4810            TAD     SOW             ADD THE WORD OFFSET
001041  040010      4820            DAC     10              SET THE POINTER TO THE FIRST GOOD WORD TO BE COPIED
001042  777400      4830            LAW     -400
001043  341123      4840            TAD     SOW             ADD THE WORD OFFSET
001044  041115      4850            DAC     DA              SET THE TOTAL NUMBER OF WORDS TO TRANSFER
001045  777777      4860            LAW     -1
001046  341121      4870            TAD     CCA
001047  040011      4880            DAC     11              SET THE START OF THE USER CORE TO TRANSFER TO
001050  220010      4890    COPY8   LAC     10,X
001051  060011      4900            DAC     11,X            COPY THE NEXT GOOD WORD
001052  441121      4910            INX     CCA             BUMP THE CORE ADDRESS POINTER
001053  441115      4920            ISZ     DA              AND COUNT THE WORD
001054  601050      4930            JMP     COPY8           NEXT ...
                    4940    *
                    4950    *       SEE IF THERE IS STILL ANY COPYING TO DO
                    4960    *
    001055          4970    COPY6   ...
001055  201122      4980            LAC     CLEN
001056  741300      4990            SNA!SPA                 SKIP IF THERE IS STILL COPYING TO DO
001057  601063      5000            JMP     COPYD           ELSE DONE
                    5010    *
```

Q COPY (705019)

```
                    5020    *       THERE IS COPYING TO DO -- THE CORRECT PARAMETERS ARE ALREADY IN
                    5030    *       CDA, CCA, & CLEN, -- SO START IT UP
                    5040    *
001060  761120      5050            LAW     CDA             LOAD A POINTER TO THE PARAMETERS
001061  101130      5060            JMS     DO              DO THE REST OF THE COPY
001062  601073      5070            JMP     ERR3            SOME SORT OF HARDWARE ERROR
                    5080    *
                    5090    *       DONE
                    5100    *
001063  601077      5110    COPYD   JMP     SPLDON
```

```
            Q                       MAIN PROGRAM

                    5120            .STITL  MAIN PROGRAM
            001447  5130            .USE    OVRLAY
                    5140    *
                    5150    *
                    5160    *       PROTECT OURSELVES FROM RESIDENT PROGRAM INTERFERENCE AND TURN THE
                    5170    *       PROGRAM INTERRUPT SYSTEM BACK ON, NOTE THAT THE THINGS THE PROGRAM INTERRUPT
                    5180    *       SYSTEM MAY CLOBBER ARE ALL STILL SAVED IN THE USER JOB TABLE.
                    5190    *
            001447  5200    SPLST   ...
001447  641002      5210            LACQ
001450  041124      5220            DAC     XFER            SAVE THE USER'S REQUESTED RESTART FROM A DISK/DECTAPE OPERATION
001451  200005      5230            LAC     $3AC
001452  041753      5240            DAC     $AC             SAVE THE USER'S AC
001453  200026      5250            LAC     $.310
001454  041723      5260            DAC     $.0+10          SAVE THE USER'S REGISTER 10
001455  200027      5270            LAC     $.311
001456  041724      5280            DAC     $.0+11          SAVE THE USER'S AUTO-INDEX REGISTER 11
001457  201776      5290            LAC     $RSTRT          LOAD THE PC SO THAT IT CAN BE SAVED FOR THE CALLER
001460  040702      5300            DAC     $OC0            PASS THE OLD PC  BACK TO THE CALLER
001461  200000      5310            LAC     0
001462  041776      5320            DAC     $RSTRT          SAVE THE USER'S EXTENDED PC FOR RESTART
001463  750001      5330            CLC
001464  040266      5340            DAC     $DKLOK          TIE UP THE DISK TO INHIBIT INTERRUPTS FROM AFFECTING US
001465  200035      5350            LAC     $RCORE
001466  140035      5360            DZM     $RCORE          TELL THE RESIDENT PROGRAM THERE IS NO MEMORY PROTECTION OVERLAY IN CORE
001467  041125      5370            DAC     CCU             AND SAVE THE CURRENT CORE USER NAME
001470  700042      5380            ION                     AT LAST IT IS SAFE TO TURN THE INTERRUPT SYSTEM BACK ON
                    5390    *
                    5400    *       DECIPHER THE TYPE OF SPECIAL IOT GIVEN <0:377>. CHECK THAT IT DOESN'T
                    5410    *       EXCEED THE MAXIMUM PERMISSIBLE (-SPMAX), WHICH WOULD RESULT IN AN UNDEFINED
                    5420    *       TRANSFER. IF LEGAL, TRANSFER THROUGH THE SPECIAL TABLE, ELSE PRINT AN ERROR MESSAGE.
                    5430    *
001471  777777      5440            LAW     -1
001472  341776      5450            TAD     $RSTRT
001473  041120      5460            DAC     TEMP0           SAVE THE ADDRESS OF THE SPECIAL
001474  500651      5470            AND     $ADRSS
001475  041711      5480            DAC     $UTEM5          SAVE JUST THE ADDRESS IN CASE OF ILLEGAL CALL
001476  221120      5490            LAC     TEMP0,X         NOW RECOVER THE SPECIAL
001477  501637      5500            AND     ($SPCOD)        RECOVER THE SPECIAL CODE
001500  041121      5510            DAC     TEMP1
001501  341520      5520            TAD     SPMAX           CHECK FOR LEGALITY
001502  750300      5530    SPL1    SMA!SZA!CLA
001503  601073      5540            JMP     ERR3            ILLEGAL SPECIAL CALL
001504  201121      5550            LAC     TEMP1           RELOAD THE CODE
001505  341640      5560            TAD     (SPTABL)
001506  041122      5570            DAC     TEMP2           SET UP THE TRANSFER
001507  621122      5580            JMP     TEMP2,X
                    5590    *
                    5600    *
                    5610    *       TRANSFER TABLE FOR LEGAL SPECIAL IOT CODES
                    5620    *
001510  601300      5630    SPTABL  JMP     SP000           MPOFF
```

Q MAIN PROGRAM

```
001511  601307  5640          JMP     SP001           TERMINATE
001512  601317  5650          JMP     SP002           READ
001513  601314  5660          JMP     SP003           PREAD
001514  601321  5670          JMP     SP004           WRITE
001515  601311  5680          JMP     SP005           PWRITE
001516  601333  5690          JMP     S0006           OPEN
001517  601343  5700          JMP     SP007           COPY
001520  777771  5710  SPMAX   SPTABL-.+1              MINUS THE GREATEST LEGAL SPECIAL NUMBER
```

```
            Q                       MISCELLANEOUS SUBROUTINES

                 5720               .STITL  MISCELLANEOUS SUBROUTINES
     001521      5730               .USE    OVRLAY
                 5740    *
                 5750    *
                 5760    *       CHECK TO SEE IF THE CALLER IS A PHANTOM PROGRAM, IF SO, RETURN
                 5770    *       NORMALLY, OTHERWISE FALL THROUGH AND PRINT AN ERROR MESSAGE ABOUT
                 5780    *       HIS ILLEGAL SPECIAL IOT INSTRUCTION.
                 5790    *
     001521      5800    PHCHK   ENTER
                                 .PMC    SAVE,ON
001521  740040                   XX
001522  201774  5810             LAC     $TYPE           FIND OUT WHAT TYPE OF PROGRAM THE USER IS
001523  740200  5820             SZA                     SKIP IF HE IS A USER PROGRAM
001524  621521  5830             RET     PHCHK,X         RETURN NORMALLY IF HE IS A PHANTOM PROGRAM
001525  601073  5840             JMP     ERR3            ELSE THE SPECIAL CALL WAS ILLEGAL
     001064     5850             .USE    PERM
                 5860    *
                 5870    *
                 5880    *       SET UP TO PRINT THE APPROPRIATE ERROR MESSAGE FOR THE USER
                 5890    *
001064  760003  5900    ERR11   LAW     3
001065  741000  5910             SKP
001066  760001  5920    ERR9    LAW     1
001067  341641  5930    ERR8    TAD     (1)
001070  341641  5940    ERR7    TAD     (1)
001071  341642  5950    ERR6    TAD     (6)
001072  741000  5960             SKP
001073  760003  5970    ERR3    LAW     3
001074  041706  5980    ERR      DAC     $UTEM2          SET THE MESSAGE NUMBER
001075  761004  5990             LAW     $SWERR          LOAD THE SWAPPER ENTRANCE TO GET MONITOR/SYSTEM MESSAGES
001076  741000  6000             SKP
                 6010    *
                 6020    *
                 6030    *       RESET THE NECESSARY REGISTERS BEFORE RETURNING TO THE SWAPPER
                 6040    *
                 6050    *
     001077      6060    SPLDON  ...
001077  761006  6070             LAW     $SXSPL          LOAD THE SWAPPER ENTRANCE TO EXIT THE SPECIAL HANDLER
001100  040654  6080             DAC     $DO             SET THE SWAPPER ENTRANCE
001101  700002  6090             IOF                     SWAPPER MUST BE ENTERED WITH THE INTERRUPT SYSTEM OFF
001102  201125  6100             LAC     CCU
001103  040035  6110             DAC     $RCORE          RESTORE THE CURRENT CORE USER'S NAME
001104  201753  6120             LAC     $AC
001105  040005  6130             DAC     $3AC            RESET THE USER'S AC
001106  201723  6140             LAC     $,0+10
001107  040026  6150             DAC     $,310           RESET THE USER'S LOCATION 10
001110  201724  6160             LAC     $,0+11
001111  040027  6170             DAC     $,311           RESET THE USER'S LOCATION 11
001112  201776  6180             LAC     $RSTRT
001113  040000  6190             DAC     0               RESET THE USER'S RESTART DATA
001114  600336  6200             JMP     $SWAP1          GET THE SWAPPER
     001526     6210             .USE    OVRLAY
```

```
          Q                         DISK/DECTAPE PARAMETER CHECKING

                    6220            .STITL  DISK/DECTAPE PARAMETER CHECKING
                    6230    *
                    6240    *       PARAMETER SETUP AND CHECKING FOR THE DISK/DECTAPE HANDLER IS
                    6250    *       DIVORCED FROM THE ACTUAL HANDLER SO THAT THE CODE FOR CHECKING CAN
                    6260    *       BE PUT INTO THE BUFFER AREA. THE HANDLER ITSELF MUST BE IN THE
                    6270    *       PERMANENT AREA.
                    6280    *
    001526          6290    PARAM   ENTER                   THIS ENTRANCE WILL RESTRICT USERS TO THEIR "PHYSICAL DISK"
                                    .PMC    SAVE,ON
001526  740040                      XX
001527  041263  6300               DAC     CMND            SET THE COMMAND POINTER
001530  201767  6310               LAC     $UDISK          LOAD THE STARTING BLOCK OF HIS "PHYSICAL DISK"
001531  041126  6320               DAC     DKMIN           RESET THE DISK BASE ADDRESS
001532  341643  6330               TAD     ($DKLENB)       ADD THE LENGTH (IN BLOCKS) OF HIS "PHYSICAL DISK"
001533  341144  6340               TAD     M1              SUBTRACT 1
001534  740001  6350               CMA                     AC = TWO'S COMPLEMENT OF MAXIMUM BLOCK NUMBER
001535  057601  6360               DAC     DKMAX           SET IT FOR THE HANDLER
                    6370
    001536          6380    PAR2    ...                     COPY THE USER'S PARAMETERS
001536  777777  6390               LAW     -1              LOAD (-1)
001537  341753  6400               TAD     $AC             ADD THE USER'S PARAMETER POINTER
001540  040010  6410               DAC     10              SET AN AUTO-INDEX POINTER TO THE USER'S PARAMETERS
001541  220010  6420               LAC     10,X
001542  041120  6430               DAC     TEMP0           SET THE USER'S DEVICE ADDRESS
001543  220010  6440               LAC     10,X
001544  041121  6450               DAC     TEMP1           SET THE USER'S CORE ADDRESS
001545  220010  6460               LAC     10,X
001546  041122  6470               DAC     TEMP2           SET THE USER'S WORD COUNT
                    6480    *
                    6490    *       NOW THE USER'S PARAMETERS ARE SET UP IN TEMP0-TEMP2, NEXT DO A
                    6500    *       SERIES OF TESTS FOR PARAMETER ERRORS. IF ONE IS FOUND, PRINT AN
                    6510    *       ERROR MESSAGE, OTHERWISE RETURN TO THE CALLER. THE DISK/DECTAPE
                    6520    *       WILL DO NO FURTHER ERROR CHECKING
                    6530    *
    001547          6540    PAR4    ...                     SEPARATE DISK AND DECTAPE FOR LEGAL HANDLER CHECKS
001547  201120  6550               LAC     CDA             LOAD THE DEVICE ADDRESS
001550  640603  6560               LLS     3               MOVE THE DEVICE TYPE BIT TO THE SIGN BIT
001551  740100  6570               SMA                     SKIP FOR DISK
001552  601560  6580               JMP     PART            ELSE IT IS A TAPE OPERATION
                    6590    *
                    6600    *       THE DISK NUMBER IS PHYSICAL, WE ONLY HAVE PHYSICAL DISK 0.
                    6610    *       ANY OTHER DISK NUMBER IS A PARAMETER ERROR; NON-EXISTANT DISK REFERENCE.
                    6620    *
001553  201120  6630               LAC     CDA             RELOAD THE DEVICE ADDRESS
001554  501264  6640               AND     TPMSK           RECOVER JUST THE HANDLER NUMBER
001555  750200  6650               SZA!CLA                 SKIP IF LEGAL
001556  601067  6660               JMP     ERR8            ELSE PARAMETER ERROR: NON-EXISTANT DISK REFERENCE
001557  601575  6670               JMP     PAR5            CONTINUE
                    6680    *
                    6690    *       THE DECTAPE HANDLER MUST HAVE BEEN ASSIGNED TO THIS USER TO
                    6700    *       BE LEGAL. NO DISTINCTION IS MADE BETWEEN A HANDLER NOT YET ASSIGNED
                    6710    *       TO ANYONE AND ONE ASSIGNED TO SOMEONE ELSE.
```

Q DISK/DECTAPE PARAMETER CHECKING

```
                    6720    *
    001560          6730    PART    ...                     CHECK FOR AN UNASSIGNED DECTAPE HANDLER
001560  201120      6740            LAC     CDA             RELOAD THE DEVICE ADDRESS
001561  501264      6750            AND     TPMSK           RECOVER JUST THE HANDLER NUMBER
001562  744020      6760            CLL!RAR
001563  742020      6770            RTR                     MOVE THE HANDLER NUMBER TO THE PROPER POSITION
001564  341771      6780            TAD     $NUMBR          FORM THE DECTAPE ALLOCATION TAG
001565  540032      6790            SAD     $RDT0           CHECK THE FIRST HANDLER
001566  601572      6800            JMP     PAR6            OK -- MATCH FOUND -- CONTINUE
001567  540033      6810            SAD     $RDT1           CHECK THE OTHER HANDLER
001570  741000      6820            SKP                     OK -- MATCH FOUND -- CONTINUE
001571  601066      6830            JMP     ERR9            ELSE PARAMETER ERROR: DEVICE NOT ASSIGNED TO THIS USER
001572  141126      6840    PAR6    DZM     DKMIN           DECTAPE IS LEGAL FROM BLOCK 0
001573  776700      6850            LAW     -1100
001574  041127      6860            DAC     TDMAX           SET THE MAXIMUM LEGAL DECTAPE BLOCK
                    6870    *
                    6880    *       CHECK FOR AN ATTEMPT TO TRANSFER PAST THE HIGH END OF A DECTAPE
                    6890    *       OR A LOGICAL DISK -- EITHER SYSTEM LOGICAL DISK OR
                    6900    *       "USER PHYSICAL DISK"
                    6910    *
    001575          6920    PAR5    ...                     DEVICE NUMBER/TYPE HAS BEEN FOUND LEGAL
001575  744000      6930            CLL
001576  201122      6940            LAC     CLEN            LOAD THE LENGTH OF THE TRANSFER
001577  650510      6950            CLQ!LRS 8.              DIVIDE BY 400 TO GET NUMBER OF BLOCKS
001600  041130      6960            DAC     D0              SAVE IT FOR FURTHER CHECKS
001601  201120      6970            LAC     CDA             RELOAD THE DEVICE ADDRESS
001602  501265      6980            AND     BMSK            RETAIN JUST THE STARTING BLOCK NUMBER
001603  341126      6990            TAD     DKMIN           ADD THE DEVICE BASE TO GET THE STARTING BLOCK
001604  341130      7000            TAD     D0              ADD IN THE NUMBER OF BLOCKS ASKED FOR TO GET THE MAXIMUM ADDRESS REFERENCED
001605  341127      7010            TAD     TDMAX           SUBTRACT OFF THE MAXIMUM LEGAL BLOCK
001606  750100      7020            SMA!CLA                 SKIP IF THE TRANSFER IS LEGAL
001607  601071      7030            JMP     ERR6            ELSE PARAMETER ERROR: ATTEMPT TO TRANSFER DATA OVER THE END OF THE DEVICE
                    7040    *
                    7050    *       CHECK FOR AN ATTEMPT TO TRANSFER DATA TO/FROM BELOW THE MEMORY PROTECTION BOUNDARY
                    7060    *
001610  776001      7070            LAW     -BOUNDARY+1     SUBTRACT THE BOUNDARY
001611  341121      7080            TAD     CCA             FROM THE DESIRED CORE ADDRESS
001612  741300      7090            SNA!SPA                 SKIP IF THE ADDRESS IS INDEED ABOVE THE BOUNDARY
001613  101521      7100            JMS     PHCHK           ELSE CHECK TO SEE IF THE USER IS A PHANTOM (THEN IT IS LEGAL)
                    7110    *
                    7120    *       EITHER THE TRANSFER WAS TO/FROM ABOVE THE BOUNDARY, OR ELSE THE
                    7130    *       USER IS A PHANTOM PROGRAM
                    7140    *
                    7150    *       NOW CHECK FOR AN ATTEMPT TO TRANSFER DATA OVER THE END OF CORE --
                    7160    *       THIS WOULD WRAP CORE, AND ON A WRITE COULD DESTROY THE EXECUTIVE.
                    7170    *
001614  762000      7180            LAW     -CORMAX         SUBTRACT THE END OF CORE
001615  341121      7190            TAD     CCA             FROM THE DESIRED CORE ADDRESS
001616  341122      7200            TAD     CLEN            PLUS THE DESIRED LENGTH
001617  750100      7210            SMA!CLA                 SKIP IF OK
001620  601070      7220            JMP     ERR7            ELSE IS  PARAMETER ERROR: TRANSFER OF DATA PAST CORE MAX
001621  621526      7230            RET     PARAM,X         EXIT
```

Q DISK/DECTAPE PARAMETER CHECKING

```
                    7240    *
                    7250    *       SET UP THE PARAMETERS FOR THE DISK/DECTAPE HANDLER ALLOWING
                    7260    *       DISK OPERATIONS ON THE WHOLE  DISK
                    7270    *
     001622         7280    PARAM1  ENTER
                                    .PMC    SAVE,ON
001622  740040                      XX
001623  041263     7290             DAC     CMND            SET THE COMMAND POINTER
001624  201622     7300             LAC     PARAM1
001625  041526     7310             DAC     PARAM           SET UP THE EXIT
001626  601536     7320             JMP     PAR2            FROM HERE, THE ROUTINES ARE IDENTICAL
                    7330    *
                    7340    *       CHECK THE PARAMETERS BEFORE A DISK/DECTAPE OPERATION TO THE BUFFER
                    7350    *
     001627         7360    PARAM2  ENTER
                                    .PMC    SAVE,ON
001627  740040                      XX
001630  201627     7370             LAC     PARAM2
001631  041526     7380             DAC     PARAM           SET UP THE EXIT
001632  601547     7390             JMP     PAR4            GO CHECK THE PARAMETERS
                    7400
                    7410
```

Q DISK/DECTAPE PARAMETER CHECKING

```
                    7420            .EJECT
          001115    7430            .USE    PERM
                    7440    *
                    7450    *
                    7460    *       MISCELLANEOUS STORAGE
                    7470    *
001115  000000      7480    DA      .DSA                    DEVICE ADDRESS FOR A DISK/DECTAPE TRANSFER TO THE BUFFER BLOCK
001116  001300      7490    BUFADD  BUFFER                  BUFFER CORE ADDRESS
001117  000400      7500            400
001120  000000      7510    TEMP0   .DSA
001121  000000      7520    TEMP1   .DSA
001122  000000      7530    TEMP2   .DSA
001123  000000      7540    TEMP3   .DSA
001124  000000      7550    XFER    .DSA
001125  000000      7560    CCU     .DSA
001126  000000      7570    DKMIN   0                       MINIMUM DISK ADDRESS IS ZERO
001127  776001      7580    TDMAX   -1777                   MINUS THE MAXIMUM DISK ADDRESS
                    7590            .INSRT  INSERT!GRODEFIN
                    100             .IFUND  $DEBUG
                    1250            .LIST   ON
                    1260            .END
```

Q DECTAPE AND DISK SUBROUTINES

```
                7600            .STITL  DECTAPE AND DISK SUBROUTINES
                7610    *
                7620    *       PROGRAMMED BY JAMES CRUCE '72
                7630    *                     WARREN MONTGOMERY '73
                7640    *
                7650    *       THESE SUBROUTINES ALLOW A PDP-9 PROGRAM TO BE ABLE TO READ OR WRITE
                7660    *       TO DECTAPE OR TO THE RS09 DISK BY SIMPLY DOING A JMS TO ONE OF
                7670    *       THE ENTRANCES WITH THE AC POINTING TO A LIST OF 3 PARAMETERS,
                7680    *       THE TWO ENTRANCES ARE 'DECIN' AND 'DECOT' WHICH ARE USED TO
                7690    *       READ INTO CORE OR WRITE OUT CORE RESPECTIVELY.
                7700    *       IF THE SUBROUTINES DETECT AN ERROR WHILE THEY
                7710    *       ARE EITHER WRITING OR READING THEY WILL HALT THE OPERATION IN PROGRESSS
                7720    *       AND PLACE THE STATUS IN THE AC AND TRANSFER TO A HLT AT 17504,
                7730    *
                7740    *       PARAMETER LIST
                7750    *       ==============
                7760    *            THE PARAMETER LIST CONSISTS OF 3 CONSECUTIVE WORDS.
                7770    *       THE LIST WOULD LOOK SOMETHING LIKE THIS:
                7780    *
                7790    *       WORD 1----TAPE OR DISK HANDLER NUMBER, BLOCK NUMBER
                7800    *       WORD 2----CORE ADDRESS TO START WRITING TO/FROM
                7810    *       WORD 3----LENGTH TO BE WRITTEN
                7820    *
                7830    *       THE LAST TWO ARE SELF EXPLANATORY BUT THE FIRST WILL REQUIRE A LITTLE
                7840    *       BIT MORE EXPLANATION.  BITS 0-2 ARE THE HANDLER OR DISK NUMBER.
                7850    *       BIT 3 SHOULD BE 1 IF THE PROGRAM WANTS TO DO I/O WITH THE DISK
                7860    *       AND IT SHOULD BE 0 IF THE PROGRAM WANTS TO USE THE DECTAPE.
                7870    *       BITS 8-17 ARE USED TO DETERMINE THE BLOCK NUMBER TO BEGIN THE READING
                7880    *       OR WRITING AT.
                7890    *
                7900    *            THE PROGRAM WILL ONLY BE ABLE TO WRITE TO A SELECTED PORTION OF THE
                7910    *       DISK BUT WILL BE ABLE TO READ ANYWHERE ON THE DISK THAT IS PAST
                7920    *       THE PLACEMENT OF THE PROGRAM'S BLOCK 0.  THE TWO PARAMETERS TO MODIFY
                7930    *       TO GAIN ACCESS TO A SPECIFIC PORTION OF THE DISK 'HPAR' WHICH
                7940    *       GIVES THE HIGHEST BLOCK NUMBER THAT THE PROGRAM WILL BE ABLE TO WRITE.
                7950    *       THE ADDRESS GIVEN IN 'DSKAD' DEFINES WHERE BLOCK 0 WILL BE ON THE
                7960    *       DISK AND ALL OF THE REST OF THE BLOCKS ARE IN RELATION TO THIS.
                7970    *
                7980    *
                7990    *
000030          8000    TAPWC   .EQU    30
000031          8010    TAPCA   .EQU    31
000036          8020    DSKWC   .EQU    36
000037          8030    DSKCA   .EQU    37
000010          8040    DSKLN   .EQU    8.
```

```
                Q                       ROUTINES TO GET THE POINTERS FOR DECTAPE

                        8050            .STITL  ROUTINES TO GET THE POINTERS FOR DECTAPE
            001130      8060            .USE    PERM
                        8070    *
                        8080    *       THIS ROUTINE WILL CALL THE GTCBLK AND STPNTR ROUTINES TO
                        8090    *       GET THE COMMAND EXECUTED.
                        8100    *
            001130      8110    DO      ENTER
          001131  040010  8120            DAC     10              SET THE COMMAND POINTER
                        8130    *
                        8140    *       NOW SET THE PHYSICAL ADDRESS FOR BOTH DEVICES
                        8150    *
          001132  220010  8160            LAC     10,X            LOAD THE FIRST PARAMETER WORD
          001133  041176  8170            DAC     WTINT           SAVE IT FOR NOW
          001134  501265  8180            AND     BMSK            RECOVER JUST THE BLOCK NUMBER
          001135  041270  8190            DAC     RBLK            SET IT FOR THE DECTAPE ROUTINES
          001136  341126  8200            TAD     DKMIN           ADD IN THE RELOCATION CONSTANT FOR THE BOTTOM OF THE LOGICAL DISK
          001137  660710  8210            ALSS    DSKLN           CONVERT THE BLOCK COUNT TO WORD COUNT
          001140  707024  8220            DLAL                    PLACE IT INTO THE DISK ADDRESS REGISTER
                        8230    *
                        8240    *       SET UP THE CORE ADDRESS FOR THE DISK ONLY -- THE DECTAPE DATA CHANNEL
                        8250    *       CELL IS STILL NEEDED FOR OTHER THINGS IF IT IS A DECTAPE TRANSFER.
                        8260    *
          001141  777777  8270            LAW     -1              LOAD A MINUS 1
          001142  360010  8280            TAD     10,X            ADD THE SECOND PARAMETER TO GET THE STARTING CORE ADDRESS FOR THE TRANSFER
          001143  040037  8290            DAC     DSKCA           SET IT FOR A DISK OPERATION; SAVE IT FOR A DECTAPE OPERATION
                        8300    *
                        8310    *       SET UP THE WORD COUNT FOR BOTH DEVICES
                        8320    *
          001144  777777  8330    M1      LAW     -1              LOAD A MINUS 1
          001145  360010  8340            TAD     10,X            ADD THE THIRD POINTER TO GET THE LENGTH TO BE COMPLEMENTED
          001146  740001  8350            CMA                     FORM THE TWO'S COMPLEMENT LENGTH
          001147  040030  8360            DAC     TAPWC           SET THE DECTAPE WORD COUNT
          001150  040036  8370            DAC     DSKWC           SET THE DISK WORD COUNT
                        8380    *
                        8390    *
                        8400    *       DECIDE WHETHER TO DO A DISK OR A DECTAPE OPERATION
                        8410    *
          001151  201176  8420            LAC     WTINT           RELOAD THE FIRST PARAMETER WORD
          001152  640603  8430            LLS     3               MOVE THE TYPE BIT TO AC(0)
          001153  741100  8440            SPA                     SKIP IF THE DEVICE IS A DECTAPE
          001154  601242  8450            JMP     DODSK           ELSE TRANSFER TO THE DISK ROUTINES
                        8460    *
                        8470    *       THIS SECTION WILL GET THE BLOCK THAT IS SPECIFIED IN THE
                        8480    *       PARAMTERS UNDER THE WRITE OR READ HEADS OF THE DECTAPE
                        8490    *       UNIT.
                        8500    *
          001155  761256  8510    DTO     LAW     TCA1            GET A POINTER TO THE BUFFER
          001156  040031  8520            DAC     TAPCA           PUT IT IN THE RIGHT PLACE
          001157  201176  8530            LAC     WTINT           GET THE PARAMETER WORD BACK
          001160  501264  8540            AND     TPMSK           AND IT DOWN TO ONLY THE TAPE HANDLER NUMBER
          001161  041260  8550            DAC     STPTP           PUT IT INTO THE STOP TAPE INSTRUCTION
          001162  241262  8560            XOR     SNST1           OR IN THE REST OF THE INSTRUCTION
```

```
                Q                       ROUTINES TO GET THE POINTERS FOR DECTAPE

001163  707545  8570            DTLA                    ISSUE THE INSTRUCTION
001164  101176  8580    DT1     JMS     WTINT           WAIT FOR DECTAPE TO STOP
001165  601206  8590            JMP     SEA1            ANALYSE IN CASE OF ERROR FROM BOT
001166  201256  8600            LAC     TCA1            GET TAPE ADDRESS
001167  740001  8610            CMA
001170  341270  8620            TAD     RBLK            ADD BLOCK WE ARE AT
001171  341144  8630            TAD     M1              CHANGE TO ONE LESS
001172  744100  8640            SMA!CLL                 SEE IF WE ARE PAST DO (AND KEEP THE LINK CLEAR
001173  601212  8650            JMP     REV             YES -- REVERSE
001174  707554  8660            DTXA+10                 CLEAR DTF
001175  601164  8670            JMP     DT1             AND TRY AGAIN
                8680    *
                8690    *       WAIT FOR DT INTERRUPT
                8700    *
                8710    *       RETURN IS TO **1 IF ERROR, **2 IF NOT
                8720    *
    001176      8730    WTINT   ENTER
001177  707573  8740            DTEF!DTRB               CHECK FOR DECTAPE ERROR
001200  741000  8750            SKP                     NO ERROR
001201  621176  8760            JMP     WTINT,X         ERROR, RETURN
001202  707601  8770            DTDF                    CHECK FOR DECTAPE OPERATION COMPLETE
001203  601177  8780            JMP     WTINT+1         JUST LOOP IF NOT
001204  441176  8790            ISZ     WTINT           RETURN PROPERLY
001205  621176  8800            JMP     WTINT,X         RETURN
                8810    *
                8820    *       ERROR OCCURRED, CHECK FOR BOT
                8830    *
    001206      8840    SEA1    ...                     GET THE STATUS REGISTER
001206  501266  8850            AND     BITB            CHECK FOR OBVIOUS EOT ERROR
001207  743220  8860            SNA!RTR                 CHECK FOR EOT, AND ROTATE BIT FOR START TAPE BIT
001210  621130  8870            JMP     DO,X            DECTAPE ERROR
001211  707544  8880            DTXA                    SET IT
                8890    *
                8900    *       REVERSE THE TAPE
                8910    *
001212  201261  8920    REV     LAC     REVDR           TURN TAPE ARROUND
001213  707544  8930            DTXA
001214  101176  8940    DT2     JMS     WTINT           WAIT FOR AN INTERRUPT
001215  741000  8950            SKP                     SKIP IF ERROR
001216  601221  8960            JMP     .+3             NO ERROR - OK
001217  201267  8970            LAC     DTST            RESTART THE TAPE
001220  707544  8980            DTXA                    SET IT
001221  201256  8990            LAC     TCA1            MAKE SURE BLOCK IS THE SAME
001222  541270  9000            SAD     RBLK            CHECK AGAINST WHERE WE WANT TO BE
001223  601226  9010            JMP     DT3             HERE
001224  707554  9020            DTXA+10                 CLEAR THE FLAG
001225  601214  9030            JMP     DT2             AND WAIT AGAIN
    001226      9040    DT3     ...
                9050    *
                9060    *        WE HAVE FOUND THE PLACE ON THE TAPE SO CHANGE THE SWITCH WORD
                9070    *        AND THE OTHER PARAMETERS NECESSARY TO EXECUTE THE COMMAND.
                9080    *
```

```
                0                       ROUTINES TO GET THE POINTERS FOR DECTAPE

001226  200036  9090          LAC     DSKWC           LOAD THE OLD WORD COUNT
001227  040030  9100          DAC     TAPWC           RESTORE IT
001230  200037  9110          LAC     DSKCA           LOAD THE CORE ADDRESS FROM WHERE IT WAS SAVED FOR THE DISK
001231  040031  9120          DAC     TAPCA           AND SET IT FOR THE DECTAPE
001232  221263  9130          LAC     CMND,X          GET THE COMMAND THAT IS TO BE ISSUED
001233  707544  9140          DTXA                    ISSUE THE INSTRUCTION
001234  101176  9150          JMS     WTINT           WAIT FOR OPERATION TO COMPLETE
001235  621130  9160          JMP     DO,X            DECTAPE ERROR
001236  201260  9170          LAC     STPTP           GET THE STOP INSTRUCTION
001237  707545  9180          DTLA                    STOP THE TAPE
001240  441130  9190          INX     DO
001241  621130  9200          JMP     DO,X            RETURN TO THE CALLER +2 FOR A SUCCESSFUL OPERATION
```

```
                Q                       DISK ROUTINES

                        9210            .STITL  DISK ROUTINES
                        9220    *
                        9230    *
          001242        9240    DQDSK   ...                     DISK AND DECTAPE USE THE SAME PASS LIST
                        9250    *
                        9260    *        ISSUE THE OPERATION
                        9270    *
     001242   441263     9280            ISZ     CMND            MOVE THE POINTER TO THE DISK COMMANDS
     001243   221263     9290            LAC     CMND,X          GET THE COMMAND
     001244   707047     9300            DSCF!DSFX!DSCN          ISSUE THE OPERATION
     001245   707001     9310            DSSF                    SEE IF THE OPERATION IS DONE
     001246   601245     9320            JMP     .-1             IF NOT THEN WAIT A LITTLE LONGER
                        9330    *
                        9340    *        CHECK THE OPERATION AND RETURN TO THE APPROPRIATE PLACE
                        9350    *
     001247   707272     9360            DSRS+10                 CLEAR THE AC AND GET THE STATUS OF THE OPERATION
     001250   707242     9370            DSCD                    CLEAR THE FLAGS
     001251   741100     9380            SPA                     SEE IF OK
     001252   621130     9390            JMP     DQ,X            IT WAS BAD SO TELL THE USER
     001253   441130     9400            ISZ     DQ
     001254   621130     9410            JMP     DQ,X            RETURN TO THE CALLER +2 FOR A SUCCESSFUL OPERATION
```

```
                Q                       STORAGE AREA

                     9420               .STITL  STORAGE AREA
                     9430       *
                     9440       *        STORAGE USED BY ALL OF THE ABOVE ROUTINES
                     9450       *
001255  015000       9460       WRITE   015000                  WRITE COMMAND
001256  000004       9470       TCA1    4                       COMMAND FOR A DISK WRITE; ALSO USED FOR DECTAPE TEMPORARY STORAGE
001257  013000       9480       READ    013000                  READ COMMAND
001260  000002       9490       STPTP   2                       COMMAND FOR A DISK READ; ALSO USED FOR DECTAPE TEMPORARY STORAGE
001261  040000       9500       REVDR   040000                  REVERSE COMMAND
001262  061000       9510       SNST1   61000                   INSTRUCTION TO START THE TAPE
001263  705002       9520       CMND    READ                    STORAGE FOR THE COMMAND POINTER
001264  700000       9530       TPMSK   700000                  WORD TO MASK DOWN TO ONLY HANDLER NUMBER
001265  001777       9540       BMSK    1777                    MASK TO SAVE ONLY THE BLOCK NUMBER
001266  100000       9550       BITB    100000                  BIT FOR END ERROR ON TAPE
001267  020000       9560       DTST    20000                   MOTION BIT FOR TAPE
001270  000000       9570       RBLK    0
    001271           9580       CHECK   .EQU    .
    001633           9590               .USE    OVRLAY
001633  677777       9600               .END    OVSTRT
001634  000004
001635  700000
001636  077777
001637  000377
001640  001510
001641  000001
001642  000006
001643  000034

TRANSFER ADDRESS  601000
```

Q

CROSS REFERENCE TABLE

1713	.Q	4510	4520	5260	5280	6140	6160			
26	.310	3400	5250	6150						
27	.311	3410	5270	6170						
4464	.DT	570								
6460	.TP	550								
2023	10SAVE	1870	1880							
2024	11SAVE	1880	1920							
5	3AC	3370	5230	6130						
305	3REST	3870	3880							
51	3TEM0	3530	3540							
52	3TEM1	3540	3550							
53	3TEM2	3550	3560							
54	3TEM3	3560	3570							
55	3TEM4	3570	3580							
56	3TEM5	3580	3590							
57	3TEM6	3590	3600							
50	3TM20	3520	3530							
2	3TM21	3350								
3	3TM22	3360								
14000	7K	1030								
16000	8K	1020	910	1010	2640	2650				
1753	AC	4520	4530	2700	3770	5240	6120	6400		
1756	ACS	4550	4560							
2015	ACSAVE	1810	1820							
2022	ACSW	1860	1870							
651	ADRSS	4100	4110	2150	5470					
246	AMPRSN	290								
300	AT	2950								
300	ATSGN	2900								
422030	BAS	420								
2151	BCNTRL	2330	2340							
634	BIT0	3970	3980							
641	BIT17	4020	4030							
635	BIT36	3980	3990							
636	BIT5	3990	4000							
637	BIT6	4000	4010							
640	BIT7	4010	4020							
642	BL7	4030	4040							
643	BL8	4040	4050							
400	BLKLEN	630								
1777	BLKMSK	610								
2000	BOUNDA	440	960	980	990	1000	1630	5040	5080	7070
377	BRK	5550								
334	BSLASH	390								
1116	BUFADD	7490	4800							
1300	BUFFER	600	2550	610	2660	2680	7490			
1000	BUFLEN	2500	2550							
1	CATBLK	550								
400	CATLEN	560								
17000	CATLOG	540	590							
777716	CATMAX	620								
644	CB0	4050	4060							

0 CROSS REFERENCE TABLE

645	CB1	4060	4070		
646	CB5	4070	4080		
647	CB7	4080	4090		
650	CBL8	4090	4100		
6	CHRMAX	3180	3200		
2	CHRPAK	3130	3200		
50	CLKMAX	2840	3180		
60	CLKSPD	3160	3170		
1757	CLOCK	4560	4570		
45	CMP1	3490	3500		
46	CMP2	3500	3510		
6	CNTRL	3380	3390		
2053	COMFLG	2200	2210		
2150	COMSTO	2270	2280		
16000	CORMAX	910	980	7180	
17005	CPARAM	590			
47	CSPL	3510	3520		
44	CSWP	3480	3490		
60	CTBFR	3600	3630	3640	
100	CTBIN	3640	3650	3670	4250
2000	CTEMP0	1630			
2001	CTEMP1	1640			
2002	CTEMP2	1650			
2003	CTEMP3	1660			
2004	CTEMP4	1670			
2005	CTEMP5	1680			
2006	CTEMP6	1690			
2007	CTEMP7	1700			
2010	CTEMP8	1710			
2011	CTEMP9	1720			
102	CTFLG	3650	3660		
104	CTNAM	3660			
2043	D PC	2120	2130		
2154	D BCA	2370	2380		
2153	D BDA	2360	2370		
2163	D FDA	2440	2450		
2042	D LOC	2110	2120		
2022	D ACSW	1860			
2156	D BALT	2390	2400		
2155	D BLEN	2380	2390		
2161	D BMAX	2420	2430		
2157	D BMIN	2400	2410		
2162	D BPTR	2430	2440		
2167	D FMAX	2480	2490		
2165	D FMIN	2460	2470		
2046	D MASK	2150	2160		
2164	D MFDA	2450	2460		
2036	DADRSW	2070	2080		
1762	DAP0	4590	4600		
1763	DAP1	4600	4610		
653	DBK	4120	4130		
24	DBKNUM	2220	2270		

Q CROSS REFERENCE TABLE

2054	DBKTAB	2210	2270	
2035	DBSTOR	2050	2060	
422027	DDT	410		
12000	DDTST	5000		
2037	DDUMSW	2080	2090	
1761	DFLAG	4580	4590	
1764	DFN	4610	4620	
2151	DFTYPE	2340	2350	
2045	DHICOR	2140	2150	
2050	DINDIR	2170	2180	
100	DK0	4270		
127	DK1	4310		
156	DK2	4350		
37	DKCA	2750		
675	DKDON	4170	4180	
16000	DKLEN	2650	2660	
34	DKLENB	2660	6330	
266	DKLOK	3830	3840	5340
672	DKOVR	4160	4170	
2	DKRD	2760		
36	DKWC	2740		
4	DKWRT	2770		
2041	DLIMIT	2100	2110	
2044	DLOCOR	2130	2140	
2160	DMBMIN	2410	2420	
2166	DMFMIN	2470	2480	
654	DO	4130	4140	6080
662	DO2	4140	4150	
663	DO3	4150	4160	
2152	DOFTYP	2350	2360	
244	DOLLAR	280		
2032	DPACSW	1980		
2040	DPATSW	2090	2100	
2051	DPCMSK	2180	2190	
2052	DREGBR	2190	2200	
2035	DREGSW	2060	2070	
2047	DRELOC	2160	2170	
1765	DSTAT	4620	4630	
446400	DT.	560		
2000	DTEMP0	1630		
2001	DTEMP1	1640		
2002	DTEMP2	1650		
2003	DTEMP3	1660		
2004	DTEMP4	1670		
2005	DTEMP5	1680		
2006	DTEMP6	1690		
2007	DTEMP7	1700		
2010	DTEMP8	1710		
2011	DTEMP9	1720		
740000	DVCMSK	600		
275	EQUAL	2910		
241	EXCLAM	260		

Q CROSS REFERENCE TABLE

5	FCBLEN	570	2740						
602	FGET	3950	3960						
1701	FRCA	4410	4420	2880	4010	4040	4130	4240	4330
1700	FRDA	4400	4410	2860	4460				
1702	FRLEN	4420	4430	2900	4140	4250			
1703	FRSTA	4430	4440	2920					
2	FUDGE	3190	3200						
276	GREAT	2930							
4	HDRLEN	580							
1700	IMPLEN	990							
3170	IMPSTR	2550							
422020	INT	320							
513	IO.IN	3910	3920						
525	IO.OT	3920	3930						
300000	IOBLK	2830							
1760	IORS	4570	4580						
1002	IOTO	4900	4910						
652	JMP	4110	4120						
100	JTLEN	960							
1700	JTSTRT	950	940	960	1000	4400			
16	KBLEN	3610	3630	3640	3680	3690	3730	3740	
10	KBNUM	3620	3670	3720					
76	LOLOK	3630							
107	L1BFR	3670	3680	3690					
127	L1BIN	3690	3700	3720	4290				
131	L1FLG	3700	3710						
125	L1LOK	3680							
133	L1NAM	3710							
136	L2BFR	3720	3730	3740					
156	L2BIN	3740	3750	4330					
160	L2FLG	3750	3760						
154	L2LOK	3730							
162	L2NAM	3760	3770						
422026	LDR	390							
2080	LDRST	5040							
274	LES$	2920							
2022	M ACSW	1860							
10	MINBUF	3200	3610						
422023	MP1	350							
422024	MP2	360							
2032	MPACSW	1980							
1004	MPOPR	4920							
1000	MPST	4880	4890						
1754	MQ	4530	4540						
2016	MQSAVE	1820	1830						
2000	MTEMP0	1630							
2001	MTEMP1	1640							
2002	MTEMP2	1650							
2003	MTEMP3	1660							
2004	MTEMP4	1670							
2005	MTEMP5	1680							
2006	MTEMP6	1690							

0 CROSS REFERENCE TABLE

2007	MTEMP7	1700							
2010	MTEMP8	1710							
2011	MTEMP9	1720							
422025	MTR	370							
2000	MTRST	5080							
1772	NAME	4670	4680						
540	NEWBR	3930	3940						
1771	NUMBR	4660	4670	6780					
243	NUM$GN	270							
623	NXPTR	3960	3970						
702	OC0	4180	4190	5300					
703	OC1	4190	4200	930					
704	OC2	4200	4210	940					
705	OC3	4210							
574646	OFF	2730							
575600	ON	2720							
1773	OVER	4680	4690						
700	OVLEN	940							
1000	OVSTRT	930	920	940	4750	4880	4960	550	9600
2033	P10SAV	1990	2000						
2034	P11SAV	2000	2050						
2025	PAC$AV	1930	1940						
2032	PAC$W	1980	1990						
1622	PARAM1	7280	2000	7300					
1627	PARAM2	7360	4590	7370					
241	PBFLAG	3810	3820						
2017	PCSAVE	1830	1840						
256	PERIOD	340	350						
227	PFLAG	3770	3780						
77	PH0	4260	4270						
126	PH1	4300	4310						
155	PH2	4340	4350						
1	PHANTO	2780							
2150	PHFLAG	2280	2330						
1700	PHLEN	2640							
2025	PHSTOR	1920	1930						
274	PIDN2	3850	3860						
270	PIDON	3840	3850						
1001	PINT	4890	4900						
303	PIOUT	3860	3870						
602026	PLDR	400							
2026	PMQ$AV	1940	1950						
602025	PMTR	380							
2027	PPC$AV	1950	1960						
606064	PPT	520							
2031	PSC$AV	1970	1980						
2030	PST$AV	1960	1970						
606460	PTP	510							
606462	PTR	500							
12100	PURLEN	1010							
1775	PURNM	4700	4710						
3700	PURSTR	2560	990	1010	2560				

Q CROSS REFERENCE TABLE

```
   546  PUTIN     3940    3950
   215  Q   CR     220
  1115  Q   DA     7480    2520    2530    2570    4490    4700    4850    4920
  1130  Q   DO     8110    2120    2600    4730    5060    6960    7000    8870    9160    9190    9200    9390    9400
                        9410
   242  Q   LF     210
  1144  Q   M1     8330    6340    8630
  4453  Q  ,DK     770
  4464  Q  ,DT     750
  6460  Q  ,TP     760
414263  Q  ABS     850
425156  Q  BIN     860
  1121  Q  CCA     3650    4050    4220    4350    4870    4910    7080    7190
  1125  Q  CCU     7560    5370    6100
  1120  Q  CDA     3630    4480    4530    5050    5550    6630    6740    6970
435762  Q  COR     880
436454  Q  CTL     810
445300  Q  DK,     740
445320  Q  DK0     780
446400  Q  DT,     720
  1155  Q  DT0     8510
  1164  Q  DT1     8580    8670
  1214  Q  DT2     8940    9030
  1226  Q  DT3     9040    9010
  1074  Q  ERR     5980    1100
476257  Q  GRO     870
606064  Q  PPT     690
606460  Q  PTP     710
606462  Q  PTR     700
  1121  Q  RCA     3640    3870    3990
  1212  Q  REV     8920    8650
  1120  Q  SOB     3620    4390    4470
  1123  Q  SOW     3690    4420    4550    4630    4810    4840
  1323  Q  SP4     2100    2010
646000  Q  TP,     730
  1286  Q BITB     9550    8850
  1265  Q BMSK     9540    2840    6980    8180
 17740  Q BOOT     520     530
    11  Q CATX     500
  1122  Q CLEN     3670    4280    4570    4580    4980    6940    7200
    12  Q CMDX     510
  1263  Q CMND     9520    6300    7290    9130    9280    9290
  1267  Q DTST     9560    8970
  1073  Q ERR3     5970    4270    4740    5070    5540    5840
  1071  Q ERR6     5950    7030
  1070  Q ERR7     5940    7220
  1067  Q ERR8     5930    6660
  1066  Q ERR9     5920    6830
  1536  Q PAR2     6380    7320
  1547  Q PAR4     6540    7390
  1575  Q PAR5     6920    6670
  1572  Q PAR6     6840    6800
```

Q CROSS REFERENCE TABLE

1560	Q PART	6730	6580			
253	Q PLUS	310				
1270	Q RBLK	9570	8190	8620	9000	
1257	Q READ	9480	1990	2040		
1122	Q RLEN	3660	3900	4110		
1206	Q SEA1	8840	8590			
1592	Q SPL1	5530				
252	Q STAR	300				
1256	Q TCA1	9470	8510	8600	8990	
1124	Q XFER	7550	2140	3710	3790	5220
17777	QADRSS	430				
337	QBKARR	240				
1271	QCHECK	9580	610			
272	QCOLON	370				
254	QCOMMA	320				
230	QCONTX	230				
1374	QCOPY2	4090	4030			
1414	QCOPY4	4320	4160			
1055	QCOPY6	4970	4650			
1034	QCOPY7	4730	4710			
1050	QCOPY8	4890	4930			
1063	QCOPYD	5110	5000			
777601	QDKMAX	650	6360			
1126	QDKMIN	7570	6320	6840	6990	8200
1242	QDODSK	9240	8450			
37	QDSKCA	8030	8290	9110		
40	QDSKLN	8040	8210			
36	QDSKWC	8020	8370	9090		
776701	QDTMAX	640				
1064	QERR11	5900	2780			
40	QINDEX	490				
255	QMINUS	330				
1001	QOPEN2	2600	2580			
1010	QOPEN4	2700	2770			
1021	QOPEN6	2830	2720			
1526	QPARAM	6290	2080	7230	7310	7380
1521	QPHCHK	5800	890	1950	5830	7100
256	QPOINT	350				
17505	QRECOV	470				
1261	QREVDR	9500	8920			
257	QSLASH	360				
1262	QSNST1	9510	8560			
1333	QSO006	2420	5690			
1300	QSP000	880	5630			
1307	QSP001	1080	5640			
1317	QSP002	2030	5650			
1314	QSP003	1980	5660			
1321	QSP004	2060	5670			
1311	QSP005	1940	5680			
1343	QSP007	3700	5700			
240	QSPACE	250				
1447	QSPLST	5200	570			

Q CROSS REFERENCE TABLE

1520	QSPMAX	5710	5520										
1123	QSTOFF	3680	4360	4400									
1260	QSTPTP	9490	8550	9170									
31	QTAPCA	8010	8520	9120									
17500	QTAPIN	450											
17502	QTAPOT	460											
30	QTAPWC	8000	8360	9100									
1127	QTDMAX	7580	6860	7010									
1120	QTEMP0	7510	2110	2510	2850	3620	3630	5460	5490	6430			
1121	QTEMP1	7520	2690	2760	3640	3650	5510	5550	6450				
1122	QTEMP2	7530	3660	3670	5570	5580	6470						
1123	QTEMP3	7540	3680	3690	3880	4100							
1264	QTPMSK	9530	2160	6640	6750	8540							
17777	QVFLAG	480											
1255	QWRITE	9460	1960	2070									
1176	QWTINT	8730	8170	8420	8530	8580	8760	8780	8790	8800	8940	9150	
34	RACS	3440											
6	RCNT	3390											
35	RCORE	3450	5350	5360	6110								
1003	RDBLK	4910	4920										
32	RDT0	3420	6790										
33	RDT1	3430	6810										
422021	RES	330											
40	RESCAT	3470	3480										
1000	RESLEN	920											
234	RFLAG	3790	3800										
230	RPTR	3780	3790										
235	RPTR	3800	3810										
242	RSCO	3820	3830										
1776	RSTRT	4710	900	920	2170	2490	2500	2620	3730	3760	3810	3820	5290	5320
			5450	6180										
1755	SC	4540	4550										
273	SCOLON	380											
640000	SCRSTR	2670											
2021	SCSAVE	1850	1860										
243	SHARP	2890											
377	SPCOD	5410	5500										
422122	SPL	430											
1077	SPLDON	6060	950	2130	2180	2610	2930	5110					
1090	SPLST	4960											
777400	SPMSK	5390											
1510	SPTABL	5630	5560	5710									
2020	STSAVE	1840	1850										
335	SWAP	3880	3890										
336	SWAP1	3890	3900	6200									
340	SWAP3	3900	3910										
1000	SWCAT	4750	4760										
1003	SWCLK	4780	4790										
1004	SWERR	4790	4800	5990									
1007	SWMP1	4820	4830										
1010	SWMP2	4830	4840										
1002	SWMTR	4770	4780										

Q CROSS REFERENCE TABLE

1011	SWOPR	4840		
422022	SWP	340		
1001	SWPPR	4760	4770	
40	SWPS	3460	3470	
1005	SWSPL	4800	4810	
1006	SXSPL	4810	4820	6070
1300	SYSBAS	2800	2810	
41300	SYSDA	2810		
17735	SYSDEV	530		
1777	SYSMAX	2820		
100	TABLEN	2630	2640	
2000	TEMP0	1630	1640	
2001	TEMP1	1640	1650	
2012	TEMP10	1730	1740	
2013	TEMP11	1740	1750	
2014	TEMP12	1750	1800	
2002	TEMP2	1650	1660	
2003	TEMP3	1660	1670	
2004	TEMP4	1670	1680	
2005	TEMP5	1680	1690	
2006	TEMP6	1690	1700	
2007	TEMP7	1700	1710	
2010	TEMP8	1710	1720	
2011	TEMP9	1720	1730	
646000	TP.	540		
376	TRCOFF	5540		
375	TRCON	5530		
2000	TTEMP0	1630		
2001	TTEMP1	1640		
2002	TTEMP2	1650		
2003	TTEMP3	1660		
2004	TTEMP4	1670		
2005	TTEMP5	1680		
2006	TTEMP6	1690		
2007	TTEMP7	1700		
2010	TTEMP8	1710		
2011	TTEMP9	1720		
6	TTYCLK	3170	3180	
3	TTYNUM	3140		
10	TTYSPD	3150	3170	
1774	TYPE	4690	4700	5810
1766	UCORE	4630	4640	
1767	UDISK	4640	4650	6310
336	UPARR	2940		
76	US0	4250	4260	4280
125	US1	4290	4300	4320
154	US2	4330	4340	4360
0	USER	2790		
3	USERS	2850	3200	
14000	USLEN	980	2640	
2015	USTORE	1800	1810	
75	UT0	4280		

Q CROSS REFERENCE TABLE

124	UT1	4320		
153	UT2	4360		
1704	UTEM0	4440	4450	
1705	UTEM1	4450	4460	
1706	UTEM2	4460	4470	5980
1707	UTEM3	4470	4480	
1710	UTEM4	4480	4490	
1711	UTEM5	4490	4500	5480
1712	UTEM6	4500	4510	
1770	VALID	4650	4660	

Q UNDEFINED SYMBOLS

#1	5630											
#2	5640											
#3	5650											
#4	5660											
#5	5680											
DEBUG	100	100	120	120								
DEFINS	100	100	120	120	580	580	2680	2680	3210	3210	5330	5330
PURCOD	5140	5270										
Q BAD	620											
Q LINE	1210											
Q MESS	1190	1200										

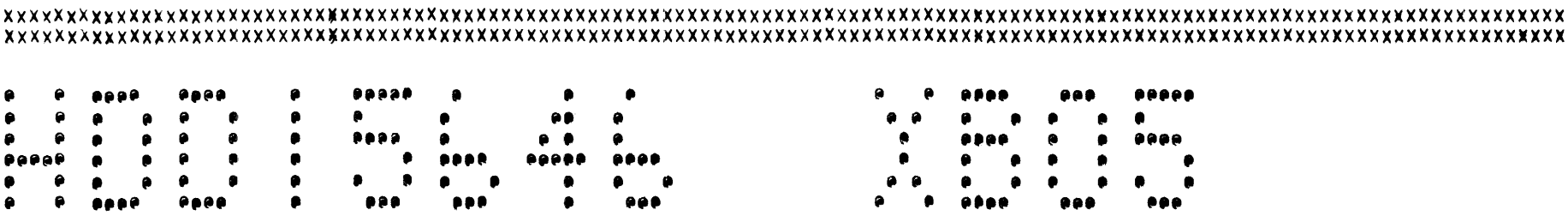
Q

MACRO CROSS REFERENCE TABLE

ENTER	920	8110	8730
FORMAT	1060		
LOOP	960		
MPOFF	5430		
NEG	1010		
START	1100		
SWAP	5610		

0 USE CROSS REFERENCE TABLE

```
    0   .!.!.!
    0   .REL.
 1000   PERM      560   2590   4720   5850   7430   8060
 1300   OVRLAY    580    640   1120   1930   2200   2950   5130   5730   6210   9590
```



05/31/72 02:04:32

```
**********************************
**********************************
**                              **
**  PDP-9 MINI TIME-SHARING SYSTEM  **
**       MTS$ SYSTEM MONITOR        **
**       MTS$ SYSTEM MESSAGES       **
**    GROWTH SYSTEM DISK BOOTSTRAP  **
**              DTS$:XMTR            **
**              MTS$:B05             **
**                              **
**********************************
**********************************
```

```
                100                 ,TITLE  PDP9 TIME-SHARING SYSTEM MONITOR AND MESSAGE ROUTINES
000001          110     PURCOD      ,EQU    1
                120                 ,NAME   MTR--805
                130     *
                140     *
                150     *           PASSED PARAMETERS:
                160     *                $UTEM5: NAME OF THE CALLING PROGRAM
                170     *                $UTEM5: OCTAL VALUE TO PRINT (OPTIONAL)
                180     *                $UTEM6: MESSAGE NUMBER
                190     *
                200
                210                 ,HEAD   M
                220                 ,PMC    ON
                230
                240                 ,INSRT  DEFINS
                100                 ,IFUND  DEFINS
```

```
                5720                ,LIST   ON
                5730                ,END
        000000  250         DEBUG   ,EQU    0
                260                 ,INSRT  :DLIBRARY:PDP9LIB:GRODEFIN
                100                 ,INE    $DEBUG,1
                1250                ,LIST   ON
                1260                ,END
                270                 ,INSRT  :DLIBRARY:PDP9LIB:LIBMACRO
                100                 ,INE    DEBUG,1
```

TELETYPE INPUT/OUTPUT MACROS

```
1940            ,LIST   ON
1950            ,END
```

MONITOR MACRO DEFINITIONS

```
280             ,STITL  MONITOR MACRO DEFINITIONS
290     *
300     *
310     *       MACRO TO SET UP PURE-CODED SUBROUTINE ENTRANCES
320     *
330     ENTER   ,DEFIN
340             ,PMC    SAVE,OFF
350     9MAPBUG ,EQU    .
360             ,USE    IMPURE          SUBROUTINE ENTRANCES CANNOT BE PURE CODE
370             ,PMC    SAVE,ON
380     #1      ...
390             ,PMC    RESTORE
400             HLT
410             JMP     9MAPBUG
420             ,USE    PREVIOUS
430             ,PMC    RESTORE
440             ,ENDM
450     *
460     *
470     *       THE FOLLOWING MACROS ARE USED TO BUILD THE TABLES WHICH CONTROL THE MONITOR
480     *
490     *
500     *       REGISTER PUTS AN ELEMENT IN THE TABLE OF AVAILABLE SOFTWARE REGISTERS
510     *
520     REGISTER ,DEFIN
530             ,ACI6   S#1$            REGISTER NAME: #1
540             JMP     R#1             DUMP REGISTER #1
550             ,CRSM   SAVE,OFF
560             ,INE    'A#2','A',2
570             ,ACI6   S#2$            ALTERNATE REGISTER NAME: #2
580             JMP     R#1             DUMP REGISTER #1
590             ,CRSM   RESTORE
600             ,ENDM
610     *
620     *       COMMAND PUTS AN ELEMENT IN THE TABLE OF AVAILABLE SOFTWARE REGISTERS
630     *
640     COMMAND ,DEFIN
650             ,ACI6   S#1$            COMMAND IS #1
660             JMP     #1              EXECUTE THE COMMAND
670             ,CRSM   SAVE,OFF
680             ,IDRP   #2
690             ,INE    'A#2','A',2
700             ,ACI6   S#2$            ALTERNATE COMMAND IS #2
710             JMP     #1              EXECUTE THE COMMAND #1
720             ,IDRP
730             ,CRSM   RESTORE
740             ,ENDM
750     *
760     *       DMODE PUTS ELEMENTS IN THE DDT MODE TABLE
770     *
780     DMODE   ,DEFIN
790             ,ACI6   S#1$            DUMP MODE NAME
```

MONITOR MACRO DEFINITIONS

```
800             JMP     M#1
810             .ACI6   $A#1$           ADDRESS MODE NAME
820             JMP     AM#1
830             .ACI6   $R#1$           REGISTER MODE NAME
840             JMP     RM#1
850             .CRSM   SAVE,OFF
860             .IDRP   #2
870             .INE    'A#2','A!.6
880             .ACI6   $#2$            ALTERNATE DUMP FORMAT NAME
890             JMP     M#1
900             .ACI6   $A#2$           ALTERNATE ADDRESS MODE NAME
910             JMP     AM#1
920             .ACI6   $R#2$           ALTERNATE REGISTERS MODE NAME
930             JMP     RM#1
940             .IDRP
950             .CRSM   RESTORE
960             .ENDM
970     *
980     *       RESOURCE PUTS AN ELEMENT IN THE TABLE OF AVAILABLE ALLOCATABLE RESOURCES
990     *
1000    RESOURCE .DEFIN
1010            .ACI6   $#1$            RESOURCE NAME
1020            LAW     $R#1            LOAD A POINTER TO THE RECORD OF RESOURCE #1
1030            JMP     #1ON            TRY TO ALLOCATE THE #1
1040            JMP     #1OFF           TRY TO DE-ALLOCATE THE #1
1050            .ENDM
1060    *
1070    *       WHAT JUMPS TO THE ERROR MESSAGE "WHAT: WORD #N"
1080    *
1090    WHAT    .DEFIN
1100            .PMC    SAVE,OFF
1110            JMP     MSG50
1120            .PMC    RESTORE
1130            .ENDM
1140    *
1150    *       FORMAT JUMPS TO THE ERROR MESSAGE "FORMAT ERROR: WORD #N"
1160    *
1170    FORMAT  .DEFIN
1180            .PMC    SAVE,OFF
1190            JMP     MSG56
1200            .PMC    RESTORE
1210            .ENDM
1220    *
1230    *       CMDERR PRINTS THE "WORD #N" IN THE ABOVE MESSAGES
1240    *
1250    CMDERR  .DEFIN
1260            .PMC    SAVE,OFF
1270            JMP     MSG57
1280            .PMC    RESTORE
1290            .ENDM
```

MONITOR CONVENTIONS AND TEMPORARY STORAGE

```
1300            .STITL  MONITOR CONVENTIONS AND TEMPORARY STORAGE
1310
1320
1330
1340    *       TEMP0 -- CALLING PROGRAM'S NAME
1350    *       TEMP1 -- CALLING PROGRAM'S EXTENDED PROGRAM COUNTER
1360    *       TEMP2 -- CALLING PROGRAM'S INSTRUCTION AT THAT LOCATION
1370    *       TEMP3 -- CALLING PROGRAM'S OVERLAY NAME
1380    *       TEMP4 -- UNUSED
1390    *       TEMP5 -- UNUSED
1400    *       TEMP6 -- PASSED PARAMETER #1 (DESIRED MESSAGE NUMBER)
1410    *       TEMP7 -- PASSED PARAMETER #2 (PROGRAM COUNTER FOR THE ERROR MESSAGE)
1420    *       TEMP8 -- PASSED PARAMETER #3 (ILLEGAL INSTRUCTION FOR THE ERROR MESSAGE)
```

MONITOR INITIALIZATION

```
                    1430            .STITL  MONITOR INITIALIZATION
                    1440            .PMC    OFF
                    1450    *
                    1460    *
                    1470    *       ARRANGE THE USE COUNTERS IN ORDER -- IMPURE CANNOT EXCEED $PURSTR
                    1480    *
    000000          1490            .USE    IMPURE          PHANTOM'S SAVED IMPURE CODE
    003170          1500            .LOC    IMPSTR          START IMPURE CODE AT THE END OF COMMON STORAGE
    003700          1510            .USE    PURE            ALL CODE ABOVE $PURSTR MUST BE PURE
                    1520    *
                    1530    *       CHECK FOR OVERLENGTH IMPURE CODE
                    1540    *
                    1550            .IFG    CHECK,3600
                    1570    *
                    1580    *
                    1590    *       ENTER THE MONITOR/MESSAGES/DDT HERE WITH MEMORY PROTECTION OFF
                    1600    *       TURN OFF ALL INTERRUPTS; SAVE THE USER OR PHANTOM REGISTERS;
                    1610    *       TURN THE INTERRUPTS BACK ON.
                    1620    *
003700  602025      1630            PMTR                    NAME OF PURE CODE PORTION
    003701          1640    START   ...,                    ALL EXTERNAL ENTRANCES TO THIS PROGRAM START HERE
    003701          1650            MPOFF
                                    .PMC    SAVE,ON
003701  705000                      $SPECIAL+0              TURN OFF MEMORY PROTECT
003702  700002      1660            IOF                     DISABLE ALL INTERRUPTS UNTIL WE DECIDE WHAT TO DO
003703  200702      1670            LAC     SQC0            LOAD THE CALLER'S PC
003704  041776      1680            DAC     SRSTRT          AND SET IT FOR THE REGISTERS SAVES TO PICK UP
003705  042011      1690            DAC     TEMP9           SAVE THE AC UNTIL WE DECIDE WHOSE IT IS
003706  202150      1700            LAC     PHFLAG          LOAD THE PHANTOM PROGRAM FLAG
003707  740200      1710            SZA                     SKIP IF WE ARE COMING FROM A USER PROGRAM
003710  603731      1720            JMP     STRT1           ELSE DON'T DESTROY THE REAL USER'S REGISTERS
003711  202011      1730            LAC     TEMP9           LOAD THE USER'S AC
003712  042015      1740            DAC     ACSAVE          SAVE THE USER'S AC
003713  641002      1750            LACQ
003714  042016      1760            DAC     MQSAVE          SAVE THE USER'S MQ REGISTER
003715  641001      1770            LACS
003716  042021      1780            DAC     SCSAVE          SAVE THE USER'S STEP COUNTER REGISTER
003717  201776      1790            LAC     SRSTRT
003720  042017      1800            DAC     PCSAVE          SAVE THE USER PROGRAM'S EXTENDED PROGRAM COUNTER
003721  201760      1810            LAC     SIORS
003722  042020      1820            DAC     STSAVE          SAVE THE USER PROGRAM'S PROGRAM INTERRUPT STATUS
003723  200010      1830            LAC     10
003724  042023      1840            DAC     10SAVE          SAVE THE USER'S AUTO-INDEX REGISTER 10
003725  200011      1850            LAC     11
003726  042024      1860            DAC     11SAVE          SAVE THE USER'S AUTO-INDEX REGISTER 11
003727  442150      1870            INX     PHFLAG          RAISE THE PHANTOM FLAG TO PROTECT USER REGISTERS
003730  603747      1880            JMP     STRT2           SKIP TO THE COMMON INITIALIZATION
                    1890    *
                    1900    *
                    1910    *       THE INTERRUPT WAS FROM A PHANTOM PROGRAM -- SAVE THE REGISTERS
                    1920    *
    003731          1930    STRT1   ...
```

MONITOR INITIALIZATION

```
003731  202011  1940            LAC     TEMP9
003732  042025  1950            DAC     PACSAV          SAVE THE PHANTOM AC
003733  641002  1960            LACQ
003734  042026  1970            DAC     PMQSAV          SAVE THE PHANTOM MQ
003735  641001  1980            LACS
003736  042031  1990            DAC     PSCSAV          SAVE THE PHANTOM STEP COUNTER
003737  201776  2000            LAC     SRSTRT
003740  042027  2010            DAC     PPCSAV          SAVE THE PHANTOM PROGRAM COUNTER
003741  201760  2020            LAC     SIORS
003742  042030  2030            DAC     PSTSAV          SAVE THEE PHANTOM PROGRAM INTERRUPT STATUS
003743  200010  2040            LAC     10
003744  042033  2050            DAC     P10SAV          SAVE THE PHANTOM AWTO-INDEX REGISTERS 10
003745  200011  2060            LAC     11
003746  042034  2070            DAC     P11SAV          SAVE THE PHANTOM AWTO-INDEX REGISTER 11
                2080    *
                2090    *
                2100    *       THE INITIALIZATION IS  COMPLETED AS FAR AS SYSTEM pROBLEMS GO,
                2110    *       SO RE-ENABLE INTERRUPTS AND START THE PROGRAM INITIALIZATION.
                2120    *
   003747       2130    STRT2   ...,
003747  201704  2140            LAC     SUTEM0
003750  042000  2150            DAC     TEMP0           SAVE THE ERROR LOCATION
003751  201705  2160            LAC     SUTEM1
003752  042001  2170            DAC     TEMP1           SAVE THE ERROR OP CODE
003753  201706  2180            LAC     SUTEM2
003754  042002  2190            DAC     TEMP2           SAVE THE MESSAGE NUMBER
003755  700042  2200            ION                     RE-ENABLE ALL INTERRUPTS
003756  701742  2210            MPEU                    GO INTO USER MODE
003757  142156  2220            DZM     D$BALT          FLAG THE BUFFER NOT YET ALTERED
003760  703302  2230            CAF                     CLEAR ALL FLAGS
003761  700002  2240            IOF                     TURN OFF THE USER PROGRAM INTERRUPT SYSTEM
003762  700416  2250            TLS+10                  INITIALIZE THE TELETYPE
                2260    *
                2270    *
                2280    *       THERE ARE SEVERAL REASONS TO CALL THIS MODULE. DETERMINE WHICH IT
                2290    *       IS THIS TIME. THE DISTINCTIONS ARE:
                2300    *            DDT BREAKPOINT -- THE ILLEGAL INSTRUCTION IS SPECIAL+BRK
                2310    *            MONITOR REQUEST -- MESSAGE NUMBER IS ZERO
                2320    *            ERROR MESSAGE REQUEST -- MESSAGE NUMBER IS NON-ZERO
                2330    *
003763  202001  2340            LAC     TEMP1           LOAD THE USER'S LAST INSTRUCTION
003764  553273  2350            SAD     (SPECIAL+BRK)   CHECK FOR A DDT BREAKPOINT
003765  740040  2360            HLT                     BREAKPOINTS NOT YET ENABLED
003766  553274  2370            SAD     (SPECIAL+TRCON)
003767  740040  2380            HLT                     TRACE ON NOT YET ENABLED
003770  553275  2390            SAD     (SPECIAL+TRCOFF)
003771  740040  2400            HLT                     TRACE OFF NOT YET ENABLED
003772  202002  2410            LAC     TEMP2           LOAD THE NUMBER OF THE POTENTIAL MESSAGE
003773  513276  2420            AND     (17777)         NO MESSAGE NUMBER EXCEEDS THIS!
003774  042002  2430            DAC     TEMP2           SET THE MESSAGE NUMBER
003775  353277  2440            TAD     (MSGMAX)        SUBTRACT THE LARGEST LEGAL MESSAGE NUMBER
003776  740100  2450            SMA                     SKIP UNLESS MESSAGE NUMBER IS TOO LARGE
```

MONITOR INITIALIZATION

```
003777  611573  2460          JMP     MSG53           IN WHICH CASE PRINT A WARNING MESSAGE
                  2470    *
                  2480    *   SYSTEM MESSAGE OUTpUT CONTROL, TRANSFER TO REQUESTED MESSAGE OUTpUT.
                  2490    *
004000  202002  2500          LAC     TEMp2           LOAD THE MESSAGE NUMBER
004001  353300  2510          TAD     (MSGPTR)
004002  042002  2520          DAC     TEMP2           SET THE ENTRY TO THE TRANSFER VECTOR
004003  622002  2530          JMP     TEMP2,X         AND MAKE THE TRANSFER
```

MAIN MONITOR ROUTINES

```
                    2540            .STITL  MAIN MONITOR ROUTINES
                    2550            .HEAD   M
                    2560    *
                    2570    *       START OF SYSTEM MONITOR
                    2580    *
    004004          2590    MONMSG  MESS    <TSSMON >,7.
    004012          2600    MONSYM  ...
004012  142002      2610            DZM     TEMP2           KILL ANY RECORD OF THE ERROR MESSAGE JUST PRINTED
    004013          2620            MESS    <#>,1
    004017          2630            LINE                    GET A LINE OF TELETYPE INPUT
    004020          2640            MPOFF
                                    .PMC    SAVE,ON
004020  705000                      SPECIAL+0               TURN OFF MEMORY PROTECT
004021  140006      2650            DZM     SCNTRL          DE-ALLOCATE CONTROL AT THE END OF EACH LINE
004022  701742      2660            MPEU
004023  103172      2670    MONXT   JMS     EOL             WAS THE PREVIOUS WORD THE LAST ONE ON THIS LINE? RETURN IF NOT.
    004024          2680            WORD                    GET THE NEXT WORD -- HOPEFULLY A COMMAND
004025  004023      2690            JMP     MONXT           IGNORE VACUOUS COMMANDS
                    2700    *
                    2710    *       IDENTIFY THE MONITOR COMMAND
                    2720    *
004026  764752      2730    MONX2   LAW     CLIST-1         LOAD A POINTER TO THE COMMAND LIST
004027  103174      2740            JMS     SEARCH          AND FIND THE COMMAND
    004030          2750            WHAT                    COULDN'T FIND THE COMMAND
004031  020010      2760            JMP     10,X            ELSE EXECUTE THE COMMAND
```

```
                    2770            ,EJECT
                    2780    *
                    2790    *
                    2800    *       TRY TO ALLOCATE AN AVAILABLE SYSTEM RESOURCE
                    2810    *
    004032          2820    ON      ...
004032  213301      2830            LAC     (NOP)           THIS PERMITS "ON" AND "OFF" TO USE A COMMON ROUTINE
004033  741000      2840            SKP
                    2850    *
                    2860    *       TRY TO DE-ALLOCATE AN AVAILABLE SYSTEM RESOURCE
                    2870    *
    004034          2880    OFF     ...
004034  213302      2890            LAC     (INX 10)        THIS PERMITS "ON" AND "OFF" TO USE A COMMON ROUTINE
004035  042001      2900            DAC     TEMP1           SET WHETHER THE COMMAND IS "ON" OR "OFF"
                    2910    *
                    2920    *       COMMON ROUTINE TO ALLOCATE OR TO DE-ALLOCATE AN AVAILABLE SYSTEM RESOURCE
                    2930    *
    004036          2940            WORD                    GET THE RESOURCE NAME
    004037          2950            WHAT                    NO NAME -- DON'T BELIEVE IT
004040  765051      2960            LAW     RSRC5-1         LOAD A POINTER TO THE RESOURCES LIST
004041  103174      2970            JMS     SEARCH          FIND THE RESOURCE
004042  604047      2980            JMP     RSRC5           DONE -- NOT A STANDARD RESOURCE -- IS IT A NUMBERED ONE?
004043  420010      2990            XCT     10,X            LOAD A POINTER TO THE  RESOURCE ALLOCATION TABLE
004044  042000      3000            DAC     TEMP0           AND PASS IT TO THE ALLOCATION/DE-ALLOCATION ROUTINES
004045  402001      3010            XCT     TEMP1           MAKE THE ON/OFF DISTINCTION
004046  620010      3020            JMP     10,X            EXECUTE THE REQUEST
                    3030
    004047          3040    RSRC5   ...
    004047          3050            WORD1                   LOAD THE RESOURCE NAME
004050  042000      3060            DAC     TEMP0           SAVE IT
004051  513303      3070            AND     (777700)        STRIP OFF THE DIGIT
004052  043326      3080            DAC     TSWORDB         REPLACE IT
004053  200010      3090            LAC     10              LOAD A POINTER TO THE NUMBERED RESOURCES LIST
004054  103174      3100            JMS     SEARCH          LOCATE THE NUMBERED RESOURCE NAME
004055  604061      3110            JMP     RSRC7           DONE -- NOT A LEGAL RESOURCE
004056  440010      3120            INX     10              SKIP THE NONSENSE ENTRY
004057  402001      3130            XCT     TEMP1           MAKE THE ON/OFF DISTINCTION
004060  620010      3140            JMP     10,X            EXECUTE THE REQUEST
                    3150
    004061          3160    RSRC7   ...
004061  202000      3170            LAC     TEMP0           RELOAD THE ORIGINAL WORD
004062  043326      3180            DAC     TSWORDB         REPLACE IT
004063  604026      3190            JMP     MONX2           NOT A LEGAL RESOURCE -- WAS IT A COMMAND?
                    3200
    004064          3210    RON     MPOFF                   TURN OFF MEMORY PROTECT
                                    .PMC    SAVE,ON
004064  705000                      SPECIAL+0               TURN OFF MEMORY PROTECT
004065  222000      3220            LAC     TEMP0,X
004066  541771      3230            SAD     SNUMBR
004067  604074      3240            JMP     RON1            IGNORE THE COMMAND IF THE RESOURCE ALREADY BELONGS TO THIS USER
004070  740200      3250            SZA                     SKIP IF THE RESOURCE IS FREE
004071  611540      3260            JMP     MSG51           ELSE ERROR MESSAGE
```

M MAIN MONITOR ROUTINES

```
004072  201771  3270          LAC     SNUMBR          LOAD THE USER #
004073  062000  3280          DAC     TEMP0,X         RESERVE THE RESOURCE FOR THIS USER
004074  701742  3290  RQN1    MPEU
004075  604023  3300          JMP     MONXT           GET THE NEXT MONITOR COMMAND
                3310
    004076      3320  RQFF    MPOFF                   TURN OFF MEMORY PROTECT
                        ,PMC    SAVE,ON
004076  705000                SPECIAL+0               TURN OFF MEMORY PROTECT
004077  200006  3330          LAC     SCNTRL
004100  541771  3340          SAD     SNUMBR          DOES THIS USER HAVE THE CONTROL LINE?
004101  604110  3350          JMP     RQK             IF SO, ALLOW THE DE-ALLOCATION WITHOUT FURTHER CHECKS
004102  222000  3360          LAC     TEMP0,X
004103  741200  3370          SNA                     SKIP IF THE RESOURCE IS ALLOCATED
004104  604110  3380          JMP     RQK             ELSE CONSIDER IT SUCCESSFULLY DE-ALLOCATED
004105  541771  3390          SAD     SNUMBR          SKIP IF ALLOCATED TO ANOTHER USER
004106  741000  3400          SKP
004107  611557  3410          JMP     MSG52           ELSE ERROR MESSAGE
004110  162000  3420  RQK     DZM     TEMP0,X         DE-ALLOCATE THE RESOURCE
004111  701742  3430          MPEU
004112  604023  3440          JMP     MONXT           GET THE NEXT MONITOR COMMAND
```

```
M                                   MTSS CATALOG MODULE

                    3450            .STITL  MTSS CATALOG MODULE
    003170          3460            .USE    IMPURE
                    3470
                    3480
003170  000000      3490    CNAM    .DSA
003171  000000      3500    NHED    .DSA
    004113          3510            .USE    PURE
004113  565045      3520    NHE     .ACI6   *NHE*
    002170          3530    CATLOG  .EQU    BUFFER
    002170          3540    CFREE   .EQU    CATLOG          DEVICE ADDRESS OF THE FIRST FREE BLOCK
    002172          3550    CNUM    .EQU    CATLOG+2        MINUS THE NUMBER OF ENTRIES IN THIS CATLOG
    002173          3560    CMAX    .EQU    CATLOG+3        MAXIMUM NUMBER OF BLOCKS ON THE DEVICE
                    3570
    004114          3580            .USE    PURE
    004114          3590    CAT     ...
    004114          3600            MESS    <CAT>,3         PRINT THE HEADER MESSAGE
004121  103314      3610            JMS     FORCE           AND FORCE THE OLD BUFFER BEFORE OVER-WRITING IT
    004122          3620    CNXL    ...
    004122          3630            CRLF
    004123          3640            MESS    <?>,1
004127  143170      3650            DZM     CNAM            INITIALIZE THE FILE NAME REQUEST
004130  143171      3660            DZM     NHED
    004131          3670            LINE                    GET THE NEXT LINE OF INPUT -- PRESUMABLY A CATALOG COMMAND
    004132          3680    CAT2    ...
    004132          3690            WORD                    GET THE NEXT WORD OF INPUT
    004133          3700            FORMAT
004134  544113      3710            SAD     NHE             CHECK FOR NO HEADER OPTION
004135  604372      3720            JMP     NHEAD           YES -- SET THE FLAG
    004136          3730            CRLF
004137  764143      3740            LAW     CAT4            LOAD THE RETURN ADDRESS
004140  043532      3750            DAC     CSDEVCV         SET IT IN THE SUBROUTINE
    004141          3760            WORD1                   RECOVER THE WORD
004142  611121      3770            JMP     CSDEVC3         MAKE IT INTO A DEVICE ADDRESS
004143  604263      3780    CAT4    JMP     CAT8            EITHER IT WAS A PAPER TAPE REQUEST (ILLEGAL) OR AN EXIT
004144  103510      3790            JMS     CSRCAT          GET THE REQUESTED CATALOG
    004145          3800            DELIM                   GET THE DELIMITER
004146  553304      3810            SAD     (SCOLON)        CHECK FOR A COLON (INDICATES A FILENAME)
004147  741000      3820            SKP                     YES
004150  604154      3830            JMP     CAT6            NO -- CARRY ON
    004151          3840            WORD                    GET THE FILENAME
004152  740000      3850            NOP                     DON'T CARE ABOUT A NULL FILENAME
004153  043170      3860            DAC     CNAM            SET THE FILENAME
    004154          3870    CAT6    ...
    004154          3880            CRLF
004155  202172      3890            LAC     CNUM
004156  040011      3900            DAC     11              INITIALIZE THE NUMBER OF ENTRIES IN THE CATALOG
004157  762173      3910            LAW     CMAX
004160  040010      3920            DAC     10              INITIALIZE THE POINTER TO THE FILES IN THE CATALOG
                    3930    *
                    3940    *       PRINT THE HEADER
                    3950    *
004161  203171      3960            LAC     NHED            LOAD THE NO-HEADER FLAG
```

```
M                                MTSS CATALOG MODULE

004162  740200  3970           SZA                     SKIP UNLESS IT IS SET
004163  604204  3980           JMP     CNEXT           IT IS SET -- SKIP PRINTING THE HEADER
004164          3990           MESS    <NAME BLOCK ADDRESS LENGTH XFER>,30.
004202          4000           CRLF                    SPACE DOWN FROM THE HEADING ONE LINE
004203          4010           CRLF
                4020    *
                4030    *      PRINT THE NEXT NAME
                4040    *
004204          4050    CNEXT  ...
004204  220010  4060           LAC     10,X            LOAD THE NAME
004205  741200  4070           SNA                     SKIP IF THERE IS A NAME
004206  604215  4080           JMP     CAT5            ELSE DON'T PRINT THIS ENTRY
004207  042001  4090           DAC     TEMP1           SAVE THE FILE NAME
004210  203170  4100           LAC     CNAM            LOAD THE NAME TO MATCH (ZERO IF NONE)
004211  741200  4110           SNA                     SKIP IF THERE IS ONE
004212  604222  4120           JMP     CAT65           ELSE PRINT ALL ENTRIES
004213  542001  4130           SAD     TEMP1           CHECK FOR A MATCH
004214  604223  4140           JMP     CAT7            MATCH -- PRINT IT
004215          4150    CAT5   ...
004215  440010  4160           INX     10              NO MATCH -- AVOID THE UNWANTED BLOCK NUMBER
004216  440010  4170           INX     10              AVOID THE UNWANTED CORE ADDRESS
004217  440010  4180           INX     10              AVOID THE UNWANTED FILE LENGTH
004220  440010  4190           INX     10              AVOID THE UNWANTED TCD
004221  604263  4200           JMP     CAT8            CHECK FOR DONE
004222          4210    CAT65  ...
004222  202001  4220           LAC     TEMP1           RELOAD THE FILE NAME
004223          4230    CAT7   ...
004223  103223  4240           JMS     DSAMOD          PRINT THE SIXBIT NAME
                4250    *
                4260    *      PRINT THE BLOCK NUMBER
                4270    *
004224  103316  4280           JMS     DSSPACE
004225  103316  4290           JMS     DSSPACE         START IN ITS OWN FIELD
004226  220010  4300           LAC     10,X            GET THE BLOCK NUMBER
004227  640512  4310           LRS     10.             SAVE THE BLOCK NUMBER
004230  641601  4320           EAECLA!LLS 1
004231  353305  4330           TAD     (260)
004232  103503  4340           JMS     TSTTYOT         PRINT THE FIRST DIGIT
                4350           .DUP    3,3
004233  641603  4360           EAECLA!LLS 3            GET THE NEXT DIGIT
004234  353305  4370           TAD     (260)           MAKE IT ASCII
004235  103503  4380           JMS     TSTTYOT         PRINT IT
004236  641603                 EAECLA!LLS 3            GET THE NEXT DIGIT
004237  353305                 TAD     (260)           MAKE IT ASCII
004240  103503                 JMS     TSTTYOT         PRINT IT
004241  641603                 EAECLA!LLS 3            GET THE NEXT DIGIT
004242  353305                 TAD     (260)           MAKE IT ASCII
004243  103503                 JMS     TSTTYOT         PRINT IT
                4390    *
                4400    *      PRINT THE STARTING ADDRESS
                4410    *
004244  103316  4420           JMS     DSSPACE
```

M MTSS CATALOG MODULE

```
004245  103316  4430            JMS     D$SPACE         START IT IN ITS OWN FIELD
004246  220010  4440            LAC     10,X            LOAD THE STARTING CORE ADDRESS
    004247      4450            OCT                     AND PRINT IT
                4460    *
                4470    *       PRINT THE FILE'S LENGTH
                4480    *
004251  103316  4490            JMS     D$SpACE
004252  103316  4500            JMS     D$SPACE         START IT IN ITS OWN FIELD
004253  220010  4510            LAC     10,X            LOAD THE LENGTH
    004254      4520            OCT                     AND PRINT IT
                4530    *
                4540    *       PRINT THE TRANSFER CARD
                4550    *
004256  103316  4560            JMS     D$SpACE         START IT IN ITS OWN FIELD
004257  220010  4570            LAC     10,X            LOAD THE TRANSFER
    004260      4580            OCT                     AND PRINT IT
    004262      4590            CRLF
                4600    *
                4610    *       CHECK FOR DONE -- IF So, PRINT THE TRAILER
                4620    *
004263  440011  4630    CAT8    ISZ     11              COUNT THE FILE
004264  604204  4640            JMP     CNEXT           NOT YET DONE -- DO THE NEXT ENTRY
    004265      4650            CRLF
004266  203171  4660            LAC     NHED            LOAD THE NO-HEADER FLAG
004267  740200  4670            SZA                     SKIP UNLESS SET
004270  604122  4680            JMP     CNXL            IF SET, DON'T PRINT A TRAILER, EITHER
    004271      4690            CRLF                    ELSE SPACE DOWN
004272  202170  4700            LAC     CFREE           LOAD THE FIRST FREE BLOCK POINTER
004273  513306  4710            AND     (037777)
    004274      4720            OCTZ
    004276      4730            NMESS   < IS THE NEXT FREE BLOCK>,23.
    004310      4740            MESS    <CATALOG LAST MODIFIED ON >,25.
004324  202170  4750            LAC     CATLOG          GET THE DEVICE ADDRESS
004325  513307  4760            AND     (040000)        KEEP THE DEVICE BIT
004326  741200  4770            SNA
004327  604335  4780            JMP     C4
    004330      4790            NMESS   <DISK >,5
004334  604342  4800            JMP     C2
    004335      4810    C4      NMESS   <DECTAPE >,8,
004342  202170  4820    C2      LAC     CATLOG          RELOAD THE DEVICE ADDRESS
004343  744000  4830            CLL                     PROTECT THE ROTATE
004344  640517  4840            LRS     18.-3
004345  353305  4850            TAD     (260)           MAKE THE DIGIT INTO ASCII
004346  103503  4860            JMS     TSTTYOT         AND PRINT IT
    004347      4870            CRLF
    004350      4880            MESS    <      ***END OF CATALOG***>,26.
004365  604122  4890            JMP     CNXL            SEE IF THERE IS ANOTHER COMMAND
                4900
    004366      4910    CAT9    ,,,
    004366      4920            DELIM                   TELL BY THE DELIMITER WHICH IT IS
004367  553310  4930            SAD     (SUPARR)
004370  604012  4940            JMP     MONSYM          EXIT ON UP-ARROW
```

```
                M                               MTSS CATALOG MODULE

    004371  604023  4950                JMP     MONXT           SEE IF IT IS A MONITOR COMMAND
                    4960    *
                    4970    *
                    4980    *           CATALOG MODULE OPTIONAL COMMANDS
                    4990    *
        004372      5000    NHEAD   ...,
    004372  443171  5010                INX     NHED            FLAG THE REQUEST
    004373  604132  5020                JMP     CAT2            RETURN FOR ANOTHER COMMAND
```

M MISCELLANEOUS SUBROUTINES AND STORAGE

```
                    5030            ,STITL  MISCELLANEOUS SUBROUTINES AND STORAGE
    004374          5040            ,USE    PURE
                    5050    *
                    5060    *
                    5070    *       CHECK TO SEE WHETHER THE DELIMITER OF THE LAST WORD WAS A
                    5080    *       CARRIAGE RETURN, IF SO, RETURN TO PRINT THE MONITOR SYMBOL
                    5090    *       AND GET THE NEXT LINE OF INPUT; ELSE RETURN TO THE CALLER.
                    5100    *
    004374          5110            ENTER   EOL
                                    ,PMC    SAVE,ON
    003172                  EOL     ...
    004374          5120            DELIM                   GET THE DELIMITER OF THE PREVIOUS WORD
004375  553311      5130            SAD     ($CR)           CHECK FOR A CARRIAGE RETURN
004376  604012      5140            JMP     MONSYM          YES -- SO GO GET THE NEXT LINE
004377  623172      5150            RET     EOL,X           NO -- SO RETURN TO THE CALLER
                    5160    *
                    5170    *
                    5180    *       SEARCH THE TABLE INDICATED BY THE POINTER PASSED IN THE
                    5190    *       ACCUMULATOR FOR A MATCH TO THE WORD IN TSWORDB. THE END OF THE
                    5200    *       TABLE IS THE FIRST LOCATION CONTAINING MINUS 1. RETURN TO THE
                    5210    *       CALLER +1 IF NO MATCH IS FOUND; +2 IF A MATCH IS FOUND.
                    5220    *
    004400          5230            ENTER   SEARCH
                                    ,PMC    SAVE,ON
    003174                  SEARCH  ...
004400  040010      5240            DAC     10              SET THE PASSED TABLE POINTER
004401  220010      5250    SRCH2   LAC     10,X            LOAD THE NEXT TABLE ENTRY
004402  645051      5260            SAD     M1              CHECK FOR DONE
004403  623174      5270            RET     SEARCH,X        YES -- NO MATCH -- RETURN +1
004404  543326      5280            SAD     TSWORDB         ELSE CHECK FOR A MATCH
004405  741000      5290            SKP
004406  604401      5300            JMP     SRCH2           NO MATCH FOUND -- TRY THE NEXT ENTRY
004407  443174      5310            INX     SEARCH          MATCH FOUND -- BUMP THE RETURN
004410  623174      5320            RET     SEARCH,X        SUCCESSFUL RETURN +2
```

```
M                                   REQUESTS FOR OTHER PROGRAMS

                    5330            ,STITL  REQUESTS FOR OTHER PROGRAMS
        004411      5340            ,USE    PURE
                    5350    *
                    5360    *
                    5370    *       REQUESTS FOR SYSTEM-SUPPLIED USER PROGRAMS
                    5380    *
                    5390
004411  213312      5400    DEB     LAC     ($DDT)
004412  042000      5410            DAC     TEMP0
004413  604421      5420            JMP     MX1
                    5430
                    5440
004414  213313      5450    LDR     LAC     ($LDR)
004415  042000      5460            DAC     TEMP0
004416  604425      5470            JMP     MX2
                    5480
004417  213314      5490    BAS     LAC     ($BAS)
004420  042000      5500            DAC     TEMP0           PASS THE PROGRAM NAME
                    5510
                    5520    *
                    5530    *
                    5540    *       THIS IS THE ONLY NORMAL EXIT FROM THE MONITOR, BEFORE EXIT ALL
                    5550    *       USER REGISTERS MUST BE RESET AND THE SWAPPER PARAMETER WORDS
                    5560    *       MUST BE SET.
                    5570    *
                    5580    *       SET THE SWAPPER PARAMETER WORDS
                    5590    *
        004421      5600    MX1     ;;;
004421  142150      5610            DZM     PHFLAG          SET THE NEXT PROGRAM TO BE A USER PROGRAM
        004422      5620            MPOFF
                                    ,PMC    SAVE,ON
004422  705000                      SPECIAL+0               TURN OFF MEMORY PROTECT
004423  213315      5630            LAC     (506000)        S-USER SWAP CONTROL WORD
004424  604427      5640            JMP     MX3
        004425      5650    MX2     ;;;
        004425      5660            MPOFF
                                    ,PMC    SAVE,ON
004425  705000                      SPECIAL+0               TURN OFF MEMORY PROTECT
004426  213316      5670            LAC     (512000)        PHANTOM SWAP CONTROL WORD
        004427      5680    MX3     ;;;
004427  042001      5690            DAC     TEMP1           SET THE SWAPPER CONTROL WORD
004430  202000      5700            LAC     TEMP0
004431  042000      5710    MX5     DAC     TEMP0           SET THE NAME TO SWAP
004432  701742      5720            MPEU                    ENABLE MEMORY PROTECTION FOR THE FORCE ROUTINE
004433  103314      5730            JMS     FORCE           FLUSH ANY PATCHES BEFORE LEAVING THIS MODULE
        004434      5740            MPOFF
                                    ,PMC    SAVE,ON
004434  705000                      SPECIAL+0               TURN OFF MEMORY PROTECT
                    5750    *
                    5760    *       NOW THE SPECIALS ARE DONE WE CAN SET THE CONTROL WORDS SAFELY
                    5770    *
004435  202001      5780            LAC     TEMP1
```

```
                M                       REQUESTS FOR OTHER PROGRAMS

004436  040702  5790            DAC     $OC0            SET THE SWAPPER CONTROL WORD
004437  202000  5800            LAC     TEMP0
004440  040703  5810            DAC     $OC1            SET THE FILENAME
004441  202002  5820            LAC     TEMP2
004442  040704  5830            DAC     $OC2            SET THE RESTART ADDRESS
004443  700002  5840    MX4     IOF
                5850    *
                5860    *       SET THE USER'S REGISTERS
                5870    *
004444  202022  5880            LAC     ACSW
004445  041756  5890            DAC     $ACS            RESTORE THE SER'S SOFTWARE AC SWITCHES REGISTER
004446  202020  5900            LAC     STSAVE
004447  041760  5910            DAC     $IORS           RESTORE THE USER'S PROGRAM INTERRUPT STATUS
                5920    *
                5930    *       RESTORE THE USER'S STEP COUNTER
                5940    *
004450  202021  5950            LAC     SCSAVE          RELOAD THE OLD STEP COUNT
004451  253317  5960            XOR     (77)            COMPLEMENT THE STEP COUNT
004452  353320  5970            TAD     (640402)        DEVELOP A PSEUDO-NORMALIZE INSTRUCTION
004453  513321  5980            AND     (640477)        DELETE A POSSIBLE STEP COUNT OVERFLOW
004454  043176  5990            DAC     MST2            SET THE NORMALIZE INSTRUCTION
    003176      6000            .USE    IMPURE
003176  740040  6010    MST2    XX
    004455      6020            .USE    PURE
004455  403176  6030            XCT     MST2            STEP COUNT TO THE SC REGISTER
                6040
004456  202016  6050            LAC     MQSAVE
004457  652000  6060            LMQ                     RESTORE THE USER'S MQ
004460  202015  6070            LAC     ACSAVE
004461  040005  6080            DAC     $3AC            RESTORE THE USER'S AC
004462  202023  6090            LAC     10SAVE
004463  040026  6100            DAC     $,310           RESTORE THE USER'S AUTO-INDEX REGISTER 10
004464  202024  6110            LAC     11SAVE
004465  040027  6120            DAC     $,311           RESTORE THE USER'S AUTO-INDEX REGISTER 11
004466  201771  6130            LAC     $NUMBR
004467  040055  6140            DAC     $3TEM4          SET THE CURRENT USER TO ALSO BE THE NEXT USER
004470  761001  6150            LAW     $SWPPR
004471  600335  6160            JMP     $SWAP           GET THE SWAPPER
```

```
M                                   DECTAPE ALLOCATION/DE-ALLOCATION

                    6170            .STITL  DECTAPE ALLOCATION/DE-ALLOCATION
     004472         6180            .USE    PURE
                    6190    *
                    6200    *
                    6210    *
                    6220
     004472         6230    DTON    ...                     TRY TO ALLOCATE THE DECTAPE UNIT
     004472         6240            MPOFF
                                    .PMC    SAVE,ON
     004472  705000                 SPECIAL+0               TURN OFF MEMORY PROTECT
     004473  103203 6250            JMS     DTNUM           SET UP THE ALLOCATION TAG AND CHECK FOR A FORMAT ERROR
     004474  744000 6260            CLL                     INITIALIZE FREE HANDLER FLAG
     004475  200032 6270            LAC     SRDT0           LOAD THE CURRENT STATUS OF ONE HANDLER
     004476  103177 6280            JMS     DTON6           CHECK ITS ALLOCATION
     004477  200033 6290            LAC     SRDT1           LOAD THE CURRENT STATUS OF THE OTHER HANDLER
     004500  103177 6300            JMS     DTON6           AND CHECK ITS ALLOCATION
                    6310
                    6320    *       NO ONE HAS ALREADY BEEN ALLOCATED THE DECTAPE TRANSPORT REQUESTED
                    6330
     004501  740400 6340            SNL                     SKIP IF THERE IS A FREE HANDLER
     004502  611624 6350            JMP     MSG54           BOTH HANDLERS HAVE BEEN ALLOCATED
     004503  200032 6360            LAC     SRDT0
     004504  740200 6370            SZA                     SKIP IF THE FIRST HANDLER IS FREE
     004505  604511 6380            JMP     DTON1           ELSE USE THE OTHER
     004506  202000 6390            LAC     TEMP0           LOAD THE TAG
     004507  040032 6400            DAC     SRDT0           AND ALLOCATE THE HANDLER
     004510  604513 6410            JMP     DTON9           EXIT
     004511  202000 6420    DTON1   LAC     TEMP0           LOAD THE TAG
     004512  040033 6430            DAC     SRDT1           ALLOCATE THE OTHER HANDLER
     004513  701742 6440    DTON9   MPEU
     004514  604023 6450            JMP     MONXT
                    6460
     004515         6470            ENTER   DTON6           ROUTINE TO SEE WHETHER OR NOT A DECTAPE HANDLER CAN BE ALLOCATED
                                    .PMC    SAVE,ON
     003177                 DTON6   ...
     004515  740200 6480            SZA                     SKIP IF THE HANDLER IS FREE
     004516  604521 6490            JMP     DTON7           ELSE CHECK WHO HAS WHAT
     004517  744002 6500            STL                     FLAG THERE IS A FREE HANDLER
     004520  623177 6510            RET     DTON6,X
     004521  542000 6520    DTON7   SAD     TEMP0
     004522  604513 6530            JMP     DTON9           EXIT -- IT ALREADY BELONGS TO THIS USER
     004523  513322 6540            AND     (770000)
     004524  542001 6550            SAD     TEMP1
     004525  611540 6560            JMP     MSG51           RESOURCE ALREADY ALLOCATED TO ANOTHER USER
     004526  623177 6570            RET     DTON6,X
                    6580
     004527         6590    DTOFF   ...                     TRY TO DE-ALLOCATE THE DECTAPE UNIT
     004527         6600            MPOFF
                                    .PMC    SAVE,ON
     004527  705000                 SPECIAL+0               TURN OFF MEMORY PROTECT
     004530  200006 6610            LAC     SCNTRL
     004531  541771 6620            SAD     SNUMBR          SEE IF THE CURRENT USER HAS A CONTROL LINE
```

M DECTAPE ALLOCATION/DE-ALLOCATION

```
004532  741000  6630            SKP
004533  604536  6640            JMP     DTOF4           NO -- CONTINUE NORMALLY
004534  770000  6650            LAW     10000           LOAD UNIT NUMBER MASK
004535  741000  6660            SKP
004536  777777  6670  DTOF4     LAW     -1              LOAD WHOLE WORD ALLOCATION MASK
004537  043326  6680            DAC     T$WORDB
004540  103203  6690            JMS     DTNUM           SET UP THE ALLOCATION TAG AND CHECK FOR A FORMAT ERROR
004541  202000  6700            LAC     TEMP0           RECOVER THE DEVICE NAME
004542  503326  6710            AND     T$WORDB         MODIFY IT BY THE MASK
004543  042000  6720            DAC     TEMP0
004544  760032  6730            LAW     $RDT0           LOAD A POINTER TO ONE OF THE HANDLERS
004545  103201  6740            JMS     DTOF6           AND CHECK ON IT
004546  760033  6750  DTOF8     LAW     $RDT1           LOAD A POINTER TO THE OTHER HANDLER
004547  103201  6760            JMS     DTOF6           AND CHECK ON IT
004550  701742  6770  DTOF9     MPEU
004551  604023  6780            JMP     MONXT
                6790
    004552      6800            ENTER   DTOF6           ROUTINE TO SEE WHETHER OR NOT A DECTAPE HANDLER CAN BE DE-ALLOCATED
                                ,PMC    SAVE,ON
    003201              DTOF6   ...
004552  043177  6810            DAC     DTON6           SET THE POINTER
004553  223177  6820            LAC     DTON6,X         LOAD THE ALLOCATION
004554  741200  6830            SNA                     SKIP IF IT IS SOMEONE'S RESOURCE
004555  623201  6840            RET     DTOF6,X         ELSE EXIT NOW
004556  503326  6850            AND     T$WORDB         RECOVER JUST THAT PART OF THE TAG WE ARE INTERESTED IN

004557  542000  6860            SAD     TEMP0
004560  604565  6870            JMP     DTOF7           FOUND THIS USER'S ALLOCATION -- REMOVE IT
004561  513322  6880            AND     (770000)
004562  542001  6890            SAD     TEMP1
004563  611557  6900            JMP     MSG52           NOT THIS USER'S RESOURCE
004564  623201  6910            RET     DTOF6,X
                6920
004565  163177  6930  DTOF7     DZM     DTON6,X         RELEASE THE DECTAPE HANDLER
004566  604546  6940            JMP     DTOF8           AND EXIT
                6950  *
                6960  *
                6970  *       DTNUM IS A SUBROUTINE USED BY THE DECTAPE ALLOCATION AND DE-ALLOCATION
                6980  *       ROUTINES TO SET UP THE ALLOCATION TAG AND TO CHECK FOR INPUT
                6990  *       OF A NON-OCTAL DIGIT.
                7000  *
    004567      7010            ENTER   DTNUM
                                ,PMC    SAVE,ON
    003203              DTNUM   ...
004567  202000  7020            LAC     TEMP0           RECOVER THE DEVICE NAME
004570  650614  7030            CLQ!LLS 12.             RETAIN JUST THE THIRD SIXBIT CHARACTER
004571  513323  7040            AND     (570000)        RETAIN JUST THE DIGIT (PLUS A FLAG FOR NON-OCTAL DIGIT)
004572  042001  7050            DAC     TEMP1           SAVE THE DIGIT
004573  341771  7060            TAD     $NUMBR          FORM THE ALLOCATION TAG
004574  042000  7070            DAC     TEMP0           SET THE ALLOCATION TAG
004575  513324  7080            AND     (700000)
004576  741200  7090            SNA
004577  623203  7100            RET     DTNUM,X         GOOD EXIT -- THE DIGIT WAS OCTAL
```

M DECTAPE ALLOCATION/DE-ALLOCATION

```
004600  701742  7110        MPEU        ELSE RE-ENABLE USER MODE BEFORE TRYING TO PRINT
004601          7120        FORMAT      AND GO PRINT AN ERROR MESSAGE
```

```
                M                       CONSOLE SWITCHES TYPE SERVICES

                        7130            .STITL  CONSOLE SWITCHES TYPE SERVICES
            004602      7140            .USE    PURE
                        7150    *
                        7160    *       REQUESTS FOR CONSOLE-SWITCH TYPE SERVICES
                        7170    *
004602  703302          7180    BYE     CAF                     CLEAR ALL OF THE USER'S FLAGS
004603  103207          7190            JMS     ZCORE           ZERO ALL OF THE USER'S CORE
004604  103211          7200            JMS     ZCOR1           ZERO THE FIRST 40 LOCATIONS
004605  103213          7210            JMS     ZDISK           ZERO ALL OF THE USER'S "PHYSICAL DISK"
            004606      7220            MPOFF
                                        .PMC    SAVE,ON
004606  705000                          SPECIAL+0               TURN OFF MEMORY PROTECT
004607  760034  7230                    LAW     $RACS
004610  103205  7240                    JMS     B5              DE-ALLOCATE AC SWITCHES
004611  760230  7250                    LAW     $RPTP
004612  103205  7260                    JMS     B5              DE-ALLOCATE PAPER TAPE PUNCH
004613  760235  7270                    LAW     $RPTR
004614  103205  7280                    JMS     B5              DE-ALLOCATE PAPER TAPE READER
004615  760242  7290                    LAW     $RSCO
004616  103205  7300                    JMS     B5              DE-ALLOCATE THE GRAPHICS II PERIPHERALS
004617  760032  7310                    LAW     $RDT0
004620  103205  7320                    JMS     B5              DE-ALLOCATE THE FIRST DECTAPE HANDLER
004621  760033  7330                    LAW     $RDT1
004622  103205  7340                    JMS     B5              DE-ALLOCATE THE SECOND DECTAPE HANDLER
004623  701742  7350                    MPEU
004624  604004  7360                    JMP     MONMSG          ALL DONE -- PRINT A FRESH MONITOR MESSAGE
                        7370    *
                        7380    *
                        7390    *       ENTER WITH A POINTER TO A RESOURCE ALLOCATION WORD IN THE AC.
                        7400    *       DE-ALLOCATE THE RESOURCE IF IT BELONGS TO THE CURRENT USER.
                        7410    *       OTHERWISE LEAVE IT ALONE.
                        7420    *
            004625      7430            ENTER   B5
                                        .PMC    SAVE,ON
            003205                      B5      ...
004625  042000  7440                    DAC     TEMP0           SAVE THE POINTER TO THE RESOURCE ALLOCATION WORD
004626  222000  7450                    LAC     TEMP0,X         LOAD THE CURRENT ALLOCATION BITS
004627  513325  7460                    AND     (777)           KEEP JUST THE USER NUMBER PORTION
004630  541771  7470                    SAD     $NUMBR          COMPARE TO THIS USER'S ID
004631  162000  7480                    DZM     TEMP0,X         IT'S HIS -- DE-ALLOCATE IT
004632  623205  7490                    RET     B5,X            EXIT
                        7500    *
                        7510    *
004633  703302  7520    CAF     CAF                             CLEAR ALL OF THE USER'S FLAGS
004634  142020  7530            DZM     STSAVE                  ALSO CLEAR ALL SOFTWARE FLAGS
004635  604023  7540            JMP     MONXT                   GET THE NEXT COMMAND
                        7550    *
                        7560    *
                        7570    *       REQUEST TO ZERO A USER FILE -- DETERMINE WHICH ONE
                        7580    *
                        7590
004636  435762  7600    CORE    .AC16   +COR+
```

M CONSOLE SWITCHES TYPE SERVICES

```
004637  445163  7610    DISK    .ACI6   *DIS*
                7620
    004640      7630    ZER     ...
004640  103314  7640            JMS     DSFORCE
    004641      7650            WORD                    GET THE NAME OF THE FILE TO ZERO
    004642      7660            WHAT                    NULL FILE NAME IS ILLEGAL
004643  544637  7670            SAD     DISK
004644  604653  7680            JMP     ZDIS            ZERO THE USER DISK
004645  544636  7690            SAD     CORE
004646  604650  7700            JMP     ZCOR            ZERO USER CORE
    004647      7710            WHAT                    NO OTHER LEGAL FILENAMES EXIST
                7720    *
004650  103207  7730    ZCOR    JMS     ZCORE           ZERO MOST OF USER CORE
004651  103211  7740            JMS     ZCOR1           ZERO LOCATIONS 0-37
004652  604023  7750            JMP     MONXT           GET THE NEXT COMMAND
                7760    *
004653  103213  7770    ZDIS    JMS     ZDISK           ZERO THE USER "PHYSICAL DISK"
004654  604023  7780            JMP     MONXT           GET THE NEXT COMMAND
                7790    *
                7800    *
                7810    *       ZERO ALL USER CORE, EXCEPT THE 40 LOCATIONS IN THE USER JOB TABLE
                7820    *
    004655      7830            ENTER   ZCORE
                                ,PMC    SAVE,ON
    003207                      ZCORE   ...
004655  764000  7840            LAW     BOUNDARY-CORMAX  TWO'S COMPLEMENT NUMBER OF LOCATIONS TO ZERO
004656  042013  7850            DAC     TEMP11
004657  762000  7860            LAW     BOUNDARY
004660  042014  7870            DAC     TEMP12          SET THE STARTING LOCATION TO ZERO
004661  103324  7880            JMS     DSBINIT         INITIALIZE THE BUFFER
004662  103306  7890            JMS     DSUCORE         OPEN USER CORE
004663  604705  7900            JMP     ZER2            DO THE ZEROING OPERATION
                7910    *
                7920    *
                7930    *       ZERO THE FIRST 40 LOCATIONS
                7940    *
    004664      7950            ENTER   ZCOR1
                                ,PMC    SAVE,ON
    003211                      ZCOR1   ...
004664  203211  7960            LAC     ZCOR1
004665  043207  7970            DAC     ZCORE           SAVE THE RETURN ADDRESS
004666  777740  7980            LAW     -40
004667  042013  7990            DAC     TEMP11          SET THE LENGTH TO ZERO
004670  750000  8000            CLA
004671  042014  8010            DAC     TEMP12          SET THE START ADDRESS TO ZERO
004672  604705  8020            JMP     ZER2            DO IT -- ZERO THE FIRST 40 CORE LOCATIONS (IN THE JOB TABLE)
                8030    *
                8040    *
                8050    *       ZERO THE USER'S "PHYSICAL DISK"
                8060    *
    004673      8070            ENTER   ZDISK
                                ,PMC    SAVE,ON
```

M CONSOLE SWITCHES TYPE SERVICES

```
    003213              8070    ZDISK   ...,
004673  203213  8080                    LAC     ZDISK
004674  043207  8090                    DAC     ZCORE           SAVE THE RETURN ADDRESS
004675  762000  8100                    LAW     -$DKLEN
004676  042013  8110                    DAC     TEMP11          SET THE DISK LENGTH
004677  142014  8120                    DZM     TEMP12          SET THE START ADDRESS
004700  103324  8130                    JMS     D$BINIT         INITIALIZE THE BUFFER AREA
004701  103310  8140                    JMS     D$UDISK         OPEN THE USER DISK FILE
004702  142151  8150                    DZM     D$FTYPE         SET THE FILE TYPE
004703  750001  8160                    CLC
004704  042051  8170                    DAC     D$PCMSK         SET A FULL WORD MASK FOR THE DISK
                8180
    004705      8190    ZER2    ...,
004705  103304  8200                    JMS     D$SNCOP         FINISH OPENING THE FILE
                8210
    004706      8220    ZER4    ...,
004706  202014  8230                    LAC     TEMP12          LOAD THE NEXT LOCATION TO CLEAR
004707  103312  8240                    JMS     D$LOCAT         LOCATE IT IN THE BUFFER
004710  162162  8250                    DZM     D$BPTR,X        ZERO IT
004711  442156  8260                    INX     D$BALT          SET THE ALTERS FLAG
004712  442014  8270                    INX     TEMP12          INCREMENT THE POINTER
004713  442013  8280                    ISZ     TEMP11          TEST FOR DONE
004714  604706  8290                    JMP     ZER4            NO -- LOOP
004715  103314  8300                    JMS     D$FORCE         YES -- FORCE THE LAST BUFFER FULL OUT
004716  623207  8310                    RET     ZCORE,X         EXIT
                8320    *
                8330    *
                8340    *       EXPLAIN SYSTEM COMMANDS
                8350    *
                8360    *       LIST OF AVAILABLE EXPLAIN FUNCTIONS
                8370    *
004717  435755  8380    COMND   ,ACI6   +COM+
004720  624563  8390    RESRC   ,ACI6   +RES+
    004721      8400    EXP     ...,
    004721      8410            WORD                            GET THE ITEM TO EXPLAIN
004722  611762  8420                    JMP     MSG63           NO NULL COMMAND FOR NOW
004723  544717  8430                    SAD     COMND
004724  612043  8440                    JMP     MSG64           COMMAND LIST
004725  544720  8450                    SAD     RESRC
004726  612676  8460                    JMP     MSG65           RESOURCE LIST
004727  611762  8470                    JMP     MSG63           ELSE GENERAL MESSAGE
```

```
M                                   MTSS DEBUGGER -- X-RATED COMMANDS

                    8480            ,STITL  MTSS DEBUGGER -- X-RATED COMMANDS
                    8490    *
                    8500    *
                    8510    *       COMMANDS WHICH WORK ON ACTUAL CORE FIRST SET THE FILE TYPE
                    8520    *       TO ACTUAL CORE IF IT IS NOT ALREADY SET, AND SAVE THE OLD FILE TYPE.
                    8530    *
                    8540    *       USE EXIT (EXI,X) TO GET BACK TO THE PREVIOUSLY OPEN FILE.
                    8550    *
    004730          8560    XDU     ...                             DUMP ACTUAL CORE
    004730          8570            NUM
    004731          8580            WHAT
004732  042001      8590            DAC     TEMP1
    004733          8600            CRLF
    004734          8610            MPOFF
                                    ,PMC    SAVE,ON
004734  705000                      SPECIAL+0                       TURN OFF MEMORY PROTECT
004735  222001      8620            LAC     TEMP1,X
004736  701742      8630            MPEU
    004737          8640            OCT
004741  604023      8650            JMP     MONXT
                    8660
                    8670
    004742          8680    XPA     ...                             PATCH ACTUAL CORE
    004742          8690            NUM
    004743          8700            WHAT
004744  042001      8710            DAC     TEMP1
    004745          8720            NUM
    004746          8730            WHAT
    004747          8740            MPOFF
                                    ,PMC    SAVE,ON
004747  705000                      SPECIAL+0                       TURN OFF MEMORY PROTECT
004750  062001      8750            DAC     TEMP1,X
004751  701742      8760            MPEU
004752  604023      8770            JMP     MONXT
```

```
M                                TABLE OF COMMANDS,RESOURCES, AND REGISTERS

                    8780                .STITL  TABLE OF COMMANDS,RESOURCES, AND REGISTERS
                    8790        *
                    8800        *       DEFINITIONS TO ALLOW FOR UN-IMPLIMENTED FEATURES AND OTHER ANOMALIES
                    8810        *
          004064    8820        SCOON   .EQU    RON
          004076    8830        SCOOFF  .EQU    ROFF
          004064    8840        PTRON   .EQU    RON
          004076    8850        PTROFF  .EQU    ROFF
          004064    8860        PTPON   .EQU    RON
          004076    8870        PTPOFF  .EQU    ROFF
          004064    8880        ACSON   .EQU    RON
          004076    8890        ACSOFF  .EQU    ROFF
          004064    8900        CNTON   .EQU    RON
          004076    8910        CNTOFF  .EQU    ROFF
          004472    8920        TPON    .EQU    DTON
          004527    8930        TPOFF   .EQU    DTOFF
                    8940                .HEAD
          000032    8950        RDT     .EQU    $RDTO
          000032    8960        RTP     .EQU    $RDTO
                    8970                .HEAD   M
                    8980        *
                    8990        *
                    9000        *       MONITOR COMMAND TABLE
                    9010        *
          004753    9020                .USE    PURE
                    9030                .PMC    SAVE,OFF
          004753    9040        CLIST   ...
          004753    9050                COMMAND DEB
          004755    9060                COMMAND LDR,L
          004761    9070                COMMAND CAT
          004763    9080                COMMAND BAS
          004765    9090                COMMAND GRO
          004767    9100                COMMAND ON
          004771    9110                COMMAND OFF
          004773    9120                COMMAND BYE,<GOO,HEL>
          005001    9130                COMMAND EXP,E
          005005    9140                COMMAND CAF
          005007    9150                COMMAND ZER,Z
          005013    9160                COMMAND VAL,V
          005017    9170                COMMAND XDU,XD
          005023    9180                COMMAND XPA,XP
                    9190                .HEAD   D
          005027    9200                COMMAND TRA,<T,JMP,J>
          005037    9210                COMMAND DDT
          005041    9220                COMMAND CON,C
          005045    9230                COMMAND EXI,X
                    9240                .HEAD   M
005051    777777    9250        M1      -1                      END OF MONITOR COMMAND TABLE
                    9260        *
                    9270        *       MONITOR RESOURCE TABLE
                    9280        *
          005052    9290        RSRCS   RESOURCE PTR
```

M TABLE OF COMMANDS,RESOURCES, AND REGISTERS

```
005056       9300            RESOURCE PTP
005062       9310            RESOURCE ACS
005066       9320            RESOURCE CNT
005072       9330            RESOURCE SCO
005076 777777 9340           -1                          END OF STANDARD RESOURCE NAMES
005077       9350            RESOURCE DT
005103       9360            RESOURCE TP
005107 777777 9370           -1                          END OF NUMBERED RESOURCES
             9380    *
             9390    *       MONITOR SOFTWARE REGISTERS TABLE
             9400    *
             9410            ,HEAD   D
005110       9420    REGLIS  REGISTER AC
005112       9430            REGISTER MQ
005114       9440            REGISTER PC
005116       9450            REGISTER LK
005120       9460            REGISTER STS
005122       9470            REGISTER ACS
005124       9480            REGISTER SC
005126       9490            REGISTER ALL
005130       9500            REGISTER VAL
             9510    *
             9520    *       DDT COMMAND TABLE
             9530    *
005132       9540    DDTCOM  ..,
005132       9550            COMMAND PAT,P
005136       9560            COMMAND DUM
005140       9570            COMMAND REG
005142       9580            COMMAND ALT
005144       9590            COMMAND TRA,<T,JMP,J>
005154       9600            COMMAND NSU,<N,NON>
005162       9610            COMMAND EXI,X
005166       9620            COMMAND CON,C
005172       9630            COMMAND CLO
005174       9640            COMMAND PRE
005176       9650            COMMAND LIM
005200       9660            COMMAND MAS
005202       9670            COMMAND SEA
005204       9680            COMMAND BAS
005206       9690            COMMAND VAL,V
             9700    *
             9710    *       DDT DATA MODES TABLE
             9720    *
005212       9730    DDTMOD  ..,
005212       9740            DMODE   0
005220       9750            DMODE   A
005226       9760            DMODE   6
005234       9770            DMODE   H
005242       9780            DMODE   7
005250       9790            DMODE   8
005256       9800            DMODE   D
005264       9810            DMODE   S
```

D TABLE OF COMMANDS, RESOURCES, AND REGISTERS

```
005272  777777  9820            -1                      END OF DDT TABLES
                9830    *
                9840    *       FILES THAT CAN BE "OPEN"ED
                9850    *
    005273      9860    FILES   COMMAND COR,C
    005277      9870            COMMAND DIS,D
    005303      9880            COMMAND SYS,S
    005307      9890            COMMAND V5A,V
    005313      9900            COMMAND XCO,X
    005317      9910            COMMAND BLO,B
005323  777777  9920            -1
                9930            .PMC    RESTORE
                9940            .HEAD   M
                9950            .INSRT  DEBUG
```

```
M                       MTSS MONITOR DEBUGGING PACKAGE

                ,STITL  MTSS MONITOR DEBUGGING PACKAGE
                ,HEAD   D
        *           9120-9130
        *
        *       THIS PACKAGE SHOULD EVENTUALLY PROBABLY BE EXPANDED ENOUGH TO
        *       ELIMINATE THE REQUIREMENT FOR A SEPARATE DDT.
        *
        *       ADVANTAGES OF HAVING THE DEBUGGING DONE FROM THE MONITOR:
        *            1) NO USER CORE IS OVERLAID (AS IT MUST BE WITH AN S-USER DDT)
        *            2) BECAUSE IT IS A PHANTOM PROGRAM THE DEBUGGER CAN HAVE MORE PRIVILEGES
        *            3) CERTAIN THINGS, SUCH AS LOW-CORE MAPPING AND SAVING OF REGISTERS
        *               ARE AUTOMATICALLY DONE FOR THE DEBUGGER BY THE SYSTEM
        *
        *       DISADVANTAGES OF HAVING A PHANTOM DEBUGGER
        *            1) SOME REDUCTION IN SPEED -- NOT VERY NOTICEABLE TO THE USER, HOWEVER
        *            2) ADDED SYSTEM OVERHEAD IN THE FORM OF MEMORY PROTECTION
        *               RELEASES AND ALL PATCHES, DUMPS, ETC REQUIRE DISK OPERATIONS.
        *
        *
        *       DEFINITIONS:
        *            1) A LETTER IS ANY UPPER-CASE OR LOWER-CASE LETTER
        *            2) OCTAL DIGITS ARE THE DIGITS 0-7
        *            3) DECIMAL DIGITS ARE THE DIGITS 0-9
        *            4) ANY OTHER CHARACTER IS A DELIMITER
        *            5) A WORD IS ANY SEQUENCE OF LETTERS AND/OR DIGITS AND IS
        *               TERMINATED BY A DELIMITER
        *            6) AN OCTAL NUMBER IS ANY SEQUENCE OF OCTAL DIGITS AND IS
        *               TERMINATED BY A DELIMITER OTHER THAN A PERIOD (.)
        *            7) A DECIMAL NUMBER IS ANY SEQUENCE OF DIGITS WHICH IS TERMINATED
        *               BY A PERIOD FOLLOWED BY ANY DELIMITER
        *            8) LEGAL PHYSICAL DEVICE NAMES AND THEIR DEVICES:
        *               PTR -- PAPER TAPE READER
        *               PTP -- PAPER TAPE PUNCH
        *               PPT -- PAPER TAPE (EITHER READER OR PUNCH, ACCORDING TO CONTEXT)
        *               TPN -- DECTAPE HANDLER #N
        *               DTN -- DECTAPE HANDLER #N (IDENTICAL TO TPN)
        *               DKN -- PHYSICAL DISK #N
        *            9) LEGAL LOGICAL DEVICE NAMES AND THEIR MEANINGS:
        *               COR -- THE USER'S CORE FILE ON THE DISK
        *               DIS -- THE USER'S "PHYSICAL DISK" FILE ON THE DISK
        *               SYS -- THE SYSTEM LOGICAL DISK
        *               V5A 0 -- DEC V5A LOGICAL DISK #0
        *               V5A 1 -- DEC LOGICAL DISK #1
        *            9) A LEGAL FILENAME IS OF THE FORM <DEVICE NAME>:<WORD>
        *               WHERE <WORD> WILL BE TRUNCATED TO THREE CHARACTERS.
        *               <FILENAME> REFERS TO A SYSTEM FORMAT FILE ON EITHER THE SYSTEM
        *               DISK OR ON DECTAPE.
        *           10) RANGE IS A SPECIFICATION OF THE ADDRESSES TO BE AFFECTED
        *               BY A COMMAND. RANGE IS OF THE FORM:
        *               <NUMBER> -- THE SINGLE LOCATION GIVEN BY THE NUMBER
        *               <NUM1>,<NUM2> -- THE LOCATIONS FROM NUM1 TO NUM2, INCLUSIVE
        *               <NUM1> <NUM2> -- STARTING FROM NUM1 FOR NUM2 LOCATIONS
```

D MTSS MONITOR DEBUGGING PACKAGE

```
620     *
630     *          11) FIELD IS <RANGE> OR <FIELD>;RANGE>
640     *
650     *
660     *
670     *       DEBUGGER COMMANDS ARE OF THE FORMAT
680     *            <COMMAND><ARG1><ARG2>...<ARGN><FIELD>
690     *       WHERE EACH ITEM IS SEPARATED FROM THE NEXT BY ANY DELIMITER
700     *
710     *       IN THE FOLLOWING COMMAND DESCRIPTIONS, FIRST THE COMMAND
720     *       IS NAMED, THEN ANY LEGAL ABBREVIATIONS ARE GIVEN, THEN A DESCRIPTION
730     *       OF THE COMMAND IS GIVEN. LEGAL DEBUGGER COMMANDS AS OF THIS
740     *       ASSEMBLY ARE:
750     *
760     *            VALIDATE (VAL,V) PROVIDES AN UNDERPRINTED AREA ON WHICH THE
770     *       USER CAN TYPE HIS PASSWORD. IF THE PASSWORD IS CORRECT, THE NEXT
780     *       COMMAND IS REQUESTED. OTHERWISE A "VALIDATION ERROR" MESSAGE IS
790     *       PRINTED AND THEN THE NEXT COMMAND IS REQUESTED.
800     *
810     *            OPEN (OPE,O) <DEVICE NAME> OR OPEN <FILENAME> OPENS THE
820     *       REQUESTED DEVICE OR FILE. ALL SUBSEQUENT SENSE AND ALTER MEMORY
830     *       COMMANDS REFER TO THIS FILE. AN ATTEMPT TO REFERENCE BELOW THE
840     *       MINIMUM ADDRESS IN THE FILE OR TO REFERENCE ABOVE THE MAXIMUM
850     *       ADDRESS IN THE FILE GENERATES AN "OUT OF BOUNDS" ERROR MESSAGE.
860     *       OPEN DOES CLOSE WHATEVER FILE WAS OPEN BEFORE IT OPENS THE REQUESTED
870     *       FILE, OTHERWISE THE USER HAS NO WAY OF KNOWING WHAT CHANGES
880     *       HAVE BEEN SAVED AND WHICH ONES HAVE NOT BEEN.
890     *
900     *            CLOSE (CLO,C) CLOSES WHICHEVER FILE IS CURRENTLY OPEN,
900     *       WRITING OUT THE LATEST CHANGES IN IT. THEN AN "OPEN CORE" IS DONE.
910     *
920     *
930     *            READ (REA,R) <NUMBER> CLOSES THE CURRENT FILE AND OPENS
940     *       ON THE CURRENT DEVICE A FILE WHICH IS THE BLOCK WHOSE <NUMBER>
950     *       WAS GIVEN.
960     *
```

```
D                                   MTSS DEBUGGER -- NEXT SYNTACTIC UNIT

                    970             ,STITL  MTSS DEBUGGER -- NEXT SYNTACTIC UNIT
     005324         980             ,USE    PURE
                    990     *
                    1000    *
                    1010    *       ENTER THE DEBUGGER HERE, ABANDONING ANY UNPROCESSED MONITOR INPUT
                    1020    *
     005324         1030    DDT     ...,
     005324         1040            MESS    <XDDT HERE>,9,
                    1050    *
                    1060    *       INITIALIZE THE FILE TO BE USER CORE
                    1070    *
005333  103324      1080            JMS     BINIT
005334  103306      1090            JMS     UCORE           SET MFDA,FDA,FMIN,MFMIN,FMAX,FTYPE,PCMSK FOR THE USER CORE FILE
005335  103304      1100            JMS     SNCOP           SET BMAX,BMIN,MBMIN,BDA FOR THE USER CORE FILE
                    1110    *
                    1120    *       INITIALIZE MODE SWITCHES
                    1130    *
005336  763221      1140            LAW     OMOD            INITIALIZE TO OCTAL MODE
005337  042036      1150            DAC     ADRSW           SET THE ADDRESS MODE
005340  042035      1160            DAC     REGSW           SET THE REGISTERS MODE
005341  763241      1170            LAW     SMOD            INITIALIZE DUMP TO BE SYMBOLIC
005342  042037      1180            DAC     DUMSW           SET THE DUMP MODE
005343  213326      1190            LAC     (20,)
005344  042041      1200            DAC     LIMIT           INITIALIZE THE LIMITS FOR SYMBOLIC DUMP
                    1210    *
                    1220    *
                    1230    *       INITIALIZATION NOW COMPLETE -- GET THE COMMAND
                    1240    *
                    1250            ,HEAD   D,M
     005345         1260    NEXTL   ...,
                    1270            ,HEAD   D
     005345         1280    NXLIN   ...,
     005345         1290            MESS    <?>,1
     005351         1300    NXLN1   LINE                    GET THE NEXT LINE OF INPUT
005352  605357      1310            JMP     NSU2            WE ARE GUARANTEED TO BE STARTING A SYNTACTIC UNIT
                    1320    *
                    1330    *       START TO PARSE THE NEXT SYNTACTIC UNIT, AFTER GETTING A NEW LINE IF NECESSARY
                    1340    *
     005353         1350    NSU     ...,                    PARSE THE NEXT SYNTACTIC UNIT
     005353         1360            DELIM                   FUDGE FOR THE ASSEMBLER -- GET THE DELIMITER WHICH SENT US HERE
005354  650324      1370            SAD     ENDSN           CHECK FOR A LEGAL ARRIVAL
005355  741000      1380            SKP                     OK
005356  605671      1390            JMP     DONE            DIDN'T BELONG HERE IN THE FIRST PLACE
     005357         1400    NSU2    ...,
005357  142053      1410            DZM     COMFLG          SET NO COMMAND YET COMPLETED
005360  213327      1420            LAC     (SKP)
005361  042052      1430            DAC     REGBR           SET UP THE REGISTERS/ALTERS BRANCH
005362  142007      1440            DZM     TEMP7           SET THAT OP CODES ARE NOT ALLOWED
005363  103251      1450            JMS     INVAL           GET THE FIRST WORD
005364  741000      1460            SKP                     NULL WORD IS AMBIGUOUS -- DO FURTHER CHECKS
005365  605506      1470            JMP     FLD08           NON-NULL WORD HAS TO BE A FIELD SPECIFIER
     005366         1480            DELIM                   GET THE NULL WORD DELIMITER
```

D MISS DEBUGGER -- NEXT SYNTACTIC UNIT

```
005367  553311  1490        SAD     ($CR)
005370  606227  1500        JMP     DNXT            CARRIAGE RETURN -- DUMP THE NEXT LOCATION
005371  550324  1510        SAD     ENDSN
005372  606227  1520        JMP     DNXT            END SIGN -- DUMP THE NEXT LOCATION
005373  550317  1530        SAD     FDEL            CHECK IT FOR A FILE NAME DELIMITER
005374  741000  1540        SKP                     YES
005375  605537  1550        JMP     FLD05           ELSE IT MIGHT BE A FIELD AFTER ALL
```

```
D                       MTSS DEBUGGER -- OPEN A FILE/DEVICE

                1560            .STITL  MTSS DEBUGGER -- OPEN A FILE/DEVICE
    005376      1570            .USE    PURE
                1580    *
                1590    *
                1600    *       OPEN SETS UP THE FILE AND BUFFER PARAMETERS FOR THE DEVICE OR
                1610    *       FILE SPECIFIED.
                1620    *
    005376      1630    OPEN    ...
005376  103314  1640            JMS     FORCE           FORCE OUT ANY CHANGES THAT ARE IN THE BUFFER CURRENTLY
    005377      1650            WORD                    GET THE NAME OF THE FILE TO OPEN
005400  613063  1660            JMP     MSG81           NO NAME -- DON'T BELIEVE IT
005401  765272  1670            LAW     DSFILES-1       LOAD A POINTER TO THE DSFILES LIST
005402  103174  1680            JMS     SEARCH          LOCATE THE CURRENT REQUEST
005403  745000  1690            SKPICLL                 NOT FOUND
005404  620010  1700            JMP     10,X            FOUND -- GO OPEN IT
                1710    *
                1720    *       THE REQUEST WAS NOT FOR A LOGICAL DEVICE -- PERHAPS IT WAS FOR A SYSTEM FILE?
                1730    *
    005405      1740            DELIM                   GET THE DELIMITER
005406  553304  1750            SAD     ($COLON)        IT MUST BE A COLON (:) TO BE A SYSTEM FILE
005407  741000  1760            SKP                     YES -- TRY TO OPEN A SYSTEM FILE
005410  605451  1770            JMP     OPE2            NOT A SYSTEM FILE -- MAYBE A PHYSICAL DEVICE
                1780    *
                1790    *       NOW GET THE CATALOG OF THE REQUESTED DEVICE AND FIND THE FILENAME
                1800    *       IN IT.
                1810    *
005411  765414  1820            LAW     OPE1
005412  043527  1830            DAC     CSGNAME         SET THE RESTART ADDRESS
005413  611061  1840            JMP     CSGNAM2         READ IN THE CATALOG IF NOT ALREADY IN CORE AND RETURN FILENAME IN THE AC
005414  613063  1850    OPE1    JMP     MSG81           ATTEMPT TO OPEN A PAPER TAPE
005415  103515  1860            JMS     CSCATL          NOW LOOK UP THE FILENAME IN THE CATALOG
005416  613077  1870            JMP     MSG82           FILE NOT FOUND
                1880    *
                1890    *       THE FILE HAS BEEN LOCATED -- NOW SET ITS PARAMETERS
005417  220011  1900            LAC     SCATX,X
005420  103322  1910            JMS     TDVAL           CHECK ON THE USER'S VALIDATION IF APPROPRIATE
005421  042163  1920            DAC     FDA             SAVE THE DEVICE ADDRESS
005422  513307  1930            AND     (040000)        SAVE THE DISK/DECTAPE BIT
005423  740200  1940            SZA                     SKIP FOR DECTAPE, WHICH BEGINS AT BLOCK ZERO
005424  213330  1950            LAC     ($SYSBAS)       ELSE LOAD THE SYSTEM DISK BASE ADDRESS
005425  342163  1960            TAD     FDA             ADD THE DEVICE ADDRESS YIELDS THE CORRECT DEVICE ADDRESS OF THE FILE
005426  042163  1970            DAC     FDA             SET ITS DEVICE ADDRESS
005427  740001  1980            CMA
005430  042164  1990            DAC     MFDA
005431  442164  2000            INX     MFDA            SET MINUS THE FILE DEVICE ADDRESS
005432  220011  2010            LAC     SCATX,X
005433  042165  2020            DAC     FMIN            SET THE FILES MINIMUM CORE ADDRESS
005434  740001  2030            CMA
005435  042166  2040            DAC     MFMIN
005436  442166  2050            INX     MFMIN           SET MINUM THE FILE MINIMUM CORE ADDRESS
005437  777777  2060            LAW     -1
005440  342165  2070            TAD     FMIN            RELOAD THE MINIMUM CORE ADDRESS
```

D MTSS DEBUGGER -- OPEN A FILE/DEVICE

```
005441  360011  2080            TAD     $CATX,X         ADD THE FILE LENGTH
005442  740001  2090            CMA
005443  042167  2100            DAC     FMAX            SET MINUS THE FILE'S MAXIMUM CORE ADDRESS
005444  210334  2110            LAC     LOGF            LOAD THE LOGICAL FILE FLAG
005445  042151  2120            DAC     FTYPE           AND SET IT
005446  213276  2130            LAC     (17777)         LOAD THE COREMAX MASK
005447  042051  2140            DAC     PCMSK           AND SET IT
005450  605475  2150            JMP     NCOP            SET UP THE CORE BUFFER PARAMETERS
                2160    *
                2170    *       THE REQUEST IS FOR NO SORT OF FILE -- COULD IT BE FOR A PHYSICAL
                2180    *       DISK OR DECTAPE? IF NOT, PRINT A FORMAT ERROR MESSAGE.
                2190    *
   005451       2200    OPE2    ...
005451  765456  2210            LAW     OPE4
005452  043532  2220            DAC     C$DEVCV         SET THE RETURN ADDRESS
   005453       2230            WORD1                   RELOAD THE DEVICE MNEMONIC
005454  611121  2240            JMP     C$DEVC3         EITHER SET UP THE DEVICE ADDRESS OR PRINT A "FORMAT ERROR" MESSAGE
005455  613063  2250            JMP     MSG81           IT IS A PAPER TAPE
   005456       2260    OPE4    ...
005456  103322  2270            JMS     TDVAL           CHECK ON THE USER'S VALIDATION IF APPROPRIATE
005457  042163  2280            DAC     FDA             SET THE FILE DEVICE ADDRESS
005460  740001  2290            CMA
005461  042164  2300            DAC     MFDA
005462  442164  2310            INX     MFDA            SET MINUS THE FILE DEVICE ADDRESS
005463  142165  2320            DZM     FMIN            ALL PHYSICAL DEVICES BEGIN WITH WORD 0
005464  142166  2330            DZM     MFMIN
005465  202163  2340            LAC     FDA             RELOAD THE FILE DEVICE ADDRESS
005466  640703  2350            ALS     3               MOVE THE DEVICE TYPE BIT TO AC(0)
005467  750100  2360            SMAICLA                 SKIP FOR THE DISK
005470  210331  2370            LAC     DTCT            ELSE LOAD THE DECTAPE MAX
005471  042167  2380            DAC     FMAX            SET MINUS THE MAXIMUM ADDRESS ON THE DEVICE
                2390    *
                2400    *
                2410    *
   005472       2420    BOPEN   ...
005472  142151  2430            DZM     FTYPE           CLEAR THE SPECIAL FILE TYPE FLAG
005473  750001  2440            CLC                     SET THE FULL-WORD MASK FOR ALL PHYSICAL DEVICES
005474  042051  2450            DAC     PCMSK
   005475       2460    NCOP    ...                     ENTER HERE TO LEAVE THE FLAG ALONE
005475  103304  2470            JMS     SNCOP           SET UP BMAX, BMIN, MBMIN, BDA FOR THE USER CORE FILE
                2480    *
                2490    *       OPEN FILE FIELD MUST END IN THE DELIMITER FDEL
                2500    *
   005476       2510    OPDON   ...
   005476       2520            DELIM                   GET THE LAST DELIMITER
005477  350317  2530            SAD     FDEL            CHECK FOR THE CORRECT END-DELIMITER
005500  605504  2540            JMP     FIELD           IF SO, EXIT -- NOW SET UP THE FIELD
   005501       2550            WORD                    ELSE SEE IF THERE IS ANOTHER DELIMITER NEXT
005502  605476  2560            JMP     OPDON           YES -- SEE IF IT IS THE CORRECT END OF THE OPEN
005503  613111  2570            JMP     MSG83           ELSE IS AN ERROR
```

D MTSS DEBUGGER -- FIELD SPECIFICATION SETUP

```
                    2580            .STITL  MTSS DEBUGGER -- FIELD SPECIFICATION SETUP
    005504          2590            .USE    PURE
                    2600    *
                    2610    *
                    2620    *       FIELD PICKS UP THE NEXT FIELD TO BE OPERATED ON, STORES THE LOWER BOUNDARY
                    2630    *       IN LOCOR AND STORES THE UPPER BOUNDARY IN HICOR.
                    2640    *
                    2650    *            1) IF THE FIELD CONTAINS A SINGLE VALUE, HICOR := LOCOR := <VALUE>
                    2660    *
                    2670    *            2) IF VAL1 < VAL2 THEN LOCOR := VAL1 AND HICOR := VAL2
                    2680    *
                    2690    *            3) IF VAL1 > VAL2 THEN HICOR := LOCOR := VAL1
                    2700    *
                    2710    *
    005504          2720    FIELD   ...                     FIND OUT WHETHER WE REALLY HAVE A FIELD
005504  103251      2730            JMS     INVAL           GET THE FIRST VALUE
005505  605537      2740    FLD06   JMP     FLD05           EVALUATE A NULL INPUT
    005506          2750    FLD08   ...
005506  202000      2760            LAC     TEMP0           LOAD THE VALUE FOUND
005507  042044      2770            DAC     LOCOR           SET IT AS THE LOW END OF THE FIELD
    005510          2780            DELIM                   GET THE DELIMITER
005511  550322      2790            SAD     BYSGN
005512  605520      2800            JMP     FLD20           COMMA DENOTES SETTING THE FIELD BY BOUNDARIES
005513  550321      2810            SAD     LNSGN
005514  605531      2820            JMP     FLD30           SPACE DENOTES TO SET THE FIELD BY LENGTH
                    2830    *
                    2840    *       SET THE FIELD TO JUST ONE WORD
                    2850    *
005515  202044      2860    FLD10   LAC     LOCOR           LOAD THE LOW END
005516  042045      2870    FLD11   DAC     HICOR           SET IT IN THE HIGH END
005517  605560      2880            JMP     MODE            FIELD SPECIFICATION IS DONE
                    2890    *
                    2900    *       SET THE FIELD BY BOUNDARIES
                    2910    *
    005520          2920    FLD20   ...
005520  103251      2930            JMS     INVAL           GET THE END VALUE
005521  605515      2940            JMP     FLD10           WASN'T ONE -- SET UP JUST ONE LOCATION AND EXIT
005522  042045      2950            DAC     HICOR           SET THE HIGH END BOUNDARY
    005523          2960            NEG                     NEGATE IT
005525  342044      2970            TAD     LOCOR           ADD IN THE LOW END
005526  741300      2980            SPA!SNA                 SKIP IF THE LOW END IS THE GREATER
005527  605560      2990            JMP     MODE            FIELD SPECIFICATION IS DONE
005530  605515      3000            JMP     FLD10           THE LOW END IS GREATER, SO SET UP JUST ONE LOCATION
                    3010    *
                    3020    *       SET THE FIELD BY LENGTH
                    3030    *
    005531          3040    FLD30   ...
005531  103251      3050            JMS     INVAL           GET THE LENGTH
005532  605515      3060            JMP     FLD10           NO LENGTH SPECIFIED -- SET UP JUST ONE LOCATION
005533  513276      3070            AND     (17777)         CAN'T BE LONGER THAN 8K
005534  342044      3080            TAD     LOCOR           ADD THE START ADDRESS
005535  353332      3090            TAD     (-1)
```

D MTSS DEBUGGER -- FIELD SPECIFICATION SETUP

```
005536  605516  3100            JMP     FLD11           SET THE END ADDRESS AND EXIT
                3110    *
                3120    *       EVALUATE AN ORIGINAL NULL INPUT -- INDIRECTION, CURRENT PC, OR NO FIELD
                3130    *
    005537      3140    FLD05   ...
    005537      3150            DELIM                   GET THE NULL FIELD DELIMITER
005540  550316  3160            SAD     INDSN           CHECK FOR THE INDIRECT ADDRESSING SIGN
005541  605547  3170            JMP     FIND            SET THE INDIRECT WORD ON
005542  550325  3180            SAD     PCSGN           CHECK FOR THE CURRENT PROGRAM COUNTER VALUE
005543  605551  3190            JMP     FPC             YES --  INSERT THE CURRENT VALUE
                3200    *
                3210    *       NO NEW FIELD SPECIFICATION, SO LOCOR := HICOR := PC, WHERE PC
                3220    *       IS THE ADDRESS OF THE LAST CONTENTS TYPED IN OR OUT.
                3230    *
    005544      3240    FLD07   ...
005544  202043  3250            LAC     PC              LOAD THE PC
005545  042044  3260            DAC     LOCOR           RESET LOCOR
005546  605515  3270            JMP     FLD10           GO DO THE REST
                3280    *
                3290    *       ALREADY HAVE THE FIRST WORD, SO INITIALIZE INVAL AND TRANSFER INTO IT
                3300    *
                3310    *
                3320    *       AN INDIRECT ADDRESS HAS BEEN REQUESTED. SET THE PROPER FLAG FOR IT.
                3330    *
    005547      3340    FIND    ...
005547  442050  3350            INX     INDIR           SET THE FLAG
005550  605504  3360            JMP     FIELD           GO GET THE NEXT VALUE
                3370    *
                3380    *
                3390    *       USE OF THE CURRENT PROGRAM COUNTER VALUE HAS BEEN REQUESTED
                3400    *
    005551      3410    FPC     ...
005551  765505  3420            LAW     FLD06           LOAD THE DESIRED RESTART ADDRESS
005552  043251  3430            DAC     INVAL           SET IT FOR THE SUBROUTINE
005553  142000  3440            DZM     TEMP0           INITIALIZE THE SUBROUTINE'S ACCUMULATED VALUE
005554  767456  3450            LAW     INPLU
005555  042002  3460            DAC     TEMP2           INITIALIZE THE DEFAULT OPERATOR TO BE PLUS
005556  142006  3470            DZM     TEMP6           INITIALIZE THE VALUE-RECEIVED FLAG
005557  607443  3480            JMP     INPC            ACCUMULATE A VALUE, STARTING WITH THE CURRENT PROGRAM COUNTER
```

```
D                                MTSS DEBUGGER -- MODE SETTING COMMANDS

                    3490                .STITL  MTSS DEBUGGER -- MODE SETTING COMMANDS
                    3500    *
                    3510    *
     005560         3520    MODE        ...                     FIND OUT WHETHER OR NOT WE REALLY HAVE A MODE
     005560         3530                DELIM                   GET THE DELIMITER WHICH BROUGHT US HERE
 005561  550323     3540                SAD     MCSGN
 005562  605575     3550                JMP     MOD10           MODE/COMMAND SIGN -- HANDLE IT
                    3560    *
                    3570    *           CERTAIN DELIMITERS ARE ALSO DDT COMMANDS -- CHECK THEM NOW
                    3580    *
     011531         3590    BRE         .EQU    MSG50
 005563  550326     3600                SAD     PATSN
 005564  606174     3610                JMP     PAT             PATCH COMMAND
 005565  550327     3620                SAD     BKSN
 005566  611531     3630                JMP     BRE             BREAKPOINT
 005567  550330     3640                SAD     JSGN
 005570  606577     3650                JMP     TRA             JUMP/TRANSFER
 005571  550324     3660                SAD     ENDSN
 005572  606234     3670                JMP     DUM             END OF SYNTACTIC UNIT -- DUMP IS DEFAULT
 005573  553311     3680                SAD     (SCR)
 005574  606234     3690                JMP     DUM             END OF SYNTACTIC UNIT -- DUMP IS DEFAULT
                    3700    *
     005575         3710    MOD10       ...
     005575         3720                WORD                    GET THE MODE/COMMAND
 005576  605560     3730                JMP     MODE            IGNORE VACUOUS WORDS HERE
 005577  765107     3740                LAW     REGLIS-1        LOAD A POINTER TO THE DDT TABLE
 005600  103174     3750                JMS     SEARCH          AND TRY FOR A MATCH
 005601  613122     3760                JMP     MSG84           FORMAT ERROR -- COULDN'T FIND IT
 005602  620010     3770                JMP     10,X            ELSE GO DO THE COMMAND
                    3780    *
                    3790    *           ACTUAL MODE-SETTING COMMANDS
                    3800    *
                    3810    *
                    3820    MODSET      .DEFIN
                    3830    M#1         LAW     #1MOD           LOAD A POINTER TO MODE #1
                    3840                JMP     S9
                    3850    AM#1        LAW     #1MOD           LOAD A POINTER TO MODE #1
                    3860                JMP     AS9
                    3870    RM#1        LAW     #1MOD           LOAD A POINTER TO MODE #1
                    3880                JMP     RS9
                    3890                .ENDM
                    3900    *
                    3910    *
                    3920    *
     005603         3930                MODSET  O               OCTAL
     005611         3940                MODSET  A               ACI6 SIXBIT
     005617         3950                MODSET  6               TRIMMED SIXBIT
     005625         3960                MODSET  H               BITS 0-8, 9-17
     005633         3970                MODSET  7               STEXT: BITS 4-10, 11-17
     005641         3980                MODSET  8               SINGLE ASCII CHARACTER IN AC(10-17)
     005647         3990                MODSET  D               DECIMAL
     005655         4000                MODSET  S               SYMBOLIC
```

D MTSS DEBUGGER -- MODE SETTING COMMANDS

```
                    4010
                    4020
005663  042037      4030    S9      DAC     DUMSW           SET THE DUMP FORMAT
005664  605560      4040            JMP     MODE
                    4050
005665  042036      4060    AS9     DAC     ADRSW           SET THE ADDRESS FORMAT
005666  605560      4070            JMP     MODE
                    4080
005667  042035      4090    RS9     DAC     REGSW           SET THE REGISTER FORMAT
005670  605560      4100            JMP     MODE
                    4110    *
                    4120    *       DONE -- NOW WAIT FOR END OF THE SYNTACTIC UNIT, AND THEN PROCESS NEXT ONE
                    4130    *
    005671          4140    DONE    ...
    005671          4150            CRLF
    005672          4160    DON1    DELIM                   GET THE LAST DELIMITER
005673  553311      4170            SAD     (SCR)           CHECK FOR END OF LINE
005674  605345      4180            JMP     NXLIN           IF SO, GET NEXT LINE
005675  550324      4190            SAD     ENDSN           CHECK FOR END OF SYNTACTIC UNIT
005676  605353      4200            JMP     NSU             IF SO, GET THE NEXT ONE
    005677          4210            WORD                    ELSE THROW AWAY ANOTHER WORD
005700  740000      4220            NOP                     NULL INPUT IS IGNORED
005701  605672      4230            JMP     DON1            AND LOOP
```

D OPEN FILE/DEVICE ROUTINES

```
                    4240            .STITL  OPEN FILE/DEVICE ROUTINES
                    4250    *
                    4260    *
                    4270    *       THE USER WANTS TO OPEN HIS USER CORE FILE ON THE DISK
                    4280    *
                    4290    *       FILL IN THE FILE PARAMETERS FOR HIM
                    4300    *
        005702      4310    COR     ...
005702  103306      4320            JMS     UCORE           SET UP THE FILE PARAMTERS FOR THE USER CORE FILE
005703  605475      4330            JMP     NCOP            SET UP THE CORE BUFFER PARRAMETERS AND EXIT
                    4340    *
                    4350    *
                    4360    *       OPEN ACTUAL CORE, BUT DO NOT LOSE THE ORIGINAL
                    4370    *       FILE PARAMETERS -- THUS AFTER THE COMMAND,THE PARAMETERS CAN
                    4380    *       BE SET BACK TO THE FILE WHICH WAS PREVIOUSLY OPEN.
                    4390    *
        005704      4400    XCO     ...
005704  103215      4410            JMS     VALCHK          MUST BE VALIDATED FOR THIS ACTIVITY
005705  202151      4420            LAC     FTYPE           LOAD THE TYPE OF FILE CURRENTLY OPEN
005706  550333      4430            SAD     ACF             SKIP IF NOT ACTUAL CORE
005707  605504      4440            JMP     FIELD           JOB DONE -- NOW SET UP THE FIELD SPECIFICATIONS
005710  042152      4450            DAC     OFTYP           SAVE THE OLD FILE TYPE SO WE DON'T HAVE TO FORCE THE BUFFER
005711  210333      4460            LAC     ACF             LOAD THE ACTUAL CORE FLAG
005712  042151      4470            DAC     FTYPE           AND USE IT TO RESET THE TYPE OF FILE
005713  213276      4480            LAC     (17777)         LOAD A STANDARD ADDRESS MASK
005714  042051      4490            DAC     PCMSK           AND SET IT
005715  605476      4500            JMP     OPDON           NOW SET UP THE OPDON SPECIFICATIONS
                    4510    *
                    4520    *       THE USER WANTS TO OPEN HIS "PHYSICAL DISK" FILE ON THE DISK
                    4530    *
        005716      4540    DIS     ...
005716  103310      4550            JMS     UDISK           SET UP THE USER DISK INITIALIZATION
005717  605472      4560            JMP     BOPEN           SET UP THE CORE BUFFER PARAMETERS AND EXIT
005720  762000      4570    DISCT   -$DKLEN
                    4580    *
                    4590    *       THE USER WANTS TO OPEN THE SYSTEM LOGICAL DISK
                    4600    *
        005721      4610    SYS     ...
005721  103215      4620            JMS     VALCHK          WE MUST BE PROPERLY VALIDATED TO DO SO
005722  213333      4630            LAC     ($SYSDA)
005723  042163      4640            DAC     FDA             SET THE DEVICE ADDRESS OF THE SYSTEM LOGICAL DISK
005724  740001      4650            CMA
005725  042164      4660            DAC     MFDA
005726  442164      4670            INX     MFDA            SET MINUS THE FILE DEVICE ADDRESS
005727  142165      4680            DZM     FMIN            SYSTEM DISK STARTS FROM WORD ZERO
005730  142166      4690            DZM     MFMIN
005731  205734      4700            LAC     SYSCT
005732  042167      4710            DAC     FMAX            SET MINUS THE MAXIMUM ADDRESS ON THE SYSTEM DEVICE
005733  605472      4720            JMP     BOPEN           SET UP THE CORE BUFFER PARAMETERS AND EXIT
005734  000401      4730    SYSCT   -$SYSMAX*400+1
                    4740    *
                    4750    *       THE USER WANTS TO OPEN ONE OF THE V5A LOGICAL DISKS
```

D OPEN FILE/DEVICE ROUTINES

```
                    4760    *
    005735          4770    V5A     ...
005735  103215      4780            JMS     VALCHK          HE MUST BE VALIDATED TO DO THIS
005736  213307      4790            LAC     (040000)
005737  042163      4800            DAC     FDA             SET THE PHYSICAL DISK DEVICE ADDRESS
    005740          4810            NUM                     GET THE NUMBER OF THE LOGICAL DISK
005741  613111      4820            JMP     MSG83           NO NUMBER -- FORMAT ERROR
005742  744020      4830            CLL!RAR                 PUT ANY LEGAL DISK NUMBER IN THE LINK
005743  740200      4840            SZA                     SKIP IF THE DISK NUMBER WAS LEGAL
005744  613137      4850            JMP     MSG85
005745  741400      4860            SZL                     SKIP FOR DISK ZERO
005746  213334      4870            LAC     (1000)          ELSE LOAD THE START OF DISK #1 (BLOCK 1000)
005747  342163      4880            TAD     FDA
005750  042163      4890            DAC     FDA             SET THE LOGICAL DISK'S DEVICE ADDRESS
005751  740001      4900            CMA
005752  042164      4910            DAC     MFDA
005753  442164      4920            INX     MFDA            SET MINUS THE FILE DEVICE ADDRESS
005754  142165      4930            DZM     FMIN            LOGICAL DISK STARTS WITH WORD ZERO
005755  142166      4940            DZM     MFMIN
005756  213335      4950            LAC     (-400000-1)
005757  042167      4960            DAC     FMAX            SET THE FILE MAXIMUM ADDRESS
005760  605472      4970            JMP     BOPEN           SET UP THE CORE BUFFER PARAMETERS AND EXIT
                    4980    *
                    4990    *       THE USER WANTS TO OPEN HIS PREVIOUS FILE -- VALID ONLY WHEN XCORE IS OPEN
                    5000    *
    005761          5010    PRE     ...
005761  202151      5020            LAC     FTYPE           LOAD THE CURRENTLY OPEN FILE TYPE
005762  550333      5030            SAD     ACF             SKIP IF AN ACTUAL CORE FILE WAS NOT PREVIOUSLY OPEN
005763  741000      5040            SKP
005764  605671      5050            JMP     DONE            OTHERWISE IGNORE
005765  202152      5060            LAC     OFTYP           ELSE LOAD THE PREVIOUS FILE TYPE
005766  042151      5070            DAC     FTYPE           RESTORE IT
005767  740200      5080            SZA                     CHECK FOR A CORE LENGTH FILE
005770  605671      5090            JMP     DONE            IN WHICH CASE THE MASK IS STILL OK
005771  750001      5100            CLC                     ELSE GET THE FULL-WORD MASK
005772  042051      5110            DAC     PCMSK           AND SET IT
005773  605671      5120            JMP     DONE            EXIT
                    5130    *
                    5140    *
                    5150    *       BLOCK FORCES THE BUFFER AND OPENS THE SPECIFIED BLOCK ON THE
                    5160    *       CURRENT DEVICE AS A NEW FILE
                    5170    *
    005774          5180    BLO     ...
    005774          5190            NUM                     GET THE BLOCK NUMBER
005775  613111      5200            JMP     MSG83           SOME SORT OF FORMAT ERROR
005776  513336      5210            AND     (1777)          MASK TO JUST THE BLOCK NUMBER
005777  042001      5220            DAC     TEMP1
006000  760000      5230            LAW     0               LOAD A DEVICE NUMBER/TYPE MASK
006001  502163      5240            AND     FDA             RECOVER THE CURRENT DEVICE NUMBER/TYPE
006002  342001      5250            TAD     TEMP1           ADD IN THE REQUESTED BLOCK NUMBER
006003  042163      5260            DAC     FDA             SET THE NEW FILE DEVICE ADDRESS
006004  740001      5270            CMA
```

D OPEN FILE/DEVICE ROUTINES

```
006005  042164  5280          DAC     MFDA
006006  442164  5290          INX     MFDA            SET MINUS THE FILE DEVICE ADDRESS
006007  142165  5300          DZM     FMIN            BLOCK STARTS WITH WORD ZERO
006010  142166  5310          DZM     MFMIN
006011  777401  5320          LAW     -377
006012  042167  5330          DAC     FMAX            SET MINUS THE MAXIMUM BLOCK ADDRESS
006013  605472  5340          JMP     BOPEN           SET THE CORE PARAMETERS AND EXIT
```

D MTSS DEBUGGER -- VALIDATE COMMAND

```
                    5350            ,STITL  MTSS DEBUGGER -- VALIDATE COMMAND
    006014          5360            .USE    PURE
006014  525252      5370    VALWD   525252                  CONSTANT TO VALIDATE A USER
006015  442125      5380    VAL1    ,ACI6   $D15$
006016  262426      5390    VAL2    ,ACI6   $646$
006017  444555      5400    VAL3    ,ACI6   $DEM$
                    5410            ,HEAD   D,M
    006020          5420    VAL     ...
                    5430            ,HEAD   D
    006020          5440            NMESS   <[MMMMMMMMM[WWWWWWWW[BBBBBBBB[>,28.
    006034          5450            LINE                    GET THE VALIDATION LINE ON INPUT
    006035          5460            WORD                    GET THE FIRST THREE CHARACTERS OF THE PASSWORD
    006036          5470            FORMAT                  FORMAT ERROR IF NONE
006037  546017      5480            SAD     VAL3            CHECK THE PUBLIC VALIDATION WORD
006040  606050      5490            JMP     VAL9
006041  546015      5500            SAD     VAL1            CHECK THEM
006042  741000      5510            SKP                     OK -- CHECK NEXT TAREE
006043  611667      5520            JMP     MSG58           ELSE VALIDATION ERROR
    006044          5530            WORD2                   GET THE NEXT THREE CHARACTERS
006045  546016      5540            SAD     VAL2            CHECK THEM
006046  741000      5550            SKP                     VALIDATION SUCCESSFUL
006047  611667      5560            JMP     MSG58           ELSE VALIDATION ERROR
    006050          5570    VAL9    MPOFF
                                    ,PMC    SAVE,ON
006050  705000                      SPECIAL+0               TURN OFF MEMORY PROTECT
006051  206014      5580            LAC     VALWD
006052  041770      5590            DAC     $VALID          VALIDATE THIS USER
006053  701742      5600            MPEU
006054  604023      5610            JMP     MSMONXT         GET THE NEXT COMMAND
                    5620    *
                    5630    *
                    5640    *
                    5650    *       SEE IF THE CURRENT USER HAS BEEN VALIDATED, IF SO, RETURN TO THE
                    5660    *       CALLER. OTHERWISE PRINT A "WHAT" ERROR MESSAGE.
                    5670    *
    006055          5680            ENTER   VALCHK
                                    ,PMC    SAVE,ON
    003215          5690    VALCHK  ...
    006055          5700            MPOFF
                                    ,PMC    SAVE,ON
006055  705000                      SPECIAL+0               TURN OFF MEMORY PROTECT
006056  201770      5710            LAC     $VALID          LOAD THE VALIDATION WORD
006057  701742      5720            MPEU
006060  546014      5730            SAD     VALWD           IS THIS USER VALIDATED
006061  623215      5740            RET     VALCHK,X        YES -- RETURN
    006062          5750            FORMAT                  NO -- PRETEND WE DON'T KNOW FORMAT HE IS TALKING ABOUT
```

```
                D                       MTSS DEBUGGER -- CLOSE -- LIMIT -- MASK -- BASE COMMANDS

                        5750            .STITL  MTSS DEBUGGER -- CLOSE -- LIMIT -- MASK -- BASE COMMANDS
                        5760    *
                        5770    *
                        5780    *       CLOSE THE CURRENTLY OPEN FILE
                        5790    *
     006063             5800    CLO     ...
006063  103314          5810            JMS     FORCE           COPY OUT THE CURRENT BUFFER IF IT HAS BEEN ALTERED
006064  103306          5820            JMS     UCORE           SET UP THE USER CORE FILE PARAMETERS
006065  103304          5830            JMS     SNCOP           SET UP THE BUFFER PARAMETERS
006066  605671          5840            JMP     DONE            EXIT
                        5850    *
                        5860    *
                        5870    *
     006067             5880    LIM     ...
006067  202044          5890            LAC     LOCOR
006070  042041          5900            DAC     LIMIT
006071  605671          5910            JMP     DONE
                        5920    *
                        5930    *
                        5940    *
     006072             5950    MAS     ...
006072  202044          5960            LAC     LOCOR
006073  042046          5970            DAC     MASK
006074  605671          5980            JMP     DONE
                        5990    *
                        6000    *
                        6010    *
     006075             6020    BAS     ...
006075  202044          6030            LAC     LOCOR
006076  042047          6040            DAC     RELOC
006077  605671          6050            JMP     DONE
```

```
D                       MTSS DEBUGGER -- SEARCH COMMAND

                6060            .STITL  MTSS DEBUGGER -- SEARCH COMMAND
                6070    *
                6080    *       SEARCH FOR WORDS IDENTICAL TO THE SEARCH VALUE WHEN BOTH ARE MASKED
                6090    *
    006100      6100    SEA     ...
    006100      6110            CRLF                    PUT THE OUTPUT ON A FRESH LINE
006101  202044  6120            LAC     LOCOR
006102  042042  6130            DAC     LOC             INITIALIZE THE WORKING LOCATION TO THE LOW END OF THE RANGE
006103  750001  6140            CLC
006104  042007  6150            DAC     TEMP7           SET THAT OP CODES ARE LEGAL
                6160
    006105      6170    SEA1    ...
006105  103251  6180            JMS     INVAL           GET THE VALUE TO SEARCH FOR
006106  606136  6190            JMP     SEA5            DONE IF A FORMAT ERROR OR NULL INPUT
006107  502046  6200            AND     MASK            MODIFY BY THE SPECIFIED MASK
006110  042014  6210            DAC     TEMP12          SAVE THE ADJUSTED VALUE FOR THE SEARCH
    006111      6220    SEA2    ...
006111  202042  6230            LAC     LOC             LOAD THE ADDRESS OF THE NEXT LOCATION TO BE EXAMINED
006112  103312  6240            JMS     LOCAT           LOCATE IT
006113  222162  6250            LAC     BPTR,X          LOAD THE VALUE TO BE EXAMINED
006114  502046  6260            AND     MASK            MODIFY BY THE SPECIFIED MASK
006115  542014  6270            SAD     TEMP12          CHECK FOR A GOOD FIND
006116  606125  6280            JMP     SEA4            GOOD FIND -- OUTPUT IT
                6290
    006117      6300    SEA3    ...
006117  202042  6310            LAC     LOC             LOAD THE LOCATION POINTER
006120  542045  6320            SAD     HICOR           SEE IF DONE
006121  606136  6330            JMP     SEA5            YES
006122  442042  6340            INX     LOC             NOT YET DONE -- ADVANCE THE POINTER
006123  740000  6350            NOP                     SKIP INSURANCE
006124  606111  6360            JMP     SEA2            LOOP TO DO THE NEXT VALUE
                6370
    006125      6380    SEA4    ...                     OUTPUT THE GOOD LOCATION AND CONTENTS
006125  202042  6390            LAC     LOC             LOAD THE SUCCESSFUL ADDRESS
006126  502051  6400            AND     PCMSK           MASK TO THE ADDRESS FIELD
006127  042043  6410            DAC     PC              UPDATE THE PC TO THE LATEST LOCATION OUTPUT
006130  122036  6420            JMS     ADRSW,X         PRINT THE ADDRESS IN THE CORRECT FORMAT
006131  103320  6430            JMS     COLSP           FOLLOWED BY A COLON AND SPACE
006132  222162  6440            LAC     BPTR,X          LOAD THE CONTENTS
006133  122037  6450            JMS     DUMSW,X         PRINT THE CONTENTS IN THE CORRECT FORMAT
    006134      6460            CRLF                    END THE LINE
006135  606117  6470            JMP     SEA3            CONTINUE
                6480
    006136      6490    SEA5    ...                     DONE
    006136      6500            DELIM                   GET THE TERMINATING DELIMITER FOR THE LAST VALUE
006137  553311  6510            SAD     ($CR)
006140  605671  6520            JMP     DONE            CARRIAGE RETURN ENDS THE COMMAND
006141  550324  6530            SAD     ENDSN
006142  605671  6540            JMP     DONE            SYNTACTIC UNIT SEPARATOR ENDS THE COMMAND
006143  606100  6550            JMP     SEA             ELSE DO THE NEXT SEARCH
```

```
D                                   MTSS DEBUGGER -- HARDWARE READ-IN PAPER TAPE PUNCH COMMAND

                    6560            .STITL  MTSS DEBUGGER -- HARDWARE READ-IN PAPER TAPE PUNCH COMMAND
                    6570
                    6580
                    6590
    006144          6600    HRI     ...
006144  777777      6610            LAW     -1
006145  342044      6620            TAD     LOCOR
006146  040010      6630            DAC     10                  SET THE AUTO-INDEX REGISTER TO THE FIRST LOCATION TO BE PUNCHED
006147  213337      6640            LAC     (200)               8-HOLE PUNCH FOR BINARY TAPE
006150  042001      6650            DAC     TEMP1               SAVE IT FOR THE SUBROUTINE
    006151          6660    HRI2    ...
006151  220010      6670            LAC     10,X                LOAD THE NEXT WORD TO PUNCH
006152  652000      6680            LMQ                         SAVE IT
006153  103217      6690            JMS     OUTPUT              PUNCH THE FIRST SIX BITS
006154  103217      6700            JMS     OUTPUT              PUNCH THE SECOND SIX BITS
006155  200010      6710            LAC     10                  LOAD THE ADDRESS OF THE WORD
006156  542045      6720            SAD     HICOR               DONE??
006157  006162      6730            JMP     HRI6                YES
006160  103217      6740            JMS     OUTPUT              NO -- OUTPUT THE THIRD SIX BITS
006161  006151      6750            JMP     HRI2                PUNCH THE NEXT WORD
    006162          6760    HRI6    ...
006162  213340      6770            LAC     (300)               LOAD BITS 7&8 FOR THE LAST LINE ON A HARDWARE READ-IN TAPE
006163  042001      6780            DAC     TEMP1               SAVE FOR THE SUBROUTINE
006164  103217      6790            JMS     OUTPUT              PUNCH THE LAST SIX BITS OF THE TAPE
006165  740040      6800            HLT
                    6810
    006166          6820            ENTER   OUTPUT              SHIFT AND PUNCH THE NEXT SIX BITS
                                    .PMC    SAVE,ON
    003217                  OUTPUT  ...
006166  641606      6830            EAE!CLA!LLS 6               GET THE NEXT SIX BITS
006167  342001      6840            TAD     TEMP1               INCLUDE THE HIGH-ORDER BIT(S)
006170  700204      6850            PSA                         PUNCH IT
006171  700201      6860            PSF                         WAIT FOR IT TO SETTLE
006172  006171      6870            JMP     .-1
006173  623217      6880            RET     OUTPUT,X            EXIT
```

```
D                                MTSS DEBUGGER -- PATCH COMMAND

                    6890         .STITL  MTSS DEBUGGER -- PATCH COMMAND
                    6900     *
                    6910     *
    006174          6920     PAT     ...
006174  750001      6930             CLC
006175  103312      6940             JMS     LOCAT           GET A POINTER TO IT
006176  042007      6950             DAC     TEMP7           FLAG THAT OPCODES ARE LEGAL
006177  202044      6960             LAC     LOCOR           LOAD THE ADDRESS AT WHICH TO DO THE PATCH
006200  103251      6970             JMS     INVAL           GET THE VALUE TO PATCH THERE
006201  606216      6980             JMP     PAT2            DONE IF A FORMAT ERROR OR NO MORE INPUT
006202  062162      6990     PAT1    DAC     BPTR,X          ELSE DO THE PATCHING
006203  442156      7000             INX     BALT            SET THE BUFFER ALTERED FLAG
006204  202044      7010             LAC     LOCOR
006205  042043      7020             DAC     PC              UPDATE THE PC TO THIS LAST LOCATION
                    7030     *
                    7040     *       CHECK FOR A BLOCK PATCH
                    7050     *
006206  542045      7060             SAD     HICOR           SEE IF THE BLOCK IS DONE
006207  606216      7070             JMP     PAT2            IF SO, RESUME NORMAL PATCHING
006210  442044      7080             INX     LOCOR           ELSE MOVE THE POINTER TO THE NEXT WORD OF THE BLOCK
006211  740000      7090             NOP                     SKIP INSURANCE
006212  202044      7100             LAC     LOCOR           LOAD THE NEXT LOCATION TO PATCH
006213  103312      7110             JMS     LOCAT           LOCATE IT
006214  202000      7120             LAC     TEMP0           RELOAD THE SAME CONTENTS
006215  606202      7130             JMP     PAT1            AND PATCH IT
    006216          7140     PAT2    ...
    006216          7150             DELIM                   GET THE LAST DELIMITER
006217  553311      7160             SAD     (SCR)
006220  605671      7170             JMP     DONE            CARRIAGE RETURN ENDS THE COMMAND
006221  550324      7180             SAD     ENDSN
006222  605671      7190             JMP     DONE            SYNTACTIC UNIT DELIMITER ENDS THE COMMAND
006223  442044      7200             INX     LOCOR           ADVANCE THE POINTER IN CASE OF ANOTHER PATCH
006224  442045      7210             INX     HICOR           MAKE HICOR KEEP UP WITH LOCOR OR ELSE WE RUN WILD
006225  740000      7220             NOP                     IN CASE OF A SKIP
006226  606174      7230             JMP     PAT             LOOP FOR ANOTHER CONTENTS TO PATCH
```

D MTSS DEBUGGER -- DUMP COMMAND

```
                    7240            .STITL  MTSS DEBUGGER -- DUMP COMMAND
                    7250    *
                    7260    *
                    7270    *       DUMP PRINTS ON THE TELETYPE IN THE CURRENT FORMAT THE FIELD SPECIFIED
                    7280    *       FROM THE CURRENTLY OPEN FILE. DUMPING CONTINUES UNTIL A NEW COMMAND IS GIVEN.
                    7290    *       A NULL FIELD WILL DUMP THE NEXT LOCATION, DUMP RESETS THE PC AS IT DUMPS.
                    7300    *       ON EXIT FROM DUMP, THE PC WILL BE SET TO THE LAST LOCATION DUMPED.
                    7310    *       (E,G. THE COMMAND (DUM .) WILL DUMP THE CURRENT LOCATION AGAIN,
                    7320    *       WHILE THE COMMAND DUM ;) WOULD DUMP THE NEXT LOCATION.
                    7330    *
                    7340    *
                    7350    *       SET TO DUMP THE NEXT LOCATION -- GET HERE ON A VACUOUS SYNTACTIC UNIT
                    7360    *
    006227          7370    DNXT    ...
006227  442043      7380            INX     PC              ADVANCE THE COUNTER FOR THE DUMP
006230  740000      7390            NOP                     PROTECT FROM A POSSIBLE SKIP
006231  202043      7400            LAC     PC
006232  042045      7410            DAC     HICOR           SET TO DUMP ONLY THE ONE LOCATION
006233  606241      7420            JMP     TTD50
                    7430    *
                    7440    *
    006234          7450    DUM     ...                     GENERAL TELETYPE DUMP DRIVER
006234  202053      7460            LAC     COMFLG          LOAD THE COMMAND-ALREADY-PROCESSED FLAG
006235  740200      7470            SZA                     SKIP IF NONE
006236  605671      7480            JMP     DONE            ELSE EXIT
                    7490    *
                    7500    *       SET UP THE PC TO FOLLOW THE DUMP
                    7510    *
006237  202044      7520            LAC     LOCOR
006240  042043      7530            DAC     PC              PC ORIGINATES AT THE LOW END OF THE FIELD
                    7540    *
                    7550    *
                    7560    *       TTDUM PRINTS ON THE TELETYPE THE CONTENTS OF THE OPEN FILE, IN THE
                    7570    *       CURRENT FORMAT, FROM LOCOR THROUGH HICOR. THE LOCOR IS EQUAL TO THE LATEST LOCATION
                    7580    *       PRINTED AT ALL TIMES. A SYMBOLIC DUMP IS PRINTED FOUR LOCATIONS PER LINE;
                    7590    *       ALL OTHER FORMATS ARE PRINTED EIGHT LOCATIONS PER LINE. OUTPUT IS DOUBLE-
                    7600    *       SPACED BEFORE EACH LINE WHOSE STARTING ADDRESS IS AN OCTAL HUNDRED.
                    7610    *
006241  202037      7620    TTD50   LAC     DUMSW           LOAD THE DUMP FORMAT SWITCH
006242  553341      7630            SAD     (LAW SMOD)      SKIP UNLESS A SYMBOLIC DUMP IS REQUESTED
006243  606277      7640            JMP     TTD10           SPECIAL SET-UP FOR A SYMBOLIC DUMP
006244  142001      7650            DZM     TEMP1           NO OTHER FORMAT HAS A HALF LINE MASK
                    7660
    006245          7670    TTD20   ...                     START OF A NEW LINE
    006245          7680            CRLF                    GET A FRESH LINE
006246  202043      7690            LAC     PC              LOAD THE NEX T ADDRESS TO PRINT
006247  502051      7700            AND     PCMSK           MASK TO THE ADDRESS FIELD IF NEEDED
006250  122036      7710            JMS     ADRSW,X         PRINT THE ADDRESS IN THE PROPER FORMAT
006251  103320      7720            JMS     COLSP           PRINT A COLON AND SPACE AFTER THE ADDRESS
                    7730
    006252          7740    TTD30   ...                     PRINT THE NEXT CONTENTS ON THE SAME LINE
006252  202043      7750            LAC     PC              LOAD THE NEXT ADDRESS
```

```
              D                              MTSS DEBUGGER -- DUMP COMMAND

006253  103312  7760            JMS     LOCAT           GET A POINTER TO THIS CONTENTS
006254  222162  7770            LAC     BPTR,X          LOAD THE CONTENTS TO PRINT
006255  122037  7780            JMS     DUMSW,X         PRINT THE CONTENTS IN THE REQUESTED FORMAT
                7790
    006256      7800    TTD40   ...                     DECIDE WHICH THING TO DO NEXT
006256  202043  7810            LAC     PC              LOAD THE LOCATION JUST OUTPUT
006257  542045  7820            SAD     HICOR           CHECK FOR DONE
006260  605671  7830            JMP     DONE            DONE -- EXIT
006261  442043  7840            INX     PC              NOT DONE YET -- MOVE THE POINTER TO THE NEXT WORD
006262  740000  7850            NOP                     SKIP INSURANCE
006263  202043  7860            LAC     PC              NOW LOAD THE UPDATED PC
006264  513317  7870            AND     (77)            AC = 0 IF A BLOCK OF OUTPUT HAS JUST ENDED
006265  741200  7880            SNA                     HAS IT??
    006266      7890            CRLF                    YES -- DOUBLE SPACE
006267  202043  7900            LAC     PC              RELOAD THE LOCATION
006270  513342  7910            AND     (7)             AC = 0 IF A LINE OF OUTPUT HAS JUST ENDED
006271  542001  7920            SAD     TEMP1           CHECK FOR THE END OF A HALF-LINE
006272  606245  7930            JMP     TTD20           YES
006273  741200  7940            SNA                     CHECK FOR END OF A FULL LINE
006274  606245  7950            JMP     TTD20           YES
006275  103316  7960            JMS     SPACE           PRINT A SPACE AFTER THE WORD
006276  606252  7970            JMP     TTD30           NO -- CONTINUE ON THE SAME LINE
                7980
    006277      7990    TTD10   ...                     SET 4 LOCATION PER LINE FOR A SYMBOLIC DUMP
006277  213343  8000            LAC     (4)
006300  042001  8010            DAC     TEMP1           SET THE HALF-LINE FLAG
006301  606245  8020            JMP     TTD20           RESUME THE DUMP
```

D SENSE USER REGISTERS

```
                    8030            ,STITL  SENSE USER REGISTERS
     006302         8040            ,USE    PURE
                    8050    *
                    8060    *
     006302         8070    REG     ...
006302  213327      8080            LAC     (SKP)           LOAD THE REGISTERS BRANCH SWITCH
006303  042052      8090            DAC     REGBR           AND SET IT
006304  606575      8100            JMP     ADONE           NOW FIND OUT WHICH REGISTERS TO DUMP
                    8110    *
                    8120    *
                    8130    *       GET, AND PRINT, THE REQUESTED REGISTER
                    8140    *
     006305         8150    RSTS    ...
006305  402052      8160            XCT     REGBR
006306  606553      8170            JMP     ASTS            BRANCH TO ALTER THE REGISTER
     006307         8180            MESS    <STS: >,5       PRINT THE REGISTER NAME
006315  202020      8190            LAC     STSAVE          LOAD THE USER'S PROGRAM INTERRUPT STATUS REGISTER
006316  606515      8200            JMP     FREG            PRINT IT IN OCTAL
                    8210
     006317         8220    RAC     ...
006317  402052      8230            XCT     REGBR
006320  606522      8240            JMP     AAC             BRANCH TO ALTER THE ACCUMULATOR
     006321         8250            MESS    <AC: >,4        PRINT THE REGISTER NAME
006326  202015      8260            LAC     ACSAVE          LOAD THE USER'S ACCUMULATOR REGISTER
006327  606515      8270            JMP     FREG            PRINT IT IN OCTAL
                    8280
     006330         8290    RMQ     ...
006330  402052      8300            XCT     REGBR
006331  606526      8310            JMP     AMQ             BRANCH TO ALTER THE MQ REGISTER
     006332         8320            MESS    <MQ: >,4
006337  202016      8330            LAC     MQSAVE
006340  606515      8340            JMP     FREG
     006341         8350    RACS    ...
006341  402052      8360            XCT     REGBR
006342  606570      8370            JMP     AACS            BRANCH TO ALTER THE ACCUMULATOR SWITCHES REGISTER
     006343         8380            MESS    <ACS: >,5
006351  202022      8390            LAC     ACSW            LOAD THE USER'S ACCUMULATOR SWITCHES SOFTWARE REGISTER
006352  606515      8400            JMP     FREG            PRINT IT IN OCTAL
                    8410
     006353         8420    RSC     ...
006353  402052      8430            XCT     REGBR
006354  606532      8440            JMP     ASC             BRANCH TO ALTER THE STEP COUNTER REGISTER
     006355         8450    RSC2    MESS    <SC: >,4
006362  202021      8460            LAC     SCSAVE
006363  606512      8470            JMP     SREG
                    8480
     006364         8490    RVAL    ...
006364  402052      8500            XCT     REGBR
     006365         8510            FORMAT                  CAN'T ALTER THE VALIDATION REGISTER
     006366         8520            MESS    <VALIDATION: >,12.
     006376         8530            MPOFF
                                    ,PMC    SAVE,ON
```

D SENSE USER REGISTERS

```
006376  705000          8530            SPECIAL+0               TURN OFF MEMORY PROTECT
006377  201770  8540            LAC     SVALID
006400  701742  8550            MPEU
006401  006512  8560            JMP     SREG
                8570
    006402      8580    RPC     ...
006402  402052  8590            XCT     REGBR
006403  006557  8600            JMP     APC             BRANCH TO ALTER THE PROGRAM COUNTER REGISTER
    006404      8610            MESS    <PC: >,4        PRINT THE REGISTER NAME
006411  202017  8620            LAC     PCSAVE          LOAD THE USER'S PROGRAM COUNTER AND MACHINE STATE
006412  513276  8630            AND     (17777)         RECOVER JUST THE PROGRAM COUNTER
006413  006512  8640            JMP     SREG            PRINT IT IN LEADING-ZEROES-SUPPRESSED OCTAL
                8650
    006414      8660    RLK     ...
006414  402052  8670            XCT     REGBR
006415  006537  8680            JMP     ALK             BRANCH TO ALTER THE LINK REGISTER
    006416      8690            MESS    <LK: >,4        PRINT THE REGISTER NAME
006423  202017  8700            LAC     PCSAVE          LOAD THE USER'S PC AND MACHINE STATE
006424  740010  8710            RAL                     MOVE THE LINK BIT (AC (0)) TO THE LINK
006425  750010  8720            GLK                     RECOVER JUST THE LINK
006426  006512  8730            JMP     SREG
                8740
                8750
    006427      8760    RALL    ...                     DO ALL OF THE REGISTERS
006427  402052  8770            XCT     REGBR
    006430      8780            FORMAT                  CANNOT ALTER ALL REGISTERS
    006431      8790            MESS    <AC: >,4
006436  202015  8800            LAC     ACSAVE
006437  122035  8810            JMS     REGSW,X         PRINT THE REGISTER IN THE DESIRED FORMAT
    006440      8820            MESS    <MQ: >,4
006445  202016  8830            LAC     MQSAVE
006446  122035  8840            JMS     REGSW,X         PRINT THE REGISTER IN THE DESIRED FORMAT
    006447      8850            MESS    <PC: >,4
006454  202017  8860            LAC     PCSAVE
006455  513276  8870            AND     (17777)
006456  122035  8880            JMS     REGSW,X         PRINT THE REGISTER IN THE DESIRED FORMAT
    006457      8890            MESS    <LK: >,4
006464  202017  8900            LAC     PCSAVE
006465  740010  8910            RAL
006466  750010  8920            GLK
    006467      8930            OCTZ
    006471      8940            MESS    <STS: >,5
006477  202020  8950            LAC     STSAVE
006500  122035  8960            JMS     REGSW,X         PRINT THE REGISTER IN THE DESIRED FORMAT
    006501      8970            MESS    <ACS: >,5
006507  202022  8980            LAC     ACSW
006510  122035  8990            JMS     REGSW,X         PRINT THE REGISTER IN THE DESIRED FORMAT
006511  006355  9000            JMP     RSC2            PRINT THE SC REGISTER
                9010
    006512      9020    SREG    ...
    006512      9030            OCTZ                    PRINT THE AC IN LEADING-ZEROES-SUPPRESSED OCTAL
006514  006575  9040            JMP     ADONE           IS THERE ANOTHER REGISTER REQUEST?
```

D SENSE USER REGISTERS

```
                    9050
          006515    9060    FREG    ...                 PRINT THE FULL AC IN OCTAL
006515    122035    9070            JMS     REGSW,X     PRINT THE AC IN THE DESIRED FORMAT
006516    606575    9080            JMP     ADONE       IS THERE ANOTHER REGISTER REQUEST?
```

D ALTER THE USER REGISTERS

```
                    9090            .STITL  ALTER THE USER REGISTERS
    006517          9100            .USE    PURE
                    9110    *
                    9120    *
    006517          9130    ALT     ...
006517  213301      9140            LAC     (NOP)           LOAD THE ALTERS BRANCH SWITCH
006520  042052      9150            DAC     REGBR           AND SET IT
006521  606575      9160            JMP     ADONE           NOW FIND OUT WHICH REGISTER TO ALTER
                    9170    *
                    9180    *       NOW ALTER THE REQUESTED REGISTER
                    9190    *
    006522          9200    AAC     ...
    006522          9210            NUM                     GET THE NEW VALUE FOR THE USER'S AC
    006523          9220            WHAT                    FORMAT ERROR
006524  042015      9230            DAC     ACSAVE          SET THE NEW USER'S AC VALUE
006525  606575      9240            JMP     ADONE           ANOTHER REGISTER???
                    9250
    006526          9260    AMQ     ...
    006526          9270            NUM                     GET THE NEW VALUE FOR THE USER'S MQ
    006527          9280            WHAT                    FORMAT ERROR
006530  042016      9290            DAC     MQSAVE          SET THE USER'S NEW MQ VALUE
006531  606575      9300            JMP     ADONE           ANOTHER REGISTER???
                    9310
    006532          9320    ASC     ...
    006532          9330            NUM                     GET THE NEW VALUE FOR THE USER'S STEP COUNTER
    006533          9340            WHAT                    FORMAT ERROR
006534  513317      9350            AND     (77)            SC IS A SIXBIT REGISTER
006535  042021      9360            DAC     SCSAVE          SET THE USER'S NEW STEP COOUNT VALUE
006536  606575      9370            JMP     ADONE           ANOTHER REGISTER???
                    9380
    006537          9390    ALK     ...
    006537          9400            NUM                     GET THE NEW VALUE FOR THE USER'S LINK
    006540          9410            WHAT                    FORMAT ERROR
006541  744020      9420            CLLIRAR                 MOVE THE VALUE INTO THE LINK
006542  740200      9430            SZA                     SKIP IF A VALUE OF ZERO OR ONE WAS TYPED -- GOOD VALUE
    006543          9440            WHAT                    ANY OTHER VALUE IS AN ERROR
006544  740020      9450            RAR                     MOVE THE VALUE INTO AC (0)
006545  042000      9460            DAC     TEMP0           SAVE IT
006546  202017      9470            LAC     PCSAVE          LOAD THE USER'S PC AND MACHINE STATE
006547  513344      9480            AND     (377777)        GET RID OF THE OLD LINK
006550  242000      9490            XOR     TEMP0           INSERT THE NEW LINK
006551  042017      9500            DAC     PCSAVE          SAVE THE UPDATED PC AND MACHINE STATE
006552  606575      9510            JMP     ADONE           ANOTHER REGISTER???
                    9520
    006553          9530    ASTS    ...
    006553          9540            NUM                     GET THE NEW VALUE FOR THE USER'S PROGRAM INTERRUPT STATUS NUM
    006554          9550            WHAT                    FORMAT ERROR
006555  042020      9560            DAC     STSAVE          SET THE UUSER'S NEW PROGRAM INTERRUPT STATUS REGISTER
006556  606575      9570            JMP     ADONE           ANOTHER REGISTER???
                    9580
    006557          9590    APC     ...
    006557          9600            NUM                     GET THE NEW VALUE FOR THE USER'S PC
```

```
              D                         ALTER THE USER REGISTERS

     006560       9610                 WHAT                    FORMAT ERROR
006561  513276    9620                 AND      (17777)        REDUCE IT TO JUST PC
006562  042000    9630                 DAC      TEMP0
006563  202017    9640                 LAC      PCSAVE         LOAD THE USER'S OLD PC AND MACHINE STATE
006564  513324    9650                 AND      (700000)       GET RID OF THE OLD PC
006565  242000    9660                 XOR      TEMP0          PUT IN THE NEW PC
006566  042017    9670                 DAC      PCSAVE         SAVE THE UPDATED PC AND MACHINE STATUS
006567  606575    9680                 JMP      ADONE          ANOTHER REGISTER???
                  9690
     006570       9700     AACS        ...
     006570       9710                 NUM                     GET THE NEW VALUE FOR THE USER'S AC SWITCHES
     006571       9720                 WHAT                    FORMAT ERROR
006572  042022    9730                 DAC      ACSW           SET THE NEW VALUE OF THE USER'S AC SWITCHES
006573  606575    9740                 JMP      ADONE          ANOTHER REGISTER???
                  9750
     006574       9760     AVAL        ...
     006574       9770                 WHAT                    CAN'T ALTER VALIDATION REGISTER
                  9780     *
                  9790     *
                  9800     *           ALMOST DONE -- FLAG THE COMMAND AND LOOK FOR THE NEXT
                  9810     *
     006575       9820     ADONE       ...
006575  442053    9830                 INX      COMFLG         FLAG THE COMMAND
006576  605560    9840                 JMP      MODE           TRY TO PICK UP ANOTHER ONE
```

D USER JUMP/TRANSFER/CONTINUE COMMANDS

```
                    9850            .STITL  USER JUMP/TRANSFER/CONTINUE COMMANDS
                    9860    *
                    9870    *
    006577          9880    TRA     ...
006577  142150      9890            DZM     PHFLAG          SET THE NEXT PROGRAM TO BE A USER PROGRAM
    006600          9900            NUM                     GET THE ADDRESS
006601  613111      9910            JMP     MSG83           DON'T ACCEPT A NULL ADDRESS
006602  513345      9920    J1      AND     (417777)
006603  253346      9930            XOR     (100000)
006604  042002      9940            DAC     TEMP2           SET THE RESTART ADDRESS
006605  213347      9950            LAC     (507000)        LOAD THE SWAPPER CONTROL WORD
006606  042001      9960            DAC     TEMP1
    006607          9970            MPOFF
                                    .PMC    SAVE,ON
006607  705000                      SPECIAL+0               TURN OFF MEMORY PROTECT
006610  201771      9980            LAC     SNUMBR          LOAD THE USER CORE PROGRAM NAME
006611  604431      9990            JMP     MSMX5           SWAP TO IT
                    10000
006612  202017      10010   CON     LAC     PCSAVE          LOAD THE USER'S RESTART ADDRESS
006613  606602      10020           JMP     J1              AND RESTART HIM THERE
                    10030
                    10040
    006614          10050   EXI     ...                     GET OUT OF ACTUAL CORE MODE
006614  202151      10060           LAC     FTYPE           LOAD THE CURRENTLY OPEN FILE TYPE
006615  550333      10070           SAD     ACF             SKIP UNLESS WE REALLY DO HAVE THE ACTUAL CORE FILE OPEN
006616  741000      10080           SKP
006617  606622      10090           JMP     EXI2            ELSE JOB IS DONE -- EXIT
006620  202152      10100           LAC     OFTYP           LOAD THE PREVIOUS FILE TYPE
006621  042151      10110           DAC     FTYPE           AND RESTOR IT TO ITS RIGHTFUL PLACE
006622  103314      10120   EXI2    JMS     FORCE           FORCE THE BUFFER BEFORE QUITTING
006623  604023      10130           JMP     MSMONXT         NEXT COMMAND
```

D MTSS DEBUGGER -- OUTPUT SUBROUTINES

```
                    10140           ,STITL  MTSS DEBUGGER -- OUTPUT SUBROUTINES
                    10150   *
                    10160   *
                    10170   *       SUBROUTINES TO OUTPUT TO THE TELETYPE THE CONTENTS OF THE ACCUMULATOR
                    10180   *       IN ANY OF THE LEGAL FORMATS.
                    10190   *
                    10200   *
    006624          10210           ENTER   OMOD            PRINT THE AC, INCLUDING LEADING ZEROES, IN OCTAL
                                    ,PMC    SAVE,ON
    003221                  OMOD    ...
    006624          10220           OCT
006626  623221      10230           RET     OMOD,X          EXIT
                    10240
                    10250
    006627          10260           ENTER   AMOD            PRINT THE AC IN AC!6 SIXBIT ASCII
                                    ,PMC    SAVE,ON
    003223                  AMOD    ...
006627  652000      10270           LMQ                     SET UP THE WORD TO PRINT
                    10280           ,DUP    3,3             ONCE FOR EACH CHARACTER
006630  641606      10290           EAECLA!LLS 6            RECOVER THE NEXT CHARACTER
006631  353350      10300           TAD     (240)           RECONSTITUTE THE ASCII
006632  103503      10310           JMS     TSTTYOT         PRINT THE CHARACTER
006633  641606                      EAECLA!LLS 6            RECOVER THE NEXT CHARACTER
006634  353350                      TAD     (240)           RECONSTITUTE THE ASCII
006635  103503                      JMS     TSTTYOT         PRINT THE CHARACTER
006636  641606                      EAECLA!LLS 6            RECOVER THE NEXT CHARACTER
006637  353350                      TAD     (240)           RECONSTITUTE THE ASCII
006640  103503                      JMS     TSTTYOT         PRINT THE CHARACTER
006641  623223      10320           RET     AMOD,X          EXIT
                    10330
                    10340
    006642          10350           ENTER   6MOD            PRINT THE AC IN TRIMMED SIXBIT
                                    ,PMC    SAVE,ON
    003225                  6MOD    ...
006642  652000      10360           LMQ                     SAVE THE VALUE
006643  103227      10370           JMS     SIX             PRINT THE FIRST CHARACTER
006644  103227      10380           JMS     SIX             PRINT THE SECOND CHARACTER
006645  103227      10390           JMS     SIX             PRINT THE THIRD CHARACTER
006646  623225      10400           RET     6MOD,X          EXIT
                    10410
                    10420
    006647          10430           ENTER   SIX             PRINT THE NEXT CHARACTER FROM THE MQ AS TRIMMED SIXBIT ASCII
                                    ,PMC    SAVE,ON
    003227                  SIX     ...
006647  641606      10440           EAECLA!LLS 6            RECOVER THE CHARACTER
006650  353351      10450           TAD     (-40)           300'S GO NEGATIVE; 200'S GO POSITIVE
006651  741100      10460           SPA                     CHECK WHICH IT WAS
006652  353352      10470           TAD     (100)           IT WAS A 300
006653  353350      10480           TAD     (240)           IT WAS A 200
006654  103503      10490           JMS     TSTTYOT         PRINT THE CHARACTER
006655  623227      10500           RET     SIX,X           EXIT
                    10510
```

D MTSS DEBUGGER -- OUTPUT SUBROUTINES

```
                    10520
    006656          10530           ENTER   HMOD            PRINT THE AC AS ONE ASCII CHARACTER IN EACH HALF-WORD
                                    ,PMC    SAVE,ON
    003231                  HMOD    ...,
006656  640511      10540           LRS     9,              KEEP THE UPPER CHARACTER
006657  103503      10550           JMS     TSTTYOT         PRINT IT
006660  640611      10560           LLS     9,              RECOVER THE LOWER CHARACTER
006661  103503      10570           JMS     TSTTYOT         PRINT IT
006662  623231      10580           RET     HMOD,X          EXIT
                    10590
                    10600
    006663          10610           ENTER   8MOD            PRINT THE AC AS ONE ASCII CHARACTER IN AC(10-17)
                                    ,PMC    SAVE,ON
    003233                  8MOD    ...,
006663  513353      10620           AND     (377)           MASK TO THE CHARACTER
006664  103503      10630           JMS     TSTTYOT         PRINT THE CHARACTER
006665  623233      10640           RET     8MOD,X
                    10650
                    10660
    006666          10670           ENTER   7MOD            PRINT THE AC AS STEXT FORMAT (CHARS IN BITS 4-10;11-17)
                                    ,PMC    SAVE,ON
    003235                  7MOD    ...,
006666  640507      10680           LRS     7               KEEP THE UPPER CHARACTER
006667  513354      10690           AND     (177)           MASK TO SEVEN BIT CODE
006670  103503      10700           JMS     TSTTYOT         PRINT IT
006671  640607      10710           LLS     7               RECOVER THE LOWER CHARACTER
006672  513354      10720           AND     (177)           MASK TO SEVEN BIT CODE
006673  103503      10730           JMS     TSTTYOT         PRINT IT
006674  623235      10740           RET     7MOD,X          EXIT
                    10750
                    10760
    006675          10770           ENTER   DMOD            PRINT THE AC AS A SIGNED DECIMAL VALUE
                                    ,PMC    SAVE,ON
    003237                  DMOD    ...,
006675  664000      10780           GSM                     GET THE SIGN AND MAGNITUDE OF THE AC
006676  042000      10790           DAC     TEMP0           SAVE THE MAGNITUDE -- IT IS CORRECT FOR POSITIVE QUANTITIES
006677  744400      10800           SNL!CLL                 SKIP IF THE NUMBER WAS NEGATIVE
006700  006704      10810           JMP     DEC2            ELSE PREPARATIONS ARE DONE
006701  442000      10820           INX     TEMP0           ALLOW FOR GSM'S ONE'S COMPLEMENT CONVERSION
006702  760255      10830           LAW     SMINUS          LOAD A MINUS SIGN
006703  103503      10840           JMS     TSTTYOT         AND PRINT IT
                    10850
                    10860
    006704          10870   DEC2    ...,
006704  762001      10880           LAW     TEMP2-1         LOAD A POINTER TO THE OUTPUT BUFFER
006705  040010      10890           DAC     10              AND SET IT
006706  202000      10900           LAC     TEMP0           LOAD THE VALUE TO PRINT
006707  653323      10910   DEC4    IDIV                    DIVIDE BY 10
006710  000012      10920           10,
006711  060010      10930           DAC     10,X            STORE THE NEXT DIGIT -- NOTE THERE CANNOT BE MORE THAN 5
006712  641002      10940           LACQ                    LOAD THE QUOTIENT
006713  744200      10950           SZA!CLL                 SKIP IF DONE
```

D MTSS DEBUGGER -- OUTPUT SUBROUTINES

```
006714  606707  10960           JMP     DEC4            ELSE LOOP
006715  777777  10970           LAW     -1              AMOUNT TO BACK UP THE POINTER FOR OUTPUT
                10980
006716  340010  10990   DEC6    TAD     10              BACK UP THE POINTER
006717  553355  11000           SAD     (LAW TEMP2-2)   SKIP UNLESS DONE
006720  606727  11010           JMP     DEC8            DONE -- EXIT
006721  040010  11020           DAC     10              ELSE RESET THE POINTER
006722  220010  11030           LAC     10,X            LOAD THE NEXT CHARACTER TO PRINT
006723  353305  11040           TAD     (260)           MAKE IT ASCII
006724  103503  11050           JMS     TSTTYOT         PRINT IT
006725  777776  11060           LAW     -2              AMOUNT TO BACK UP THE POINTER
006726  606716  11070           JMP     DEC6            TRY FOR ANOTHER CHARACTER
                11080
    006727      11090   DEC8    ...
006727  760256  11100           LAW     SPERIOD         LOAD A PERIOD (.)
006730  103503  11110           JMS     TSTTYOT         AND PRINT IT TO SIGNIFY DECIMAL OUTPUT
006731  623237  11120           RET     DMOD,X          EXIT
                11130   *
                11140   *
                11150   *       SMOD PRINTS THE AC AS A SYMBOLIC VALUE UNDER THE FOLLOWING RULES:
                11160   *            1) IF THE OP CODE FIELD IS A MEMORY REFERENCE INSTRUCTION OR
                11170   *               A LAW INSTRUCTION, IT IS SO PRINTED. IN THIS CASE THE PRESENCE
                11180   *               OF A 1 IN BIT 4 CAUSES A ",X" TO BE PRINTED AFTER THE ADDRESS
                11190   *               TO SIGNIFY INDIRECTION. THE ADDRESS FIELD WILL BE PRINTED
                11200   *               AS THE USER SYMBOL IT IS CLOSEST TO PLUS OR MINUS THE DIFFERENCE.
                11210   *               IF THE ABSOLUTE VALUE OF THE DIFFERENCE EXCEEDS THE PRE-ASSIGNED
                11220   *               LIMIT, THEN THE ADDRESS IS PRINTED ENTIRE.
                11230   *
                11240   *            2) IF THE OPCODE FIELD IS AN EAE, THE EAE INSTRUCTION WHICH IS
                11250   *               CLOSEST TO THE VALUE WILL BE PRINTED, ALONG WITH PLUS OR MINUS
                11260   *               THE DIFFERENCE.
                11270   *
                11280   *            3) IF THE VALUE IS AN IOT INSTRUCTION OR AN OPERATE INSTRUCTION
                11290   *               IT WILL BE PRINTED AS A SEQUENCE OF MICROCODED INSTRUCTIONS.
                11300   *
                11310   *
                11320   *       TEMPORARY VARIABLE USAGE IS:
                11330   *            TEMP0 -- STORE THE VALUE TO BE OUTPUT
                11340   *            TEMP1 -- STORE THE INDIRECT BIT IF NECESSARY
                11350   *            TEMP2 -- SCRATCH
                11360   *            TEMP3 -- POINTER TO THE CLOSEST SYMBOL VALUE LOCATED
                11370   *            TEMP4 -- ABSOLUTE VALUE OF THE DIFFERENCE BETWEEN VALUE SOUGHT AND SYMBOL VALUE FOUND
                11380   *
    006732      11390           ENTER   SMOD            PRINT THE AC AS A SYMBOLIC VALUE
                                ,PMC    SAVE,ON
    003241              SMOD    ...
006732  042000  11400           DAC     TEMP0           SAVE THE VALUE
006733  744000  11410           CLL                     PROTECT THE SHIFT
006734  640516  11420           LRS     18,-4           RETAIN JUST THE OP CODE
006735  353356  11430           TAD     (OPTAB)         FORM A POINTER TO THE OPCODE TABLE
006736  042001  11440           DAC     TEMP1           SET THE POINTER
006737  353357  11450           TAD     (-OPTAB-15)     EAE, IOT, AND OPR INSTRUCTIONS STAY POSITIVE
```

D MTSS DEBUGGER -- OUTPUT SUBROUTINES

```
006740  740100  11460           SMA                     SKIP IF IT WAS A MEMORY REFERENCE INSTRUCTION
006741  622001  11470           JMP     TEMP1,X         ELSE BRANCH TO A SPECIAL HANDLING ROUTINE
006742  222001  11480           LAC     TEMP1,X         LOAD THE OPCODE
006743  103223  11490           JMS     AMOD            PRINT IT
006744  103316  11500           JMS     SPACE           FOLLOWED BY A SPACE
006745  202000  11510           LAC     TEMP0           RELOAD THE VALUE
006746  513360  11520           AND     (020000)        RECOVER THE INDIRECTION BIT
006747  042001  11530           DAC     TEMP1           SAVE IT
006750  202000  11540   SMOD2   LAC     TEMP0           RELOAD THE FULL VALUE
006751  513276  11550           AND     (17777)         RETAIN JUST THE ADDRESS FIELD
006752  042000  11560           DAC     TEMP0           REPLACE THE VALUE -- THE REST OF IT IS ALREADY PRINTED
006753  103276  11570           JMS     UVSCH           SEARCH THE USER TABLE FOR A SYMBOL FOR THE ADDRESS
006754  202004  11580           LAC     TEMP4           LOAD THE ABSOLUTE VALUE OF THE DIFFERENCE BETWEEN SOUGHT AND FOUND
006755          11590           NEG                     NEGATE IT
006757  342041  11600           TAD     LIMIT           ADD THE ALLOWED MARGIN
006760  740100  11610           SMA                     SKIP IF WE ARE OUTSIDE THE ALLOWED MARGIN
006761  607040  11620           JMP     SYM2            ELSE PRINT THE SYMBOL AND DIFFERENCE
                11630   *
                11640   *       NO SUITABLE SYMBOL, SO PRINT THE ADDRESS IN OCTAL
                11650   *
006762  202000  11660           LAC     TEMP0           RELOAD THE ADDRESS
006763          11670           OCTZ                    PRINT IT IN OCTAL
006765  202001  11680           LAC     TEMP1           LOAD THE INDIRECT BIT FLAG
006766  741200  11690           SNA                     SKIP IF THERE IS ONE
006767  623241  11700           RET     SMOD,X          ELSE DONE
006770  206773  11710           LAC     IND             LOAD THE COMMA X
006771  103245  11720           JMS     SAMOD           PRINT IT
006772  623241  11730           RET     SMOD,X          DONE
                11740
006773  001470  11750   IND     .ACI6   * ,X*
006774  623241  11760           RET     SMOD,X          EXIT
                11770   *
                11780   *       THE VALUE IS A LAW INSTRUCTION
                11790   *
006775  544167  11800   SLAW2   .ACI6   *LAW*
006776          11810   SLAW    ...
006776  206775  11820           LAC     SLAW2           LOAD THE MNEMONIC
006777  103223  11830           JMS     AMOD            PRINT THE LAW SYMBOL
007000  103316  11840           JMS     SPACE
007001  142001  11850           DZM     TEMP1           LAW INSTRUCTION HAS NO INDIRECT BIT
007002  606750  11860           JMP     SMOD2           PRINT THE ADDRESS
                11870
                11880
007003  454145  11890   SEAB    .ACI6   *EAE*
007004          11900   SEAE    ...                     EAE INSTRUCTION -- DO A BIT MATCH FOR BEST AND PRINT NUMERICAL RESIDUE
007004  213361  11910           LAC     (640000)
007005  042007  11920           DAC     TEMP7           SET THE OP CODE
007006  213301  11930           LAC     (740000)
007007  042010  11940           DAC     TEMP8           SET THE OP CODE MASK
007010  202000  11950           LAC     TEMP0
007011  513306  11960           AND     (037777)        GET RID OF THE OP CODE FROM THE OLD VALUE
007012  042000  11970           DAC     TEMP0
```

D MTSS DEBUGGER -- OUTPUT SUBROUTINES

```
007013  103302  11980          JMS     PBSCH           SEARCH FOR THE CODE WITH THE LARGEST NUMBER OF COMMON BITS
007014  202004  11990          LAC     TEMP4           LOAD THE NUMBER OF COMMON BITS
007015  740200  12000          SZA                     SKIP IF THERE WERE NONE
007016  607024  12010          JMP     SEAE2           ELSE CHECK THE FIT
007017  202000  12020  SEAE3   LAC     TEMP0           RELOAD THE VALUE
007020  253361  12030          XOR     (640000)        PUT THE OPCODE BACK
007021          12040          OCT                     PRINT IT
007023  623241  12050          RET     SMOD,X          AND EXIT
                12060
007024          12070  SEAE2   ...
007024  777777  12080  M1      LAW     -1
007025  362003  12090          TAD     TEMP3,X         LOAD THE TABLE VALUE
007026  740001  12100          CMA                     NEGATE IT
007027  342000  12110          TAD     TEMP0           ADD THE ORIGINAL MICROCODE
007030  353361  12120          TAD     (640000)        ADD THE OPCODE
007031  042004  12130          DAC     TEMP4           SAVE THE SIGNED DIFFERENCE BETWEEN THEM
007032  664000  12140          GSM                     GET SIGN AND MAGNITUDE
007033  347024  12150          TAD     M1
007034  740001  12160          CMA                     NEGATE THE MAGNITUBE OF THE DIFFERENCE
007035  342041  12170          TAD     LIMIT           ADD THE ALLOWABLE DIFFERENCE
007036  741100  12180          SPA                     SKIP IF IT IS OK
007037  607017  12190          JMP     SEAE3           ELSE PRINT IT IN OCTAL
                12200
                12210
007040  103243  12220  SYM2    JMS     SYM4            PRINT THE BEST SYMBOL FOUND
007041  202004  12230          LAC     TEMP4           LOAD THE DIFFERENCE IN VALUES
007042  741200  12240          SNA                     SKIP IF THERE IS ANY
007043  623241  12250          RET     SMOD,X          DON'T BOTHER TO PRINT A ZERO
007044  760253  12260          LAW     SPLUS           LOAD A PLUS SIGN
007045  103503  12270          JMS     TSTTYOT         AND PRINT IT
007046  202004  12280          LAC     TEMP4           RELOAD THE VALUES DIFFERENCE
007047  103237  12290          JMS     DMOD            PRINT IT IN DECIMAL
007050  623241  12300          RET     SMOD,X          EXIT
                12310
007051          12320          ENTER   SYM4            PRINT THE BEST SYMBOL FOUND
                               ,PMC    SAVE,ON
003243                 SYM4    ...
007051  142002  12330          DZM     TEMP2           INITIALIZE THE FIRST-CHARACTER-PRINTED FLAG
007052  777775  12340          LAW     -3              AMOUNT TO BACK UP THE POINTER
007053  342003  12350          TAD     TEMP3           MOVE THE POINTER BACK TO THE SYMBOL
007054  040010  12360          DAC     10              SAVE THE POINTER
007055  220010  12370          LAC     10,X            LOAD THE FIRST HALF-SYMBOL
007056  103245  12380          JMS     SAMOD           PRINT IT
007057  220010  12390          LAC     10,X            LOAD THE SECOND HALF-SYMBOL
007060  103245  12400          JMS     SAMOD           PRINT IT
007061  623243  12410          RET     SYM4,X
                12420  *
                12430  *       PRINT THE AC IN SIXBIT (ACI6) ASCII, DELETING LEADING BLANKS
                12440  *       AND CHANGING OTHER BLANKS TO DOLLAR SIGNS ($).
                12450  *
007062          12460          ENTER   SAMOD
                               ,PMC    SAVE,ON
```

D MTSS DEBUGGER -- OUTPUT SUBROUTINES

```
    003245              SAMOD   ...
007062  652000  12470           LMQ             SAVE THE VALUE TO BE PRINTED
007063  777775  12480           LAW     -3      LOAD THE CHARACTER COUNT
007064  042001  12490           DAC     TEMP1   AND SET IT
    007065      12500   SAM2    ...
007065  641606  12510           EAECLA!LLS 6    GET THE NEXT CHARACTER
007066  741200  12520           SNA             SKIP IF NON-BLANK
007067  607076  12530           JMP     SAM4    ELSE TAKE APPROPRIATE ACTION
007070  353350  12540           TAD     (240)   MAKE INTO ASCII
007071  442002  12550           INX     TEMP2   COUNT THE PRINTED CHARACTERS
    007072      12560   SAM6    ...
007072  103503  12570           JMS     TSTTYOT PRINT THE CHARACTER
    007073      12580   SAM8    ...
007073  442001  12590           ISZ     TEMP1   COUNT THE CHARACTERS IN THIS WORD
007074  607065  12600           JMP     SAM2    LOOP TO PRINT THE NEXT CHARACTER
007075  623245  12610           RET     SAMOD,X ELSE DONE -- EXIT
                12620
    007076      12630   SAM4    ...
007076  202002  12640           LAC     TEMP2   LOAD THE FIRST-CHARACTER-PRINTED FLAG
007077  741200  12650           SNA             SKIP IF THERE HAS BEEN A CHARACTER PRINTED
007100  607073  12660           JMP     SAM8    ELSE DON'T PRINT THIS ONE
007101  773362  12670           LAW     ($DOLLAR) YES -- REPLACE THE BLANK WITH A DOLLAR SIGN ($)
007102  442002  12680           DZM     TEMP2   AND SUPPRESS PRINTING FURTHER BLANKS
007103  607072  12690           JMP     SAM6    PRINT THE DOLLAR SIGN
                12700
                12710
007104  565760  12720   OPR8    .ACI6   +NOP+
    007105      12730   SOPR    ...             PRINT A STRING OF OPERATE INSTRUCTION MICROCODES
007105  202000  12740           LAC     TEMP0   LOAD THE VALUE
007106  513363  12750           AND     (LAW)   RETAIN THE OPCODE PLUS THE LAW BIT IF PRESENT
007107  553363  12760           SAD     (LAW)   CHECK THE LAW BIT
007110  606776  12770           JMP     SLAW    YES -- LAW INSTRUCTIONS ARE DIFFERENT FROM THE REST OF THE OPERATE GROUP
                12780
007111  213301  12790           LAC     (740000)
007112  042007  12800           DAC     TEMP7   SET THE OP CODE
007113  042010  12810           DAC     TEMP8   SET THE MASK FOR THE OP CODE TABLE SEARCH
007114  202000  12820           LAC     TEMP0   LOAD THE VALUE TO MATCH
007115  513306  12830           AND     (037777) GET RID OF THE OP CODE
007116  042000  12840           DAC     TEMP0   SET JUST THE MICROCODE
007117  513334  12850           AND     (001000) RECOVER THE INVERTED SKIP BIT
007120  042011  12860           DAC     TEMP9   AND SAVE IT
                12870
    007121      12880   OPR2    ...
007121  103302  12890           JMS     PBSCH   SEARCH THE PERMANENT SYMBOL TABLE FOR THE OPR INST W/LGST # OF COMMON BITS
007122  202004  12900           LAC     TEMP4   LOAD THE NUMBER OF COMMON BITS
007123  741200  12910           SNA             SKIP IF THERE ARE ANY
007124  607146  12920           JMP     OPR6    ELSE PRINT THE NUMERICAL DIFFERENCE
007125  103243  12930           JMS     SYM4    PRINT THE BEST MATCH FOUND
007126  222003  12940           LAC     TEMP3,X LOAD THE TABLE VALUE OF THE "MATCH"
007127  242000  12950           XOR     TEMP0   GET RID OF THE MICROCODE JUST PRINTED
007130  242007  12960           XOR     TEMP7   GET RID OF THE OPCODE
007131  741200  12970           SNA             SKIP UNLESS DONE
```

D MTSS DEBUGGER -- OUTPUT SUBROUTINES

```
007132  623241  12980          RET     SMOD,X          IN WHICH CASE EXIT
007133  042000  12990          DAC     TEMP0           SET THE REDUCED MICROCODE
007134  513364  13000          AND     (000700)        RECOVER ANY REMAINING SKIP BITS
007135  741200  13010          SNA                     SKIP UNLESS THERE ARE NONE
007136  607143  13020          JMP     OPR1            IN WHICH CASE THE MICROCODE IS CORRECT
007137  202000  13030          LAC     TEMP0           STILL HAVE SKIP BITS, SO RELOAD THE MICROCODE
007140  513365  13040          AND     (776777)        GET RID OF ANY INVERTED SKIP BIT
007141  242011  13050          XOR     TEMP9           PUT THE PROPER INVERTED SKIP BIT IN
007142  042000  13060          DAC     TEMP0           RESTORE THE MICROCODE
                13070
   007143       13080  OPR1    ...,
007143  760241  13090          LAW     SEXCLAM         LOAD AN EXCLAMATION MARK FOR THE MICROCODING INDICATION
007144  103503  13100          JMS     TSTTYOT         AND PRINT IT
007145  607121  13110          JMP     OPR2            DO THE REMAINDER OF THE CODE
                13120
                13130
007146  207104  13140  OPR6    LAC     OPR8            LOAD THE MNEMONIC
007147  103223  13150          JMS     AMOD            PRINT IT
007150  623241  13160          RET     SMOD,X          EXIT
                13170  *
                13180  *
                13190  *       DECODE THE IOT INSTRUCTION AND PRINT IT AS A STRING OF MICROCODES
                13200  *       PLUS ANY OCTAL OFFSET. UNRECOGNIZED DEVICES WILL BE PRINTED AS
                13210  *       IOT+<OFFSET>.
                13220  *
007151  515764  13230  SIOT8   .ACI6   +IOT+
   007152       13240  SIOT    ...,
007152  142011  13250          DZM     TEMP9           CLEAR THE CODE PRINTED FLAG
007153  213366  13260          LAC     (777760)
007154  042010  13270          DAC     TEMP8           SET THE OP CODE RETAINING MASK
007155  502000  13280          AND     TEMP0
007156  042007  13290          DAC     TEMP7           SET THE OP CODE FOR THE TABLE SEARCH
007157  202000  13300          LAC     TEMP0           LOAD THE VALUE
007160  513306  13310          AND     (037777)        GET RID OF THE OP AND DEVICE CODES
007161  042000  13320          DAC     TEMP0           SAVE THE MICROCODE
007162  103247  13330          JMS     SIOT2           PRINT THE INSTRUCTION AS A MICRO-CODED STRING
   007163       13340  SIOT4   ...,                    NO MATCH FOUND FOR THE REMAINING CODE
007163  202011  13350          LAC     TEMP9           LOAD THE CODE PRINTED FLAG
007164  740200  13360          SZA                     SKIP IF NO CODE HAS YET BEEN PRINTED
007165  607176  13370          JMP     SIOT6           ELSE PRINT THE OFFSET
007166  202000  13380          LAC     TEMP0           LOAD THE UNACCOUNTED FOR BITS
007167  741200  13390          SNA                     SKIP IF THERE ARE ANY
007170  607173  13400          JMP     SIOT3           ELSE PRINT THE IOT MNEMONIC
007171  253324  13410          XOR     (700000)        RESTORE THE OP CODE
007172  607201  13420          JMP     SIOT7           PRINT IT
   007173       13430  SIOT3   ...,
007173  207151  13440          LAC     SIOT8           LOAD THE IOT MNEMONIC
007174  103223  13450          JMS     AMOD            PRINT THE OP CODE
007175  623241  13460          RET     SMOD,X          EXIT
                13470
   007176       13480  SIOT6   ...,                    PRINT ANY REMAINING OFFSET
007176  760253  13490          LAW     SPLUS           LOAD A PLUS SIGN
```

D MTSS DEBUGGER -- OUTPUT SUBROUTINES

```
007177  103503  13500          JMS     T$TTYOT         PRINT IT
007200  202000  13510          LAC     TEMP0           LOAD THE REMAINING UNMATCHED CODE
007201          13520  SIOT7   OCTZ                    PRINT IT IS ZERO-SUPPRESSED OCTAL
007203  623241  13530          RET     SMOD,X          EXIT
                13540
                13550
007204          13560          ENTER   SIOT2           PRINT A STRING OF MICROCODES AND EXIT WHEN NO MORE MATCH CAN BE FOUND
                               .PMC    SAVE,ON
003247                 SIOT2   ...
007204  103302  13570          JMS     PBSCH           SEARCH THE PERMANENT SYMBOL TABLE FOR THE INST W/LGST # OF COMMON BITS
007205  202004  13580          LAC     TEMP4           LOAD THE NUMBER OF MATCHING BITS
007206  741200  13590          SNA                     SKIP IF THERE ARE SOME
007207  623247  13600          RET     SIOT2,X         ELSE EXIT
007210  202011  13610          LAC     TEMP9           SEE IF THERE HAS ALREADY BEEN A PIECE OF CODE PRINTED
007211  741200  13620          SNA                     SKIP IF SO
007212  607215  13630          JMP     SIOT5           ELSE DON'T PRINT AN EXCLAMATION POINT
007213  760241  13640          LAW     $EXCLAM         LOAD AN EXCLAMATION POINT FOR THE MICROCODING INDICATOR
007214  103503  13650          JMS     T$TTYOT         PRINT IT
007215  442011  13660  SIOT5   INX     TEMP9           COUNT THE PRINTED CODE
007216  103243  13670          JMS     SYM4            PRINT THE BEST MATCH FOUND
007217  202007  13680          LAC     TEMP7           LOAD THE OLD OPCODE & DEVICE NUMBER
007220  513301  13690          AND     (740000)        RECOVER JUST THE OP CODE
007221  242000  13700          XOR     TEMP0           INCLUDE THE LATEST SEARCHED-FOR BITS
007222  262003  13710          XOR     TEMP3,X         GET RID OF THE MICROC CODE JUST PRINTED
007223  741200  13720          SNA                     SKIP UNLESS DONE
007224  623241  13730          RET     SMOD,X          IN WHICH CASE, EXIT
                13740
007225  042000  13750          DAC     TEMP0           SET THE REDUCED MICROCODE
007226  603250  13760          JMP     SIOT2+1         ITERATE
                13770
                13780
007227  434154  13790  OPTAB   ,ACI6   +CAL+           TABLE OF PDP-9 OPCODES
007230  444143  13800          ,ACI6   +DAC+
007231  525563  13810          ,ACI6   +JMS+
007232  447255  13820          ,ACI6   +DZM+
007233  544143  13830          ,ACI6   +LAC+
007234  705762  13840          ,ACI6   +XOR+
007235  414444  13850          ,ACI6   +ADD+
007236  644144  13860          ,ACI6   +TAD+
007237  704364  13870          ,ACI6   +XCT+
007240  516372  13880          ,ACI6   +ISZ+
007241  415644  13890          ,ACI6   +AND+
007242  634144  13900          ,ACI6   +SAD+
007243  525560  13910          ,ACI6   +JMP+
007244  607004  13920          JMP     SEAE
007245  607152  13930          JMP     SIOT
007246  607105  13940          JMP     SOPR
```

```
D                                MTSS DEBUGGER -- INVAL SUBROUTINE

                    13950                ,STITL  MTSS DEBUGGER -- INVAL SUBROUTINE
                    13960    *
                    13970    *
                    13980    *       INVAL EVALUATES THE NEXT EXPRESSION THE USER HAS TYPED,
                    13990    *            RETURN IS +1 IF THERE IS NONE
                    14000    *                      +2 IF THE EXPRESSION IS SUCCESSFULLY EVALUATED
                    14010    *                      FORMAT ERROR MESSAGE OTHERWISE
                    14020    *
                    14030    *       TEMPORARY VARIABLE USAGE IS:
                    14040    *            TEMP0 -- ACCMULATED VALUE
                    14050    *            TEMP1 -- LATEST TERM VALUE OR HEAD SYMBOL
                    14060    *            TEMP2 -- OPERATOR SWITCH
                    14070    *            TEMP3 -- TELETYPE BUFFER POINTER TO START OF CURRENT WORD
                    14080    *            TEMP4 -- WORD COUNT AT THE START OF THE CURRENT WORD
                    14090    *            TEMP5 -- DELIMITER AT THE START OF THE CURRENT WORD
                    14100    *            TEMP6 -- FLAG FOR VALUE HAS BEEN RECEIVED (NEED FOR PC SIGN)
                    14110    *            TEMP7 -- =1 FOR OP CODE INPUT ALLOWED (PERMANENT SYMBOL TABLE WILL BE CHECKED)
                    14120    *                     =0 FOR OP CODE INPUT NOT ALLOWED (PERMANENT SYMBOL TABLE WILL NOT BE CHECKED)
                    14130    *            TEMP8 -- FLAG THAT A PREVIOUS VALUE WAS FROM THE PERMANENT SYMBOL TABLE
                    14140    *            TEMP9 -- USED BY THE SYMBOL PICK-UP ROUTINE
                    14150    *            TEMP10 -- USED BY THE SYMBOL PICK-UP ROUTINE
                    14160    *
                    14170    *       A SPACE IS A DELIMITER FOR THE VALUE UNLESS THE WORD IT IS DELIMITING IS AN
                    14180    *       OP CODE AND THE NEXT WORD IS NOT AN OP CODE, IN THAT CASE IT IS AN
                    14190    *       IMPLIED PLUS.
                    14200    *
                    14210    *       A COMMA IS A DELIMITER FOR THE VALUE UNLESS IT IS FOLLOWED BY EITHER
                    14220    *       X<DELIMITER> OR I<DELIMITER>, IN THAT CASE THE COMMA X OR COMMA I IS
                    14230    *       REPLACED BY  <EXCLAMATION PT>020000.
                    14240    *
    007247          14250                ENTER   INVAL
                                         ,PMC    SAVE,ON
    003251                       INVAL   ...
007247  142000  14260                    DZM     TEMP0           INITIALIZE THE ACCUMULATED VALUE
007250  142006  14270                    DZM     TEMP6           INITIALIZE THE VALUE RECEIVED FLAG
007251  767456  14280                    LAW     INPLU
007252  042002  14290                    DAC     TEMP2           INITIALIZE THE OPERATOR SWITCH TO PLUS
007253  142010  14300                    DZM     TEMP8           INITIALIZE THE PREVIOUS VALUE FLAG
007254  103260  14310    INV2            JMS     SYMBOL          BUILD THE ASSUMED SYMBOLIC INPUT
007255  607432  14320                    JMP     INULL           NO INPUT AVAILABLE -- REQUIRES FURTHER CHECKS
007256  442006  14330                    INX     TEMP6           FLAG A NON-NULL SYMBOL
007257  202007  14340                    LAC     TEMP7           LOAD THE OP CODES FLAG
007260  741200  14350                    SNA                     SKIP IF THEY ARE ALLOWED
007261  607264  14360                    JMP     INV3            ELSE DO NOT CHECK THE PERMANENT SYMBOL TABLE
                    14370    *
                    14380    *       SEARCH ONE OR BOTH SYMBOL TABLES FOR THE SYMBOL WE HAVE BUILT, RETURN
                    14390    *       WITH THE SYMBOL VALUE IN TEMP1.
                    14400    *
007262  103266  14410                    JMS     PSRCH           SEARCH THE PERMANENT SYMBOL TABLE
007263  607302  14420                    JMP     INV6            SUCCESS -- FURTHER CHECKS NEEDED
                    14430    *
                    14440    *       THE SYMBOL IS NOT FROM THE PERMANENT SYMBOL TABLE
```

D MTSS DEBUGGER -- INVAL SUBROUTINE

```
                    14450   *
    007264          14460   INV3    ...
007264  202010      14470           LAC     TEMP8           LOAD THE PREVIOUS SYMBOL TYPE
007265  740200      14480           SZA                     SKIP IF IT WAS NOT FROM THE PERMANENT SYMBOL TABLE
007266  607274      14490           JMP     INV4            ELSE WE ARE O.K.
007267  202002      14500           LAC     TEMP2           IF SO, LOAD THE OLD DELIMITER
007270  553367      14510           SAD     (LAW INSPA)
007271  607313      14520           JMP     INV62           OLD DELIMITER WAS A SPACE -- BACK THE POINTERS AND EXIT
007272  553370      14530           SAD     (LAW INCOM)
007273  607313      14540           JMP     INV62           OLD DELIMITER WAS A COMMA -- BACK THE POINTERS AND EXIT
    007274          14550   INV4    ...
007274  103270      14560           JMS     USRCH           NOW SEARCH THE USER SYMBOL TABLE (IF ANY)
007275  622002      14570           JMP     TEMP2,X         SUCCESS -- DO THE INDICATED OPERATION
007276  202002      14580           LAC     TEMP2           FAILURE -- LOAD THE BRANCH CONTROL
007277  553370      14590           SAD     (LAW INCOM)     CHECK FOR A PREVIOUS COMMA
007300  607531      14600           JMP     INCOM           IF SO, HANDLE IT
007301  607325      14610           JMP     INV30           ELSE SEE IF IT WAS A NUMBER
                    14620
    007302          14630   INV6    ...
007302  142007      14640           DZM     TEMP7           FLAG NO MORE OP CODES ALLOWED
007303  042001      14650           DAC     TEMP1           SAVE THE VALUE
007304  202010      14660           LAC     TEMP8           LOAD THE PREVIOUS VALUE FLAG
007305  751201      14670           SNA!CLA!CMA             SKIP IF THE PREVIOUS VALUE WAS A PERMANENT SYMBOL
007306  607322      14680           JMP     INV61           ELSE WE ARE O.K.
007307  202002      14690           LAC     TEMP2           LOAD THE BRANCH CONTROL
007310  553367      14700           SAD     (LAW INSPA)     SEE IF THE PREVIOUS DELIMITER WAS A SPACE
007311  741000      14710           SKP                     IF SO, FORGET THIS SYMBOL AND EXIT
007312  607322      14720           JMP     INV61           IF NOT! THIS SYMBOL IS O.K.
007313  202003      14730   INV62   LAC     TEMP3
007314  043451      14740           DAC     TSBPTR          RESTORE THE TELETYPE BUFFER POINTER
007315  202004      14750           LAC     TEMP4
007316  043456      14760           DAC     TSCOUNT         RESTORE THE PREVIOUS WORD COUNT
007317  202005      14770           LAC     TEMP5
007320  043455      14780           DAC     TSDLMTR         RESTORE THE PREVIOUS DELIMITER
007321  607365      14790           JMP     INV49           EXIT THIS ROUTINE
007322  042010      14800   INV61   DAC     TEMP8           SET THE FLAG THAT THIS VALUE IS FROM THE PERMANENT SYMBOL TABLE
007323  202001      14810           LAC     TEMP1           RELOAD THE VALUE FOUND
007324  622002      14820           JMP     TEMP2,X         DO THE OPERATION
                    14830   *
                    14840   *       THE SYMBOL IS UNRECOGNIZABLE -- MAYBE IT IS A NUMBER
                    14850   *
    007325          14860   INV30   ...
007325  202003      14870           LAC     TEMP3
007326  043451      14880           DAC     TSBPTR          BACK UP TO THE START OF THE LAST WORD
007327  202004      14890           LAC     TEMP4           RELOAD THE LATEST COUNT
007330  043456      14900           DAC     TSCOUNT         AND BACK IT UP TO THE PREVIOUS WORD
007331  202005      14910           LAC     TEMP5
007332  043455      14920           DAC     TSDLMTR         RESTORE THE PREVIOUS DELIMITER
    007333          14930           NUM                     AND TRY READING THE INPUT AS A NUMBER
007334  741000      14940           SKP                     FAILURE -- INPUT IS NOT RECOGNIZABLE
007335  622002      14950           JMP     TEMP2,X         DO THE OPERATION
007336  202010      14960           LAC     TEMP8
```

D MTSS DEBUGGER -- INVAL SUBROUTINE

```
007337  741200  14970           SNA
007340  613165  14980           JMP     MSG87           TRULY AN ILLEGAL SYMBOL
007341  607313  14990           JMP     INV62           ELSE EXIT NORMALLY AFTER BACKING THE POINTERS
                15000   *
                15010   *
                15020   *       NOW THE OPERATION IS DONE -- CHECK WHETHER OR
                15030   *       NOT THERE ARE MORE TERMS TO COME
                15040   *
    007342      15050   INV45   ...,
    007342      15060           DELIM                   LOAD THE DELIMITER
007343  553371  15070           SAD     ($EXCLAM)       (!)
007344  607405  15080           JMP     INV31
007345  553372  15090           SAD     ($AMPRSN)       (&)
007346  607407  15100           JMP     INV32
007347  553373  15110           SAD     ($STAR)         (*)
007350  607411  15120           JMP     INV33
007351  553374  15130           SAD     ($PLUS)         (+)
007352  607413  15140           JMP     INV34
007353  553375  15150           SAD     ($MINUS)        (-)
007354  607415  15160           JMP     INV35
007355  553376  15170           SAD     ($SLASH)        (/)
007356  607421  15180           JMP     INV36
007357  553377  15190           SAD     ($BSLASH)       (\)
007360  607423  15200           JMP     INV37
007361  553350  15210           SAD     ($SPACE)        ( )
007362  607425  15220           JMP     INV38
007363  553400  15230           SAD     ($COMMA)        (,)
007364  607427  15240           JMP     INV38_1
                15250   *
                15260   *       THE DELIMITER IS NOT AN OPERATOR, RELOAD THE VALUE AND EXIT
                15270   *
007365  202050  15280   INV49   LAC     INDIR           LOAD THE INDIRECT FLAG
007366  740200  15290           SZA                     SKIP IF NO INDIRECTION HAS BEEN INDICATED
007367  607375  15300           JMP     INV48
007370  202006  15310           LAC     TEMP6           LOAD THE VALUE RECEIVED FLAG
007371  740200  15320           SZA                     SKIP IF NONE HAS BEEN RECEIVED
007372  443251  15330           INX     INVAL           ELSE BUMP THE RETURN TO INDICATE SUCCESS
007373  202000  15340           LAC     TEMP0           RELOAD THE ACCUMULATED VALUE
007374  623251  15350           RET     INVAL,X         AND EXIT
                15360
    007375      15370   INV48   ...,
007375  777777  15380           LAW     -1
007376  342050  15390           TAD     INDIR           COUNT THE INDIRECT LEVEL DOWN
007377  042050  15400           DAC     INDIR
007400  202000  15410           LAC     TEMP0           LOAD THE ADDRESS FOR THE INDIRECTION
007401  103312  15420           JMS     LOCAT           FIND IT IN THE OPEN FILE
007402  222162  15430           LAC     BPTR,X          LOAD THE INDIRECT WORD
007403  042000  15440           DAC     TEMP0           SET THE NEW VALUE
007404  607365  15450           JMP     INV49           FINISH UP AND EXIT
                15460
007405  767522  15470   INV31   LAW     INIOR           LOGICAL OR
007406  607430  15480           JMP     INV39
```

D MTSS DEBUGGER -- INVAL SUBROUTINE

```
007407 767516 15490 INV32  LAW  INAND          LOGICAL AND
007410 607430 15500        JMP  INV39
007411 767465 15510 INV33  LAW  INMUL          MULTIPLICATION
007412 607430 15520        JMP  INV39
007413 767456 15530 INV34  LAW  INPLU          ADDITION
007414 607430 15540        JMP  INV39
007415 767462 15550 INV35  LAW  INMIN          LOAD THE SUBTRACT SWITCH
007416 542002 15560        SAD  TEMP2          CHECK THE OLD SWITCH FOR ALSO SUBTRACTION
007417 607413 15570        JMP  INV34          YES, AND TWO MINUSES MAKE A PLUS
007420 607430 15580        JMP  INV39          NO, SO SET UP THE SUBTRACTION
007421 767471 15590 INV36  LAW  INDIV          DIVISION
007422 607430 15600        JMP  INV39
007423 767520 15610 INV37  LAW  INXOR          LOGICAL EXCLUSIVE OR
007424 607430 15620        JMP  INV39
007425 767461 15630 INV38  LAW  INSPA          SPACE MAY TERMINATE THE VALUE OR MAY BE <OPCODE> <ADDRESS>
007426 607430 15640        JMP  INV39
007427 767531 15650 INV381 LAW  INCOM          COMMA MAY TERMINATE THE VALUE OR INDICATE INDIRECT ADDRESSING
               15660
007430 042002 15670 INV39  DAC  TEMP2          SET THE NEW OPERATOR SWITCH
007431 607254 15680        JMP  INV2           GET THE NEXT TERM
               15690 *
               15700 *
               15710 *     THE FIRST WORD OF INPUT IS A NULL, IF THE DELIMITER IS A DOLLAR SIGN ($)
               15720 *     THEN A SYMBOL FOLLOWS, IF IT IS AN INDIRECT SIGN, THEN INDIRECT
               15730 *     ADDRESSING IS BEING REQUESTED, OTHERWISE IT IS TRUE NULL INPUT.
               15740 *
    007432     15750 INULL ...
007432 202005 15760        LAC  TEMP5          GET THE DELIMITER
007433 550325 15770        SAD  PCSGN          CHECK FOR THE CURRENT PROGRAM COUNTER SIGN
007434 607443 15780        JMP  INPC           YES -- GO SET IT UP
    007435     15790        DELIM               GET THE LAST DELIMITER
007436 550316 15800        SAD  INDSN          CHECK FOR AN INDIRECT ADDRESS REQUEST
007437 741000 15810        SKP
007440 607342 15820        JMP  INV45          NOW CHECK FOR A LOGICAL OR ARITHMETIC OPERATOR
               15830 *
               15840 *     INDIRECT ADDRESSING HAS BEEN REQUESTED
               15850 *
007441 442050 15860 ININD  INX  INDIR          FLAG THE REQUEST FOR INDIRECTION
007442 607254 15870        JMP  INV2           AND GET THE NEXT WORD
               15880 *
               15890 *     USE THE CURRENT VALUE OF THE PC, TERMINATED BY AN IMPLIED PLUS SIGN
               15900 *
    007443     15910 INPC  ...
007443 442006 15920        INX  TEMP6          FLAG THERE HAS BEEN AT LEAST ONE COMMAND COMPLETED
007444 202002 15930        LAC  TEMP2          LOAD THE BRANCH CONTROL
007445 553367 15940        SAD  (LAW INSPA)    CHECK FOR PREVIOUS DELIMITER WAS A SPACE
007446 607453 15950        JMP  INPC1          YES -- BACK UP THE POINTERS AND EXIT
007447 553370 15960        SAD  (LAW INCOM)    CHECK FOR A COMMA
007450 607453 15970        JMP  INPC1          YES -- BACK UP THE POINTERS AND EXIT
007451 202043 15980        LAC  PC             LOAD THE PC TO USE AS THE VALUE
007452 622002 15990        JMP  TEMP2,X        AND INCLUDE IT IN THE CALCULATIONS
               16000
```

D MTSS DEBUGGER -- INVAL SUBROUTINE

```
007453  16010  INPC1  DELIM                GET THE REAL DELIMITER
007454  042005 16020         DAC    TEMP5  AND SET IT FOR RESTORATION
007455  607313 16030         JMP    INV62  BACK UP THE POINTERS AND EXIT
```

```
                    16040           .EJECT
                    16050   *
                    16060   *
                    16070   *       LOGICAL AND ARITHMETIC OPERATIONS SECTION
                    16080   *       COMBINE THE AC WITH TEMP0, STORE THE RESULT IN TEMP0, AND RETURN TO THE MAIN ROUTINE
                    16090   *
    007456          16100   INPLU   ...                     TEMP0 := TEMP0 + AC
007456  342000      16110           TAD     TEMP0
007457  042000      16120   INPL2   DAC     TEMP0
007460  607342      16130           JMP     INV45
                    16140
                    16150
    007461          16160   INSPA   ...                     SPACE -- TEMP0 := TEMP0 + AC
007461  607456      16170           JMP     INPLU
                    16180
                    16190
    007462          16200   INMIN   ...                     TEMP0 := TEMP0 - AC
    007462          16210           NEG                     NEGATE THE AC
007464  607456      16220           JMP     INPLU           THEN IS SAME AS ADDITION
                    16230
                    16240
    007465          16250   INMUL   ...                     TEMP0 := TEMP0 * AC
007465  652000      16260           LMQ                     SET THE MULTIPLICAND
007466  213401      16270           LAC     (MULS)          LOAD A MULTIPLY INSTRUCTION
007467  103253      16280           JMS     MULDIV          DO THE MULTIPLICATION
007470  741400      16290           SZL                     MULTIPLY TEST FOR A NEED TO ADJUST THE ONE'S COMPLEMENT ANSWER
                    16300
                    16310
    007471          16320   INDIV   ...                     TEMP0 := TEMP0/AC
007471  652000      16330           LMQ                     SET THE DIVISOR
007472  213402      16340           LAC     (IDIVS)         LOAD A DIVIDE INSTRUCTION
007473  103253      16350           JMS     MULDIV          DO THE DIVISION
007474  741100      16360           SPA                     DIVIDE TEST FOR A NEED TO ADJUST THE ONE'S COMPLEMENT ANSWER
                    16370
                    16380
    007475          16390           ENTER   MULDIV
                                    .PMC    SAVE,ON
    003253                  MULDIV  ...
007475  043255      16400           DAC     MD1             SET THE INSTRUCTION
007476  202000      16410           LAC     TEMP0           LOAD THE MULTIPLIER/DIVIDEND
007477  741100      16420           SPA
007500  353332      16430           TAD     (-1)            NEED TO ADJUST NEGATIVE NUMBERS TO BE ONE'S COMPLEMENT
007501  042000      16440           DAC     TEMP0           RESTORE THE MULTIPLIER/DIVIDEND
007502  641002      16450           LACQ                    LOAD THE MULTIPLICAND/DIVISOR
007503  741100      16460           SPA
007504  353332      16470           TAD     (-1)            NEED TO ADJUST NEGATIVE NUMBERS TO BE ONE'S COMPLEMENT
007505  664000      16480           GSM                     SET UP FOR THE OPERATION
007506  043256      16490           DAC     MD2             SET THE OPERAND
007507  202000      16500           LAC     TEMP0
007510  603255      16510           JMP     MD1             DO THE OPERATION
    003255          16520           .USE    IMPURE
003255  740040      16530   MD1     XX                      SET THE MULTIPLY/DIVIDE INSTRUCTION HERE
```

D MTSS DEBUGGER -- INVAL SUBROUTINE

```
003256  740040  16540  MD2     XX                      SET THE MULTIPLICAND/DIVISOR HERE
003257  607511  16550          JMP     MD4             RETURN TO PURE CODE
    007511      16560          .USE    PURE
    007511      16570  MD4     ...
007511  740010  16580          RAL                     MOVE THE SIGN OF THE PRODUCT TO THE LINK
007512  641002  16590          LACQ                    LOAD THE RESULT
007513  423253  16600          XCT     MULDIV,X        TEST THE NEED TO ADJUST TO BE ONE'S COMPLEMENT
007514  353331  16610          TAD     (1)             IF YES, MAKE THE ADJUSTMENT
007515  607457  16620          JMP     INPL2           SAVE THE RESULT AND EXIT
                16630
                16640
    007516      16650  INAND   ...                     TEMP0 t= TEMP0 (AND) AC
007516  502000  16660          AND     TEMP0           DO THE AND
007517  607457  16670          JMP     INPL2           SAVE THE RESULT AND EXIT
                16680
                16690
    007520      16700  INXOR   ...                     TEMP0 t= TEMP0 (EXCLUSIVE OR) AC
007520  242000  16710          XOR     TEMP0
007521  607457  16720          JMP     INPL2           SAVE THE RESULT AND EXIT
                16730
                16740
    007522      16750  INIOR   ...                     TEMP0 t= TEMP0 (INCLUSIVE OR) AC
007522  042001  16760          DAC     TEMP1
007523  740001  16770          CMA
007524  502000  16780          AND     TEMP0
007525  242001  16790          XOR     TEMP1
007526  607457  16800          JMP     INPL2           SAVE THE RESULT AND EXIT
                16810
                16820
007527  000070  16830  COMX    .ACI6   '  X'
007530  000051  16840  COMI    .ACI6   '  I'
    007531      16850  INCOM   ...                     PREVOIUS DELIMITER WAS COMMA -- TERMINATOR OR INDIRECT ADDRESSING?
007531  203326  16860          LAC     T$WORDB         LOAD THE UPPER HALF OF THE SYMBOL
007532  740200  16870          SZA                     SKIP IF THERE IS NONE
007533  607313  16880          JMP     INV62           OOPS -- NOT INDIRECTION -- BACK UP THE POINTERS AND EXIT
007534  203327  16890          LAC     T$WORDB+1       LOAD THE LOWER HALF OF THE SYMBOL
007535  547527  16900          SAD     COMX
007536  607542  16910          JMP     INCM2           INDIRECT ADDRESSING
007537  547530  16920          SAD     COMI
007540  607542  16930          JMP     INCM2           INDIRECT ADDRESSING
007541  607313  16940          JMP     INV62           NO INDIRECT ADDRESSING -- BACK UP AND EXIT
                16950
007542  213360  16960  INCM2   LAC     (020000)        LOAD THE INDIRECT ADDRESS BIT
007543  607522  16970          JMP     INIOR           AND PUT IT IN THE VALUE
```

```
                D                       MTSS DEBUGGER -- SYMBOL PICKUP SUBROUTINE

                    16980               .STITL  MTSS DEBUGGER -- SYMBOL PICKUP SUBROUTINE
                    16990   *
                    17000   *
                    17010   *       THIS SUBROUTINE EXAMINES THE INPUT STRING CHARACTER BY CHARACTER TO BUILD
                    17020   *       A LEGAL SYMBOL IN TSWORDB AND TSWORDB+1 IF POSSIBLE. EXITS ARE:
                    17030   *            +1 -- NULL INPUT -- FIRST CHARACTER IS A NON-SPACE DELIMITER
                    17040   *            +2 -- SYMBOL SUCCESSFULLY BUILT AND STORED IN TSWORDB & TSWORDB+1
                    17050   *            ELSE -- ILLEGAL SYMBOL ERROR MESSAGE
                    17060   *
                    17070   *       TEMPORARY VARIABLE USAGE:
                    17080   *            TEMP3  -- SAVE TTY BUFFER POINTER IN CASE RESTORATION IS LATER NEEDED
                    17090   *            TEMP4  -- SAVE WORD COUNT IN CASE LATER RESTORATION IS NEEDED
                    17100   *            TEMP9  -- NUMBER OF CHARACTERS IN THE MAIN PART OF THIS SYMBOL
                    17110   *            TEMP10 -- HEAD SYMBOL
                    17120   *
                    17130   *
                    17140   MQOR     .OPDEF  642000          OR THE AC TO THE MQ
                    17150
     007544         17160           ENTER   SYMBOL
                                    .PMC    SAVE,ON
     003260                 SYMBOL  ...
                    17170   *
                    17180   *       FIRST SAVE THE TELETYPE BUFFER POINTERS SO WE CAN BACK UP LATER IF NECESSARY
                    17190   *
007544  203451      17200           LAC     TSBPTR
007545  042003      17210           DAC     TEMP3           SET THE CHARACTER POINTER
007546  203456      17220           LAC     TSCOUNT
007547  042004      17230           DAC     TEMP4           SAVE THE COUNT
     007550         17240           DELIM
007551  042005      17250           DAC     TEMP5           SAVE THE DELIMITER
                    17260   *
                    17270   *       INITIALIZE THE INPUT, THROWING AWAY LEADING BLANKS
                    17280   *
007552  777771      17290           LAW     -7
007553  042011      17300           DAC     TEMP9           INITIALIZE THE CHARACTER COUNT
007554  142012      17310           DZM     TEMP10          INITIALIZE TO NO HEAD SYMBOL
007555  103477      17320           JMS     TSINTIN         RETURN +1 WITH A DELIMITER; ELSE +2
007556  607640      17330           JMP     SYMB1           INITIAL DELIMITER SPECIAL -- PERIOD AMBIGUOUS; DOLLAR SIGN HEAD SYMBOL
                    17340   *
                    17350   *       FIRST CHARACTER
                    17360   *
007557  443260      17370   SYMB3   INX     SYMBOL          BUMP THE RETURN FOR A NON-VACUOUS SYMBOL
007560  103475      17380           JMS     TSFGET          GET THE FIRST CHARACTER
007561  103501      17390           JMS     TSCHRID         IDENTIFY IT
007562  103262      17400           JMS     SYMB4           ROUTINELY ANALYZE THE DELIMITER
007563  740000      17410           NOP                     LETTERS ARE NOT SPECIAL
007564  353403      17420   SYMB31  TAD     (-240)          MAKE IT SIXBIT
007565  650614      17430           CLQILLS 12.             MOVE IT TO THE FIRST CHARACTER POSITION
007566  043326      17440           DAC     TSWORDB         SAVE IT
007567  442011      17450           INX     TEMP9           COUNT THE FIRST ELEMENT OF THE SYMBOL
                    17460   *
                    17470   *       THE SECOND CHARACTER IS ALSO SPECIAL -- IF IT IS A DOLLAR SIGN ($)
```

D MTSS DEBUGGER -- SYMBOL PICKUP SUBROUTINE

```
                    17480   *       THE FIRST CHARACTER WAS A HEAD SYMBOL -- OTHERWISE THE FIRST CHARACTER
                    17490   *       WAS THE TRUE FIRST CHARACTER OF THE MAIN SYMBOL.
                    17500   *
 007570  103475     17510           JMS     T$FGET          GET THE SECOND CHARACTER
 007571  103501     17520           JMS     T$CHRID         IDENTIFY IT
 007572  607671     17530           JMP     SYMB2           DELIMITERS HERE REQUIRE SPECIAL ANALYSIS
 007573  740000     17540           NOP                     LETTERS ARE NOT SPECIAL
                    17550   *
                    17560   *       THE SECOND CHARACTER IS EITHER A LETTER OR A DIGIT. THEREFORE THE
                    17570   *       PREVIOUS CHARACTER WAS NOT A HEAD SYMBOL,
                    17580   *
 007574  442011     17590   SYMB25  INX     TEMP9           COUNT THE SYMBOL ELEMENT
 007575  353403     17600           TAD     (-240)          CONVERT TO SIXBIT
 007576  660706     17610           ALSS    6               MOVE THE CHARACTER TO THE SECOND CHARACTER POSITION
 007577  243326     17620           XOR     T$WORDB         CONCATENATE WITH THE FIRST CHARACTER
 007600  043326     17630           DAC     T$WORDB         SAVE THEM
                    17640   *
                    17650   *       GET THE THIRD CHARCTER AND STORE IT; THEN STORE THE FIRST HALF OF THE SYMBOL
                    17660   *
 007601  103475     17670           JMS     T$FGET          GET THE THIRD CHARCTER
 007602  103501     17680           JMS     T$CHRID         IDENTIFY IT
 007603  103262     17690           JMS     SYMB4           DO THE STANDARD DELIMITER ANALYSIS ON IT
 007604  740000     17700           NOP                     LETTERS ARE NOT SPECIAL
 007605  442011     17710           INX     TEMP9           COUNT THE THIRD CHARACTER
 007606  353403     17720           TAD     (-240)          MAKE ASCII
 007607  243326     17730           XOR     T$WORDB         ACCUMULATE IN THE AC WITH THE FIRST TWO CHARACTERS
 007610  043326     17740           DAC     T$WORDB         SAVE THE FIRST HALF OF THE SYMBOL
                    17750   *
                    17760   *        GET THE LOWER HALF OF THE SYMBOL
                    17770   *
 007611  651000     17780           EAECLA!CLQ              INITIALIZE THE SECOND HALF ACCUMULATION
 007612  103264     17790           JMS     SYMB5           GET THE FOURTH SIXBIT CHARACTER IIN AC(12-17)
 007613  660714     17800           ALSS    12.             MOVE IT TO THE FOURTH CHARACTER POSITION
 007614  103264     17810           JMS     SYMB5           SAVE IT AND GET THE NEXT CHARACTER
 007615  660706     17820           ALSS    6               MOVE IT TO THE FIFTH CHARCTER POSITION
 007616  103264     17830           JMS     SYMB5           SAVE IT AND GET THE NEXT CHARACTER
 007617  642000     17840           MQOR                    SAVE THE SIXTH CHARACTER
 007620  142012     17850           DZM     TEMP10          THE HEAD SYMBOL IS CANCELLED DUE TO SIX OR MORE CHARS IN MAIN SYMBOL
                    17860   *
                    17870   *
 007621  103475     17880   SYMB6   JMS     T$FGET          GET THE NEXT SYMBOL
 007622  103501     17890           JMS     T$CHRID         IDENTIFY IT
 007623  103262     17900           JMS     SYMB4           DO THE STANDARD DELIMITER ANALYSIS
 007624  607621     17910           JMP     SYMB6           ELSE LOOP TO THROW AWAY THE CHARACTER
                    17920   *
                    17930   *        THE SYMBOL IS COMPLETE -- RIGHT JUSTIFY IT AND RESTORE THE HEAD SYMBOL
                    17940   *
 007625  203326     17950   SYMB7   LAC     T$WORDB         LOAD THE HIGH-ORDER HALF OF THE SYMBOL
 007626  442011     17960   SYMB71  ISZ     TEMP9           CHECK FOR JUSTIFICATION COMPLETE
 007627  745000     17970           SKP!CLL                 NO -- PROTECT THE SHIFT AND JUSTIFY BY ONE MORE CHARACTER
 007630  607633     17980           JMP     SYMB8           SYMBOL IS JUSTIFIED
 007631  640506     17990           LRS     6               JUSTIFY BY ONE MORE CHARACTER
```

D MTSS DEBUGGER -- SYMBOL PICKUP SUBROUTINE

```
007632  607626  18000           JMP     SYMB71          SEE IF MORE JUSTIFICATION IS NEEDED
                18010
007633  242012  18020   SYMB8   XOR     TEMP10           PUT ANY HEAD SYMBOL INTO THE SYMBOL
007634  043326  18030           DAC     T$WORDB         STORE THE UPPER HALF SYMBOL
007635  641002  18040           LACQ
007636  043327  18050           DAC     T$WORDB+1       STORE THE LOWER HALF OF THE SYMBOL
007637  623260  18060           RET     SYMBOL,X        EXIT
                18070
                18080
007640  553404  18090   SYMB1   SAD     ($PERIOD)
007641  607647  18100           JMP     SYMB11          FURTHER CHECKS NEEDED IF PERIOD WAS THE FIRST CHARACTER
007642  553362  18110           SAD     ($DOLLAR)
007643  607557  18120           JMP     SYMB3           DOLLAR SIGN FOR THE FIRST SYMBOL MERELY DENOTES NO HEADSYMBOL
007644  553405  18130           SAD     ($SHARP)
007645  607660  18140           JMP     SYMB13          SHARP SIGN IS A NORMAL CHARACTER IN THE SYMBOL
007646  623260  18150           RET     SYMBOL,X        IMMEDIATE RETURN FOR NULL INPUT
                18160
    007647      18170   SYMB11  ...;
007647  042005  18180           DAC     TEMP5           SAVE THE PERIOD
007650  103475  18190           JMS     T$FGET          GET THE CHARACTER AFTER THE PERIOD
007651  103501  18200           JMS     T$CHRID         IDENTIFY IT
007652  623260  18210           RET     SYMBOL,X        EXIT -- MEANS TO USE THE PC
007653  740000  18220           NOP                     LETTERS ARE NOT SPECIAL
007654  777776  18230           LAW     -2              LOAD THE AMOUNT TO BACK UP TO INCLUDE THE PERIOD IN THE SYMBOL
007655  343451  18240   SYMB14  TAD     T$BPTR          ADD THE TELETYPE BUFFER POINTER
007656  043451  18250           DAC     T$BPTR          STORE THE BACKED-UP POINTER
007657  607557  18260           JMP     SYMB3           PERIOD IS A LEGAL COMPONENT OF THE SYMBOL
007660  777777  18270   SYMB13  LAW     -1              LOAD THE AMOUNT TO BACK UP FOR A SHARP SIGN
007661  607655  18280           JMP     SYMB14          DO IT
                18290
                18300
    007662      18310           ENTER   SYMB4           STANDARD DELIMITER ANALYSIS
                                ,PMC    SAVE,ON
    003262              SYMB4   ...;
007662  553362  18320           SAD     ($DOLLAR)
007663  613177  18330           JMP     MSG88           ILLEGAL HEAD SYMBOL ATTEMPT
007664  553405  18340           SAD     ($SHARP)
007665  623262  18350           RET     SYMB4,X         SHARP SIGN IS A LEGAL SYMBOL COMPONENT
007666  553404  18360           SAD     ($PERIOD)
007667  623262  18370           RET     SYMB4,X         PERIOD IS A LEGAL SYMBOL COMPONENT
007670  607625  18380           JMP     SYMB7           ELSE THE SYMBOL HAS BEEN COMPLETED
                18390
007671  553362  18400   SYMB2   SAD     ($DOLLAR)
007672  607675  18410           JMP     SYMB21          DOLLAR SIGN MEAN THE FIRST CHARACTER WAS A HEAD SYMBOL
007673  103262  18420           JMS     SYMB4           ELSE  DO NORMAL DELIMITER ANALYSIS
007674  607574  18430           JMP     SYMB25          RETURN
                18440
007675  202012  18450   SYMB21  LAC     TEMP10          LOAD THE PREVIOUS HEAD SYMBOL
007676  740200  18460           SZA                     SKIP IF NONE
007677  613177  18470           JMP     MSG88           ELSE A SECOND DOLLAR SIGN IS AN ILLEGAL HEAD SYMBOL CONSTRUCTION
007700  641002  18480           LACQ
007701  042012  18490           DAC     TEMP10          SET THE VALID HEAD SYMBOL
```

D MTSS DEBUGGER -- SYMBOL PICKUP SUBROUTINE

```
007702  777771  18500          LAW     -7              RELOAD THE CHARCTER COUNT
007703  042011  18510          DAC     TEMP9           AND RESET IT
007704  607557  18520          JMP     SYMB3           GET A NEW FIRST CHARACTER
                18530
                18540
   007705       18550          ENTER   SYMB5           STORE THE PREVIOUS CHARACTER AND GET A NEW SIXBIT CHAR IN AC(12-17)
                               ,PMC    SAVE,ON
   003264               SYMB5  ...
007705  642000  18560          MQOR                    ACCUMULATE THE NEW CHARACTER IN THE MQ WITH THE PREVIOUS ONES
007706  103475  18570          JMS     TSFGET          GET THE NEXT CHARACTER
007707  103501  18580          JMS     TSCHRID         IDENTIFY IT
007710  103262  18590          JMS     SYMB4           ANALYZE A DELIMITER
007711  740000  18600          NOP                     LETTERS ARE NOT SPECIAL
007712  442011  18610          INX     TEMP9           COUNT THE CHARACTER
007713  353403  18620          TAD     (-240)          MAKE IT ASCII
007714  623264  18630          RET     SYMB5,X
```

```
D                              MTSS DEBUGGER -- SYMBOL TABLE LABEL SEARCH SUBROUTINES

                    18640               .STITL  MTSS DEBUGGER -- SYMBOL TABLE LABEL SEARCH SUBROUTINES
                    18650     *
                    18660     *
                    18670     *         SEARCH THE INDICATED SYMBOL TABLE FOR THE LABEL PASSED IN TSWORDB AND
                    18680     *         TSWORDB+1, RETURN TO THE CALLER +2 IF THE LABEL CANNOT BE FOUND;
                    18690     *         +1 WITH THE VALUE IN THE AC IF IT IS FOUND
                    18700     *         TEMP1 IS USED AS A COUNTER
                    18710     *
    007715          18720               ENTER   PSRCH           SEARCH THE PERMANENT SYMBOL TABLE FOR A LABEL MATCH
                                        .PMC    SAVE,ON
    003266                      PSRCH   ...
007715  773447      18730               LAW     SYM0-1          LOAD A POINTER TO THE PERMANENT SYMBOL TABLE
007716  103272      18740               JMS     SRCH            SEARCH IT
007717  443266      18750               INX     PSRCH           BUMP THE RETURN IF WE FAILED TO FIND A MATCH
007720  623266      18760               RET     PSRCH,X         EXIT
                    18770
                    18780
    007721          18790               ENTER   USRCH           SEARCH THE USER SYMBOL TABLE FOR A LABEL MATCH
                                        .PMC    SAVE,ON
    003270                      USRCH   ...
007721  774564      18800               LAW     SYM1-1          LOAD A POINTER TO THE USER SYMBOL TABLE
007722  103272      18810               JMS     SRCH            AND SEARCH IT
007723  443270      18820               INX     USRCH           BUMP THE RETURN IF WE FAILED TO FIND A MATCH
007724  623270      18830               RET     USRCH,X         EXIT
                    18840     *
                    18850     *
                    18860     *         SEARCH THE SYMBOL TABLE BEGINNING AT THE POINTER PASSED IN THE AC
                    18870     *         +1 WITH THE VALUE IN THE AC IF IT IS FOUND
                    18880     *         FOR FAILURE, RETURN +2 WITH THE VALUE IN THE AC FOR SUCCESS.
                    18890     *
    007725          18900               ENTER   SRCH
                                        .PMC    SAVE,ON
    003272                      SRCH    ...
007725  040010      18910               DAC     10              SET THE SYMBOL TABLE POINTER
007726  220010      18920               LAC     10,X
007727  042001      18930               DAC     TEMP1           SET MINUS THE NUMBER OF SYMBOLS IN THIS TABLE
    007730          18940     SRCH1     ...
007730  442001      18950               ISZ     TEMP1           DONE??
007731  741000      18960               SKP                     NO
007732  623272      18970               RET     SRCH,X          YES -- RETURN +1 FOR FAILURE
007733  220010      18980               LAC     10,X            LOAD THE FIRST HALF OF THE NEXT SYMBOL
007734  543326      18990               SAD     TSWORDB         NO -- TEST AGAINST THE GIVEN SYMBOL
007735  741000      19000               SKP                     GOOD MATCH -- KEEP ON TRYING
007736  607743      19010               JMP     SRCH2           NO MATCH -- TRY THE NEXT SYMBOL
007737  220010      19020               LAC     10,X            LOAD THE SECOND HALF OF THE SYMBOL
007740  543327      19030               SAD     TSWORDB+1       AND TEST IT
007741  607746      19040               JMP     SRCH4           EUREKA...
007742  741000      19050               SKP
                    19060
007743  440010      19070     SRCH2     INX     10              MOVE THE POINTER  BY THE SECOND HALF OF THE SYMBOL
007744  440010      19080               INX     10              MOVE THE POINTER BY THE VALUE
007745  607730      19090               JMP     SRCH1           LOOP
```

D MTSS DEBUGGER -- SYMBOL TABLE LABEL SEARCH SUBROUTINES

```
    007746      19100   SRCH4   ...
007746  220010  19110           LAC     10,X            LOAD THE VALUE OF THE FOUND SYMBOL
007747  443272  19120           INX     SRCH            BUMP THE RETURN FOR SUCCESS
007750  623272  19130           RET     SRCH,X
```

```
          D                         MTSS DEBUGGER -- SYMBOL TABLE VALUE SEARCH SUBROUTINES

                    19140           .STITL  MTSS DEBUGGER -- SYMBOL TABLE VALUE SEARCH SUBROUTINES
                    19150   *
                    19160   *
    007751          19170           ENTER   PVSCH           SEARCH THE PERMANENT SYMBOL TABLE FOR THE CLOSEST MATCH TO VALUE IN TEMP0
                                    .PMC    SAVE,ON
    003274                  PVSCH   ...
007751  773447  19180               LAW     SYM0-1          LOAD A POINTER TO THE PERMANENT SYMBOL TABLE
007752  103300  19190               JMS     VSRCH           FIND THE CLOSEST MATCH IN THE PERMANENT TABLE
007753  623274  19200               RET     PVSCH,X
                19210
                19220
    007754          19230           ENTER   UVSCH           SEARCH THE USER SYMBOL TABLE FOR THE CLOSEST MATCH TO THE VALUE IN TEMP0
                                    .PMC    SAVE,ON
    003276                  UVSCH   ...
007754  774564  19240               LAW     SYM1-1          LOAD A POINTER TO THE USER SYMBOL TABLE
007755  103300  19250               JMS     VSRCH           FIND THE CLOSEST MATCH IN THE USER TABLE
007756  623276  19260               RET     UVSCH,X
                19270   *
                19280   *
                19290   *       SEARCH THE SYMBOL TABLE WHOSE POINTER IS PASSED IN THE AC FOR
                19300   *       THE CLOSEST MATCH TO THE VALUE PASSED IN TEMP0. RETURN WITH AUTOINDEX
                19310   *       REGISTER 10 POINTING TO THE VALUE FOUND.
                19320   *
                19330   *       TEMPORARY VARIABLE USAGE:
                19340   *            TEMP0 -- THE VALUE TO BE MATCHED AS CLOSELY AS POSSIBLE
                19350   *            TEMP2 -- SYMBOL TABLE LENGTH COUNTER
                19360   *            TEMP3 -- POINTER TO THE BEST VALUE FOUND SO FAR
                19370   *            TEMP4 -- ABSOLUTE VALUE OF THE DIFFERENCE BETWEEN THE VALUE
                19380   *                     SOUGHT AND THE VALUE FOUND
                19390   *
    007757          19400           ENTER   VSRCH
                                    .PMC    SAVE,ON
    003300                  VSRCH   ...
007757  040010  19410               DAC     10              SET THE POINTER TO THE SYMBOL TABLE TO SEARCH
007760  220010  19420               LAC     10,X            LOAD THE SYMBOL TABLE COUNT
007761  042002  19430               DAC     TEMP2           SET THE SYMBOL TABLE LENGTH
007762  213344  19440               LAC     (377777)        LOAD A RIDICULOUS OFFSET
007763  042004  19450               DAC     TEMP4           SET IT
    007764          19460   VSR2    ...
007764  442002  19470               ISZ     TEMP2           CHECK FOR DONE
007765  741000  19480               SKP                     NOT YET
007766  623300  19490               RET     VSRCH,X         DONE -- EXIT
007767  440010  19500               INX     10              MOVE THE POINTER PAST THE FIRST HALF SYMBOL
007770  440010  19510               INX     10              MOVE THE POINTER PAST THE SECOND HALF-SYMBOL
007771  220010  19520               LAC     10,X            LOAD THE NEXT VALUE TO BE COMPARED
    007772          19530           NEG                     NEGATE IT
007774  342000  19540               TAD     TEMP0           SUBTRACT IT FROM THE DESIRED VALUE
007775  664000  19550               GSM                     GET SIGN AND MAGNITUDE OF THE DIFFERENCE BETWEEN THE VALUES
007776  652000  19560               LMQ                     SAVE THE MAGNITUDE
    007777          19570           NEG                     NEGATE THE MAGNITUDE
010001  342004  19580               TAD     TEMP4           ADD THE OLD BEST MAGNITUDE
010002  741100  19590               SPA
```

D MTSS DEBUGGER -- SYMBOL TABLE VALUE SEARCH SUBROUTINES

```
010003  607764  19600          JMP     VSR2            THE OLD ONE WAS BETTER -- KEEP IT AND LOOP
010004  200010  19610          LAC     10
010005  042003  19620          DAC     TEMP3           THE NEW ONE IS BETTER -- RECORD ITS POINTER
010006  641002  19630          LACQ
010007  042004  19640          DAC     TEMP4           RECORD ITS MAGNITUDE
010010  607764  19650          JMP     VSR2            LOOP
```

```
D                       MTSS DEBUGGER -- SYMBOL TABLE BIT SEARCH SUBROUTINE

                19660           .STITL  MTSS DEBUGGER -- SYMBOL TABLE BIT SEARCH SUBROUTINE
                19670   *
                19680   *
                19690   *       SEARCH THE PERMANENT SYMBOL TABLE FOR THE VALUE  WHOSE BITS
                19700   *       ARE THE LARGEST SUBSET OF THE BITS IN TEMP0,
                19710   *       AND WHOSE MASKED AREA IS IDENTICAL.
                19720   *
                19730   *       TEMPORARY VARIABLE USAGE:
                19740   *            TEMP0 -- THE VALUE WHOSE BITS ARE TO BE MATCHED AS CLOSELY AS POSSIBLE
                19750   *            TEMP1 -- SCRATCH
                19760   *            TEMP2 -- TABLE LENGTH
                19770   *            TEMP3 -- POINTER TO THE BEST VALUE FOUND SO FAR
                19780   *            TEMP4 -- NUMBER OF MATCHING BITS IN THE BEST VALUE
                19790   *            TEMP5 -- CURRENT VALUE BEING COMPARED
                19800   *            TEMP6 -- SCRATCH
                19810   *            TEMP7 -- OP CODE FOR THE VALUE
                19820   *            TEMP8 -- RETAINING MASK FOR THE OP CODE FIELD
                19830   *
    010011      19840           ENTER   PBSCH
                                .PMC    SAVE,ON
    003302              PBSCH   ...
 010011  773447  19850           LAW     SYM0-1
 010012  040010  19860           DAC     10                      SET THE TABLE POINTER
 010013  220010  19870           LAC     10,X
 010014  042002  19880           DAC     TEMP2                   SET THE TABLE COUNT
 010015  142004  19890           DZM     TEMP4                   INITIALIZE TO NO BITS MATCHED
    010016      19900   PBS2    ...                             CHECK THE NEXT LIST ELEMENT FOR A BETTER MATCH
 010016  442002  19910           ISZ     TEMP2                   CHECK FOR DONE
 010017  741000  19920           SKP                             NO
 010020  623302  19930           RET     PBSCH,X                 YES -- EXIT
 010021  440010  19940           INX     10
 010022  440010  19950           INX     10                      MOVE THE POINTER TO THE NEXT VALUE
 010023  220010  19960           LAC     10,X
 010024  042005  19970           DAC     TEMP5                   SET THE NEW VALUE TO COMPARE
 010025  502010  19980           AND     TEMP8                   APPLY THE MASK
 010026  542007  19990           SAD     TEMP7                   COMPARE THE MASKED AREAS
 010027  741000  20000           SKP                             THEY ARE IDENTICAL, SO CONTINUE
 010030  610016  20010           JMP     PBS2                    THEY ARE DIFFERENT; SO TRY THE NEXT ONE
 010031  202007  20020           LAC     TEMP7                   LOAD THE OLD EXTENDED OP CODE
 010032  513301  20030           AND     (740000)                RECOVER THE OP CODE
 010033  242000  20040           XOR     TEMP0                   ADD IN THE VALUE WE ARE TRYING TO MATCH
 010034  740001  20050           CMA                             FORM THE SUBSET MASK
 010035  502005  20060           AND     TEMP5                   KEEP THOSE BITS NOT IN THE ORIGINAL VALUE
 010036  740200  20070           SZA                             SKIP IF THERE ARE NONE (I.E. SKIP IF NEW IS A SUBSET OF THE OLD)
 010037  610016  20080           JMP     PBS2                    ELSE LOOP ON THE NEXT SYMBOL
                 20090   *
                 20100   *      COUNT THE NUMBER OF BITS THE OLD VALUE (TEMP0) AND THE VALUE BEING
                 20110   *      TESTED (TEMP5) HAVE IN COMMON, REPLACE THE BEST-VALUE POINTER
                 20120   *      IF THIS IS BETTER THAN THE PREVIOUS NUMBER OF COMMON BITS.
                 20130   *
 010040  142001  20140           DZM     TEMP1                   ZERO THE COMMON BITS COUNT
 010041  777767  20150           LAW     -9.                     LOAD THE NUMBER OF BIT PAIRS
```

D MTSS DEBUGGER -- SYMBOL TABLE BIT SEARCH SUBROUTINE

```
010042  042006  20160          DAC     TEMP6           SET THE LOOP CONTROL
010043  202000  20170          LAC     TEMP0           RELOAD THE OLD VALUE
010044  502005  20180          AND     TEMP5           KEEP THOSE BITS THE TWO WORDS HAVE IN COMMON
010045  745010  20190          CLL!RAL!SKP
    010046      20200  PBS4    ...
010046  742010  20210          RTL                     MOVE THE NEXT BIT PAIR TO THE LINK AND AC(0)
010047  741400  20220          SZL                     SKIP UNLESS THE NEXT BIT WAS ON
010050  442001  20230          INX     TEMP1           YES -- COUNT IT
010051  741100  20240          SPA                     SKIP UNLESS THE NEXT BIT WAS ON
010052  442001  20250          INX     TEMP1           YES -- COUNT IT
010053  442006  20260          ISZ     TEMP6           SEE IF DONE LOOPING
010054  610046  20270          JMP     PBS4            NO -- LOOP AGAIN
                20280
010055  777777  20290          LAW     -1
010056  342001  20300          TAD     TEMP1
010057  740001  20310          CMA                     AC = MINUS NUMBER OF COMMON BITS
010060  342004  20320          TAD     TEMP4           ADD THE PREVIOUS BEST NUMBER
010061  740100  20330          SMA                     SKIP IF THE NEW ONE IS BETTER
010062  610016  20340          JMP     PBS2            ELSE LOOP -- CHECK THE NEXT VALUE
010063  202001  20350          LAC     TEMP1
010064  042004  20360          DAC     TEMP4           REPLACE THE BIT COUNT
010065  200010  20370          LAC     10
010066  042003  20380          DAC     TEMP3           REPLACE THE POINTER TO THE BEST VALUE
010067  610016  20390          JMP     PBS2            LOOP
```

```
D                              STSS DEBUGGER -- MISCELLANEOUS SUBROUTINES

                 20400         .STITL  STSS DEBUGGER -- MISCELLANEOUS SUBROUTINES
                 20410  *
                 20420  *
                 20430  *      THE FILE PARAMETERS HAVE ALREADY BEEN FILLED IN AND NOTHING OF
                 20440  *      IMPORTANCE IS LEFT IN THE BUFFER, NOW SET UP THE BUFFER PARAMETERS
                 20450  *      WITH THE MINIMUM CORE ADDRESS PRESENT IN THE BUFFER SET TO BE
                 20460  *      GREATER THAN THE MAXIMUM, THIS WILL FORCE A PAGE OPERATION
                 20470  *      THE FIRST TIME THE BUFFER IS REFERENCED.
                 20480  *
                 20490  *
                 20500  *
    010070       20510          ENTER   SNCOP
                                .PMC    SAVE,ON
    003304               SNCOP  ...
010070  142161   20520          DZM     BMAX            SET THE MAXIMUM ADDRESS IS 0
010071  142157   20530          DZM     BMIN
010072  442157   20540          INX     BMIN            SET THE MINIMUM ADDRESS IS +1
010073  750001   20550          CLC
010074  042160   20560          DAC     MBMIN           SET MINUS THE MINIMUM BUFFER ADDRESS
010075  202163   20570          LAC     FDA
010076  042153   20580          DAC     BDA             SET THE BUFFER ADDRESS TO CANCEL OUT ANY TROUBLESOME OLD ONE
010077  623304   20590          RET     SNCOP,X
                 20600  *
                 20610  *
                 20620  *
    010100       20630          ENTER   UCORE
                                .PMC    SAVE,ON
    003306               UCORE  ...
    010100       20640          MPOFF
                                .PMC    SAVE,ON
010100  705000                 SPECIAL+0               TURN OFF MEMORY PROTECT
010101  201766   20650          LAC     SUCORE
010102  701742   20660          MPEU
010103  744000   20670          CLL                     PROTECT THE SHIFT
010104  640510   20680          LRS     8,              DIVIDE BY 400 TO GET THE BLOCK NUMBER
010105  253307   20690          XOR     (040000)        ADD IN THE DISK 0 DEVICE ADDRESS
010106  042163   20700          DAC     FDA             SET THE FILE DEVICE ADDRESS
010107  740001   20710          CMA
010110  042164   20720          DAC     MFDA
010111  442164   20730          INX     MFDA            SET MINUS THE FILE DEVICE ADDRESS
010112  213406   20740          LAC     (SBOUNDARY)
010113  042165   20750          DAC     FMIN            SET THE MINIMUM FILE CORE ADDRESS
010114  776000   20760          LAW     -SBOUNDARY
010115  042166   20770          DAC     MFMIN           SET MINUS THE MINIMUM FILE CORE ADDRESS
010116  210125   20780          LAC     CORCT
010117  042167   20790          DAC     FMAX            SET MINUS THE FILE MAXIMUM CORE ADDRESS
010120  210332   20800          LAC     UCF             LOAD THE USER CORE FILE FLAG
010121  042151   20810          DAC     FTYPE           AND SET THE CURRENT TYPE OF FILE
010122  213276   20820          LAC     (17777)         LOAD A STANDARD ADDRESS MASK
010123  042051   20830          DAC     PCMSK           AND SET IT
010124  623306   20840          RET     UCORE,X
010125  762000   20850  CORCT   -8K
```

D STSS DEBUGGER -- MISCELLANEOUS SUBROUTINES

```
                    20860
                    20870
                    20880
    010126          20890           ENTER   UDISK
                                    ,PMC    SAVE,ON
    003310                  UDISK   ...
    010126          20900           MPOFF
                                    ,PMC    SAVE,ON
010126  705000                      SPECIAL+0               TURN OFF MEMORY PROTECT
010127  201767  20910              LAC     SUDISK
010130  701742  20920              MPEU
010131  744000  20930              CLL                     PROTECT THE SHIFT
010132  640510  20940              LRS     8.              DIVIDE BY 400 TO GET THE BLOCK COUNT
010133  353307  20950              TAD     (040000)        ADD THE DISK DEVICE ADDRESS
010134  042163  20960              DAC     FDA             SET THE FILE DEVICE ADDRESS
010135  740001  20970              CMA
010136  042164  20980              DAC     MFDA
010137  442164  20990              INX     MFDA            SET MINUS THE FILE DEVICE ADDRESS
010140  142165  21000              DZM     FMIN            USER DISK STARTS FROM ZERO
010141  142166  21010              DZM     MFMIN
010142  205720  21020              LAC     DISCT
010143  042167  21030              DAC     FMAX            SET MINUS THE MAXIMUM ADDRESS
010144  623310  21040              RET     UDISK,X
                    21050   *
                    21060   *
                    21070   *       LOCAT IS ENTERED WITH THE DESIRED ADDRESS IN THE AC.
                    21080   *       LOCAT EXITS WITH THE POINTER TO THE DESIRED ADDRESS IN THE AC AND IN BPTR
                    21090   *
    010145          21100           ENTER   LOCAT
                                    ,PMC    SAVE,ON
    003312                  LOCAT   ...
                    21110   *
                    21120   *       SEE WHICH FILE WE ARE WORKING ON
                    21130   *
010145  502051  21140              AND     PCMSK           MASK TO JUST THE ADDRESS
010146  042162  21150              DAC     BPTR            SAVE THE REQUIRED LOCATION
010147  202151  21160              LAC     FTYPE           LOAD THE TYPE OF FILE WE HAVE OPEN
010150  550333  21170              SAD     ACF
010151  610210  21180              JMP     PUCF4           IT IS AN ACTUAL CORE FILE; THE CALLER'S POINTER WAS CORRECT
                    21190   *
                    21200   *       CHECK TO SEE THAT THE ADDRESS IS WITHIN THE CURRENT FILE
                    21210   *
010152  744002  21220              STL                     SET THE OVERFLOW FLAG
010153  202162  21230              LAC     BPTR            RELOAD THE DESIRED LOCATION
010154  342167  21240              TAD     FMAX            SUBTRACT THE MAXIMUM ADDRESS IN THE FILE
010155  740400  21250              SNL                     SKIP ON NO OVERFLOW -- SKIP IF REQUESTED LOCATION IS NOT PAST THE FILE END
010156  613232  21260              JMP     MSG91           YES -- THE ADDRESS IS OUT OF BOUNDS
010157  202166  21270              LAC     MFMIN           LOAD THE MINIMUM ADDRESS IN THE FILE
010160  741200  21280              SNA                     SKIP IF ALL IS CORRECT
010161  744000  21290              CLL                     ELSE CLEAR THE LINK TO MAKE THE CORRECT NUMBER
010162  342162  21300              TAD     BPTR            AND ADD THE DESIRED LOCATION
010163  740400  21310              SNL                     SKIP IF REQUESTED ADDRESS IS BEFORE THE BUFFER START
```

D STSS DEBUGGER -- MISCELLANEOUS SUBROUTINES

```
010164  610171  21320          JMP     PAGE6           NO -- PAGE IT
                21330  *
                21340  *       O.K, -- NOW SEE WHAT KIND OF A FILE WE ARE WORKING WITH
                21350  *
010165  202151  21360          LAC     FTYPE
010166  550332  21370          SAD     UCF
010167  610204  21380          JMP     PUCF            IT IS A USER CORE FILE -- CHECK FOR 0-37
010170  613232  21390          JMP     MSG91           ILLEGAL ADDRESS BEFORE THE BUFFER START
                21400  *
                21410  *       THE FILE IS A NORMAL DEVICE FILE -- IS THE DESIRED LOCATION IN CORE NOW?
                21420  *
    010171      21430  PAGE6   ...
010171  202162  21440          LAC     BPTR            RELOAD THE DESIRED LOCATION
010172  342161  21450          TAD     BMAX            SUBTRACT THE MAXIMUM LOCATION WHICH IS IN THE BUFFER
010173  740300  21460          SMA!SZA                 IS THE DESIRED LOCATION PAST THE BUFFER END?
010174  610217  21470          JMP     PNEW            YES -- GET A NEW PAGE
010175  202162  21480          LAC     BPTR            NO -- RELOAD THE DESIRED LOCATION
010176  342160  21490          TAD     MBMIN           SUBTRACT THE MINIMUM ADDRESS WHICH IS IN THE BUFFER NOW
010177  741100  21500          SPA                     IS THE DESIRED LOCATION BEFORE THE BUFFER START?
010200  610217  21510          JMP     PNEW            YES -- GET A NEW PAGE
                21520  *
                21530  *       THE DESIRED LOCATION IS IN CORE NOW -- SET UP THE POINTER, THE
                21540  *       AC ALREADY CONTAINS THE ADDRESS RELATIVE TO THE START OF THE BUFFER.
                21550  *
010201  342154  21560          TAD     BCA             ADD IN THE BUFFER ADDRESS
010202  042162  21570  PAGE8   DAC     BPTR            SET THE POINTER TO THE BUFFER LOCATION
010203  623312  21580          RET     LOCAT,X
                21590  *
                21600  *       THE FILE IS A USER CORE FILE -- LOCATIONS 0-37 ARE SPECIAL, SO CHECK FOR THEM
                21610  *
    010204      21620  PUCF    ...
010204  777741  21630          LAW     -37             LOAD MINUS THE MAXIMUM LEGAL SPECIAL ADDRESS
010205  342162  21640          TAD     BPTR            ADD THE REQUESTED ADDRESS
010206  740300  21650          SMA!SZA                 IS THE DESIRED LOCATION A LEGAL SPECIAL ADDRESS?
010207  610212  21660          JMP     PUCF2           NO -- IS IT LEGAL AT ALL?
010210  202162  21670  PUCF4   LAC     BPTR            YES, THE ADDRESS IS LEGAL -- DON'T ALTER IT
010211  623312  21680          RET     LOCAT,X
                21690
010212  776000  21700  PUCF2   LAW     -BOUNDARY       LOAD THE MINIMUM NORMAL LEGAL ADDRESS
010213  342162  21710          TAD     BPTR            ADD THE REQUESTED ADDRESS
010214  741100  21720          SPA                     IS IT LEGAL?
010215  613232  21730          JMP     MSG91           ADDRESS OUT OF BOUNDS
010216  610171  21740          JMP     PAGE6           YES -- CARRY ON
                21750  *
                21760  *
                21770  *       PNEW GETS A NEW TWO-BLOCK PAGE SURROUNDING THE LOCATION GIVEN
                21780  *       IN BPTR. THE NEW "PAGE" WILL BE THE BLOCK CONTAINING THE DESIRED
                21790  *       LOCATION AND THE FOLLOWING BLOCK.
                21800  *
    010217      21810  PNEW    ...
010217  103314  21820          JMS     FORCE           FIRST FORCE THE OLD PAGE
010220  762170  21830          LAW     BUFFER
```

D STSS DEBUGGER -- MISCELLANEOUS SUBROUTINES

```
010221  042154  21840          DAC     BCA             RESTORE THE BUFFER CORE ADDRESS
010222  213334  21850          LAC     (BUFLEN)
010223  042155  21860          DAC     BLEN            RESTORE THE BUFFER LENGTH
010224  202162  21870          LAC     BPTR            NO -- RELOAD THE DESIRED ADDRESS
010225  342166  21880          TAD     MFMIN           SUBTRACT THE MINIMUM ADDRESS IN THE FILE
010226  744000  21890          CLL                     PROTECT THE ROTATE
010227  640510  21900          LRS     8,              ELSE DIVIDE BY 400 TO GET THE BLOCK NUMBER IN THE FILE
010230  342163  21910          TAD     FDA             ADD TO THE FILE DEVICE ADDRESS
010231  042153  21920          DAC     BDA             SET THE NEW BUFFER DEVICE ADDRESS
010232  342164  21930          TAD     MFDA            SUBTRACT THE FILE DEVICE ADDRESS
010233  660710  21940          ALSS    8,              MULTIPLY BY 400 TO GET THE OFFSET FROM THE BEGINNING OF THE FILE (IN WORDS)
010234  342165  21950          TAD     FMIN            ADD THE MINIMUM CORE ADDRESS TO GET THE MINIMUM IN THE BUFFER
010235  042157  21960          DAC     BMIN            AND SET IT
010236  740001  21970          CMA
010237  353331  21980          TAD     (1)
010240  042160  21990          DAC     MBMIN           SET MINUS THE MINIMUM ADDRESS
010241  202157  22000          LAC     BMIN            RELOAD THE ADDRESS
010242  353332  22010          TAD     (-1)
010243  353334  22020          TAD     (BUFLEN)        ADD THE BUFFER LENGTH TO GET THE MAXIMUM ADDRESS IN THE BUFFER
010244  740001  22030          CMA
010245  042161  22040          DAC     BMAX
010246  442161  22050          INX     BMAX            SET MINUS THE MAXIMUM ADDRESS IN THE BUFFER
010247  770171  22060  PNEW2   LAW     PAGE6
010250  652000  22070          LMQ                     SET THE RESTART ADDRESS
010251  762153  22080          LAW     BDA             LOAD A POINTER TO THE READ PARAMETERS
010252  705003  22090          PREAD                   READ IN A NEW BUFFER
                22100  *
                22110  *       HARDWARE ERROR RETURN IS HERE
                22120  *
U 010253  100000  22130          JMS     ERROR
  010254  610247  22140          JMP     PNEW2
                22150  *
                22160  *
                22170  *       FORCE IS A ROUTINE TO OUTPUT THE CORE BUFFER IF IT HAS BEEN
                22180  *       ALTERED, AND TO ZERO THE ALTERS FLAG.
                22190  *
     010255     22200          ENTER   FORCE
                               .PMC    SAVE,ON
     003314             FORCE  ...
  010255  202156  22210          LAC     BALT            LOAD THE BUFFER ALTERS FLAG
  010256  142156  22220          DZM     BALT            CLEAR IT IN ANY CASE
  010257  741200  22230          SNA                     SKIP IF THE BUFFER HAS BEEN ALTERED
  010260  623314  22240          RET     FORCE,X         ELSE EXIT NOW
  010261  202151  22250          LAC     FTYPE           LOAD THE CURRENT TYPE OF FILE
  010262  550333  22260          SAD     ACF             SKIP IF IT CAN BE FORCED
  010263  623314  22270          RET     FORCE,X         IT IS AN ACTUAL CORE FILE, AND CANNOT BE FORCED
  010264  203314  22280  FORCE2  LAC     FORCE           LOAD THE RESTART ADDRESS
  010265  652000  22290          LMQ                     AND PASS IT TO THE EXEC
  010266  762153  22300          LAW     BDA             LOAD A POINTER TO THE PARAMETERS
  010267  705005  22310          PWRITE                  WRITE THE BUFFER OUT
                22320  *
                22330  *       ERROR RETURN IS HERE
```

D STSS DEBUGGER -- MISCELLANEOUS SUBROUTINES

```
                    22340   *
U   010270  100000  22350           JMS     ERROR
    010271  610264  22360           JMP     FORCE2          RETRY THE OPERATION
                    22370   *
                    22380   *
                    22390   *       PRINT A SPACE
                    22400   *
        010272      22410           ENTER   SPACE
                                    ,PMC    SAVE,ON
        003316              SPACE   ...
    010272  760240  22420           LAW     240
    010273  103503  22430           JMS     TSTTYOT
    010274  623316  22440           RET     SPACE,X
                    22450   *
                    22460   *       PRINT A COLON FOLLOWED BY A SPACE
                    22470   *
        010275      22480           ENTER   COLSP
                                    ,PMC    SAVE,ON
        003320              COLSP   ...
    010275  760272  22490           LAW     SCOLON
    010276  103503  22500           JMS     TSTTYOT         PRINT THE COLON
    010277  760240  22510           LAW     SSPACE
    010300  103503  22520           JMS     TSTTYOT         PRINT THE SPACE
    010301  623320  22530           RET     COLSP,X
                    22540   *
                    22550   *
        010302      22560           ENTER   TDVAL           MAKE SURE THE USER IS VALIDATED IF HE IS TRYING TO DO A DISK OPERATION
                                    ,PMC    SAVE,ON
        003322              TDVAL   ...
    010302  042000  22570           DAC     TEMP0           SAVE THE DEVICE ADDRESS
    010303  640703  22580           ALS     3               MOVE THE DEVICE-TYPE BIT TO AC(0)
    010304  741100  22590           SPA                     SKIP FOR DECTAPE
    010305  103215  22600           JMS     VALCHK          ELSE CHECK THE USER'S VALIDATION
    010306  202000  22610           LAC     TEMP0           RELOAD THE DEVICE ADDRESS
    010307  623322  22620           RET     TDVAL,X
                    22630
                    22640
                    22650
        010310      22660           ENTER   BINIT
                                    ,PMC    SAVE,ON
        003324              BINIT   ...
    010310  103314  22670           JMS     FORCE           FORCE ANY OLD BUFFER BEFORE MESSING AROUND WITH THE POINTERS
    010311  213407  22680           LAC     (BUFFER)
    010312  042154  22690           DAC     BCA             SET THE BUFFER CORE ADDRESS
    010313  213334  22700           LAC     (BUFLEN)
    010314  042155  22710           DAC     BLEN            SET THE BUFFER LENGTH
    010315  623324  22720           RET     BINIT,X
```

D MTSS DEBUGGER -- MISCELLANEOUS STORAGE

```
                    22730           ,STITL  MTSS DEBUGGER -- MISCELLANEOUS STORAGE
                    22740   *
                    22750   *       DDT CONTROL SIGN DEFINITIONS
                    22760   *
 010316  000336     22770   INDSN   $UPARR                 UP ARROW INDICATES INDIRECT ADDRESSING
 010317  000300     22780   FDEL    $AT                     AT DELIMITS FILENAMES
 010320  000243     22790   RSGN    $SHARP                  SHARP SIGN DENOTES REGISTER NAMES
 010321  000240     22800   LNSGN   $SPACE                  SPACE IN FIELD DENOTES SPECIFICATION BY LENGTH
 010322  000254     22810   BYSGN   $COMMA                  COMMA DENOTE FIELD SPECIFICATION BY BOUNDARIES
 010323  000272     22820   MCSGN   $COLON                  COMMERCIAL AT SIGN DENOTES MODE/COMMAND WHEN NEEDED
 010324  000273     22830   ENDSN   $SCOLON                 SEMI-COLON DENOTES END OF A SYNTACTIC UNIT
 010325  000256     22840   PCSGN   $PERIOD                 PERIOD  IS SYMBOL FOR PROGRAM COUNTER
 010326  000275     22850   PATSN   $EQUAL                  (=) IS PATCH COMMAND
 010327  000274     22860   BKSN    $LESS                   (<) IS BREAKPOINT COMMAND
 010330  000276     22870   JSGN    $GREAT                  (>) IS JUMP/TRANSFER COMMAND
                    22880   *
                    22890   *       MISCELLANEOUS CONSTANTS
                    22900   *
 010331  340000     22910   DTCT    -1100*400
 010332  000001     22920   UCF     1                       USER CORE FILE FLAG
 010333  000002     22930   ACF     2                       ACTUAL CORE FLAG
 010334  000003     22940   LOGF    3                       LOGICAL FILE FLAG
                    22950
                    22960           .END
```

```
                D                       MTSS DEGUGGER -- PERMANENT SYMBOL TABLE

                        9960            ,STITL  MTSS DEGUGGER -- PERMANENT SYMBOL TABLE
           013450       9970            ,USE    SYMTAB          PUT IT LAST IN CORE SO THE USER CAN ADD TO IT INDEFINITELY
                        9980            ,PMC    SAVE,OFF        DON'T EXPAND THE SYMBOL TABLE -- IT GOES ON FOR PAGES!!!
                        9990            ,INSRT  INSERT;OPS
                        100             ,CRSM   SAVE,OFF
                        110     OP      ,DEFIN
                        120             ,ACI6   /#1/
                        130             #1
                        140             ,ENDM   OP
013450  777473          150     SYMO    ,-SYM1-1/3-1            NUMBER OF SYMBOLS IN THE PERMANENT SYMBOL TABLE -1
           013451       160             OP      <   ABS>
013454  414444          170             ,ACI6   *ADDROT*
013455  625764
013456  742000          180             742000
           013457       190             OP      <   ALS>
           013462       200             OP      <  ALSS>
           013465       210             OP      <   BEG>
           013470       220             OP      <   CAF>
           013473       230             OP      <   CCK>
           013476       240             OP      <   CDF>
           013501       250             OP      <   CLA>
           013504       260             OP      <   CLC>
           013507       270             OP      <   CLL>
           013512       280             OP      <  CLOF>
           013515       290             OP      <  CLON>
           013520       300             OP      <   CLQ>
013523  435461          310             ,ACI6   *CLQLLS*
013524  545463
013525  650600          320             CLQILLS
           013526       330             OP      <  CLSF>
           013531       340             OP      <   CMA>
           013534       350             OP      <   CML>
           013537       360             OP      <   CMQ>
           013542       370             OP      <   CON>
           013545       380             OP      <   CPB>
           013550       390             OP      <   DBK>
           013553       400             OP      <   DBR>
           013556       410             OP      <   DCB>
           013561       420             OP      <  DGHB>
           013564       430             OP      <  DGSB>
           013567       440             OP      <   DIV>
           013572       450             OP      <  DIVS>
           013575       460             OP      <  DLAH>
           013600       470             OP      <  DLAL>
           013603       480             OP      <  DLBR>
           013606       490             OP      <  DLOK>
           013611       500             OP      <   DLP>
           013614       510             OP      <   DOV>
           013617       520             OP      <  DPBS>
           013622       530             OP      <  DPCF>
           013625       540             OP      <  DPEP>
           013630       550             OP      <  DPMK>
```

D MTSS DEBUGGER -- PERMANENT SYMBOL TABLE

```
013633      560          OP      <  DPOF>
013636      570          OP      <  DPON>
013641      580          OP      <  DPOP>
013644      590          OP      <  DPQT>
013647      600          OP      <  DPRC>
013652      610          OP      <  DPRS>
013655      620          OP      <  DPSF>
013660      630          OP      <  DPWC>
013663      640          OP      <  DRAH>
013666      650          OP      <  DRAL>
013671      660          OP      <  DRBR>
013674      670          OP      <  DSCC>
013677      680          OP      <  DSCD>
013702      690          OP      <  DSCF>
013705      700          OP      <  DSCN>
013710      710          OP      <  DSFX>
013713      720          OP      <  DSRS>
013716      730          OP      <  DSSC>
013721      740          OP      <  DSSF>
013724      750          OP      <  DTCA>
013727      760          OP      <  DTDF>
013732      770          OP      <  DTEF>
013735      780          OP      <  DTLA>
013740      790          OP      <  DTRA>
013743      800          OP      <  DTRB>
013746      810          OP      <  DTXA>
013751      820          OP      <EAECLA>
013754      830          OP      <   ECR>
013757      840          OP      <   ECS>
013762      850          OP      <   EEM>
013765      860          OP      <   EIS>
013770      870          OP      <   ELP>
013773      880          OP      <   EOV>
013776      890          OP      <   ESS>
014001      900          OP      < FRDIV>
014004      910          OP      <FRDIVS>
014007      920          OP      <   GLK>
014012      930          OP      <   GSM>
014015      940          OP      <   HLT>
014020      950          OP      <  IDIV>
014023      960          OP      < IDIVS>
014026      970          OP      <   IDX>
014031      980          OP      <   INX>
014034      990          OP      <   IOF>
014037      1000         OP      <   ION>
014042      1010         OP      <  IORS>
014045      1020         OP      <   ISA>
014050      1030         OP      <   KRB>
014053      1040         OP      <KRBLT1>
014056      1050         OP      <KRBLT2>
014061      1060         OP      <   KSF>
014064      1070         OP      <KSFLT1>
```

D MTSS DEBUGGER -- PERMANENT SYMBOL TABLE

```
014067      1080          OP      <KSFLT2>
014072      1090          OP      <  LACQ>
014075      1100          OP      <  LACS>
014100      1110          OP      <   LAS>
014103      1120          OP      <   LAW>
014106      1130          OP      <   LBL>
014111      1140          OP      <   LCK>
014114      1150          OP      <   LDA>
014117      1160          OP      <   LDB>
014122      1170          OP      <   LDS>
014125      1180          OP      <   LEM>
014130      1190          OP      <   LLS>
014133      1200          OP      <  LLSC>
014136      1210          OP      <  LLSS>
014141      1220          OP      <   LMQ>
014144      1230          OP      <   LPB>
014147      1240          OP      <   LPM>
014152      1250          OP      <   LRS>
014155      1260          OP      <LRSCLA>
014160      1270          OP      <  LRSS>
014163      1280          OP      <    LX>
014166      1290          OP      <    LY>
014171      1300          OP      < MPCNE>
014174      1310          OP      <  MPEU>
014177      1320          OP      <  MPCV>
014202      1330          OP      <  MPEV>
014205      1340          OP      <  MPLD>
014210      1350          OP      < MPOFF>
                          .PMC    SAVE,ON
014212  705000            $PECIAL+0                TURN OFF MEMORY PROTECT
014213      1360          OP      <  MPSK>
014216      1370          OP      < MPSNE>
014281  000055  1380      .ACI6   *  MQOR*
014282  615762
014283  642000  1390      642000
014284      1400          OP      <   MUL>
014287      1410          OP      <  MULS>
014292      1420          OP      <   NOP>
014295      1430          OP      <  NORM>
014240      1440          OP      < NORMS>
014243      1450          OP      <   OAS>
014246      1460          OP      <   OCK>
014251      1470          OP      <   OMQ>
014254      1480          OP      <   OPB>
014257      1490          OP      <   OSC>
014262      1500          OP      <   PCF>
014265      1510          OP      <   PLS>
014270      1520          OP      <   PSA>
014273      1530          OP      <   PSB>
014276      1540          OP      <   PSF>
014301      1550          OP      <  RAEF>
014304      1560          OP      <   RAL>
```

D MTSS DEBUGGER -- PERMANENT SYMBOL TABLE

```
014307      1570            OP      <   RAR>
014312      1580            OP      <   RCF>
014315      1590            OP      <   RCL>
014320      1600            OP      <   RCR>
014323      1610            OP      <   RET>
014326      1620            OP      <  RLPD>
014331      1630            OP      <   RPL>
014334      1640            OP      <  RLPE>
014337      1650            OP      <   RRB>
014342      1660            OP      <   RSA>
014345      1670            OP      <   RSB>
014350      1680            OP      <   RSF>
014353      1690            OP      <   RTL>
014356      1700            OP      <   RTR>
014361      1710            OP      <   SCK>
014364      1720            OP      <   SEM>
014367      1730            OP      <   SKP>
014372      1740            OP      <  SKP7>
014375      1750            OP      <   SMA>
014400      1760            OP      <   SNA>
014403      1770            OP      <   SNL>
014406      1780            OP      <   SPA>
014411      1790            OP      <   SPB>
014414      1800            OP      <   SPI>
014417      1810            OP      <   STL>
014422      1820            OP      <   SZA>
014425      1830            OP      <   SZL>
014430      1840            OP      <   TCF>
014433      1850            OP      <TCFLT1>
014436      1860            OP      <TCFLT2>
014441      1870            OP      <   TLS>
014444      1880            OP      <TLSLT1>
014447      1890            OP      <TLSLT2>
014452      1900            OP      <   TSF>
014455      1910            OP      <TSFLT1>
014460      1920            OP      <TSFLT2>
014463      1930            OP      <   TTS>
014466      1940            OP      <   WBL>
014471      1950            OP      <  WCGA>
014474      1960            OP      <   WDA>
014477      1970            OP      <  WDBC>
014502      1980            OP      <  WDBS>
014505      1990            OP      <   CAL>
014510      2000            OP      <   DAC>
014513      2010            OP      <   JMS>
014516      2020            OP      <   DZM>
014521      2030            OP      <   LAC>
014524      2040            OP      <   XOR>
014527      2050            OP      <   ADD>
014532      2060            OP      <   TAD>
014535      2070            OP      <   XCT>
014540      2080            OP      <   ISZ>
```

D MTSS DEGUGGER -- PERMANENT SYMBOL TABLE

```
014543          2090            OP      <   AND>
014546          2100            OP      <   SAD>
014551          2110            OP      <   JMP>
014554          2120            OP      <   EAE>
014557          2130            OP      <   IOT>
014562          2140            OP      <   OPR>
014565  777777  2150    SYM1    -1                      NUMBER OF SYMBOLS IN THE USER SYMBOL TABLE -1
                2160            .CRSM   RESTORE
                2170            .END
                10000           .PMC    RESTORE
003326          10010           .USE    IMPURE
                10020           .HEAD
000001          10030   DEBUG   .EQU    1               TURN ON THE HANDLER INSERTS LISTINGS
                10040           .HEAD   D
                10050           .INSRT  :DLIBRARY:PDP9LIB:TTYNON
                100             .INE    SDEBUG,1
                120             .IFE    SDEBUG,1
```

D MTSS-PDP9 NON-INTERRUPTING TELETYPE HANDLER

```
130                 .STITL  MTSS-PDP9 NON-INTERRUPTING TELETYPE HANDLER
140                 .HEAD   T
150       *
160       *
170       *         PROGRAMMED BY ROBERT W. BLEAN
180       *
190       *
200       *         LAST REVISED 24 MARCH 1972
210       *
220       *
230       *         THIS HANDLER PERMITS NON-INTERRUPT DRIVEN INPUT FROM AND OUTPUT
240       *         TO THE CONSOLE TELETYPE ON THE PDP-9 COMPUTER.
250       *
260       *         THIS HANDLER ALTERS THE  AC, AND MQ, IT DOES NOT ALTER ANY CORE
270       *         MEMORY OUTSIDE OF ITSELF. IN PARTICULAR IT DOES NOT ALTER ANY AUTO-INDEX REGISTER.
280       *
290       *         DATA FORMATS:
300       *
310       *              1) OCTAL
320       *
330       *              2) SIXBIT -- SIXBIT IS 8-BIT ASCII MINUS 240. THIS MAPS THE PRINTING
340       *                 CHARACTERS ONTO THE SET 0-77. ASCII VALUE 333 ([) IS USED FOR
350       *                 CARRIAGE RETURN AND 335 (]) IS USED FOR LINEFEED. NOTE THAT NEITHER
360       *                 333, 335, NOR ANY CONTROL CHARACTERS CAN BE RECOGNIZED IN SIXBIT.
370       *
380       *              3) ASCII -- ONE ASCII CHARACTER IS STORED PER WORD, LINE INPUT
390       *                 IS STORED IN THIS FORMAT, SINCE THERE IS ONLY ONE LINE-BUFFER
400       *                 THE EXTRA BUFFER LENGTH WASTES LESS SPACE THAN WOULD THE HANDLING
410       *                 ROUTINES NECESSARY FOR OTHER FORMS OF PACKING CHARACTERS.
```

T (MTSS TELETYPE HANDLER) STORAGE AREA

```
                    420             .STITL  (MTSS TELETYPE HANDLER) STORAGE AREA
                    430             .IFE    PURCOD,1
     003326         440             .USE    IMPURE
                    450
                    460
     003326         470     WORDB   .BLOCK  2               ROOM TO ACCUMULATE TWO VALID WORDS
     000120         480     STD     .EQU    80.             STANDARD IS AN 80-CHARACTER LINE BUFFER
     003330         490     BUFFR   .BLOCK  STD
                    500     *
                    510     *
                    520     *       VARIABLES
                    530     *
 003450  003447     540     BEND    .-1                     END OF THE CHARACTER BUFFER
 003451  000000     550     BPTR    .DSA                    POINTER TO CURRENTLY ACTIVE WORD IN LINE BUFFER
 003452  000000     560     T1      .DSA                    TEMPORARY VARIABLE
 003453  000000     570     T2      .DSA                    TEMPORARY VARIABLE
 003454  000000     580     CHAR    .DSA                    STORES LATEST CHARACTER FROM FGET
 003455  000000     590     DLMTR   .DSA                    STORES LATEST DELIMITER THROUGH CHRID
 003456  000000     600     COUNT   .DSA
                    610             .IFE    PURCOD,1
     010335         620             .USE    PURE
```

T (MTSS TELETYPE HANDLER) LINE BUFFER INPUT

```
                    630             .STITL  (MTSS TELETYPE HANDLER) LINE BUFFER INPUT
                    640
                    650
                    660     *
                    670     *       THE PROGRAM IS PROTECTED AGAINST OVERFLOW OR UNDERFLOW OF THE LINE
                    680     *       BUFFER. UNDERFLOW (EXCESS DELETIONS) IS IGNORED, AND OVERFLOW CHARACTERS
                    690     *       ARE LOST, EXCEPT FOR THE LAST CHARACTER TYPED.
                    700     *
                    710
    010335          720             ENTER   INLIN           SUBROUTINE TO READ IN AND BUFFER A LINE FROM THE TELETYPE
                                    .PMC    SAVE,ON
    003457                  INLIN   ...
010335  700312      730             KRB                     ONCE, ON ENTRANCE, CLEAN UP ANY PRIOR INPUT
010336  213410      740     INL     LAC     (BUFFR-1)       LOAD A POINTER TO START OF THE BUFFER MINUS ONE
010337  043451      750             DAC     BPTR            INITIALIZE THE BUFFER POINTER
010340  143456      760             DZM     COUNT           INITIALIZE THE WORD FETCHED COUNT
010341  143455      770             DZM     DLMTR           INITIALIZE THE LAST DELIMITER STORAGE
010342  700313      780     IN1     KSF!KRB                 GET THE NEXT INPUT CHARACTER
010343  610342      790             JMP     .-1
010344  553411      800             SAD     ($BKARR)
010345  610367      810             JMP     1CHAR           DELETE ONE CHARACTER IF IT WAS A BACKARROW
010346  553412      820             SAD     ($CONTX)
010347  610365      830             JMP     1LINE           DELETE THE ENTIRE LINE IF IT WAS A CONTROL X
010350  652000      840     IN4     LMQ                     SAVE THE CHARACTER
010351  203451      850             LAC     BPTR            LOAD THE CURRENT BUFFER POINTER
010352  543450      860             SAD     BEND            SKIP IF NO OVERFLOW
010353  741000      870             SKP                     AVOID DAMAGE DUE TO OVERFLOW
010354  443451      880             ISZ     BPTR            ADVANCE THE POINTER -- IT IS STILL WITHIN THE BUFFER
010355  641002      890             LACQ                    RELOAD THE CHARACTER
010356  063451      900             DAC     BPTR,X          AND PUT IT IN THE BUFFER
010357  553311      910             SAD     ($CR)
010360  741000      920             SKP                     EXIT WHEN A CARRIAGE RETURN IS FOUND
010361  610342      930             JMP     IN1             ELSE GET THE NEXT CHARACTER
010362  763327      940             LAW     BUFFR-1
010363  043451      950             DAC     BPTR            RESET THE BUFFER POINTER AT THE END OF THE LINE
010364  623457      960             JMP     INLIN,X         AND RETURN TO THE CALLER
                    970
010365  103505      980     1LINE   JMS     CRLF            PRINT THE RESPONSE TO A LINE-DELETE
010366  610336      990             JMP     INL             REREAD THE LINE
010367  203451     1000     1CHAR   LAC     BPTR            LOAD THE BUFFER POINTER
010370  550336     1010             SAD     INL             SKIP IF NO UNDERFLOW
010371  610342     1020             JMP     IN1             ELSE IGNORE THE COMMAND
010372  353332     1030             TAD     (-1)            DECREMENT THE BUFFER POINTER
010373  043451     1040             DAC     BPTR            AND SAVE IT
010374  610342     1050             JMP     IN1             GET THE NEXT CHARACTER
```

```
T                       (MTSS TELETYPE HANDLER) OCTAL WORD INPUT/OUTPUT

                1060            .STITL  (MTSS TELETYPE HANDLER) OCTAL WORD INPUT/OUTPUT
                1070
                1080    *
                1090    *       OPERATION RETURN L   AC     MQ   MEANING
                1100    *       --------- ------ - ------ ------ ------------------------------
                1110    *       INPUT       +1   0   X      X    FORMAT ERROR DISCOVERED
                1120    *                   +1   1 DELIM    X    FIRST NON-BLANK CHARACTER IS A DELIMITER
                1130    *                   +2   1 OCTAL  DELIM  SUCCESSFUL READ OF AN OCTAL NUMBER
                1140    *       OUTPUT      +1   X   X      X    SUCCESSFUL WRITE OF AN OCTAL NUMBER
                1150    *
                1160
    010375      1170            ENTER   NUMIN
                                .PMC    SAVE,ON
    003461              NUMIN   ...
010375  143453  1180            DZM     T2              INITIALIZE THE DECIMAL-DIGIT-RECEIVED FLAG
010376  103477  1190            JMS     INTIN           INITIALIZE THE INPUT STRING, ETC
010377  623461  1200            JMP     NUMIN,X         RETURN +1 FOR DELIMITER IS FIRST NON-BLANK CHARACTER
010400  103475  1210    NUM20   JMS     FGET            GET THE NEXT CHARACTER
010401  103501  1220            JMS     CHRID           IDENTIFY IT
010402  610424  1230            JMP     NUM26           IT IS A DELIMITER, SO EXIT
010403  623461  1240            JMP     NUMIN,X         IT IS A LETTER, SO EXIT +1 FOR A FORMAT ERROR
010404  741400  1250            SZL                     SKIP IF THE CHARACTER IS AN OCTAL DIGIT
010405  443453  1260            ISZ     T2              ELSE BE SURE THE DECIMAL-DIGIT-RECEIVED FLAG IS SET
010406  513413  1270            AND     (17)            RETAIN JUST THE DIGIT
010407  043452  1280            DAC     T1              AND SAVE IT FOR DECIMAL ACCUMULATION
                1290
010410  640503  1300            LRS     3               SAVE THE "OCTAL DIGIT"
010411  203326  1310            LAC     WORDB           LOAD THE PREVIOUSLY GATHERED "OCTAL NUMBER"
010412  640603  1320            LLS     3               CONCATENATE THE "OCTAL DIGITS"
010413  043326  1330            DAC     WORDB           AND SAVE THE RESULT
                1340
010414  203327  1350            LAC     WORDB+1         LOAD THE PREVIOUSLY GATHERED "DECIMAL NUMBER"
010415  744000  1360            CLL                     SET THE LINK FOR THE MULTIPLY
010416  653122  1370            MUL                     MULTIPLY THE PREVIOUS "DECIMAL VALUE"
010417  000012  1380            10,                     BY 10 FOR DECIMAL
010420  641002  1390            LACQ                    LOAD THE RESULT
010421  343452  1400            TAD     T1              ADD THE CURRENT "DECIMAL DIGIT"
010422  043327  1410            DAC     WORDB+1         AND SAVE THE TOTAL "DECIMAL NUMBER"
                1420
010423  610400  1430            JMP     NUM20           LOOP
                1440
                1450
010424  553404  1460    NUM26   SAD     ($POINT)        CHECK FOR A PERIOD
010425  610433  1470            JMP     NUM27           IF SO, PICK UP THE DECIMAL VALUE
010426  203453  1480            LAC     T2              ELSE LOAD THE DECIMAL-DIGITS-RECEIVED FLAG
010427  744200  1490            SZA!CLL                 AND SKIP IF THERE WERE NONE
010430  623461  1500            JMP     NUMIN,X         RETURN +1, LK=0 FOR A FORMAT ERROR: DECIMAL DIGITS, BUT NO PERIOD
010431  203326  1510            LAC     WORDB           LOAD THE OCTAL VALUE
010432  610442  1520            JMP     NUM29
010433  103475  1530    NUM27   JMS     FGET            GET THE NEXT CHARACTER
010434  103501  1540            JMS     CHRID           AND IDENTIFY IT
010435  610441  1550            JMP     NUM28           A DELIMITER IS LEGAL, SO EXIT
```

T (MTSS TELETYPE HANDLER) OCTAL WORD INPUT/OUTPUT

```
010436  623461  1560            JMP     NUMIN,X         A LETTER -- EXIT +1 FOR A FORMAT ERROR
010437  744000  1570            CLL                     A NUMBER -- CLEAR THE LINK FOR A FORMAT ERROR
010440  623461  1580            JMP     NUMIN,X         AND EXIT +1
010441  203327  1590    NUM28   LAC     WORDB+1         LOAD THE DECIMAL VALUE
010442  043326  1600    NUM29   DAC     WORDB           SAVE THE CORRECT VALUE
010443  443461  1610            ISZ     NUMIN           BUMP TO A RETURN +2 FOR SUCCESSFUL
010444  623461  1620            JMP     NUMIN,X
                1630
                1640
                1650
    010445      1660            ENTER   OCTOT
                                .PMC    SAVE,ON
    003463                      OCTOT   ...
010445  652000  1670    OCT42   LMQ                     SET THE VALUE TO BE OUTPUT
010446  741400  1680            SZL                     SKIP IF NO LEADING ZEROES ARE TO BE SUPPRESSED
010447  750201  1690            SZA!CLC                 SET A FLAG TO PRINT ONE CHARACTER, ANYWAY, IF THE AC IS ZERO
010450  777772  1700            LAW     -6              ELSE SET THE COUNT FOR THE STANDARD SIX CHARACTERS
010451  043452  1710            DAC     T1              SET THE NUMBER OF CHARACTERS TO BE OUTPUT
010452  641002  1720            LACQ                    RELOAD THE USER'S VALUE
010453  741200  1730            SNA                     SKIP FOR A NON-ZERO VALUE
010454  744000  1740            CLL                     ELSE FORCE A SINGLE ZERO TO PRINT
010455  641603  1750    OCT44   LLSC    3,              GET THE NEXT OCTAL DIGIT
010456  740200  1760            SZA                     IF IT IS ZERO, DON'T CHANGE PRINT-SUPPRESSION STATE
010457  744000  1770            CLL                     ELSE CLEAR THE PRINT INHIBIT AT THE FIRST NON-ZERO FOUND
010460  353305  1780            TAD     (260)           MAKE ASCII IN ANY CASE
010461  740400  1790            SNL                     BUT SKIP IF PRINT IS INHIBITED
010462  103503  1800            JMS     TTYOT           ELSE PRINT THE DIGIT
010463  443452  1810            ISZ     T1              DONE???
010464  610455  1820            JMP     OCT44           NO -- LOOP
010465  700401  1830            TSF
010466  610465  1840            JMP     .-1             WAIT FOR THE TELETYPE TO SETTLE
010467  623463  1850            JMP     OCTOT,X         YES -- EXIT
```

```
                T                       (MTSS TELETYPE HANDLER) SIXBIT WORD INPUT & SIXBIT BUFFER OUTPUT

                        1860            .STITL  (MTSS TELETYPE HANDLER) SIXBIT WORD INPUT & SIXBIT BUFFER OUTPUT
                        1870
                        1880
                        1890    *
                        1900    *       OPERATION RETURN L    AC      MQ    MEANING
                        1910    *       --------- ------ - ------ ------ ------------------------------
                        1920    *       INPUT       +1   L DELIM    X     FIRST NON-BLANK CHARACTER IS A DELIMITER
                        1930    *                   +2   L SIXBIT DELIM   SUCCESSFUL READ OF A SIXBIT WORD
                        1940    *       OUTPUT      +1   X   X      X     SUCCESSFUL WRITE OF A SIXBIT BUFFER
                        1950    *
                        1960
                010470  1970            ENTER   SIXIN
                                        .PMC    SAVE,ON
                003465                  SIXIN   ...
010470  763326  1980                    LAW     WORDB
010471  043452  1990                    DAC     T1              INITIALIZE THE SIXBIT BUFFER POINTER
010472  103477  2000                    JMS     INTIN           INITIALIZE THE INPUT
010473  623465  2010                    JMP     SIXIN,X         RETURN +1 FOR DELIMITER IS FIRST NON-BLANK CHARACTER
010474  443465  2020                    ISZ     SIXIN           ELSE BUMP TO A GOOD RETURN
010475  103467  2030    SIX2            JMS     SIX5            GET THE FIRST GOOD CHARACTER
010476  660714  2040                    ALSS    12.             AND PUT IT IN THE FIRST CHARACTER POSITION
010477  063452  2050                    DAC     T1,X            AND SAVE IT
010500  103467  2060                    JMS     SIX5            GET THE SECOND CHARACTER
010501  660706  2070                    ALSS    6.              PUT IT IN THE SECOND CHARACTER POSITION
010502  263452  2080                    XOR     T1,X            CONCATENATE THE CHARACTERS
010503  063452  2090                    DAC     T1,X            AND SAVE THE RESULT
010504  103467  2100                    JMS     SIX5            GET THE THIRD CHARACTER
010505  263452  2110                    XOR     T1,X            CONCATENATE THE CHARACTERS
010506  063452  2120                    DAC     T1,X            AND SAVE THE RESULT
010507  443452  2130                    ISZ     T1              BUMP THE STORAGE BUFFER POINTER
010510  610475  2140                    JMP     SIX2            LOOP
                        2150
010511  203326  2160    SIX9            LAC     WORDB           LOAD THE FIRST SIXBIT WORD
010512  623465  2170                    JMP     SIXIN,X         EXIT
                        2180
                010513  2190            ENTER   SIX5            SUBROUTINE TO GET THE NEXT CHARACTER, MAKE IT SIXBIT, EXIT IF A DELIMITER
                                        .PMC    SAVE,ON
                003467                  SIX5    ...
010513  103475  2200                    JMS     FGET            GET THE NEXT CHARACTER
010514  103501  2210                    JMS     CHRID           IDENTIFY IT
010515  610511  2220                    JMP     SIX9            EXIT IF IT IS A DELIMITER
010516  740000  2230                    NOP                     PERMIT LETTERS
010517  353403  2240                    TAD     (-240)          MAKE SIXBIT
010520  623467  2250                    JMP     SIX5,X
                        2260
                        2270
                010521  2280            ENTER   SIXOT
                                        .PMC    SAVE,ON
                003471                  SIXOT   ...
010521  043452  2290                    DAC     T1              SET THE NEGATIVE CHARACTER COUNT
010522  223471  2300    SIX24           LAC     SIXOT,X         LOAD THE NEXT WORD OF OUTPUT
010523  652000  2310                    LMQ                     SAVE IT FOR PRINTING
```

T (MTSS TELETYPE HANDLER) SIXBIT WORD INPUT & SIXBIT BUFFER OUTPUT

```
010524  443471  2320            ISZ     SIXOT           BUMP THE POINTER
010525  103473  2330            JMS     SIX26           OUTPUT THE FIRST CHARACTER
010526  103473  2340            JMS     SIX26           OUTPUT THE SECOND CHARACTER
010527  103473  2350            JMS     SIX26           OUTPUT THE THIRD CHARACTER
010530  610522  2360            JMP     SIX24           LOOP
                2370
    010531      2380            ENTER   SIX26
                                ,PMC    SAVE,ON
    003473              SIX26   ...
010531  641606  2390            LLSC    6.              GET THE NEXT SIXBIT CHARACTER
010532  353350  2400            TAD     (240)           MAKE IT ASCII
010533  553414  2410            SAD     (333)           CHECK FOR CARRIAGE RETURN MAPPING
010534  760215  2420            LAW     SCR
010535  553415  2430            SAD     (335)           CHECK FOR LINE FEED MAPPING
010536  760212  2440            LAW     SLF
010537  103503  2450            JMS     TTYOT           PRINT THE CHARACTER
010540  443452  2460            ISZ     T1              ALL CHARACTERS PRINTED?
010541  623473  2470            JMP     SIX26,X         NO -- LOOP
010542  700401  2480            TSF
010543  610542  2490            JMP     .-1             WAIT FOR THE TELETYPE TO SETTLE
010544  623471  2500            JMP     SIXOT,X         YES -- EXIT
                2510    *
                2520    *
```

T (MTSS TELETYPE HANDLER) MISCELLANEOUS LINE BUFFER ROUTINES

```
                    2530            .STITL  (MTSS TELETYPE HANDLER) MISCELLANEOUS LINE BUFFER ROUTINES
                    2540
                    2550
                    2560
                    2570
    010545          2580            ENTER   FGET            SUBROUTINE TO GET THE FIRST REMAINING CHARACTER FROM THE LINE BUFFER
                                    .PMC    SAVE,ON
    003475                  FGET    ...
010545  443451      2590            ISZ     BPTR            NO -- BUMP THE POINTER
010546  223451      2600            LAC     BPTR,X          LOAD THE NEXT CHARACTER
010547  043454      2610            DAC     CHAR            AND SAVE IT
010550  623475      2620    FGET9   JMP     FGET,X
                    2630
    010551          2640            ENTER   INTIN           INITIALIZE INPUT WORD-GETTING
                                    .PMC    SAVE,ON
    003477                  INTIN   ...
010551  443456      2650            ISZ     COUNT           COUNT THE WORD, SUCCESSFUL OR NOT
010552  143326      2660            DZM     WORDB           INITIALIZE THE TWO FIRST WORDS OF THE INPUT BUFFER
010553  143327      2670            DZM     WORDB+1
010554  103475      2680            JMS     FGET            GET THE NEXT CHARACTER
010555  553350      2690            SAD     ($SPACE)        CHECK IT FOR A SPACE
010556  610554      2700            JMP     .-2             THROW AWAY SPACES
010557  103501      2710            JMS     CHRID           IDENTIFY THE NON-SPACE
010560  623477      2720            JMP     INTIN,X         RETURN +1 FOR A DELIMITER
010561  740000      2730            NOP
010562  443477      2740            ISZ     INTIN           ELSE BUMP THE RETURN FOR A NUMBER OR A LETTER
010563  750001      2750            CLC
010564  343451      2760            TAD     BPTR            BACK UP THE POINTER TO POINT TO THE FIRST GOOD CHARACTER
010565  043451      2770            DAC     BPTR
010566  623477      2780            JMP     INTIN,X
```

```
T                                   (MTSS TELETYPE HANDLER) MISCELLANEOUS CHARACTER-HANDLING SUBROUTINES

                    2790            ,STITL  (MTSS TELETYPE HANDLER) MISCELLANEOUS CHARACTER-HANDLING SUBROUTINES
                    2800    *
                    2810    *
                    2820    *       CHRID -- SUBROUTINE TO CLASSIFY EIGHT-BIT ASCII CHARACTERS.
                    2830    *       ENTER WITH THE CHARACTER IN THE AC; LEAVE WITH THE EIGHT-BIT CHARACTER
                    2840    *       IN AC(0-17) AND THE LINK AS FOLLOWS:
                    2850    *
                    2860    *       RETURN LINK MEANING
                    2870    *       ------ ---- ----------------------------------------------------
                    2880    *         +1    1   THE CHARACTER IS A DELIMITER (I.E. NEITHER A DIGIT NOR A LETTER
                    2890    *         +2    0   THE CHARACTER IS EITHER AN UPPER CASE OR A LOWER CASE LETTER
                    2900    *         +3    0   THE CHARACTER IS AN OCTAL DIGIT
                    2910    *         +3    1   THE CHARACTER IS A DECIMAL DIGIT (8 OR 9)
                    2920    *
    010567          2930            ENTER   CHRID
                                    ,PMC    SAVE,ON
    003501                  CHRID   ...
010567  513353      2940            AND     (377)
010570  043503      2950            DAC     TTYOT           SAVE THE EIGHT-BIT ASCII CHARACTER
010571  353416      2960            TAD     (-260)          AC < 0 FOR DELIMITERS
010572  745102      2970            SPA!STL
010573  610611      2980            JMP     DLMR            CHARACTER IS A DELIMITER
010574  353417      2990            TAD     (-10)           AC < 0 FOR OCTAL DIGITS
010575  745100      3000            SPA!CLL
010576  610614      3010            JMP     DIGIT           CHARACTER IS AN OCTAL DIGIT
010577  353420      3020            TAD     (-2)            AC < 0 FOR DECIMAL DIGITS
010600  745102      3030            SPA!STL
010601  610614      3040            JMP     DIGIT           CHARACTER IS A DECIMAL DIGIT
010602  353421      3050            TAD     (-6)            AC <= 0 FOR DELIMITERS
010603  745302      3060            SNA!SPA!STL
010604  610611      3070            JMP     DLMR            CHARACTER IS A DELIMITER
010605  513422      3080            AND     (777737)        MAP LOWER CASE INTO UPPER CASE
010606  353423      3090            TAD     (-33)           AC < 0 FOR LETTERS -- L=1 FOR LETTERS; L=0 FOR DELIMITERS
010607  741102      3100            SPA!CML
010610  610615      3110            JMP     LETTR           THE CHARACTER IS A LETTER
                    3120
010611  203503      3130    DLMR    LAC     TTYOT           LOAD THE DELIMITER
010612  043455      3140            DAC     DLMTR           SAVE IT
010613  623501      3150            JMP     CHRID,X
                    3160
010614  443501      3170    DIGIT   ISZ     CHRID
010615  443501      3180    LETTR   ISZ     CHRID
010616  203503      3190            LAC     TTYOT           RELOAD THE CHARACTER
010617  623501      3200            JMP     CHRID,X
                    3210
                    3220
                    3230
    010620          3240            ENTER   TTYOT
                                    ,PMC    SAVE,ON
    003503                  TTYOT   ...
010620  700401      3250            TSF
010621  610620      3260            JMP     .-1             WAIT FOR THE TELEPRINTER TO BE FREE
```

T (MTSS TELETYPE HANDLER) MISCELLANEOUS CHARACTER-HANDLING SUBROUTINES

```
010622  700301  3270            KSF                     KILL-THE-OUTPUT FEATURE
010623  700406  3280            TLS                     PRINT THE CHARACTER IN THE AC
010624  623503  3290            JMP     TTYOT,X
                3300
                3310
    010625      3320            ENTER   CRLF
                                .PMC    SAVE,ON
    003505              CRLF    ...
010625  760215  3330            LAW     215
010626  103503  3340            JMS     TTYOT
010627  760215  3350            LAW     215
010630  103503  3360            JMS     TTYOT
010631  760212  3370            LAW     212
010632  103503  3380            JMS     TTYOT
010633  700401  3390            TSF
010634  610633  3400            JMP     .-1             WAIT FOR THE TTY TO SETTLE
010635  623505  3410            JMP     CRLF,X
                3420
                3430
                3440            .HEAD                   TURN OFF THE INSERT'S HEAD SYMBOL
                3450            .LIST   ON
                3460            .END
    004012      10060   NEXTL   .EQU    $MONSYM
                10070           .HEAD   M               GET BACK UNDER THE MAIN HEAD SYMBOL
    002170      10080   CATLOG  .EQU    BUFFER          REDEFINE THE CATALOG BUFFER
    003314      10090   FORCE   .EQU    DSFORCE
                10100           .INSRT  MTSSCAT
```

```
M                       DESCRIPTION OF THE GROWTH SYSTEM CATALOG STRUCTURE

100             ,STITL  DESCRIPTION OF THE GROWTH SYSTEM CATALOG STRUCTURE
110             ,HEAD   C
120
130
140     *
150     *       MAJOR REVISION -- JAN 21, 1972 BY ROBERT W. BLEAN
160     *
170     *       A GROWTH CATALOG FOR A FILE-ORIENTED DEVICE IS LOCATED IN THE 400 WORDS
180     *       OF LOGICAL BLOCK 1 OF THE LOGICAL DEVICE. THIS PERMITS DISK AND DECTAPE
190     *       TO BE USED INTERCHANGEABLY BY THE GROWTH SYSTEM PROGRAMS.
200     *
210     *       THE DEVICE ADDRESS OF A HANDLER IS THE HANDLER NUMBER IN BITS 0-2
220     *       AND THE TYPE (DISK (1) OR DECTAPE (0)) IN BIT 3.
230     *
240     *       THE DEVICE ADDRESS OF A FILE IS THE DEVICE ADDRESS OF THE HANDLER IT
250     *       IS ON PLUS IN BITS 8-17 ITS STARTING BLOCK NUMBER.
260     *
270     *       ALL DEVICE ADDRESSES IN A DECTAPE CATALOG ARE CORRECT FOR THE HANDLER
280     *       THE TAPE WAS MOUNTED ON THE LAST TIME IT WAS ALTERED.
290     *
300     *       THE FIRST FOUR WORDS OF THE CATALOG BLOCK ARE A HEADER:
310     *            1) THE DEVICE ADDRESS OF THE FIRST FREE BLOCK ON THE DEVICE
320     *            2) POINTER TO THE FIRST FREE WORD IN THE CATALOG MINUS ONE PLUS THE CATALOG'S CORE ADDRESS
330     *            3) TWOS COMPLEMENT COUNT OF THE NUMBER OF FILES CATALOGED
340     *            4) TWOS COMPLEMENT MAXIMUM BLOCK NUMBER ON THE DEVICE
350     *
360     *       THE REMAINDER OF THE CATALOG CONSISTS OF A SERIES OF FIVE WORD FILE-
370     *       CONTROL BLOCKS, THE FIRST FILE CONTROL BLOCK IS FOR THE CATALOG ITSELF.
380     *       THEN THERE IS ONE FILE CONTROL BLOCK FOR EACH FILE ON THE DEVICE.
390     *
400     *       FORMAT OF THE FILE CONTROL BLOCKS:
410     *            1) THE FIRST WORD IS THE SIXBIT ASCII (EIGHTBIT ASCII - 240)
420     *               FILENAME. THIS MEANS THE FILENAME IS RESTRICTED TO THREE
430     *               CHARACTERS, WITH NO EXTENSION OR PASSWORD.
440     *            2) THE DEVICE ADDRESS OF THE FILE.
450     *            3) THE FILE'S CORE ADDRESS
460     *            4) THE FILE'S LENGTH (IN WORDS)
470     *            5) THE PROGRAM START
480     *
490     *       THIS LEAVES TWO WORDS OF THE CATALOG BLOCK UNUSED. IT IS SUGGESTED THAT
500     *       THE SECOND OF THESE CONTAIN THE BLOCK NUMBER OF A CONTINUATION OF THE
510     *       CATALOG, SHOULD THIS EVER BE NECESSARY; IT WOULD BE ZERO IF THERE
520     *       IS NO CONTINUED CATALOG BLOCK.
```

```
C                                   GROWTH SYSTEM STANDARD CATALOG ROUTINES

                    530             ,STITL  GROWTH SYSTEM STANDARD CATALOG ROUTINES
    003507          540             ,USE    IMPURE
                    550
003507  000000      560     CTEM1   ,DSA                    CATALOG ROUTINE'S PRIVATE TEMP
    010636          570             ,USE    PURE
                    580
                    590
                    600     *
                    610     *       RCAT -- THE BASIC CATALOG ROUTINE, IT READS IN CATALOGS AND UPDATES THEM
                    620     *       FOR THE CURRENT DEVICE AND (POSSIBLY NON-STANDARD) CATALOG LOCATION.
                    630     *
                    640     *
                    650     *       A HANDLER DEVICE ADDRESS IS PASSED IN THE AC. THE REQUESTED
                    660     *       CATALOG IS READ IN AND ALL OF THE DEVICE ADDRESSES ARE UPDATED.
                    670     *       AS A RESULT, THE CATALOG IN CORE ALWAYS HAS THE PROPER DEVICE ADDRESSES
                    680     *       FOR THE DEVICE IT WAS READ FROM.
                    690     *
                    700     *       RETURN IS +1 WHEN THE DESIRED CATALOG IS IN CORE.
                    710     *
                    720     *       IN THE EVENT OF UNRECOVERABLE ERROR, EXIT IS TO AN ERROR ROUTINE.
                    730     *
    010636          740             ENTER   RCAT
                                    ,PMC    SAVE,ON
    003510                  RCAT    ...
010636  043507      750             DAC     CTEM1           SAVE THE DEVICE ADDRESS OF THE DEVICE WHOSE CATALOG IS BEING REQUESTED
010637  103314      760             JMS     MSFORCE         FORCE THE OLD BUFFER BEFORE READING A NEW ONE
                    770
    010640          780     RCAT1   ...
010640  203507      790             LAC     CTEM1
010641  513301      800             AND     (DVCMSK)        GET THE NEW HANDLER'S DEVICE ADDRESS
010642  042153      810             DAC     DSBDA           SET THE HANDLER DEVICE ADDRESS
010643  513307      820             AND     (040000)        RECOVER JUST THE DISK/DECTAPE BIT
010644  740200      830             SZA                     SKIP FOR DECTAPE
010645  213330      840             LAC     ($SYSBAS)       ELSE LOAD THE DISK SYSTEM BASE ADDRESS
010646  242153      850             XOR     DSBDA           ADD THE BASE ADDRESS INTO THE HANDLER DEVICE ADDRESS
010647  253331      860             XOR     ($CATBLK)       ADD IN THE CATALOG BLOCK NUMBER
010650  042153      870             DAC     DSBDA           SAVE THE NEW CATALOG'S DEVICE ADDRESS
010651  762170      880             LAW     BUFFER          LOAD A POINTER TO THE BUFFER
010652  042154      890             DAC     DSBCA           SET IT AS THE CORE ADDRESS
010653  213424      900             LAC     (400)           LOAD THE LENGTH
010654  042155      910             DAC     DSBLEN          SET IT AS THE BUFFER LENGTH
010655  103512      920             JMS     CSRCOVR         SET UP THE ERROR RECOVERY
010656  770663      930             LAW     RCAT3
010657  652000      940             LMQ                     LOAD THE RESTART ADDRESS
010660  762153      950             LAW     DSBDA           GET A POINTER TO THE CATALOG PARAMETERS
010661  705003      960             PREAD                   READ IN THE NEW CATALOG
010662  610716      970             JMP     RCVR4           IN CASE OF ERROR
                    980     *
                    990     *       NOW UPDATE THE DEVICE ADDRESSES
                   1000     *
010663  203507     1010     RCAT3   LAC     CTEM1
010664  513301     1020             AND     (DVCMSK)
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
010665  043507  1030            DAC     CTEM1           SET THE CURRENT DEVICE ADDRESS
                1040
010666  202170  1050            LAC     CATLOG
010667  513336  1060            AND     (BLKMSK)
010670  243507  1070            XOR     CTEM1
010671  042170  1080            DAC     CATLOG          UPDATE THE OLD DEVICE ADDRESS OF THE FIRST FREE BLOCK
                1090
010672  762175  1100            LAW     CATLOG+5
010673  043314  1110            DAC     MSFORCE
010674  043515  1120            DAC     CATL            SET POINTERS TO THE FIRST OLD DEVICE ADDRESS
010675  202172  1130            LAC     CATLOG+2
010676  043512  1140            DAC     RCOVR           SET THE COUNT OF FCB'S
                1150
010677  223314  1160    RCAT4   LAC     MSFORCE,X       LOAD THE NEXT OLD DEVICE ADDRESS
010700  513336  1170            AND     (BLKMSK)        RECOVER THE BLOCK NUMBER
010701  243507  1180            XOR     CTEM1           ADD IN THE CURRENT HANDLER DEVICE ADDRESS
010702  063515  1190            DAC     CATL,X          SAVE THE UPDATED FILE DEVICE ADDRESS
                1200
010703  443512  1210            ISZ     RCOVR           COUNT THE FILES DONE
010704  741000  1220            SKP
010705  623510  1230            JMP     RCAT,X          ALL DONE
                1240
010706  203314  1250            LAC     MSFORCE         LOAD THE FCB POINTER
010707  353425  1260            TAD     (FCBLEN)        ADVANCE IT TO THE NEXT FCB
010710  043314  1270            DAC     MSFORCE
010711  043515  1280            DAC     CATL            SAVE THE NEW POINTER
010712  610677  1290            JMP     RCAT4           LOOP
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
                    1300            .EJECT
                    1310
                    1320
                    1330
    010713          1340            ENTER   RCOVR           SUBROUTINE TO SET UP RECOVERY FROM HARDWARE ERRORS
                                    .PMC    SAVE,ON
    003512                  RCOVR   ...
010713  777776      1350            LAW     -2              SET FOR FIVE RETRIES BEFORE GIVING UP
010714  043514      1360            DAC     ERCNT
010715  623512      1370            JMP     RCOVR,X
                    1380
    010716          1390    RCVR4   MESS    <DEVICE ERROR>,12,
010726  443514      1400            ISZ     ERCNT           COUNT THE ERROR
010727  623512      1410            JMP     RCOVR,X
    010730          1420    RCVR5   MESS    <TYPE 'IGNORE' OR 'CONTINUE'! >,29,
    010746          1430            LINE                    GET THE USER'S ANSWER TO WHAT HE WANTS TO DO ABOUT IT
    010747          1440            WORD                    READ HIS ANSWER
010750  610730      1450            JMP     RCVR5           NO INPUT IS ILLEGAL
010751  550756      1460            SAD     IGN
010752  604012      1470            JMP     $NEXTL          IGNORE THE LAST COMMAND
010753  550757      1480            SAD     CON
010754  603513      1490            JMP     RCOVR+1         SET UP TO TRY AGAIN
010755  610730      1500            JMP     RCVR5           ANY OTHER ANSWER IS ILLEGAL
    003514          1510            .USE    IMPURE
003514  000000      1520    ERCNT   .DSA
    010756          1530            .USE    PURE
010756  514756      1540    IGN     .ACI6   *IGN*
010757  435756      1550    CON     .ACI6   *CON*
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
                    1560            .EJECT
                    1570    *
                    1580    *       CATL
                    1590    *
                    1600    *       CATL SEARCHES THE CATALOG IN CORE FOR THE FILENAME
                    1610    *       PASSED IN THE AC
                    1620    *
                    1630    *       RETURN +2 WITH CATX POINTING TO THE FILE NAME IF SUCCESSFUL
                    1640    *
                    1650    *       RETURN +1 WITH CATX POINTING TO THE FIRST FREE SPACE -1 IN THE
                    1660    *       CATALOG IF THE FILE NAME IS NOT FOUND
                    1670    *
        010760      1680            ENTER   CATL
                                    .PMC    SAVE,ON
        003515              CATL    ...
    010760  043326  1690            DAC     TSWORDB         SAVE CATALOG NAME
                    1700    *
                    1710    *       FIRST CHECK WHETHER OR NOT THIS IS A SPECIAL FILE
                    1720    *
    010761  203531  1730            LAC     CDFLG           LOAD THE CORE/DISK SPECIAL FILE FLAG
    010762  741200  1740            SNA                     SKIP IF IT IS SET
    010763  611005  1750            JMP     CATL1           NO -- THEREFORE IT IS A NORMAL FILE
                    1760    *
                    1770    *       FIND OUT WHICH KIND OF SPECIAL FILE WE ARE TALKING ABOUT
                    1780    *
        010764      1790            WORD1
    010765  543517  1800            SAD     CORE
    010766  611024  1810            JMP     CORE1           IT IS THE USER CORE FILE
    010767  543523  1820            SAD     DISK
    010770  611036  1830            JMP     DISK1           IT IS THE USER DISK FILE
        010771      1840            MESS    <ILLEGAL SPECIAL FILE>*20.
    011004  605345  1850            JMP     MSNEXTL         GET THE NEXT COMMAND
                    1860    *
                    1870    *       NEXT CHECK FOR NORMAL FILES
                    1880    *
    011005  762173  1890    CATL1   LAW     CATLOG+3
    011006  040011  1900            DAC     SCATX           SET A POINTER TO THE FIRST FCB IN THE CATALOG AUTO-INDEX REGISTER
    011007  202172  1910            LAC     CATLOG+2        GET CATALOG COUNT
    011010  043507  1920            DAC     CTEM1           SAVE IT
        011011      1930    CATLL   WORD1                   RESTORE NAME TO SEARCH FOR
    011012  560011  1940            SAD     SCATX,X         CHECK IT
    011013  611022  1950            JMP     CATL9           FOUND IT
    011014  200011  1960            LAC     SCATX
    011015  353343  1970            TAD     (FCBLEN-1)      FAILED -- MOVE THE POINTER TO THE NEXT FILE CONTROL BLOCK
    011016  040011  1980            DAC     SCATX
    011017  443507  1990            ISZ     CTEM1           COUNT
    011020  611011  2000            JMP     CATLL           LOOP
    011021  623515  2010            JMP     CATL,X          EXHAUSTED, NO FILE FOUND -- BAD RETURN
    011022  443515  2020    CATL9   ISZ     CATL            GOOD RETURN
    011023  623515  2030            JMP     CATL,X
                    2040    *
                    2050    *       SPECIAL CATALOG AND ROUTINES  FOR THE USER CORE IMAGE
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
                    2060    *
          003517    2070            .USE    IMPURE
003517    435762    2080    CORE    .ACI6   *COR*
003520    000000    2090    CORDA   .DSA                    DISK ADDRESS OF THE USER CORE IMAGE
003521    002000    2100    CORCA   BOUNDARY                STARTING CORE ADDRESS OF THE USER CORE
003522    014000    2110    CORLN   CORMAX-BOUNDARY         LENGTH OF USER CORE
                    2120
          011024    2130            .USE    PURE
          011024    2140    CORE1   ...                     SET UP THE USER CORE IMAGE AS A FILE
          011024    2150            MPOFF
                                    .PMC    SAVE,ON
011024    705000                    SPECIAL+0               TURN OFF MEMORY PROTECT
011025    201766    2160            LAC     SUCORE          LOAD THE USER CORE IMAGE DISK ADDRESS
011026    744000    2170            CLL                     PROTECT THE SHIFT
011027    640510    2180            LRS     8,              MAKE THE PHYSICAL ADDRESS INTO A BLOCK ADDRESS
011030    253307    2190            XOR     (040000)        SET THE DISK BIT ON
011031    043520    2200            DAC     CORDA           SET IT IN THE TEMPORARY CATALOG
011032    701742    2210            MPEU                    RE-ENABLE USER MODE
011033    763517    2220            LAW     CORE            LOAD A POINTER TO THE CATALOG
011034    040011    2230            DAC     SCATX           AND PASS IT TO THE CALLER
011035    611022    2240            JMP     CATL9           EXIT
                    2250    *
                    2260    *       SPECIAL CATALOG AND ROUTINES FOR  THE USER "PHYSICAL DISK"
                    2270    *
          003523    2280            .USE    IMPURE
003523    445163    2290    DISK    .ACI6   *DIS*
003524    000000    2300    DISDA   .DSA                    DISK ADDRESS OF THE USER "PHYSICAL DISK
003525    000000    2310    DISCA   0                       MINIMUM USER "PHYSICAL DISK" ADDRESS
003526    016000    2320    DISLN   $DKLEN                  LENGTH OF THE USER "PHYSICAL DISK"
                    2330
          011036    2340            .USE    PURE
          011036    2350    DISK1   ...
          011036    2360            MPOFF
                                    .PMC    SAVE,ON
011036    705000                    SPECIAL+0               TURN OFF MEMORY PROTECT
011037    201767    2370            LAC     SUDISK          LOAD THE USER "PHYSICAL DISK" DISK ADDRESS
011040    744000    2380            CLL                     PROTECT THE SHIFT
011041    640510    2390            LRS     8,              MAKE THE PHYSICAL ADDRESS INTO A DISK ADDRESS
011042    253307    2400            XOR     (040000)        SET THE DISK BIT ON
011043    043524    2410            DAC     DISDA           AND SET IT IN THE TEMPORARY CATALOG
011044    701742    2420            MPEU                    RE-ENABLE USER MODE
011045    763523    2430            LAW     DISK            LOAD A POINTER TO THE CATALOG
011046    040011    2440            DAC     SCATX           PASS IT TO THE CALLER
011047    611022    2450            JMP     CATL9           EXIT
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
                    2460            ,EJECT
                    2470    *
                    2480    *       GNAME
                    2490    *
                    2500    *       GNAME GETS A FILE NAME FROM THE TTY BUFFER
                    2510    *       AND READS IN THE CATALOG IF NECESSARY
                    2520    *
                    2530    *       RETURN IS +1 FOR PAPER TAPE DESIRED
                    2540    *       RETURN IS +2 FOR SUCCESS ON DISK OR DECTAPE
                    2550    *       OTHERWISE EXIT IS TO FORMAT ERROR
                    2560    *
                    2570    *       THE FILE NAME IS RETURNED IN TSWORDB AND IN THE AC.
                    2580    *
    011050          2590            ENTER   GNAME
                                    .PMC    SAVE,ON
    003527                  GNAME   ...
 011050  143531     2600            DZM     CDFLG           INITIALIZE THE SPECIAL FILE FLAG
    011051          2610            WORD                    GET A WORD OF SIX BIT ASCII
 011052  611105     2620            JMP     GNAM90          CHECK FOR A SPECIAL FILE IF A DELIMITER IS FIRST CHARACTER
    011053          2630            DELIM                   GET THE DELIMITER
 011054  553304     2640            SAD     ($COLON)        CHECK FOR COLON
 011055  611061     2650            JMP     GNAM2
 011056  103534     2660    GNAM1   JMS     PAPER           CHECK FOR PAPER TAPE
 011057  623527     2670            JMP     GNAME,X         YES -- PAPER TAPE
 011060  611071     2680            JMP     GNAM5           NO -- SO USE CURRENT CATALOG
                    2690
 011061  771065     2700    GNAM2   LAW     GNAM3
 011062  043532     2710            DAC     DEVCV
    011063          2720            WORD1                   RELOAD THE CATALOG NAME
 011064  611121     2730            JMP     DEVC3           CONVERT IT TO A DEVICE ADDRESS
 011065  623527     2740    GNAM3   JMP     GNAME,X
 011066  103510     2750            JMS     RCAT            READ IN THE CATALOG
    011067          2760            WORD                    GET ANOTHER WORD
 011070  740000     2770            NOP
    011071          2780    GNAM5   DELIM                   GET THE DELIMITER
 011072  553376     2790            SAD     ($SLASH)        CHECK FOR SLASH
 011073  611100     2800            JMP     GNAME6          LOOK FOR OCTAL
    011074          2810            WORD1                   ELSE RECOVER THE SIXBIT NAME
 011075  741200     2820            SNA                     CHECK FOR ALL SPACES
    011076          2830            FORMAT                  FORMAT ERROR -- ALL SPACES IS AN ILLEGAL NAME
 011077  611103     2840            JMP     GNAM8
    011100          2850    GNAME6  NUM                     GET THE NUMBER
    011101          2860            FORMAT
 011102  043326     2870            DAC     TSWORDB         TO BE COMPATABLE WITH SIXBIT INPUT
 011103  443527     2880    GNAM8   ISZ     GNAME           GOOD RETURN
 011104  623527     2890            JMP     GNAME,X
                    2900    *
                    2910    *
                    2920    *       CHECK FOR A SPECIAL FILE REQUEST (E.G. 'CORE' OR 'DISK')
                    2930    *
    003531          2940            ,USE    IMPURE
 003531  000000     2950    CDFLG   .DSA                    FLAG FOR PRESENCE OF SPECIAL FILE REQUEST
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
011105          2960            .USE    PURE
                2970
011105          2980    GNAM90  ...
011105          2990            DELIM   -               RECOVER THE DELIMITER
011106  553304  3000            SAD     ($COLON)        CHECK FOR A VACUOUS COLON
011107  741000  3010            SKP                     YES -- IT IS A SPECIAL FILE
011110  611056  3020            JMP     GNAM1           NO -- RETURN TO NORMAL PROCESSING
                3030
011111  213426  3040            LAC     ($DK0)          LOAD THE IMPLIED SYSTEM DISK MNEMONIC
011112  043326  3050            DAC     TSWORDB         FAKE THAT IT WAS TYPED
011113  443531  3060            INX     CDFLG           FLAG THE SPECIAL FILE REQUEST
011114  611061  3070            JMP     GNAM2           RESUME NORMAL PROCESSING OF THE FAKED INPUT
```

```
                    3080            .EJECT
                    3090    *
                    3100    *       DEVCV -- READS THE NEXT WORD,
                    3110    *       RETURN IS +1 WITH THE NAME IN THE AC IF IT IS A PAPER TAPE CALL
                    3120    *
                    3130    *       OTHERWISE IT ATTEMPTS TO CONVERT THE NAME TO DEVICE ADDRESS FORMAT,
                    3140    *       IF SUCCESSFUL, IT RETURNS +2 WITH THE HANDLER NUMBER IN AC(0-2) AND
                    3150    *       THE DEVICE TYPE (DISK (1) OR DECTAPE (0)) IN AC(3), REMAINING BITS
                    3160    *       ARE ZEROED,
                    3170    *
                    3180    *       EXIT IS TO THE FORMAT ERROR MESSAGE IF THE DEVICE IS NEITHER PAPER TAPE
                    3190    *       NOR DISK NOR DECTAPE,
                    3200    *
    011115          3210            ENTER   DEVCV
                                    ,PMC    SAVE,ON
    003532                  DEVCV   ...
    011115          3220            WORD                    GET THE DEVICE NAME
    011116          3230            FORMAT
 011117  103534     3240            JMS     PAPER           CHECK FOR PAPER TAPE
 011120  623532     3250            JMP     DEVCV,X         YES -- PAPER TAPE
 011121  513303     3260    DEVC3   AND     (777700)        REMOVE DEVICE NUMBER
 011122  553427     3270            SAD     ($TP.)          CHECK FOR DECTAPE
 011123  611133     3280            JMP     DEVC1           YES
 011124  553430     3290            SAD     ($DT.)          CHECK FOR DECTAPE
 011125  611133     3300            JMP     DEVC1
 011126  553431     3310            SAD     ($DK.)          CHECK FOR DISK
 011127  611141     3320            JMP     DEVC4           IT IS DISK -- CKECK FOR VALIDATION
 011130  604026     3330            JMP     M$MONX2         NO OTHERS -- MAYBE IT IS A COMMAND
 011131  650004     3340    DEVC35  CLQ!CMQ                 FOR DISK PUT THE SIGN BIT ON IN THE MQ
 011132  741000     3350            SKP
 011133  650000     3360    DEVC1   CLQ                     CLEAR Q FOR TAPE
    011134          3370            WORD1                   RESTORE NAME
 011135  640617     3380            LLS     18.-3           SHIFT TO POSITION
 011136  513301     3390            AND     (DVCMSK)        CONVERT TO HANDLER DEVICE ADDRESS FORMAT
 011137  443532     3400            ISZ     DEVCV           INCREMENT RETURN
 011140  623532     3410            JMP     DEVCV,X         AND NOW RETURN
                    3420    *
                    3430    *
                    3440    *       DISK FILE OPERATIONS ARE PERMITTED ONLY FOR VALIDATED USERS
                    3450    *
    011141          3460    DEVC4   ...
    011141          3470            MPOFF
                                    ,PMC    SAVE,ON
 011141  705000                     SPECIAL+0               TURN OFF MEMORY PROTECT
 011142  201770     3480            LAC     SVALID          LOAD THE USER'S VALIDATION WORD
 011143  701742     3490            MPEU
 011144  740200     3500            SZA                     SKIP IF THE USER IS NOT VALIDATED
 011145  611131     3510            JMP     DEVC35          ELSE THE OPERATION CAN PROCEED
                    3520    *
                    3530    *       CHECK FOR A SPECIAL FILE OPERATION -- IF SO, IT IS ALLOWED
                    3540    *
 011146  203531     3550            LAC     CDFLG           LOAD THE SPECIAL FILE FLAG
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
011147  740200  3560            SZA                     SKIP IF NOT SET -- THEN THE OPERATION IS ILLEGAL
011150  611131  3570            JMP     DEVC35          IT IS A SPECIAL FILE OPERATION, SO ALLOW IT
                3580    *
                3590    *       DISK OPERATION IS ILLEGAL
                3600    *
    011151      3610            MESS    <DISK OPERATION IS FORBIDDEN>,27,
011166  605345  3620            JMP     MSNEXTL         GET THE NEXT COMMAND LINE
                3630    *
                3640    *       PAPER CHECKS THE AC FOR A PAPER TAPE MNEMONIC, IT RETURNS +1 IF IT
                3650    *       FINDS ONE, ELSE RETURNS +2. THE AC IS UNCHANGED.
                3660    *
    011167      3670            ENTER   PAPER
                                .PMC    SAVE,ON
    003534              PAPER   ...
011167  553432  3680            SAD     ($PPT)
011170  623534  3690            JMP     PAPER,X
011171  553433  3700            SAD     ($PTR)
011172  623534  3710            JMP     PAPER,X
011173  553434  3720            SAD     ($PTP)
011174  623534  3730            JMP     PAPER,X
011175  443534  3740            ISZ     PAPER           NO PAPER TAPE MNEMONIC
011176  623534  3750            JMP     PAPER,X
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
                    3760            .EJECT
                    3770    *
                    3780    *       SAVE CHECKS THE CATALOG FOR THE NAME FOUND IN THE AC
                    3790    *
                    3800    *       RETURN IS +1 IF THE FILE IS ALREADY SAVED
                    3810    *       A CATALOG ENTRY IS CREATED FOR THE NAME AND RETURN IS +2 OTHERWISE
                    3820    *       EXITS TO AN ERROR MESSAGE IF THE CATALOG IS FULL
                    3830    *
                    3840    *       ON RETURN CATX POINTS TO THE FILE NAME IN THE CATALOG
                    3850    *
    011177          3860            ENTER   SAVE
                                    .PMC    SAVE,ON
    003536                  SAVE    ...
011177  103515      3870            JMS     CATL            LOOK UP NAME
011200  741000      3880            SKP
011201  623536      3890            JMP     SAVE,X          DON'T ALLOW DUPLICATES
011202  202172      3900            LAC     CATLOG+2        LOAD THE FCB COUNT
011203  553435      3910            SAD     (CATMAX)        CHECK FOR CATALOG ALREADY FULL
011204  611217      3920            JMP     CFULL           YES -- EXIT TO ERROR MESSAGE
011205  353332      3930            TAD     (-1)            COUNT THE NEW FILE
011206  042172      3940            DAC     CATLOG+2        UPDATE THE FCB COUNT
011207  202171      3950            LAC     CATLOG+1        GET FREE POINTER
011210  353425      3960            TAD     (FCBLEN)        ADD ONE FILE CONTROL BLOCK LENGTH FOR THE NEW ENTRY
011211  042171      3970            DAC     CATLOG+1
    011212          3980            WORD1                   RECOVER THE FILE NAME
011213  060011      3990            DAC     $CATX,X         SAVE IT
011214  442156      4000            ISZ     D$BALT          FLAG THE CATALOG HAS BEEN ALTERED
011215  443536      4010            ISZ     SAVE
011216  623536      4020            JMP     SAVE,X
                    4030
    011217          4040    CFULL   MESS    <CATALOG FULL>,12.
011227  604012      4050            JMP     $NEXTL
                    4060    *
                    4070    *       ALC RECEIVES A WORD COUNT IN THE AC AND CALCULATES THE LEAST INTEGER
                    4080    *       NUMBER OF BLOCKS THAT CAN HOLD THAT LENGTH, IT THEN ALLOCATES THE STORAGE
                    4090    *       IN THE CORE CATALOG HEADER AND RETURNS WITH THE DEVICE ADDRESS OF THE
                    4100    *       FIRST FREE BLOCK IN THE AC.
                    4110    *
                    4120    *       EXIT IS TO AN ERROR MESSAGE IF THIS ALLOCATION WOULD RESULT IN
                    4130    *       OVERFLOWING THE DEVICE. IN THIS CASE THE CATALOG IS UNALTERED.
                    4140    *
    011230          4150            ENTER   ALC
                                    .PMC    SAVE,ON
    003540                  ALC     ...
011230  353353      4160            TAD     (377)           ROUND UP TO A BLOCK
011231  660510      4170            LRSS    8.              AC = MINIMUM INTEGER NUMBER OF BLOCKS REQUIRED
011232  043507      4180            DAC     CTEM1           SAVE IN A GOOD RANDOM PLACE
011233  202170      4190            LAC     CATLOG          GET THE POINTER TO THE FIRST FREE BLOCK
011234  652000      4200            LMQ                     SAVE IT
011235  343507      4210            TAD     CTEM1           ADD THE REQUESTED NUMBER OF BLOCKS TO FORM A NEW POINTER
011236  043507      4220            DAC     CTEM1           SAVE THE NEW POINTER
011237  513336      4230            AND     (1777)          EXTRACT BLOCK NUMBER
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
011240  342173  4240            TAD     CATLOG+3        SEE IF WE OVERFLOWED THE DEVICE
011241  740100  4250            SMA                     NO IF SKP
011242  611247  4260            JMP     DFULL           FULL -- HELP*?!@
011243  203507  4270            LAC     CTEM1
011244  042170  4280            DAC     CATLOG          SET THE FREE FCB POINTER NOW WE KNOW IT WILL BE OK
011245  641002  4290            LACQ                    RESTORE THE DEVICE ADDRESS OF THE FIRST FREE BLOCK
011246  623540  4300            JMP     ALC,X
                4310
    011247      4320    DFULL   MESS    <DEVICE FULL>,11.
011257  604012  4330            JMP     $NEXTL
                4340            .END
```

C SYSTEM MESSAGE OUTPUT ROUTINES

```
                    10110           .SITL   SYSTEM MESSAGE OUTPUT ROUTINES
          011260    10120           .USE    PURE
                    10130           .HEAD   M
                    10140   *       MSG0: A CALL FOR MESSAGE #0 IS A MONITOR CALL
                    10150   *       MSG1-9: RESERVED FOR MEMORY PROTECTION OVERLAY #1 (MP1)
                    10160   *       MSG10-19: RESERVED FOR MEMORY PROTECTION OVERLAY #2 (MP2)
                    10170   *       MSG50-59: RESERVED FOR MONITOR MESSAGES, MONITOR TRANSFERS DIRECTLY
                    10180   *                 TO ITS OWN MESSAGE ROUTINES, NOT GOING THROUGH THE SORTING PROCEDURE.
                    10190   *
 011260   604004    10200   MSGPTR  JMP     MONMSG
 011261   611305    10210           JMP     MSG1
 011262   611321    10220           JMP     MSG2
 011263   611344    10230           JMP     MSG4
 011264   611332    10240           JMP     MSG3
 011265   611365    10250           JMP     MSG5
 011266   611377    10260           JMP     MSG6
 011267   611412    10270           JMP     MSG7
 011270   611466    10280           JMP     MSG8
 011271   611447    10290           JMP     MSG9
 011272   611424    10300           JMP     MSG10
 011273   611506    10310           JMP     MSG11
 011274   611522    10320           JMP     MSG12
                    10330
     777763         10340   MSGMAX  .EQU    MSGPTR-.        TWO'S COMP OF LENGTH OF TABLE FOR CHECKING MESSAGE REQUESTS
                    10350
     011275         10360   MSGOCT  OCTZ                    PRINT THE AC IN ZERO-SUPPRESSED OCTAL
 011277   604012    10370           JMP     MONSYM          GET THE NEXT LINE
                    10380
 011300   202000    10390   ADDOCT  LAC     TEMP0
 011301   744002    10400           STL                     SUPPRESS LEADING ZEROES
     011302         10410           OCTZ                    PRINT THE ERROR ADDRESS IN OCTAL
 011304   604004    10420           JMP     MONMSG          PRINT THE MONITOR HEADER
                    10430   *
                    10440   *       MP1 MESSAGES, GENERALLY PRINTING THE PC AS WELL
                    10450   *
                    10460
     011305         10470   MSG1    MESS    <ILLEGAL TRANSFER TO >,20.
 011320   611300    10480           JMP     ADDOCT          PRINT THE LOCATION OF THE VIOLATION
                    10490
     011321         10500   MSG2    MESS    <BAD ADDRESS: >,13.
 011331   611300    10510           JMP     ADDOCT
                    10520
     011332         10530   MSG3    MESS    <CHAINED XCT'S: >,15.
 011343   611300    10540           JMP     ADDOCT
                    10550
     011344         10560   MSG4    MESS    <ILLEGAL OP CODE >,16.
 011355   202001    10570           LAC     TEMP1           LOAD THE OP CODE
     011356         10580           OCT
     011360         10590           NMESS   < AT >,4
 011364   611300    10600           JMP     ADDOCT
                    10610
     011365         10620   MSG5    MESS    <PROGRAM HALTED: >,16.
```

M SYSTEM MESSAGE OUTPUT ROUTINES

```
011376  611300  10630           JMP     ADDOCT
                10640
    011377      10650   MSG6    MESS    <DEVICE OVERFLOW: >,17.
011411  611300  10660           JMP     ADDOCT
                10670
    011412      10680   MSG7    MESS    <CORE OVERFLOW: >,15.
011423  611300  10690           JMP     ADDOCT
                10700
    011424      10710   MSG10   MESS    <DEVICE TRANSFER TO/FROM PROTECTED MEMORY: >,42.
011446  611300  10720           JMP     ADDOCT
                10730
    011447      10740   MSG9    MESS    <UNASSIGNED DEVICE REQUESTED: >,29.
011465  611300  10750           JMP     ADDOCT
                10760
    011466      10770   MSG8    MESS    <NON-EXISTANT DISK REFERENCED: MESS>,30.
011505  611300  10780           JMP     ADDOCT
                10790
    011506      10800   MSG11   MESS    <FILE NOT FOUND: MESS>,16.
011521  611300  10810           JMP     ADDOCT
                10820
    011522      10830   MSG12   MESS    <CAL: >,5
011530  611300  10840           JMP     ADDOCT
```

M SYSTEM MESSAGE OUTPUT ROUTINES

```
                    10850           ,EJECT
                    10860           ,HEAD   M,C,T,D,0
                    10870   *
                    10880   *       MONITOR MESSAGES RETURN TO PRINT MONITOR-READY MESSAGE WHEN COMPLETED.
                    10890   *
                    10900
    011531          10910   MSG50   MESS    <WHAT: >,6
    011537          10920           CMDERR
                    10930
    011540          10940   MSG51   ..,
011540  701742      10950           MPEU
    011541          10960           MESS    <RESOURCE ALREADY ALLOCATED>,26,
    011556          10970           CMDERR
                    10980
    011557          10990   MSG52   ..,
011557  701742      11000           MPEU
    011560          11010           MESS    <NOT YOUR RESOURCE>,17,
    011572          11020           CMDERR
                    11030
    011573          11040   MSG53   MESS    <[]*** WARNING -- ILLEGAL MESSAGE NUMBER GENERATED: >,50,
011620  202002      11050           LAC     TEMP2           LOAD THE ILLEGAL MESSAGE NUMBER
    011621          11060           OCTZ                    AND PRINT IT
011623  604004      11070           JMP     MONMSG
                    11080
    011624          11090   MSG54   ..,
011624  701742      11100           MPEU
    011625          11110           MESS    <BOTH DECTAPE HANDLERS ALREADY ALLOCATED>,39,
    011646          11120           CMDERR
                    11130
                    11140
    011647          11150   MSG56   MESS    <FORMAT ERROR: >,14,
                    11160
    011660          11170   MSG57   NMESS   < WORD # >,8,
    011665          11180           COUNT                   GET THE NUMBER OF THE WORD PROVOKING THE ERROR MESSAGE
011666  611275      11190           JMP     MSGOCT
                    11200
    011667          11210   MSG58   MESS    <VALIDATION ERROR>,16,
    011700          11220           MPOFF
                                    ,PMC    SAVE,ON
011700  705000                      $PECIAL+0               TURN OFF MEMORY PROTECT
011701  141770      11230           DZM     $VALID          INSURE THE USER IS NOT VALIDATED
011702  701742      11240           MPEU
011703  604012      11250           JMP     MONSYM          GET THE NEXT LINE OF INPUT
                    11260
    011704          11270   MSG59   MESS    <ADDRESS OUT OF BOUNDS>,21,
    011717          11280           CMDERR
                    11290
    011720          11300           ENTER   MSG60           SAVE THE LOCATION OF THE SYSTEM ERROR HERE
                                    ,PMC    SAVE,ON
    003542                  MSG60   ..,
    011720          11310           MESS    <SYSTEM ERROR: >,14,
011731  777777      11320           LAW     -1
```

M C T D SYSTEM MESSAGE OUTPUT ROUTINES

```
011732  343542  11330           TAD     MSG60           RECOVER THE LOCATION OF THE SYSTEM ERROR
    011733      11340           OCTZ
011735  604012  11350           JMP     MONSYM          GET THE NEXT COMMAND
                11360
    011736      11370   MSG61   MESS    <FILE NOT FOUND: >,16.
    011747      11380           CMDERR
                11390
    011750      11400   MSG62   MESS    <CAN'T CATALOG IT>,16.
    011761      11410           CMDERR
                11420
    011762      11430   MSG63   MESS    <FOR A LIST OF MONITOR COMMANDS TYPE 'EXPLAIN COMMANDS'>,54.
    012010      11440           MESS    <FOR A LIST OF MTSS ALLOCATABLE RESOURCES TYPE 'EXPLAIN RESOURCES'>,65.
012042  604023  11450           JMP     MONXT           PICK UP THE NEXT COMMAND
                11460
    012043      11470   MSG64   ...,
    012043      11480           CRLF
    012044      11490           MESS    <BAS -- CALLS S-USER PROGRAM 'BASIC INTERPRETER'>,47.
    012070      11500           MESSR   <BYE -- CLEARS USER FLAGS, DE-ALLOCATES USER RESOURCES,>,54.
    012115      11510           MESSR   <       ZEROES USER CORE AND USER DISK>,37.
    012134      11520           MESSR   <CAF -- CLEARS ALL USER HARDWARE FLAGS>,37.
    012153      11530           MESSR   <CAT -- CALLS PHANTOM PROGRAM 'CATALOG'>,38.
    012173      11540           MESSR   <CON -- CONTINUE BEGINS TO EXECUTE CURRENT USER CORE>,51.
    012217      11550           MESSR   <       AT CURRENT REGISTER VALUES>,33.
    012235      11560           MESSR   <DEB -- CALLS S-USER PROGRAM 'DEBUGGER'>,38.
    012255      11570           MESSR   <DDT -- CALLS PHANTOM PROGRAM 'DDT'>,34.
    012273      11580           MESSR   <EXP -- LIST EXPLAINABLE TOPICS>,31.
    012310      11590           MESSR   <GOO -- GOODBYE IS THE SAME AS BYE>,34.
    012326      11600           MESSR   <GRO -- ABORTS MTSS AND BOOTSTRAPS GROWTH. REQUIRES CONTROL LINE>,64.
    012356      11610           MESSR   <HEL -- HELLO HAS THE SAME EFFECT AS BYE>,40.
    012376      11620           MESSR   <JMP -- JMP ADDRESS REPLACES USER PC WITH THE ADDRESS AND CONTINUES>,66.
    012427      11630           MESSR   <LDR -- CALLS PHANTOM PROGRAM 'LOADER'>,37.
    012446      11640           MESSR   <OFF -- 'OFF RESOURCE' DE-ALLOCATES THE NAMED RESOURCE>,53.
    012473      11650           MESSR   <ON  -- 'ON RESOURCE' ATTEMPTS TO ALLOCATE THE NAMED RESOURCE>60.
    012522      11660           MESSR   <TRA -- TRANSFER IS THE SAME AS JMP>,34.
    012540      11670           MESSR   <VAL -- REQUESTS UNDERPRINTING FOR A VALIDATION PASSWORD>,55.
    012565      11680           MESSR   <XDU -- SIMPLE OCTAL DUMP FOR SYSTEM DEBUGGING>,45.
    012607      11690           MESSR   <XPA -- SIMPLE OCTAL PATCH FOR SYSTEM DEBUGGING>,46.
    012631      11700           MESSR   <ZER -- 'ZERO CORE' ZEROES USER CORE>,35.
    012650      11710           MESSR   <       'ZERO DISK' SETS THE ENTIRE USER DISK TO ZERO>,52.
    012674      11720           CRLF
012675  604023  11730           JMP     MONXT           PICK UP THE NEXT COMMAND
                11740
    012676      11750   MSG65   ...,
    012676      11760           CRLF
    012677      11770           MESS    <PTR -- PAPER TAPE READER>,24.
    012713      11780           MESS    <PTP -- PAPER TAPE PUNCH>,23.
    012727      11790           MESS    <ACS -- HARDWARE ACCUMULATOR SWITCHES>,36.
    012747      11800           MESS    <CNT -- CONTROL LINE>,19.
    012761      11810           MESS    <SCO -- GRAPHICS II PERIPHERALS>,30.
    012777      11820           MESS    <DTN -- DECTAPE HANDLER #N>,25.
    013013      11830           MESS    <TPN -- DECTAPE HANDLER #N>,25.
    013027      11840           CRLF
```

M C T D SYSTEM MESSAGE OUTPUT ROUTINES

```
013030  604023  11850          JMP     MONXT           PICK UP THE NEXT COMMAND
```

```
M C T D                                 MTSS DEBUGGER MESSAGES

                    11860               ,STITL  MTSS DEBUGGER MESSAGES
                    11870               ,HEAD   D
                    11880
                    11890
                    11900
    013031          11910   MSGLOC      ...
013031  042000      11920               DAC     TEMP0           SAVE THE AC (= LOCATION OF THE ERROR)
    013032          11930               NMESS   < AT >,4
013036  202000      11940               LAC     TEMP0
    013037          11950   MSGCT       OCTZ
013041  605345      11960               JMP     DSNXLIN
                    11970
    013042          11980   MSGWRD      ...
    013042          11990               NMESS   < WORD >,6      PRINT THE WORD COUNT AT THE ERROR
    013046          12000               COUNT
013047  613037      12010               JMP     MSGCT
                    12020
    013050          12030               ENTER   MSG80
                                        ,PMC    SAVE,ON
    003544                  MSG80       ...
    013050          12040               MESS    <DDT SYSTEM ERROR>,16.
013061  203544      12050               LAC     MSG80
013062  613031      12060               JMP     MSGLOC
                    12070
    013063          12080   MSG81       MESS    <FILE CAN'T BE OPENED>,20.
013076  613042      12090               JMP     MSGWRD
                    12100
    013077          12110   MSG82       MESS    <FILE NOT FOUND>,14.
013110  613042      12120               JMP     MSGWRD
                    12130
    013111          12140   MSG83       MESS    <FORMAT ERROR>,12.
013121  613042      12150               JMP     MSGWRD
                    12160
    013122          12170   MSG84       MESS    <ILLEGAL MODE OR COMMAND>,23.
013136  613042      12180               JMP     MSGWRD
                    12190
    013137          12200   MSG85       MESS    <ILLEGAL DEVICE NUMBER>,21.
013152  613042      12210               JMP     MSGWRD
                    12220
    013153          12230   MSG86       MESS    <VALIDATION ERROR>,16.
013164  613042      12240               JMP     MSGWRD
                    12250
    013165          12260   MSG87       MESS    <UNDEFINED SYMBOL>,16.
013176  613042      12270               JMP     MSGWRD
                    12280
    013177          12290   MSG88       MESS    <ILLEGAL HEAD SYMBOL>,19.
013211  613042      12300               JMP     MSGWRD
                    12310
    013212          12320   MSG89       MESS    <TABLE FULL>,10.
013221  613042      12330               JMP     MSGWRD
                    12340
    013222          12350   MSG90       MESS    <NOT FOUND>,9.
```

D MTSS DEBUGGER MESSAGES

```
013231  613042  12360           JMP     MSGWRD
                12370
    013232      12380   MSG91   MESS    <ADDRESS OUT OF BOUNDS>,21.
013245  613042  12390           JMP     MSGWRD
```

```
D                                   GROWTH SYSTEM BOOTSTRAP

                    12400           .STITL  GROWTH SYSTEM BOOTSTRAP
                    12410           .HEAD   M,C,T,D
                    12420   *
                    12430   *
                    12440   *
    013246          12450   GRO     MPOFF                   LEAVING TSS, SO TURN OFF MEMORY PROTECT
                                    .PMC    SAVE,ON
013246  705000                      SPECIAL+0               TURN OFF MEMORY PROTECT
013247  700006      12460           IOFICLOF                INSURE NO INTERRUPTS WHILE TRYING TO SHUT DOWN THE SYSTEM
013250  200006      12470           LAC     $CNTRL
013251  541771      12480           SAD     $NUMBR          SEE IF THE CURRENT USER HAS A CONTROL LINE
013252  613256      12490           JMP     .+4
013253  700046      12500           IONICLON
013254  701742      12510           MPEU                    NO -- SO DON'T LET HIM CRASH THE SYSTEM
    013255          12520           WHAT                    AND PRINT HIS ERROR MESSAGE
013256  703302      12530           CAF
013257  777377      12540           LAW     -401
013260  040036      12550           DAC     36              SET TO READ IN ONE BLOCK
013261  040037      12560           DAC     37
013262  707074      12570           DLAH+10                 SET PHYSICAL DISK 0
013263  213272      12580           LAC     C1
013264  707024      12590           DLAL                    SET THE DISK ADDRESS
013265  760002      12600           LAW     2
013266  707047      12610           DSCF!DSFX!DSCN          NON-INTERRUPTING DISK READ
013267  707001      12620           DSSF
013270  613267      12630           JMP     .-1
013271  617740      12640           JMP     17740           DONE -- ENTER GROWTH SYSTEM
013272  540000      12650   C1      540000                  START ADDRESS OF GROWTH SYSTEM
                    12660
                    12670           .HEAD
    003546          12680           .USE    IMPURE
    003546          12690   CHECK   .EQU    .
    003700          12700           .LOC    PURSTR
    013273          12710           .USE    PURE
013273  705377      12720           .END    $START
013274  705375
013275  705376
013276  017777
013277  777763
013300  011260
013301  740000
013302  440010
013303  777700
013304  000272
013305  000260
013306  037777
013307  040000
013310  000336
013311  000215
013312  422027
013313  422026
013314  422030
```

GROWTH SYSTEM BOOTSTRAP

```
013315  506000
013316  512000
013317  000077
013320  640402
013321  640477
013322  770000
013323  570000
013324  700000
013325  000777
013326  000024
013327  741000
013330  001300
013331  000001
013332  777777
013333  041300
013334  001000
013335  400001
013336  001777
013337  000200
013340  000300
013341  763241
013342  000007
013343  000004
013344  377777
013345  417777
013346  100000
013347  507000
013350  000240
013351  777740
013352  000100
013353  000377
013354  000177
013355  762000
013356  007227
013357  770534
013360  020000
013361  640000
013362  000244
013363  760000
013364  000700
013365  776777
013366  777760
013367  767461
013370  767531
013371  000241
013372  000246
013373  000252
013374  000253
013375  000255
013376  000257
013377  000334
013400  000254
```

GROWTH SYSTEM BOOTSTRAP

```
013401  657122
013402  657323
013403  777540
013404  000256
013405  000243
013406  002000
013407  002170
013410  003327
013411  000337
013412  000230
013413  000017
013414  000333
013415  000335
013416  777520
013417  777770
013420  777776
013421  777772
013422  777737
013423  777745
013424  000400
013425  000005
013426  445320
013427  646000
013430  446400
013431  445300
013432  606064
013433  606462
013434  606460
013435  777716
013436  000000
013437  000000
013440  000000
013441  000000

TRANSFER ADDRESS  603701
```

CROSS REFERENCE TABLE

1713	.0	4510	4520		
11534	.046.	10910			
11544	.047.	10960			
11563	.048.	11010			
11576	.049.	11040			
11630	.050.	11110			
11652	.051.	11150			
11662	.052.	11170			
11672	.053.	11210			
11707	.054.	11270			
11723	.055.	11310			
11741	.056.	11370			
11753	.057.	11400			
11765	.058.	11430			
12013	.059.	11440			
12047	.060.	11490			
12072	.061.	11500			
12117	.062.	11510			
12136	.063.	11520			
12155	.064.	11530			
12175	.065.	11540			
12221	.066.	11550			
12237	.067.	11560			
12257	.068.	11570			
12275	.069.	11580			
12312	.070.	11590			
12330	.071.	11600			
12360	.072.	11610			
12400	.073.	11620			
12431	.074.	11630			
12450	.075.	11640			
12475	.076.	11650			
12524	.077.	11660			
12542	.078.	11670			
12567	.079.	11680			
12611	.080.	11690			
12633	.081.	11700			
12652	.082.	11710			
12702	.083.	11770			
12716	.084.	11780			
12732	.085.	11790			
12752	.086.	11800			
12764	.087.	11810			
13002	.088.	11820			
13016	.089.	11830			
26	.310	3400	6100		
27	.311	3410	6120		
4453	.DK	770			
4464	.DT	750			
6460	.TP	760			
2023	10SAVE	1870	1880	1840	6090
2024	11SAVE	1880	1920	1860	6110

CROSS REFERENCE TABLE

CROSS REFERENCE TABLE

13272	C C1	12650								
3540	C ALC	4150	4300							
10757	C CON	1550	1480							
13246	C GRO	12450								
10796	C IGN	1540	1460							
3515	C CATL	1680	1860	1120	1190	1280	2010	2020	2030	3870
3517	C CORE	2080	1800	2220						
3523	C DISK	2290	1820	2430						
3510	C RCAT	740	3790	1230	2750					
3536	C SAVE	3860	3890	4010	4020					
10721	C,027,	1390								
10733	C,028,	1420								
10774	C,029,	1840								
11154	C,030,	3610								
11222	C,031,	4040								
11252	C,032,	4320								
11534	C,046,	10910								
11544	C,047,	10960								
11563	C,048,	11010								
11576	C,049,	11040								
11630	C,050,	11110								
11652	C,051,	11150								
11662	C,052,	11170								
11672	C,053,	11210								
11707	C,054,	11270								
11723	C,055,	11310								
11741	C,056,	11370								
11753	C,057,	11400								
11765	C,058,	11430								
12013	C,059,	11440								
12047	C,060,	11490								
12072	C,061,	11500								
12117	C,062,	11510								
12136	C,063,	11520								
12155	C,064,	11530								
12175	C,065,	11540								
12221	C,066,	11550								
12237	C,067,	11560								
12257	C,068,	11570								
12275	C,069,	11580								
12312	C,070,	11590								
12330	C,071,	11600								
12360	C,072,	11610								
12400	C,073,	11620								
12431	C,074,	11630								
12450	C,075,	11640								
12475	C,076,	11650								
12524	C,077,	11660								
12542	C,078,	11670								
12567	C,079,	11680								
12611	C,080,	11690								
12633	C,081,	11700								

CROSS REFERENCE TABLE

CROSS REFERENCE TABLE

45	CMP1	3490	3500							
46	CMP2	3500	3510							
11531	CMSG50	10910								
11540	CMSG51	10940								
11557	CMSG52	10990								
11573	CMSG53	11040								
11624	CMSG54	11090								
11647	CMSG56	11150	2830	2860	3230					
11660	CMSG57	11170								
11667	CMSG58	11210								
11704	CMSG59	11270								
3542	CMSG60	11300								
11736	CMSG61	11370								
11750	CMSG62	11400								
11762	CMSG63	11430								
12043	CMSG64	11470								
12676	CMSG65	11750								
4076	CNTOFF	8910	9320							
6	CNTRL	3380	3390	2650	3330	6610	12470			
272	COLON	370	3810	1750	22490	22820	2640	3000		
2053	COMFLG	2200	2210	1410	7460	9830				
254	COMMA	320	15230	22810						
2150	COMSTO	2270	2280							
230	CONTX	230	820							
435762	COR	880								
16000	CORMAX	910	980	7840	2110					
3534	CPAPER	3670	2660	3240	3690	3710	3730	3740	3750	
17095	CPARAM	590								
215	CR	220	5130	1490	3680	4170	6510	7160	910	2420
10640	CRCAT1	780								
10663	CRCAT3	1010	930							
10677	CRCAT4	1160	1290							
3512	CRCOVR	1340	920	1140	1210	1370	1410	1490		
10716	CRCVR4	1390	970							
10730	CRCVR5	1420	1450	1500						
67	CSPL	3510	3520							
44	CSWP	3480	3490							
60	CTBFR	3600	3630	3640						
100	CTBIN	3640	3650	3670	4250					
2000	CTEMP0	1630								
2001	CTEMP1	1640								
2002	CTEMP2	1650								
2003	CTEMP3	1660								
2004	CTEMP4	1670								
2005	CTEMP5	1680								
2006	CTEMP6	1690								
2007	CTEMP7	1700								
2010	CTEMP8	1710								
2011	CTEMP9	1720								
102	CTFLG	3650	3660							
436454	CTL	810								
104	CTNAM	3660								

CROSS REFERENCE TABLE

13272	D	C1	12650											
6602	D	J1	9920	10020										
7024	D	M1	12080	12150										
5617	D	M6	3950	9760										
5633	D	M7	3970	9780										
5641	D	M8	3980	9790										
5631	D	MA	3940	9750										
5647	D	MD	3990	9800										
5625	D	MH	3960	9770										
5603	D	MQ	3930	9740										
5655	D	MS	4000	9810										
2043	D	PC	2120	2130	3250	6410	7020	7380	7400	7530	7690	7750	7810	7840	7860
				7900	15980										
5663	D	S9	4030	3930	3940	3950	3960	3970	3980	3990	4000				
6522	D	AAC	9200	8240											
10333	D	ACF	22930	4430	4460	5030	10070	21170	22260						
6537	D	ALK	9390	8680											
6517	D	ALT	9130	9580											
5621	D	AM6	3950	9760											
5635	D	AM7	3970	9780											
5643	D	AM8	3980	9790											
5613	D	AMA	3940	9750											
5651	D	AMD	3990	9800											
5627	D	AMH	3960	9770											
5605	D	AMQ	3930	9740											
6526	D	AMQ	9260	8310											
5657	D	AMS	4000	9810											
6597	D	APC	9590	8600											
5665	D	AS9	4060	3930	3940	3950	3960	3970	3980	3990	4000				
6532	D	ASC	9320	8440											
6075	D	BAS	6020	9680											
2154	D	BCA	2370	2380	21560	21840	22690	890							
2153	D	BDA	2360	2370	20580	21920	22080	22380	810	850	870	950			
5774	D	BLQ	5180	9910	9910										
11531	D	BRE	3590	3630											
6063	D	CLO	5800	9630											
6612	D	CON	10010	9220	9220	9620	9620								
5782	D	COR	4310	9860	9860										
5324	D	DDT	1030	9210											
5736	D	DIS	4540	9870	9870										
6234	D	DUM	7450	9560	3670	3690									
6614	D	EXI	10050	9230	9230	9610	9610								
2163	D	FDA	2440	2450	1920	1960	1970	2280	2340	4640	4800	4880	4890	5240	5260
				20570	20700	20960	21910								
5551	D	FPC	3410	3190											
13246	D	GRO	12450												
6164	D	HRI	6600												
6773	D	IND	11750	11710											
6067	D	LIM	5880	9650											
2042	D	LOC	2110	2120	6130	6230	6310	6340	6390						
6072	D	MAS	5950	9660											
3255	D	MD1	16530	16400	16510										

CROSS REFERENCE TABLE

3256	D	MD2	16540	16490									
7541	D	MD4	16570	16550									
5353	D	NSU	1350	9600	9600	9600	4200						
6174	D	PAT	6920	9550	9550	3610	7230						
5761	D	PRE	5010	9640									
6347	D	RAC	8220	9420									
6302	D	REG	8070	9570									
6414	D	RLK	8660	9450									
5623	D	RM6	3950	9760									
5637	D	RM7	3970	9780									
5645	D	RM8	3980	9790									
5615	D	RMA	3940	9750									
5653	D	RMD	3990	9800									
5631	D	RMH	3960	9770									
5607	D	RMO	3930	9740									
6330	D	RMQ	8290	9430									
5661	D	RMS	4000	9810									
6402	D	RPC	8580	9440									
5667	D	RS9	4090	3930	3940	3950	3960	3970	3980	3990	4000		
6353	D	RSC	8420	9480									
6100	D	SEA	6100	9670	6550								
3247	D	SIX	10430	10370	10380	10390	10500						
5721	D	SYS	4610	9880	9880								
6577	D	TRA	9880	9200	9200	9200	9200	9590	9590	9590	9590	3650	
10332	D	UCF	22920	20800	21370								
5735	D	V5A	4770	9890	9890								
6020	D	VAL	5420	9690	9690								
5704	D	XCO	4400	9900	9900								
3225	D	6MOD	10350	3950	3950	3950	10400						
3235	D	7MOD	10670	3970	3970	3970	10740						
3233	D	8MOD	10610	3980	3980	3980	10640						
6570	D	AACS	9700	8370									
2022	D	ACSW	1860	8390	9980	9730							
3223	D	AMOD	10260	4240	3940	3940	3940	10320	11490	11830	13150	13450	
6553	D	ASTS	9530	8170									
6574	D	AVAL	9760										
2156	D	BALT	2390	2400	2220	8260	7000	22210	22220	4000			
10327	D	BKSN	22860	3620									
2155	D	BLEN	2380	2390	21860	22710	910						
2161	D	BMAX	2420	2430	20520	21450	22040	22050					
2157	D	BMIN	2400	2410	20530	20540	21960	22000					
2162	D	BPTR	2430	2440	8250	6250	6440	6990	7770	15430	21150	21230	21300	21440	21480
				21570	21640	21670	21710	21870							
7530	D	COMI	16840	16920									
7527	D	COMX	16830	16900									
6704	D	DEC2	10870	10810									
6707	D	DEC4	10910	10960									
6716	D	DEC6	10990	11070									
6727	D	DEC8	11090	11010									
3237	D	DMOD	10770	3990	3990	3990	11120	12290					
6227	D	DNXT	7370	1500	1520								
5572	D	DON1	4160	4230									

CROSS REFERENCE TABLE

5671	D DONE	4140	1390	5050	5090	5120	5840	5910	5980	6050	6520	6540	7170	7190
			7480	7830										
10331	D DTCT	22910	2370											
6622	D EXI2	10120	10090											
10317	D FDEL	22780	1530	2530										
5547	D FIND	3340	3170											
2167	D FMAX	2480	2490	2100	2380	4710	4960	5330	20790	21030	21240			
2165	D FMIN	2460	2470	2020	2070	2320	4680	4930	5300	20750	21000	21950		
6515	D FREG	9060	8200	8270	8340	8400								
3231	D HMOD	10530	3960	3960	3960	10580								
6151	D HRI2	6660	6750											
6162	D HRI6	6760	6730											
7443	D INPC	15910	3480	15780										
7254	D INV2	14310	15680	15870										
7264	D INV3	14460	14360											
7274	D INV4	14550	14490											
7302	D INV6	14630	14420											
10330	D JSGN	22870	3640											
10334	D LOGF	22940	2110											
2086	D MASK	2150	2160	5970	6200	6260								
2164	D MFDA	2450	2460	1990	2000	2300	2310	4660	4670	4910	4920	5280	5290	20720
			20730	20980	20990	21930								
5560	D MODE	3520	2880	2990	3730	4040	4070	4100	9840					
5475	D NCOP	2460	2150	4330										
5357	D NSU2	1400	1310											
3221	D OMOD	10210	1140	3930	3930	3930	10230							
5414	D OPE1	1850	1820											
5451	D OPE2	2200	1770											
5456	D OPE4	2260	2210											
5376	D OPEN	1630												
7143	D OPR1	13080	13020											
7121	D OPR2	12880	13110											
7146	D OPR6	13140	12920											
7104	D OPR8	12720	13140											
6202	D PAT1	6990	7130											
6216	D PAT2	7140	6980	7070										
10016	D PBS2	19900	20010	20080	20340	20390								
10046	D PBS4	20200	20270											
10217	D PNEW	21810	21470	21510										
10204	D PUCF	21620	21380											
6341	D RACS	8350	9470											
6427	D RALL	8760	9490											
6355	D RSC2	8450	9000											
10320	D RSGN	22790												
6305	D RSTS	8150	9460											
6364	D RVAL	8490	9500											
7065	D SAM2	12500	12600											
7076	D SAM4	12630	12530											
7072	D SAM6	12560	12690											
7073	D SAM8	12580	12660											
6105	D SEA1	6170												
6111	D SEA2	6220	6360											

CROSS REFERENCE TABLE

```
 6117  D SEA3    6300   6470
 6125  D SEA4    6380   6280
 6136  D SEA5    6490   6190   6330
 7003  D SEA8   11890
 7004  D SEAE   11900  13920
 7152  D SIOT   13240  13930
 6776  D SLAW   11810  12770
 3241  D SMOD   11390   1170   4000   4000   4000   7630  11700  11730  11760  12050  12250  12300  12980
                       13160  13460  13530  13730
 7105  D SOPR   12730  13940
 3272  D SRCH   18900  18740  18810  18970  19120  19130
 6512  D SREG    9020   8470   8560   8640   8730
13450  D SYM0     150  18730  19180  19850
14565  D SYM1    2150  18800  19240    150
 7040  D SYM2   12220  11620
 3243  D SYM4   12320  12220  12410  12930  13670
 6015  D VAL1    5380   5500
 6016  D VAL2    5390   5540
 6017  D VAL3    5400   5480
 6050  D VAL9    5570   5490
 7764  D VSR2   19460  19600  19650
 5327  D,010,    1040
 5350  D,011,    1290
 6022  D,012,    5440
 6312  D,013,    8180
 6324  D,014,    8250
 6335  D,015,    8320
 6346  D,016,    8380
 6360  D,017,    8450
 6371  D,018,    8520
 6407  D,019,    8610
 6421  D,020,    8690
 6434  D,021,    8790
 6443  D,022,    8820
 6452  D,023,    8850
 6462  D,024,    8890
 6474  D,025,    8940
 6504  D,026,    8970
11534  D,046,   10910
11544  D,047,   10960
11563  D,048,   11010
11576  D,049,   11040
11630  D,050,   11110
11652  D,051,   11150
11662  D,052,   11170
11672  D,053,   11210
11707  D,054,   11270
11723  D,055,   11310
11741  D,056,   11370
11753  D,057,   11400
11765  D,058,   11430
12013  D,059,   11440
```

CROSS REFERENCE TABLE

12047	D,060,	11490										
12072	D,061,	11500										
12117	D,062,	11510										
12136	D,063,	11520										
12155	D,064,	11530										
12175	D,065,	11540										
12221	D,066,	11550										
12237	D,067,	11560										
12257	D,068,	11570										
12275	D,069,	11580										
12312	D,070,	11590										
12330	D,071,	11600										
12360	D,072,	11610										
12400	D,073,	11620										
12431	D,074,	11630										
12450	D,075,	11640										
12475	D,076,	11650										
12524	D,077,	11660										
12542	D,078,	11670										
12567	D,079,	11680										
12611	D,080,	11690										
12633	D,081,	11700										
12652	D,082,	11710										
12702	D,083,	11770										
12716	D,084,	11780										
12732	D,085,	11790										
12752	D,086,	11800										
12764	D,087,	11810										
13002	D,088,	11820										
13016	D,089,	11830										
13034	D,090,	11930										
13044	D,091,	11990										
13053	D,092,	12040										
13066	D,093,	12080										
13102	D,094,	12110										
13114	D,095,	12140										
13125	D,096,	12170										
13142	D,097,	12200										
13156	D,098,	12230										
13170	D,099,	12260										
13202	D,100,	12290										
13215	D,101,	12320										
13235	D,102,	12350										
13235	D,103,	12380										
6575	DADONE	9820	8100	9040	9080	9160	9240	9300	9370	9510	9570	9680	9740
2036	DADRSW	2070	2080	1150	4060	6420	7710						
1762	DAP0	4590	4600										
1763	DAP1	4600	4610										
3324	DBINIT	22660	7880	8130	1080	22720							
653	DBK	4120	4130										
24	DBKNUM	2220	2270										
2054	DBKTAB	2210	2270										

CROSS REFERENCE TABLE

5472	DBOPEN	2420	4560	4720	4970	5340							
2035	DBSTOR	2050	2060										
10322	DBYSGN	22810	2790										
3320	DCOLSP	22480	6430	7720	22530								
10125	DCORCT	20850	20780										
5720	DDISCT	4570	21020										
422027	DDT	410	5400										
5132	DDTCOM	9540											
5212	DDTMOD	9730											
12090	DDTST	5000											
2037	DDUMSW	2080	2090	1180	4030	6450	7620	7780					
1	DEBUG	10030	100	120	100	120	100	120					
10324	DENDSN	22830	1370	1510	3660	4190	6530	7180					
11131	DEVC35	3340	3510	3570									
5504	DFIELD	2720	2540	3360	4440								
5273	DFILES	9860	1670										
1761	DFLAG	4580	4590										
5537	DFLD05	3140	1550	2740									
5505	DFLD06	2740	3420										
5544	DFLD07	3240											
5506	DFLD08	2750	1470										
5515	DFLD10	2860	2940	3000	3060	3270							
5516	DFLD11	2870	3100										
5520	DFLD20	2920	2800										
5531	DFLD30	3040	2820										
1764	DFN	4610	4620										
3314	DFORCE	22200	7640	8300	1640	5810	10120	21820	22240	22270	22280	22670	10090
2151	DFTYPE	2340	2350	8150	2120	2430	4420	4470	5020	5070	10060	10110	20810
			21360	22250									21160
2045	DHICOR	2140	2150	2870	2950	6320	6720	7060	7210	7410	7820		
7516	DINAND	16650	15490										
7542	DINCM2	16960	16910	16930									
7531	DINCOM	16850	14530	14590	14600	15650	15960						
2050	DINDIR	2170	2180	3350	15280	15390	15480	15860					
7471	DINDIV	16320	15590										
10316	DINDSN	22770	3160	15800									
7441	DININD	15860											
7522	DINIOR	16750	15470	16970									
7462	DINMIN	16200	15550										
7465	DINMUL	16250	15510										
7453	DINPC1	16010	15950	15970									
7457	DINPL2	16120	16620	16670	16720	16800							
7456	DINPLU	16100	3450	14280	15530	16170	16220						
7461	DINSPA	16160	14510	14700	15630	15940							
7432	DINULL	15750	14320										
7325	DINV30	14860	14610										
7405	DINV31	15470	15080										
7407	DINV32	15490	15100										
7411	DINV33	15510	15120										
7413	DINV34	15530	15140	15570									
7415	DINV35	15550	15160										
7421	DINV36	15590	15180										

CROSS REFERENCE TABLE

7423	DINV37	15610	15200										
7425	DINV38	15630	15220										
7430	DINV39	15670	15480	15500	15520	15540	15580	15600	15620	15640			
7342	DINV45	15050	15820	16130									
7375	DINV48	15370	15300										
7365	DINV49	15280	14790	15450									
7322	DINV61	14800	14680	14720									
7313	DINV62	14730	14520	14540	14990	16030	16880	16940					
3251	DINVAL	14250	1450	2730	2930	3050	3430	6180	6970	15330	15350		
7520	DINXOR	16700	15610										
445300	DK.	740	3310										
445320	DK0	780	3040										
127	DK1	4310											
156	DK2	4350											
37	DKCA	2750											
675	DKDON	4170	4180										
16000	DKLEN	2650	2660	8100	4570	2320							
34	DKLENB	2660											
266	DKLOK	3830	3840										
777601	DKMAX	650											
672	DKOVR	4160	4170										
2	DKRD	2760											
36	DKWC	2740											
4	DKWRT	2770											
2041	DLIMIT	2100	2110	1200	5900	11600	12170						
10321	DLNSGN	22800	2810										
3312	DLOCAT	21100	8240	6240	6940	7110	7760	15420	21580	21680			
2044	DLOCOR	2130	2140	2770	2860	2970	3080	3260	5890	5960	6030	6120	6620	6960
			7010	7080	7100	7200	7520						
2160	DMBMIN	2410	2420	20560	21490	21990							
10323	DMCSGN	22820	3540										
2166	DMFMIN	2470	2480	2040	2050	2330	4690	4940	5310	20770	21010	21270	21880
5575	DMOD10	3710	3550										
11531	DMSG50	10910	3590	9220	9280	9340	9410	9440	9550	9610	9720	9770	
11540	DMSG51	10940											
11557	DMSG52	10990											
11573	DMSG53	11040											
11624	DMSG54	11090											
11647	DMSG56	11150	5470	5740	8510	8780							
11660	DMSG57	11170											
11667	DMSG58	11210	5520	5560									
11704	DMSG59	11270											
3542	DMSG60	11300											
11736	DMSG61	11370											
11750	DMSG62	11400											
11762	DMSG63	11430											
12043	DMSG64	11470											
12676	DMSG65	11750											
3544	DMSG80	12030	12050										
13063	DMSG81	12080	1660	1850	2250								
13077	DMSG82	12110	1870										
13111	DMSG83	12140	2570	4820	5200	9910							

CROSS REFERENCE TABLE

CROSS REFERENCE TABLE

CROSS REFERENCE TABLE

1702	FRLEN	4420	4430						
1703	FRSTA	4430	4440						
2	FUDGE	3190	3200						
11185	GNAM90	2980	2620						
11100	GNAME6	2850	2800						
276	GREAT	2930	22870						
476257	GRO	870							
4	HDRLEN	580							
1700	IMPLEN	990							
3180	IMPSTR	2550	1500						
10	INDEX	490							
422020	INT	320							
7427	INV381	15650	15240						
513	IO.IN	3910	3920						
525	IO.OT	3920	3930						
300000	IOBLK	2830							
1760	IORS	4570	4580	1810	2020	5910			
1002	IOTO	4900	4910						
652	JMP	4110	4120						
100	JTLEN	960							
1700	JTSTRT	950	940	960	1000	4400			
16	KBLEN	3610	3630	3640	3680	3690	3730	3740	
10	KBNUM	3620	3670	3720					
76	LOLOK	3630							
107	L1BFR	3670	3680	3690					
127	L1BIN	3690	3700	3720	4290				
131	L1FLG	3700	3710						
125	L1LOK	3680							
133	L1NAM	3710							
136	L2BFR	3720	3730	3740					
156	L2BIN	3740	3750	4330					
160	L2FLG	3750	3760						
154	L2LOK	3730							
162	L2NAM	3760	3770						
422026	LDR	390	5450						
2000	LDRST	5040							
274	LESS	2920	22860						
212	LF	210	2440						
3205	M B5	7430	7240	7260	7280	7300	7320	7340	7490
13272	M C1	12650	12580						
4342	M C2	4820	4800						
4335	M C4	4810	4780						
5051	M M1	9250	5260						
4032	M ON	2820	9100						
4417	M BAS	5490	9080						
4602	M BYE	7180	9120	9120	9120				
4633	M CAF	7520	9140						
4114	M CAT	3590	9070						
4411	M DEB	5400	9050						
3172	M EOL	5110	2670	5150					
4721	M EXP	8400	9130	9130					
13246	M GRO	12450	9090						

CROSS REFERENCE TABLE

4414	M	LDR	5450	9060	9060		
4421	M	MX1	5600	5420			
4425	M	MX2	5650	5470			
4427	M	MX3	5680	5640			
4443	M	MX4	5840				
4481	M	MX5	5710	9990			
4113	M	NHE	3520	3710			
4034	M	OFF	2880	9110			
4110	M	ROK	3420	3350	3380		
4064	M	RON	3210	8820	8840	8860	8880	8900
6020	M	VAL	5420	9160	9160		
4730	M	XDU	8560	9170	9170		
4742	M	XPA	8680	9180	9180		
4640	M	ZER	7630	9150	9150		
2022	M	ACSW	1860	5880			
4132	M	CAT2	3680	5020			
4143	M	CAT4	3780	3740			
4215	M	CAT5	4150	4080			
4154	M	CAT6	3870	3830			
4223	M	CAT7	4230	4140			
4263	M	CAT8	4630	3780	4200		
4366	M	CAT9	4910				
2173	M	CMAX	3560	3910			
3170	M	CNAM	3490	3650	3860	4100	
2172	M	CNUM	3550	3890			
4122	M	CNXL	3620	4680	4890		
4636	M	CORE	7600	7690			
4637	M	DISK	7610	7670			
4472	M	DTON	6230	8920	9350		
11305	M	MSG1	10470	10210			
11321	M	MSG2	10500	10220			
11332	M	MSG3	10530	10240			
11344	M	MSG4	10560	10230			
11365	M	MSG5	10620	10250			
11377	M	MSG6	10650	10260			
11412	M	MSG7	10680	10270			
11466	M	MSG8	10770	10280			
11447	M	MSG9	10740	10290			
3176	M	MST2	6010	5990	6030		
3171	M	NHED	3500	3660	3960	4660	5010	
4076	M	ROFF	3320	8830	8850	8870	8390	8910
4074	M	RON1	3290	3240			
4472	M	TPON	8920	9360			
4650	M	ZCOR	7730	7700			
4653	M	ZDIS	7770	7680			
4795	M	ZER2	8190	7900	8020		
4796	M	ZER4	8220	8290			
4007	M,000,		2590				
4016	M,001,		2620				
4117	M,002,		3600				
4126	M,003,		3640				
4167	M,004,		3990				

CROSS REFERENCE TABLE

4300	M,005,	4730
4313	M,006,	4740
4332	M,007,	4790
4337	M,008,	4810
4353	M,009,	4880
11310	M,033,	10470
11324	M,034,	10500
11335	M,035,	10530
11347	M,036,	10560
11362	M,037,	10590
11370	M,038,	10620
11402	M,039,	10650
11415	M,040,	10680
11427	M,041,	10710
11452	M,042,	10740
11471	M,043,	10770
11511	M,044,	10800
11525	M,045,	10830
11534	M,046,	10910
11544	M,047,	10960
11563	M,048,	11010
11576	M,049,	11040
11630	M,050,	11110
11652	M,051,	11150
11662	M,052,	11170
11672	M,053,	11210
11707	M,054,	11270
11723	M,055,	11310
11741	M,056,	11370
11753	M,057,	11400
11765	M,058,	11430
12013	M,059,	11440
12047	M,060,	11490
12072	M,061,	11500
12117	M,062,	11510
12136	M,063,	11520
12155	M,064,	11530
12175	M,065,	11540
12221	M,066,	11550
12237	M,067,	11560
12257	M,068,	11570
12275	M,069,	11580
12312	M,070,	11590
12330	M,071,	11600
12360	M,072,	11610
12400	M,073,	11620
12431	M,074,	11630
12450	M,075,	11640
12475	M,076,	11650
12524	M,077,	11660
12542	M,078,	11670
12567	M,079,	11680

CROSS REFERENCE TABLE

CROSS REFERENCE TABLE

12641	M.080.	11690											
12633	M.081.	11700											
12652	M.082.	11710											
12702	M.083.	11770											
12716	M.084.	11780											
12732	M.085.	11790											
12752	M.086.	11800											
12764	M.087.	11810											
13002	M.088.	11820											
13016	M.089.	11830											
4064	MACSON	8880	9310										
4222	MCAT65	4210	4120										
2170	MCFREE	3540	4700										
4753	MCLIST	9040	2730										
4204	MCNEXT	4050	3980	4640									
4064	MCNTON	8900	9320										
4717	MCOMND	8380	8430										
3203	MDTNUM	7010	6250	6690	7100								
4536	MDTOF4	6670	6640										
3201	MDTOF6	6800	6740	6760	6840	6910							
4565	MDTOF7	6930	6870										
4546	MDTOF8	6750	6940										
4550	MDTOF9	6770											
4527	MDTOFF	6590	8930	9350									
4541	MDTON1	6420	6380										
3177	MDTON6	6470	6280	6300	6510	6570	6810	6820	6930				
4521	MDTON7	6520	6490										
4513	MDTON9	6440	6410	6530									
3314	MFORCE	10090	3610	5730	760	1110	1160	1250	1270				
10	MINBUF	3200	3610										
255	MINUS	330	10830	15150									
4026	MMONX2	2730	3190	3330									
4023	MMONXT	2670	2690	3300	3440	4950	6450	6780	7540	7750	7780	8650	8770	5610
			10130	11450	11730	11850							
11424	MMSG10	10710	10300										
11506	MMSG11	10800	10310										
11522	MMSG12	10830	10320										
11531	MMSG50	10910	2750	2950	7660	7710	8580	8700	8730	12520			
11540	MMSG51	10940	3260	6560									
11557	MMSG52	10990	3410	6900									
11573	MMSG53	11040											
11624	MMSG54	11090	6350										
11647	MMSG56	11150	3700	7120									
11660	MMSG57	11170	10920	10970	11020	11120	11280	11380	11410				
11667	MMSG58	11210											
11704	MMSG59	11270											
3542	MMSG60	11300	11330										
11736	MMSG61	11370											
11750	MMSG62	11400											
11762	MMSG63	11430	8420	8470									
12043	MMSG64	11470	8440										
12676	MMSG65	11750	8460										

CROSS REFERENCE TABLE

```
  5345  MNEXTL   1260  1850  3620
  4372  MNHEAD   5000  3720
  4004  MONMSG   2590  7360  10200  10420  11070
  4012  MONSYM   2600  4940  5140  10060  10370  11250  11350
422023  MP1       350
422024  MP2       360
  2032  MPACSW   1980
  1004  MPOPR    4920
  1000  MPST     4880  4890
  4064  MPTPON   8860  9300
  4064  MPTRON   8840  9290
  1754  MQ       4530  4540
  2016  MQSAVE   1820  1830  1760  6050  8330  8830  9290
  4720  MRESRC   8390  8450
  4047  MRSRC5   3040  2980
  4061  MRSRC7   3160  3110
  5052  MRSRCS   9290  2960
  4064  MSCOON   8820  9330
 11531  MSG50    10910
 11540  MSG51    10940
 11557  MSG52    10990
 11573  MSG53    11040  2460
 11624  MSG54    11090
 11647  MSG56    11150
 11660  MSG57    11170
 11667  MSG58    11210
 11704  MSG59    11270
  3542  MSG60    11300
 11736  MSG61    11370
 11750  MSG62    11400
 11762  MSG63    11430
 12043  MSG64    11470
 12676  MSG65    11750
 13031  MSGLOC   11910  12060
777763  MSGMAX   10340  2440
 11275  MSGOCT   10360  11190
 11260  MSGPTR   10200  2510  10340
 13042  MSGWRD   11980  12090  12120  12150  12180  12210  12240  12270  12300  12330  12360  12390
  4491  MSRCH2   5250  5300
  2000  MTEMP0   1630  3000  3060  3170  3220  3280  3360  3420  5410  5460  5500  5700  5710
                       5800  6390  6420  6520  6700  6720  6860  7020  7070  7440  7450  7480
                      10390
  2001  MTEMP1   1640  2900  3010  3130  4090  4130  4220  5690  5780  6550  6890  7050  8590
                       8620  8710  8750  10570
  2002  MTEMP2   1650  2610  5820  11050
  2003  MTEMP3   1660
  2004  MTEMP4   1670
  2005  MTEMP5   1680
  2006  MTEMP6   1690
  2007  MTEMP7   1700
  2010  MTEMP8   1710
  2011  MTEMP9   1720
```

CROSS REFERENCE TABLE

CROSS REFERENCE TABLE

4076	PTPOFF	8870	9300							
606462	PTR	700	3700							
4076	PTROFF	8850	9290							
1	PURCOD	110	5140	5270	430	610				
12100	PURLEN	1010								
1775	PURNM	4700	4710							
3700	PURSTR	2560	990	1010	2560	12700				
546	PUTIN	3940	3950							
34	RACS	3440	7230	9310						
6	RCNT	3390	9320							
35	RCORE	3450								
1003	RDBLK	4910	4920							
32	RDT	8950	9350							
32	RDT0	3420	6270	6360	6400	6730	7310	8950	8960	
33	RDT1	3430	6290	6430	6750	7330				
17505	RECOV	470								
5110	REGLIS	9420	3740							
422021	RES	330								
40	RESCAT	3470	3480							
1000	RESLEN	920								
234	RFLAG	3790	3800							
230	RPTP	3780	3790	7250	9300					
235	RPTR	3800	3810	7270	9290					
242	RSCO	3820	3830	7290	9330					
1776	RSTRT	4710	1680	1790	2000					
32	RTP	8960	9360							
1755	SC	4540	4550							
273	SCOLON	380	22830							
4076	SCOOFF	8830	9330							
640000	SCRSTR	2670								
2021	SCSAVE	1850	1860	1780	5950	8460	9360			
3134	SEARCH	5230	2740	2970	3100	5270	5310	5320	1680	3750
243	SHARP	2890	18130	18340	22790					
257	SLASH	360	15170	2790						
240	SPACE	250	15210	22510	22800	2690				
377	SPCOD	5410								
422122	SPL	430								
1000	SPLST	4960								
777400	SPMSK	5390								
252	STAR	300	15110							
3701	START	1640	12720							
3731	STRT1	1930	1720							
3747	STRT2	2130	1880							
2020	STSAVE	1840	1850	1820	5900	7530	8190	8950	9560	
335	SWAP	3880	3890	6160						
336	SWAP1	3890	3900							
340	SWAP3	3900	3910							
1000	SWCAT	4750	4760							
1003	SWCLK	4780	4790							
1004	SWERR	4790	4800							
1007	SWMP1	4820	4830							
1010	SWMP2	4830	4840							

CROSS REFERENCE TABLE

1002	SWMTR	4770	4780										
1011	SWOPR	4840											
422022	SWP	340											
1001	SWPPR	4760	4770	6150									
40	SWPS	3460	3470										
1005	SWSPL	4800	4810										
1006	SXSPL	4810	4820										
7647	SYMB11	18170	18100										
7660	SYMB13	18270	18140										
7655	SYMB14	18240	18280										
7675	SYMB21	18450	18410										
7574	SYMB25	17590	18430										
7564	SYMB31	17420											
7626	SYMB71	17960	18000										
3260	SYMBOL	17160	14310	17370	18060	18150	18210						
1300	SYSBAS	2800	2810	1950	840								
41300	SYSDA	2810	4630										
17735	SYSDEV	530											
1777	SYSMAX	2820	4730										
13272	T C1	12650											
3452	T T1	560	1280	1400	1710	1810	1990	2050	2080	2090	2110	2120	2130	2290
			2460											
3453	T T2	570	1180	1260	1480									
13246	T GRO	12450												
10342	T IN1	780	930	1020	1050									
10350	T IN4	840												
10336	T INL	740	990	1010										
120	T STD	480	490											
3450	T BEND	540	860											
3451	T BPTR	550	14740	14880	17200	18240	18250	750	850	880	900	950	1000	1040
			2590	2600	2760	2770								
3454	T CHAR	580	2610											
3505	T CRLF	3320	3630	3730	3880	4000	4010	4590	4650	4690	4870	8600	4150	6110
			6460	7680	7890	980	3410	11480	11720	11760	11840			
10611	T DLMR	3130	2980	3070										
3475	T FGET	2580	17380	17510	17670	17880	18190	18570	1210	1530	2200	2620	2680	
10475	T SIX2	2030	2140											
3467	T SIX5	2190	2030	2060	2100	2250								
10511	T SIX9	2160	2220											
11534	T,046,	10910												
11544	T,047,	10960												
11563	T,048,	11010												
11576	T,049,	11040												
11630	T,050,	11110												
11652	T,051,	11150												
11662	T,052,	11170												
11672	T,053,	11210												
11707	T,054,	11270												
11723	T,055,	11310												
11741	T,056,	11370												
11753	T,057,	11400												
11765	T,058,	11430												

CROSS REFERENCE TABLE

```
12013  T,059,    11440
12047  T,060,    11490
12072  T,061,    11500
12117  T,062,    11510
12136  T,063,    11520
12155  T,064,    11530
12175  T,065,    11540
12221  T,066,    11550
12237  T,067,    11560
12257  T,068,    11570
12275  T,069,    11580
12312  T,070,    11590
12330  T,071,    11600
12360  T,072,    11610
12400  T,073,    11620
12431  T,074,    11630
12450  T,075,    11640
12475  T,076,    11650
12524  T,077,    11660
12542  T,078,    11670
12567  T,079,    11680
12611  T,080,    11690
12633  T,081,    11700
12652  T,082,    11710
12702  T,083,    11770
12716  T,084,    11780
12732  T,085,    11790
12752  T,086,    11800
12764  T,087,    11810
13002  T,088,    11820
13016  T,089,    11830
10367  T1CHAR     1000     810
10365  T1LINE      980     830
  100  TABLEN     2630    2640
17500  TAPIN       450
17502  TAPOT       460
 3330  TBUFFR      490     740     940
 3501  TCHRID     2930   17390   17520   17680   17890   18280   18580    1220    1540    2210    2710    3150    3170
                         3180    3200
 3456  TCOUNT      600   14760   14900   17220     760    2650   11180   12000
10614  TDIGIT     3170    3010    3040
 3455  TDLMTR      590    3800    4920    5120    1360    1480    1740    2520    2780    3150    3530    4160    6500
                         7150   14780   14920   15060   15790   16010   17240     770    3140    2630    2780    2990
 2000  TEMP0      1630    1640    2150
 2001  TEMP1      1640    1650    2170    2340
 2012  TEMP10     1730    1740   17310   17850   18020   18450   18490
 2013  TEMP11     1740    1750    7850    7990    8110    8280
 2014  TEMP12     1750    1800    7870    8010    8120    8230    8270    6210    6270
 2002  TEMP2      1650    1660    2190    2410    2430    2500    2520    2530
 2003  TEMP3      1660    1670
 2004  TEMP4      1670    1680
 2005  TEMP5      1680    1690
```

CROSS REFERENCE TABLE

2006	TEMP6	1690	1700										
2007	TEMP7	1700	1710										
2010	TEMP8	1710	1720										
2011	TEMP9	1720	1730	1690	1730	1940							
10550	TFGET9	2620											
3457	TINLIN	720	2630	3670	1300	5450	960	1430					
3477	TINTIN	2640	17320	1190	2000	2720	2740	2780					
10615	TLETTR	3180	3110										
11531	TMSG50	10910											
11540	TMSG51	10940											
11557	TMSG52	10990											
11573	TMSG53	11040											
11624	TMSG54	11090											
11647	TMSG56	11150											
11660	TMSG57	11170											
11667	TMSG58	11210											
11704	TMSG59	11270											
3542	TMSG60	11300											
11736	TMSG61	11370											
11750	TMSG62	11400											
11762	TMSG63	11430											
12043	TMSG64	11470											
12676	TMSG65	11750											
10400	TNUM20	1210	1430										
10424	TNUM26	1460	1230										
10433	TNUM27	1530	1470										
10441	TNUM28	1590	1550										
10442	TNUM29	1600	1520										
3461	TNUMIN	1170	8570	8690	8720	4810	5190	9210	9270	9330	9400	9540	9600	9710
			9900	14930	1200	1240	1500	1560	1580	1610	1620	2850		
10445	TOCT42	1670												
10455	TOCT44	1750	1820											
3463	TOCTOT	1660	4450	4520	4580	4720	8640	8930	9030	10220	11670	12040	13520	1850
			10360	10410	10580	11060	11340	11950						
646000	TP.	730	3270											
376	TRCOFF	5540	2390											
375	TRCON	5530	2370											
10522	TSIX24	2300	2360											
3473	TSIX26	2380	2330	2340	2350	2470								
3465	TSIXIN	1970	2680	2940	3690	3840	7650	8410	1650	2550	3720	4210	5460	2010
			2020	2170	1440	2610	2760	3220						
3471	TSIXOT	2280	2590	2620	3600	3640	3990	4730	4740	4790	4810	4880	1040	1290
			5440	8180	8250	8320	8380	8450	8520	8610	8690	8790	8820	8850
			8890	8940	8970	2300	2320	2500	1390	1420	1840	3610	4040	4320
			10470	10500	10530	10560	10590	10620	10650	10680	10710	10740	10770	10800
			10830	10910	10960	11010	11040	11110	11150	11170	11210	11270	11310	11370
			11400	11430	11440	11490	11500	11510	11520	11530	11540	11550	11560	11570
			11580	11590	11600	11610	11620	11630	11640	11650	11660	11670	11680	11690
			11700	11710	11770	11780	11790	11800	11810	11820	11830	11930	11990	12040
			12080	12110	12140	12170	12200	12230	12260	12290	12320	12350	12380	
2000	TTEMP0	1630												
2001	TTEMP1	1640												

CROSS REFERENCE TABLE

UNDEFINED SYMBOLS

#1	5630											
#2	5640											
#3	5650											
#4	5660											
#5	5680											
BAD	1560											
D #1	130											
DEFINS	100	100	120	120	580	580	2680	2680	3210	3210	5330	5330
DERROR	22130	22350										
LINE	1210											
MESS	1190	1200										
OCTZ	1790											

MACRO CROSS REFERENCE TABLE

CHAR	1380												
CHRQT	1340												
CMDERR	1250	10920	10970	11020	11120	11280	11380	11410					
COMMAN	640	9050	9060	9070	9080	9090	9100	9110	9120	9130	9140	9150	9160
		9170	9180	9200	9210	9220	9230	9550	9560	9570	9580	9590	9600
		9610	9620	9630	9640	9650	9660	9670	9680	9690	9860	9870	9880
		9890	9900	9910									
COUNT	1460	11180	12000										
CRLF	1300	3630	3730	3880	4000	4010	4590	4650	4690	4870	8600	4150	6110
		6460	7680	7890	11480	11720	11760	11840					
DELIM	1420	3800	4920	5120	1360	1480	1740	2520	2780	3150	3530	4160	6500
		7150	15060	15790	16010	17240	2630	2780	2990				
DMODE	780	9740	9750	9760	9770	9780	9790	9800	9810				
EMESS	1750												
ENTER	330	5110	5230	6470	6800	7010	7430	7830	7950	8070	5680	6820	10210
		10260	10350	10430	10530	10610	10670	10770	11390	12320	12460	13560	14250
		16390	17160	18310	18550	18720	18790	18900	19170	19230	19400	19840	20510
		20630	20890	21100	22200	22410	22480	22560	22660	720	1170	1660	1970
		2190	2280	2380	2580	2640	2930	3240	3320	740	1340	1680	2590
		3210	3670	3860	4150	11380	12030						
FORMAT	1170	3700	7120	5470	5740	8510	8780	2830	2860	3230			
LINE	1100	2630	3670	1300	5450	1430							
LOOP	960												
MESS	1620	2590	2620	3600	3640	3990	4740	4880	1040	1290	8180	8250	8320
		8380	8450	8520	8610	8690	8790	8820	8850	8890	8940	8970	1390
		1420	1840	3610	4040	4320	10470	10500	10530	10560	10620	10650	10680
		10710	10740	10770	10800	10830	10910	10960	11010	11040	11110	11150	11210
		11270	11310	11370	11400	11430	11440	11490	11770	11780	11790	11800	11810
		11820	11830	12040	12080	12110	12140	12170	12200	12230	12260	12290	12320
		12350	12380										
MESSR	1520	2590	2620	3600	3640	3990	4740	4880	1040	1290	8180	8250	8320
		8380	8450	8520	8610	8690	8790	8820	8850	8890	8940	8970	1390
		1420	1840	3610	4040	4320	10470	10500	10530	10560	10620	10650	10680
		10710	10740	10770	10800	10830	10910	10960	11010	11040	11110	11150	11210
		11270	11310	11370	11400	11430	11440	11490	11500	11510	11520	11530	11540
		11550	11560	11570	11580	11590	11600	11610	11620	11630	11640	11650	11660
		11670	11680	11690	11700	11710	11770	11780	11790	11800	11810	11820	11830
		12040	12080	12110	12140	12170	12200	12230	12260	12290	12320	12350	12380
MODSET	3820	3930	3940	3950	3960	3970	3980	3990	4000				
MPOFF	5430	1650	2640	3210	3320	5620	5660	5740	6240	6600	7220	8610	8740
		5570	5690	8530	9970	20640	20900	1350	2150	2360	3470	11220	12450
NEG	1010	2960	11590	16210	19530	19570							
NMESS	1670	4730	4790	4810	5440	10590	11170	11930	11990				
NUM	1260	8570	8690	8720	4810	5190	9210	9270	9330	9400	9540	9600	9710
		9900	14930	2850									
OCT	1890	4450	4520	4580	8640	10220	12040	10580					
OCTZ	1840	4720	8930	9030	11670	13520	10360	10410	11060	11340	11950		
OP	110	160	190	200	210	220	230	240	250	260	270	280	290
		300	330	340	350	360	370	380	390	400	410	420	430
		440	450	460	470	480	490	500	510	520	530	540	550
		560	570	580	590	600	610	620	630	640	650	660	670
		680	690	700	710	720	730	740	750	760	770	780	790

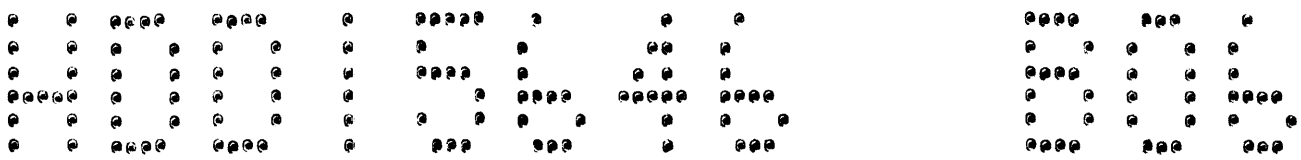
MACRO CROSS REFERENCE TABLE

		800	810	820	830	840	850	860	870	880	890	900	910
		920	930	940	950	960	970	980	990	1000	1010	1020	1030
		1040	1050	1060	1070	1080	1090	1100	1110	1120	1130	1140	1150
		1160	1170	1180	1190	1200	1210	1220	1230	1240	1250	1260	1270
		1280	1290	1300	1310	1320	1330	1340	1350	1360	1370	1400	1410
		1420	1430	1440	1450	1460	1470	1480	1490	1500	1510	1520	1530
		1540	1550	1560	1570	1580	1590	1600	1610	1620	1630	1640	1650
		1660	1670	1680	1690	1700	1710	1720	1730	1740	1750	1760	1770
		1780	1790	1800	1810	1820	1830	1840	1850	1860	1870	1880	1890
		1900	1910	1920	1930	1940	1950	1960	1970	1980	1990	2000	2010
		2020	2030	2040	2050	2060	2070	2080	2090	2100	2110	2120	2130
		2140											
REGIST	520	9420	9430	9440	9450	9460	9470	9480	9490	9500			
RESOUR	1000	9290	9300	9310	9320	9330	9340	9360					
START	1100												
SWAP	5610												
WHAT	1090	2750	2950	7660	7710	8580	8700	8730	9220	9280	9340	9410	9440
		9550	9610	9720	9770	12520							
WORD	1140	2680	2940	3690	3840	7650	8410	1650	2550	3720	4210	5460	1440
		2610	2760	3220									
WORD1	1180	3050	3760	2230	1790	1930	2720	2810	3370	3980			
WORD2	1220	5530											

USE CROSS REFERENCE TABLE

```
    0   .!.!.!
    0   .REL.
    0   IMPURE      1490    3460   5110   5230   6000   6470   6800   7010   7430   7830   7950   8070   5680
                        6820  10210  10260  10350  10480  10530  10610  10670  10770  11390  12320  12460
                       13560  14250  16390  16520  17160  18310  18550  18720  18790  18900  19170  19230
                       19400  19840  20510  20630  20890  21100  22200  22410  22480  22560  22660  10010
                         440    720   1170   1660   1970   2190   2280   2380   2580   2640   2930   3240
                        3320    540    740   1340   1510   1680   2070   2280   2590   2940   3210   3670
                        3860   4150  11300  12030  12680
 3700   PURE         1510    3510   3580   5040   5110   5230   5340   6020   6180   6470   6800   7010   7140

                        7430   7830   7950   8070   9020    980   1570   2590   5360   5680   6820   8040
                        9100  10210  10260  10350  10480  10530  10610  10670  10770  11390  12320  12460
                       13560  14250  16390  16560  17160  18310  18550  18720  18790  18900  19170  19230
                       19400  19840  20510  20630  20890  21100  22200  22410  22480  22560  22660    620
                         720   1170   1660   1970   2190   2280   2380   2580   2640   2930   3240   3320
                         570    740   1340   1530   1680   2130   2340   2590   2960   3210   3670   3860
                        4150  10120  11300  12030  12710
13450   SYMTAB       9970
```



05/31/72 02:01:17

```
**********************************
**********************************
**                              **
**  PDP-9 MINI TIME-SHARING SYSTEM  **
**       MTSS SYSTEM LOADER     **
**             DTSS:LDR         **
**             MTSS:B06         **
**                              **
**********************************
**********************************
```

LOADER INITIALIZATION

```
100             .STITL  LOADER INITIALIZATION
110             .NAME   LDR--806
120             .TITLE  GROWTH SYSTEM LOADER
130
140
150     *       REVISED 20 JAN 1971 BY ROBERT W. BLEAN
160
170             .INSRT  DEFINS
100             .IFUND  DEFINS
```

LOADER INITIALIZATION

```
5720            .LIST   ON
5730            .END
180             .HEAD   M
```

```
                M                       LOADER INITIALIZATION

                        190             ,STITL  LOADER INITIALIZATION
                        200     ENTER   .DEFIN
                        210             ,PMC    SAVE,OFF
                        220             ,CRSM   SAVE,ON
                        230             ,PMC    ON
                        240             ,USE    IMPURE
                        250     #1      HLT
                        260             JMP     #2
                        270             ,USE    PREVIOUS
                        280     #2      ,EQU    .
                        290             ,PMC    OFF
                        300             ,CRSM   RESTORE
                        310             ,PMC    RESTORE
                        320             ,ENDM
                        330
                        340     PREAD   ,OPDEF  705003
                        350     PWRITE  ,OPDEF  705005
000001                  360     PURCOD  ,EQU    1
                        370             ,HEAD
000001                  380     DEBUG   ,EQU    1
                        390             ,HEAD   M
014000                  400     BASE    ,EQU    14000           LOADER STARTING LOCATION
002170                  410     BUF     ,EQU    BUFFER
002170                  420     TBUF    ,EQU    BUFFER
001000                  430     BMAX    ,EQU    1000
001000                  440     TBFL    ,EQU    1000
000012                  450     CMDX    ,EQU    12
                        460             ,HEAD
003714                  470     NEXTL   ,EQU    MSNEXTL
                        480             ,HEAD   M
003714                  490     MONXT   ,EQU    NEXTL
                        500     *
                        510     *       ARRANGE THE USE COUNTERS IN ORDER
                        520     *
003170                  530     IMPSTR  ,EQU    3170            START OF THE IMPURE CODE
003700                  540     PURSTR  ,EQU    3700            START OF THE PURE CODE
000000                  550             ,USE    IMPURE
003170                  560             ,LOC    IMPSTR
003700                  570             ,USE    PURE
                        580     *
                        590     *       CHECK FOR OVERLENGTH PURE CODE
                        600     *
                        610             ,IFG    CHECK,3600
003700  602026          630             ,ACI6   *P06*           SET THE NAME OF THE PURE CODE PORTION OF THIS PROGRAM
                        640
                        650             ,HEAD
                        660             ,INSRT  :DLIBRARY:PDP9LIB:LIBMACRO
                        100             ,INE    DEBUG,1
                        120             ,IFE    $DEBUG          AVOID FORM-FEED UNLESS LISTING IS BEING PRINTED
                        140
                        150
                        160     *
```

LOADER INITIALIZATION

```
170      *       THESE MACROS ARE FOR USE WITH THE PROGRAM PDP9LIB***;TTY-NON
180      *       TTY-NON IS A NON-INTERRUPT DRIVEN TELETYPE HANDLER FOR THE CONSOLE
190      *       TELETYPE ON THE PDP-9.
200      *
210      *       LINE INPUT MACRO IS:
220      *
230      *            LINE  -- GETS THE NEXT LINE FROM THE TELETYPE, PACKS IT IN THE
240      *                     INCLUDED LINE BUFFER, AND RETURNS TO THE USER. USE BACK-ARROW
250      *                     FOR CHARACTER DELETION AND CONTROL X FOR LINE DELETION.
260      *                     THE ROUTINE PROTECTS AGAINST BUFFER UNDERFLOW OR OVERFLOW.
270      *
280      *       WORD INPUT MACROS ALL DELETE LEADING BLANKS, RETURNING TO THE USER
290      *       AT +1 WITH THE DELIMITER IN THE AC IF A DELIMITER IS THE FIRST NON-
300      *       BLANK CHARACTER, THEY ALL UTILIZE WORDB AND WORDB+1 FOR STORAGE, AND
310      *       ANY VALUE ACCUMULATED THERE REMAINS UNTIL THE NEXT TIME A WORD-PACKING
320      *       MACRO IS USED ('WORD' OR 'NUM'). THE DELIMITER THAT ENDED THE WORD
330      *       IS STORED IN DLMTR UNTIL THE NEXT TIME A WORD PACKING MACRO IS USED
340      *       OR UNTIL THE USER PROGRAM USES THE ROUTINE 'CHRID'.
350      *       THE AVAILABLE MACROS ARE:
360      *
370      *            WORD  -- PACKS CHARACTERS, IN A LEFT-JUSTIFIED SIXBIT PACK,
380      *                     INTO WORDB, WORDB+1, .... RETURNS THE FIRST THREE (OR
390      *                     FEWER) CHARACTERS LEFT JUSTIFIED IN THE AC.
400      *
410      *            NUM   -- GETS A NUMBER, AND RETURNS IT IN THE AC. A FORMAT ERROR
420      *                     IS CAUSED BY A LETTER BEING FOUND OR BY A DECIMAL DIGIT
430      *                     (8 OR 9) BEING FOUND WITHOUT A TRAILING DECIMAL POINT.
440      *                     THAT THE DECIMAL VALUE IS DESIRED IS SIGNALLED BY THE
450      *                     DELIMITER BEING A PERIOD, OTHERWISE THE OCTAL VALUE IS
460      *                     RETURNED. THE VALUE RETURNED REMAINS AVAILABLE IN WORDB.
470      *                     THIS IS THE VALUE FOUND MOD 2^18 -- I.E. OVERFLOW IS LOST.
480      *
490      *                     RETURN IS:
500      *                          +1 WITH LINK = 0 FOR A FORMAT ERROR
510      *                          +1 WITH LINK = 1 FOR THE FIRST NON-BLANK CHARACTER A DELIMITER
520      *                          +2 FOR SUCCESS
530      *
540      *            WORD1 -- GETS THE CONTENTS FROM WORDB. THIS IS THE FIRST THREE
550      *                     SIXBIT CHARACTERS OR THE VALUE.
560      *            WORD2 -- GETS THE CONTENTS OF WORDB+1. THIS IS THE SECOND THREE
570      *                     SIXBIT CHARACTERS OR THE "DECIMAL" VALUE. NOTE THAT THE
580      *                     "DECIMAL" VALUE WILL BE GARBAGE IF AN OCTAL NUMBER WAS INPUT.
590      *
600      *                     IN THE CASE OF SIXBIT INPUT, FURTHER INPUT WILL BE LOST.
610      *
620      *            COUNT -- GETS THE OCTAL COUNT OF THE NUMBER OF TIMES 'WORD' AND
630      *                     'NUM' HAVE BEEN CALLED SINCE THE LINE WAS INPUT. THIS
640      *                     IS THE COUNT OF THE NUMBER OF WORDS EXTRACTED SO FAR
650      *                     FROM THE CURRENT LINE BUFFER.
660      *
670      *            DELIM -- GETS THE LAST DELIMITER SEEN BY 'CHRID'. THIS WILL BE
680      *                     THE DELIMITER THAT ENDED THE LAST WORD FETCHED UNLESS
```

LOADER INITIALIZATION

```
690     *                    THE USER PROGRAM IS ACCESSING 'CHRID' ITSELF.
700     *
710     *       MISCELLANEOUS CHARACTER-ORIENTED MACROS:
720     *
730     *            CHAR  -- GETS THE OLDEST REMAINING CHARACTER FROM THE LINE BUFFER.
740     *                     THIS PERMITS THE USER PROGRAM TO EXAMINE THE ENTIRE INPUT
750     *                     STRING, WHICH IS A HARD THING TO DO USING 'WORD'.
760     *                  RETURNS +1 WITH THE CHARACTER IN THE AC
770     *
780     *            CRLF  -- PRINTS A CARRIAGE RETURN AND LINE FEED. IT DISTURBS NO
790     *                     STORAGE OR POINTERS.
800     *
810     *            CHROT -- PRINTS THE SINGLE ASCII CHARACTER IN THE AC.
820     *
830     *
840     *       OUTPUT MACROS ARE:
850     *
860     *            OCT   -- OUTPUTS AS SIX DIGIT OCTAL THE CONTENTS OF THE AC.
870     *
880     *            OCTZ  -- OUTPUTS AS OCTAL WITH LEADING ZEROES SUPPRESSED THE CONTENTS OF THE AC.
890     *
900     *            MESS <TEXT>,<CHARACTER COUNT> USES SIXBIT FORMAT TO OUTPUT THE
910     *                     CARRIAGE RETURN AND LINE FEED, FOLLOWED BY THE TEXT. IT
920     *                     FIRST DOES A 'KRB' INSTRUCTION TO CLEAR ANY PRINT-INHIBIT.
930     *
940     *            MESSR <TEXT>,<CHARACTER COUNT> IS THE SAME AS 'MESS', BUT NO
950     *                     'KRB' IS SUPPLIED. THIS PERMITS CONTINUATION OF A SINGLE
960     *                     MESSAGE.
970     *
980     *            NMESS <TEXT>,<CHARACTER COUNT> IS THE SAME AS 'MESSR' EXCEPT
990     *                     NO CARRIAGE RETURN NOR LINE FEED IS SUPPLIED. THIS PERMITS
1000    *                     CONTINUING THE MESSAGE ON THE SAME LINE.
1010    *
1020    *       HITTING ANY KEY ON THE TELETYPE DURING OUTPUT WILL INHIBIT THE ACTUAL
1030    *       PRINTING OF THE REST OF THE MESSAGE UNTIL THE NEXT 'MESS' OR KRB
1040    *       INSTRUCTION, NOTE THAT EXCEPT THE CHARACTER IS NOT PRINTED, THE REST
1050    *       OF THE PROGRAM CARRIES ON AS USUAL.
1060    *
1070
1080
1090
1100    LINE    ,DEFIN
1110            JMS     T$INLIN
1120            ,ENDM

1130
1140    WORD    ,DEFIN
1150            JMS     T$SIXIN
1160            ,ENDM
1170
1180    WORD1   ,DEFIN
1190            LAC     T$WORDB
1200            ,ENDM
```

LOADER INITIALIZATION

```
1210
1220    WORD2   .DEFIN
1230            LAC     TSWORDB+1
1240            .ENDM
1250
1260    NUM     .DEFIN
1270            JMS     TSNUMIN
1280            .ENDM
1290
1300    CRLF    .DEFIN
1310            JMS     TSCRLF
1320            .ENDM
1330
1340    CHROT   .DEFIN
1350            JMS     TSTTYOT
1360            .ENDM
1370
1380    CHAR    .DEFIN
1390            JMS     TSFGET
1400            .ENDM
1410
1420    DELIM   .DEFIN
1430            LAC     TSDLMTR
1440            .ENDM
1450
1460    COUNT   .DEFIN
1470            LAC     TSCOUNT
1480            .ENDM
1490
1500
1510
1520    MESSR   .DEFIN
1530            .CRSM   SAVE,ON
1540            LAW     -#2-2
1550            JMS     T$SIXOT
1560            .PMC    SAVE,OFF
1570    #5      .AC16   '!!#1'
1580            .PMC    RESTORE
1590            .CRSM   RESTORE
1600            .ENDM
1610
1620    MESS    .DEFIN
1630            KRB
1640            MESSR   <#1>,#2
1650            .ENDM
1660
1670    NMESS   .DEFIN
1680            .CRSM   SAVE,ON
1690            LAW     -#2
1700            JMS     T$SIXOT
1710    #5      .AC16   '#1'
1720            .CRSM   RESTORE
```

GROWTH SYSTEM STANDARD DEFINITIONS

```
                130             .STITL  GROWTH SYSTEM STANDARD DEFINITIONS
                140
                150     *       PROGRAMMED BY ROBERT W. BLEAN
                160
                170     *       LATEST REVISION 20 JAN 1971
                180
                190     *       ASCII CHARACTERS
                200
000212          210     LF      .EQU    212
000215          220     CR      .EQU    215
000230          230     CONTX   .EQU    230
000337          240     BKARR   .EQU    337
000240          250     SPACE   .EQU    240
000241          260     EXCLAM  .EQU    241             EXCLAMATION POINT
000243          270     NUMSGN  .EQU    243
000244          280     DOLLAR  .EQU    244             $
000246          290     AMPRSN  .EQU    246             &
000252          300     STAR    .EQU    252             ASTERISK (*)
000253          310     PLUS    .EQU    253
000254          320     COMMA   .EQU    254
000255          330     MINUS   .EQU    255
000256          340     PERIOD  .EQU    256             .
000256          350     POINT   .EQU    PERIOD
000257          360     SLASH   .EQU    257
000272          370     COLON   .EQU    272
000273          380     SCOLON  .EQU    273
000334          390     BSLASH  .EQU    334             BACK SLASH (\)
                400
                410     *       CONSTANTS
                420
017777          430     ADRSS   .EQU    17777           ADRESS FIELD MASK
002000          440     BOUNDA  .EQU    2000            TSS USER CORE START
017500          450     TAPIN   .EQU    17500
017502          460     TAPOT   .EQU    17502
017505          470     RECOV   .EQU    17505
017777          480     VFLAG   .EQU    17777
000010          490     INDEX   .EQU    10              GENERAL PURPOSE AUTO-INDEX REGISTER
000011          500     CATX    .EQU    11              CATALOG ROUTINES' AUTO-INDEX REGISTER
000012          510     CMDX    .EQU    12
017740          520     BOOT    .EQU    17740           BOOTSTRAP LOADER STARTING ADDRESS
017735          530     SYSDEV  .EQU    BOOT-3          HOLDS DEVICE ADDRESS OF CATALOG BLOCK ON THE SYSTEM DEVICE
017000          540     CATLOG  .EQU    17000           START OF THE RESIDENT CATALOG BLOCK
000001          550     CATBLK  .EQU    1               CATALOG IS AT LOGICAL BLOCK 1 OF ANY DEVICE

000400          560     CATLEN  .EQU    400             CATALOG LENGTH IS 400 WORDS MAXIMUM
000005          570     FCBLEN  .EQU    5               FILE CONTROL BLOCK IS FIVE WORDS LONG
000004          580     HDRLEN  .EQU    4               CATALOG HEADER IS FOUR WORDS LONG
017005          590     CPARAM  .EQU    CATLOG+5        POINTER TO PARAMETERS FOR CATALOG READ/WRITE
740000          600     DVCMSK  .EQU    740000          MASK TO EXTRACT HANDLER NUMBER AND TYPE FROM DEVICE ADDRESS
001777          610     BLKMSK  .EQU    1777            MASK TO RETRIEVE DEVICE BLOCK NUMBER
777716          620     CATMAX  .EQU    -50.            MAXIMUM NUMBER OF FILE CONTROL BLOCKS IN A CATALOG
000400          630     BLKLEN  .EQU    400             NUMBER OF WORDS IN ONE LOGICAL BLOCK
776701          640     DTMAX   .EQU    -1077           MAXIMUM NUMBER OF USABLE BLOCKS ON A DECTAPE
```

GROWTH SYSTEM STANDARD DEFINITIONS

```
777601          650     DKMAX   .EQU    -177            MAXIMUM NUMBER OF USABLE BLOCKS ON A LOGICAL DISK
                660
                670     *       DEVICE NAMES
                680
606064          690     PPT     .EQU    606064
606462          700     PTR     .EQU    606462
606460          710     PTP     .EQU    606460
446400          720     DT.     .EQU    446400
646000          730     TP.     .EQU    646000
445300          740     DK.     .EQU    445300
004464          750     .DT     .EQU    004464
006460          760     .TP     .EQU    006460
004453          770     .DK     .EQU    004453
445320          780     DK0     .EQU    445320
                790
                800     *       FILENAMES
436454          810     CTL     .EQU    436454          CATALOG BLOCK
                820
                830     *       FORMATS
                840
414263          850     ABS     .EQU    414263          LOADSTRING BINARY
425156          860     BIN     .EQU    425156          BINARY
476257          870     GRO     .EQU    476257          GROWTH SYSTEM FORMAT (CORE IMAGE)
435762          880     COR     .EQU    435762          CORE
                890
                900     *       MACROS
                910
                920     ENTER   .DEFIN
                930     #1      XX
                940             .ENDM
                950
                960     LOOP    .DEFIN
                970             ISZ     #1
                980             JMP     #2
                990             .ENDM
                1000
                1010    NEG     .DEFIN
                1020            CMA
                1030            TAD     (1              )
                1040            .ENDM
                1050
                1060    FORMAT  .DEFIN
                1070            JMP     FORMAT
                1080            .ENDM
                1090
                1100    START   .DEFIN                  STANDARD INITIALIZATION MACRO FOR THE GROWTH SYSTEM
                1110            .PMC    SAVE,ON         PRINT THIS ONE MACRO, AT LEAST
                1120            CAF
                1130            IOF!CLOF
                1140            LAC     (700000         )
                1150            ISA                     API ON, NO PAPER TAPE READER ATTACHED
                1160            TLS+10
```

GROWTH SYSTEM STANDARD DEFINITIONS

```
                1170            DLP                     DISABLE THE LIGHT PEN, ON GENERAL PRINCIPLES
                1180            DZM     CATALT          WE WON'T MESS WITH SOMEONE ELSE'S ALTERED CATALOG
                1190            MESS    <#1             HERE>,#2-5
                1200    NEXTL   MESS    <               )>;1 "PRINT THE INPUT REQUEST
                1210            LINE                    GET THE USER'S INPUT
                1220            ,PMC    RESTORE
                1230            ,ENDM
                1240
                1250            ,LIST   ON
                1260            ,END
    002175      680     CPARAM  ,EQU    BUFFER+5
    002170      690     CATLOG  ,EQU    BUFFER
                700     RET     ,OPDEF  JMP+020000
                710             ,HEAD   M
```

```
M                                   MAIN PROGRAM

                    720             .STITL  MAIN PROGRAM
                    730     *
                    740     *
                    750     *       MACRO TO SET UP PURE-CODED SUBROUTINE ENTRANCES
                    760     *
                    770     ENTER   .DEFIN
                    780             .PMC    SAVE,OFF
                    790     9MAPBUG .EQU    .
                    800             .USE    IMPURE          SUBROUTINE ENTRANCES CANNOT BE PURE CODE
                    810             .PMC    SAVE,ON
                    820     #1      ...
                    830             .PMC    RESTORE
                    840             HLT
                    850             JMP     9MAPBUG
                    860             .USE    PREVIOUS
                    870             .PMC    RESTORE
                    880             .ENDM
                    890     *
                    900     *
                    910     *
003701  703302      920     START   CAF
003702  700002      930             IOF
003703  700416      940             TLS+10
003704  142156      950             DZM     DSBALT          DON'T MESS AROUND WITH SOMEONE ELSE'S ALTERED BUFFER
    003705          960             CRLF
    003706          970             MESS    <LOADER>,6
    003714          980     NEXTL   MESS    <?>,1           REQUEST THE NEXT LINE OF INPUT
    003720          990             LINE                    AND GET IT
                    1000    *
                    1010    *
                    1020    *       SCAN NEXT COMMAND
                    1030    *
    003721          1040            WORD                    GET THE NEXT COMMAND
003722  603714      1050            JMP     NEXTL           IGNORE VACUOUS LINES
003723  765112      1060    MONX2   LAW     COMTB-1         POINT TO COMMAND TABLE
003724  040012      1070            DAC     CMDX            SAVE IT
003725  777754      1080            LAW     COMTB-COME
003726  043246      1090            DAC     C$CTEM1         SAVE COUNT
    003727          1100            WORD1                   RECOVER THE COMMAND
003730  560012      1110    COML    SAD     CMDX,X          CHECK(1)WORD
003731  620012      1120            JMP     CMDX,X          GO TO IT
    003732          1130            LOOP    C$CTEM1,COML
    003734          1140    ERROR   MESS    <COMMAND ERROR>,13.
003744  603714      1150            JMP     NEXTL
    003745          1160    FORMAT  MESS    <FORMAT ERROR WORD # >+20.
    003760          1170            COUNT
    003761          1180            OCTZ
003763  603714      1190            JMP     NEXTL
    003764          1200    HARD    MESS    <DEVICE ERROR>,12.
003774  603714      1210            JMP     NEXTL
    003775          1220    NSAVE   MESS    <FILE NOT SAVED>,14.
004006  603714      1230            JMP     NEXTL
```

M MAIN PROGRAM

```
    004007      1240    DSAVE   MESS    <FILE ALREADY SAVED>,18.
004021  603714  1250            JMP     NEXTL
```

M MAIN PROGRAM

```
                    1260            .EJECT
                    1270    *
                    1280    *       COPY SUBROUTINE
                    1290    *
                    1300    *       COPIES FROM DEVICE INDA TO DEVICE OUTDA FOR LEN WORDS
                    1310    *
    004022          1320            ENTER   COPY
                                    .PMC    SAVE,ON
    003170                  COPY    ...
004022  103212  1330            JMS     FORCE           FORCE THE BUFFER OUT IF ALTERED
004023  762170  1340            LAW     BUFFER
004024  042154  1350            DAC     D$BCA           BE SURE THE CORE ADDRESS IS SET CORRECTLY
004025  203230  1360    COPL    LAC     LEN             GET LENGTH REMAINING
004026  741200  1370            SNA
004027  623170  1380            RET     COPY            RETURN IF DONE
004030  346066  1390            TAD     (-BMAX)         SUBTRACT AMOUNT WE CAN COPY IN (1) OPERATION
004031  741100  1400            SPA
004032  604036  1410            JMP     COPL2
004033  043230  1420            DAC     LEN             RESTORE LENGTH REMAINING
004034  206067  1430            LAC     (BMAX)          GET AMOUNT FOR CURRENT COPY
004035  604040  1440            JMP     COPL4           SKIP THE OTHER BRANCH
004036  203230  1450    COPL2   LAC     LEN             GET LENGTH FOR COPY
004037  143230  1460            DZM     LEN             NONE REMAINING
004040  042155  1470    COPL4   DAC     D$BLEN          SAVE NEW LENGTH TO COPY
004041  203227  1480            LAC     INDA            GET INPUT DA
004042  042153  1490            DAC     D$BDA           SAVE IT
004043  543232  1500            SAD     OUTDA           CHECK FOR NOTHINGISH COPIES
004044  623170  1510            RET     COPY
004045  346070  1520            TAD     (BMAX/BLKLEN)   COMPUTE AMOUNT TO COPY IN BLKLENKS
004046  043227  1530            DAC     INDA            RESTORE FOR NEXT COPY
004047  103251  1540            JMS     C$RCVR          SET UP THE ERROR RECOVERY
004050  764055  1550            LAW     .+5
004051  652000  1560            LMQ                     SET THE RESTART ADDRESS
004052  762153  1570            LAW     D$BDA           LOAD THE PARAMETER POINTER
004053  705003  1580            PREAD                   DO THE OPERATION
004054  605223  1590            JMP     C$RCVR4         ERROR RETURN
004055  203232  1600            LAC     OUTDA           GET OUTPUT DA
004056  741200  1610            SNA
004057  623170  1620            RET     COPY            RETURN IF INPUT ONLY
004060  042153  1630            DAC     D$BDA           SAVE IT
004061  346070  1640            TAD     (BMAX/BLKLEN)
004062  043232  1650            DAC     OUTDA           SET THE UPDATED OUTPUT DEVICE ADDRESS FOR NEXT TIME
004063  762153  1660            LAW     D$BDA           GET PARAMETERS
004064  103172  1670            JMS     TPOT            CORY OUT
004065  604025  1680            JMP     COPL            LOOP
```

M MAIN PROGRAM

```
                    1690            ,EJECT
                    1700    *
                    1710    *       DEVICE WRITE SETUP ROUTINE
                    1720    *
     004066         1730            ENTER   TPOT
                                    ,PMC    SAVE,ON
     003172                 TPOT    ...
004066  043235      1740            DAC     PARW            SAVE CALLING PARAMETERS
004067  103251      1750            JMS     CSRCOVR         SET UP THE ERROR RECOVERY
004070  764075      1760            LAW     .+5
004071  652000      1770            LMQ                     SET THE RESTART ADDRESS
004072  203235      1780    TPOT1   LAC     PARW            GET PARAMETER WORD
004073  705005      1790            PWRITE                  DO THE WRITE OPERATION
004074  605223      1800            JMP     CSRCVR4         ERROR RETURN
004075  623172      1810            RET     TPOT
                    1820    *
                    1830    *       READ A WORD OF PAPER TAPE
                    1840    *
                    1850    *       RETURN IS +1 IF TIMEOUT
                    1860    *       RETURN IS +2 IF OK, WITH THE CHARACTER IN THE AC
                    1870    *
     004076         1880            ENTER   GETW
                                    ,PMC    SAVE,ON
     003174                 GETW    ...
004076  700144      1890            RSB                     SLELECT BINARY
     004077         1900    GEWL    ...
004077  700101      1910            RSF                     WAIT FOR READER
004100  741000      1920            SKP
004101  604107      1930            JMP     GW1             WE GOT IT
004102  700314      1940            IORS                    NOT YET -- LOOK FOR STATUS
004103  506067      1950            AND     (001000)        RECOVER OUT-OF-TAPE BIT
004104  740200      1960            SZA                     SKIP IF NOT TRUE
004105  623174      1970            RET     GETW            RETURN TO CALLER WHEN OUT-OF-TAPE
004106  604077      1980            JMP     GEWL            ELSE WAIT MORE
004107  700112      1990    GW1     RRB                     GET THE CHAR
004110  043245      2000            DAC     WT              SAVE
004111  243225      2010            XOR     CKSUM           COMPUTE CHECKSUM
004112  043225      2020            DAC     CKSUM
004113  203245      2030            LAC     WT              RESTORE WORD
004114  443174      2040            ISZ     GETW            SHOW OUR SUCESS
004115  623174      2050            RET     GETW
     003176         2060            ,USE    IMPURE
003176  740040      2070    TIME    XX
     004116         2080            ,USE    PURE
```

M LOADER COMMANDS

```
                    2090            .STITL  LOADER COMMANDS
                    2100    *
                    2110    *       CLEAR
                    2120    *
                    2130    *       CLE <DEVICE>
                    2140    *
                    2150    *       CLEARS THE CATALOG FOR A DEVICE
                    2160    *
004116  103212      2170    CLE     JMS     FORCE           FORCE OUT THE OLD CATALOG
004117  103271      2180            JMS     C$DEVCV         GET THE DEVICE NAME AND CONVERT IT TO DEVICE ADDRESS FORMAT
004120              2190            FORMAT                  FORMAT ERROR -- PAPER TAPE NOT LEGAL FOR THIS OPERATION
004121  103177      2200            JMS     NEWHDR          GO CLEAR THE HEADER AND CATALOG FILE CONTROL BLOCK
004122  603714      2210            JMP     NEXTL
                    2220
                    2230
                    2240
                    2250    *
                    2260    *       NEWHDR THE HEADER AND THE CATALOG FILE CONTROL BLOCK OF THE CORE CATALOG
                    2270    *       FOR THE HANDLER WHOSE DEVICE ADDRESS IS PASSED IN THE AC.
                    2280    *
004123              2290            ENTER   NEWHDR          SUBROUTINE TO INITIALIZE THE CATALOG HEADER AND FIRST FILE CONTROL BLOCK
                                    .PMC    SAVE,ON
003177                      NEWHDR  ...
004123  346071      2300            TAD     (1)
004124  042175      2310            DAC     CPARAM          SET THE CATALOG DEVICE ADDRESS
004125  042153      2320            DAC     D$BDA           SET IT ALSO TO BE THE BUFFER DEVICE ADDRESS
004126  346071      2330            TAD     (1)
004127  042170      2340            DAC     CATLOG          SET THE DEVICE ADDRESS OF THE FIRST FREE BLOCK
004130  776701      2350            LAW     $DTMAX
004131  042173      2360            DAC     CATLOG+3        SET THE DECTAPE MAXIMUM BLOCK NUMBER
004132  202175      2370            LAC     CPARAM          LOAD THE CATALOG DEVICE ADDRESS
004133  506072      2380            AND     (40000)         CHECK FOR  DISK
004134  741200      2390            SNA                     YES IF SKP
004135  604140      2400            JMP     NEW2
004136  777601      2410            LAW     $DKMAX
004137  042173      2420            DAC     CATLOG+3        SET THE MAXIMUM DISK BLOCK NUMBER
004140  777010      2430    NEW2    LAW     17010
004141  042171      2440            DAC     CATLOG+1        SET THE INITIAL POINTER TO THE FIRST FREE FCB
004142  762200      2450            LAW     CATLOG+10
004143  040012      2460            DAC     CMDX
004144  777777      2470            LAW     -1
004145  042172      2480            DAC     CATLOG+2        SET THE INITIAL FCB COUNT
004146  206073      2490            LAC     ($CTL)
004147  042174      2500            DAC     CATLOG+4        SET THE CATALOG BLOCK'S NAME (CTL)
004150  777000      2510            LAW     17000
004151  042176      2520            DAC     CATLOG+6        SET THE CATALOG BLOCK'S CORE ADDRESS
004152  762170      2530            LAW     CATLOG
004153  042154      2540            DAC     D$BCA           SET THE REAL CORE ADDRESS
004154  206074      2550            LAC     (CATLEN)
004155  042177      2560            DAC     CATLOG+7        SET THE CATALOG BLOCK LENGTH
004156  042155      2570            DAC     D$BLEN          SET THE BUFFER LENGTH AS WELL
004157  777740      2580            LAW     $B00
```

M LOADER COMMANDS

```
004160  042200  2590          DAC     CATLOG+10       SET THE CATALOG BLOCK'S TRANSFER IN CASE IT GETS LOADED
004161  442156  2600          ISZ     D$BALT          SET THE ALTERED CATALOG FLAG
004162  623177  2610          RET     NEWHDR
```

M LOADER COMMANDS

```
                    2620            ,EJECT
                    2630    *
                    2640    *       UNSAVE
                    2650    *
                    2660    *       UNSAVE <TREE NAME>
                    2670    *
                    2680    *       UNSAVE DELETES AN ENTRY FROM A CATALOG
                    2690    *
004163  103266      2700    UNS     JMS     C$GNAME         GET THE CTL
004164              2710            FORMAT                  FORMAT ERROR -- PAPER TAPE NOT LEGAL FOR THIS OPERATION
004165  103254      2720            JMS     C$CATL          LOOK IT UP
004166  603775      2730            JMP     NSAVE           FILE NOT SAVED
004167  200011      2740            LAC     $CATX           GET THE POINTER
004170  043240      2750            DAC     TEMP
004171  163240      2760            DZM     TEMP,X          A NAME OF ZERO INDICATES NOTHING THERE
004172  442156      2770            ISZ     D$BALT          SET CATALOG ALTERED FLAG
004173  603714      2780            JMP     NEXTL
```

M LOADER COMMANDS

```
                    2790            .EJECT
004174  103212      2800    EXIT    JMS     FORCE           CLEAR ANY REMAINING BUFFER ALTERATIONS
004175  705001      2810            TERMINAT
```

```
                    2820            ,EJECT
                    2830    *
                    2840    *       PURGE
                    2850    *
                    2860    *       PURGE <DEVICE>
                    2870    *
                    2880    *       PURGE COMPACTS STORAGE FOR A GIVEN DEVICE
                    2890    *
    004176          2900    PUR     ...
004176  103271      2910            JMS     CSDEVCV         GET THE DEVICE NAME AND CONVERT IT TO HANDLER DEVICE ADDRESS FORMAT
    004177          2920            FORMAT                  FORMAT ERROR -- PAPER TAPE NOT LEGAL FOR THIS OPERATION
004200  043227      2930            DAC     INDA
004201  043232      2940            DAC     OUTDA           INPUT AND OUTPUT BOTH
004202  604211      2950            JMP     PCOPY           COPY OVER THE FILES
                    2960    *
                    2970    *       LDUMP
                    2980    *
                    2990    *       LDUMP <DEVICE> <DEVICE>
                    3000    *
                    3010    *       LDUMP DUMPS THE FILES ON THE FIRST DEVICE TO THE SECOND DEVICE
                    3020    *
    004203          3030    LDU     ...
004203  103271      3040            JMS     CSDEVCV         GET THE DEVICE NAME AND CONVERT IT TO HANDLER DEVICE ADDRESS FORMAT
    004204          3050            FORMAT                  FORMAT ERROR -- PAPER TAPE NOT LEGAL FOR THIS OPERATION
004205  043227      3060            DAC     INDA            SET THE INPUT HANDLER DEVICE ADDRESS
004206  103271      3070            JMS     CSDEVCV         GET THE DEVICE NAME AND CONVERT IT TO HANDLER DEVICE ADDRESS FORMAT
    004207          3080            FORMAT                  FORMAT ERROR -- PAPER TAPE NOT LEGAL FOR THIS OPERATION
004210  043232      3090            DAC     OUTDA           SET THE OUTPUT HANDLER DEVICE ADDRESS
                    3100    *
                    3110    *       ROUTINE TO COPY CATALOG FOR PURGE AND LDUMP
                    3120    *
    004211          3130    PCOPY   ...
004211  203227      3140            LAC     INDA
004212  103247      3150            JMS     CSRCAT          READ THE INPUT DEVICE CATALOG
004213  202172      3160            LAC     CATLOG+2
004214  043237      3170            DAC     PTMP            SAVE THE FCB COUNT
004215  203232      3180            LAC     OUTDA
004216  103177      3190            JMS     NEWHDR          CLEAR THE OUTPUT DEVICE CATALOG
                    3200    *
                    3210    *       LOOP TO RECOPY FILES
                    3220    *
                    3230    *       CMDX RUNS DOWN THE INPUT DEVICE CATALOG
                    3240    *       CATX RUNS DOWN THE OUTPUT DEVICE CATALOG
                    3250    *
004217  443237      3260    PURL    ISZ     PTMP            CHECK FOR DONE
004220  741000      3270            SKP
004221  603714      3280            JMP     NEXTL
004222  220012      3290            LAC     CMDX,X          GET THE NEXT FILE
004223  741200      3300            SNA
004224  604252      3310            JMP     PURZ            NOT THERE
004225  103275      3320            JMS     CSSAVE          SAVE IT
004226  740040      3330            HLT                     %#&*!  THE FILE CANNOT POSSIBLY BE SAVED  !*&#%
```

M LOADER COMMANDS

```
004227  220012  3340        LAC     CMDX,X
004230  043227  3350        DAC     INDA            SET THE INPUT FILE'S CURRENT DEVICE ADDRESS
004231  220012  3360        LAC     CMDX,X
004232  043240  3370        DAC     TEMP            SAVE THE FILE'S CORE ADDRESS
004233  220012  3380        LAC     CMDX,X
004234  043230  3390        DAC     LEN             SAVE THE FILE'S LENGTH
004235  103277  3400        JMS     C$ALC           ALLOCATE SPACE ON THE DEVICE FOR IT
004236  060011  3410        DAC     $CATX,X         SET ITS NEW DEVICE ADDRESS
004237  043232  3420        DAC     OUTDA           SAVE FOR OUTPUT
004240  203240  3430        LAC     TEMP
004241  060011  3440        DAC     $CATX,X         SET IT'S CORE ADDRESS
004242  203230  3450        LAC     LEN
004243  060011  3460        DAC     $CATX,X         SET IT'S LENGTH
004244  220012  3470        LAC     CMDX,X
004245  060011  3480        DAC     $CATX,X         SET ITS TRANSFER CARD
004246  103170  3490        JMS     COPY
004247  203232  3500        LAC     OUTDA
004250  103247  3510        JMS     C$RCAT          GET THE OUTPUT CATALOG BACK
004251  604217  3520        JMP     PURL            2LOOP
004252  206075  3530  PURZ  LAC     (FCBLEN-1)
004253  340012  3540        TAD     CMDX
004254  040012  3550        DAC     CMDX            SAVE NEW POSITION
004255  604217  3560        JMP     PURL            LOOP
```

M LOADER COMMANDS

```
                    3570            .EJECT
                    3580    *
                    3590    *       SAVE
                    3600    *
                    3610    *       SAVE <TREE NAME> <START> <END> <FORMAT> <DEVICE>
                    3620    *
                    3630    *       SAVE CREATES A NEW CATALOG ENTRY AND LOADS
                    3640    *       IT WITH A FILE
                    3650    *
004256  103266      3660    SAV     JMS     CSGNAME         GET A NAME
    004257          3670            FORMAT                  FORMAT ERROR -- PAPER TAPE NOT LEGAL FOR THIS OPERATION
004260  103275      3680            JMS     CSSAVE          SAVE IF POSSIBLE
004261  604007      3690            JMP     DSAVE           DUPLICATE
004262  200011      3700            LAC     $CATX
004263  043223      3710            DAC     CATP            SET A POINTER TO THE FILENAME IN THE CATALOG
    004264          3720            NUM
    004265          3730            FORMAT
004266  043222      3740            DAC     BOTM            SET THE START ADDRESS
004267  440011      3750            ISZ     $CATX
004270  060011      3760            DAC     $CATX,X         SAVE IT
    004271          3770            NEG
004273  043230      3780            DAC     LEN
    004274          3790            NUM                     GET THE END ADDRESS
    004275          3800            FORMAT
004276  343230      3810            TAD     LEN             SUBTRACT THE START ADDRESS
004277  741100      3820            SPA
    004300          3830            FORMAT                  FORMAT ERROR -- END ADDRESS LESS THAN START ADDRESS
004301  043230      3840            DAC     LEN
004302  060011      3850            DAC     $CATX,X         SET THE LENGTH
004303  103277      3860            JMS     CSALC           ALLOCATE FOR IT
004304  443223      3870            ISZ     CATP            INDEX POINTER
004305  063223      3880            DAC     CATP,X          SET THE FILE'S DEVICE ADDRESS
004306  604322      3890            JMP     REP1            JOIN REPLACE
```

M LOADER COMMANDS

```
                    3900            .EJECT
                    3910    *
                    3920    *       REPLACE
                    3930    *
                    3940    *       REPLACE <TREE NAME> <FORMAT> <DEVICE>
                    3950    *
004307  103266      3960    REP     JMS     CSGNAME         GET A CATALOG
004310              3970            FORMAT                  FORMAT ERROR -- PAPER TAPE NOT LEGAL FOR THIS OPERATION
004311  103254      3980            JMS     CSCATL
004312  603775      3990            JMP     NSAVE           NOT THERE
004313  440011      4000            ISZ     SCATX           SKIP THE DEVICE ADDRESS
004314  200011      4010            LAC     SCATX
004315  043223      4020            DAC     CATP            SAVE A POINTER TO THE DEVICE ADDRESS IN THE CATALOG
004316  220011      4030            LAC     SCATX,X
004317  043222      4040            DAC     BOTM            SET THE CORE ADDRESS
004320  220011      4050            LAC     SCATX,X
004321  043230      4060            DAC     LEN             SET THE LENGTH
                    4070
004322              4080    REP1    WORD                    GET FORMAT
004323              4090            FORMAT
004324  546076      4100            SAD     (CAL SABS)      NECESSARY TO DO IT THIS WAY OR ELSE ASSEMBLER TAKES ABS OPCODE
004325  604445      4110            JMP     ABSF
004326  546077      4120            SAD     (SBIN)
004327  604355      4130            JMP     BINF
004330  546100      4140            SAD     (SGRO)
004331  604333      4150            JMP     GROF
004332              4160            FORMAT                  NO OTHER FORMATS
```

M LOADER COMMANDS

```
                    4170            .EJECT
                    4180    *
                    4190    *       LOAD FORMATS -- GROWTH
                    4200    *
                    4210    *       GROWTH TAKES A FILE FROM ANOTHER GROWTH DEVICE
                    4220    *
004333  202175      4230    GROF    LAC     CPARAM          GET THE CURRENT CATALOG DEVICE ADDRESS
004334  043240      4240            DAC     TEMP
004335  103266      4250            JMS     C$GNAME         GET A NAME
    004336          4260            FORMAT                  FORMAT ERROR -- PAPER TAPE NOT LEGAL FOR THIS OPERATION
004337  103254      4270            JMS     C$CATL
004340  603775      4280            JMP     NSAVE           NOT THERE
004341  220011      4290            LAC     $CATX,X         GET THE DEVICE ADDRESS
004342  043227      4300            DAC     INDA            SAVE INPUT DA
004343  440011      4310            ISZ     $CATX           BYPASS LOC
004344  440011      4320            ISZ     $CATX           AND LEN
004345  220011      4330            LAC     $CATX,X         GET TCD
004346  043245      4340            DAC     WT              SAVE IT
004347  203240      4350            LAC     TEMP
004350  103247      4360            JMS     C$RCAT          READ IN OLD CATALOG
    004351          4370    GRO1    ...
004351  223223      4380            LAC     CATP,X
004352  043232      4390            DAC     OUTDA           SET THE OUTPUT DEVICE ADDRESS
004353  103170      4400            JMS     COPY            COPY OVER FILE
004354  604771      4410            JMP     NXLT1
```

M LOADER COMMANDS

```
                    4420            .EJECT
                    4430    *       LOADER FORMATS -- BINARY
                    4440    *
                    4450    *       BINARY LOADS ABSOLUTE BINARY DATA FROM
                    4460    *       PAPER TAPE, DISK, OR TAPE
                    4470    *
004355  103271      4480    BINF    JMS     CSDEVCV         GET THE DEVICE NAME AND CONVERT IT TO HANDLER DEVICE ADDRESS FORMAT
004356  604366      4490            JMP     BINPPT          PAPER TAPE
004357  043227      4500            DAC     INDA            SET THE INPUT HANDLER DEVICE ADDRESS
004360              4510            NUM                     GET BLOCK NUMBER
004361              4520            FORMAT
004362  343227      4530            TAD     INDA
004363  043227      4540            DAC     INDA            SET THE INPUT FILE DEVICE ADDRESS
004364  143245      4550            DZM     WT              MAKE SURE WE GET A TCD
004365  604351      4560            JMP     GRO1            HANDLE LIKE GROWTH
                    4570    *
                    4580    *       LOAD FROM PAPER TAPE
                    4590    *
004366  777000      4600    BINPPT  LAW     -BMAX           GET BUFFER LENGTH
004367  043217      4610            DAC     BCOUNT          SAVE IT
004370  762167      4620            LAW     BUF-1           GET BUFFER POINTER
004371  040012      4630            DAC     CMDX            SAVE IT
004372  777777      4640            LAW     -1              COMPLEMENT LEN
004373  343230      4650            TAD     LEN
004374  740001      4660            CMA
004375  043230      4670            DAC     LEN
004376  223223      4680            LAC     CATP,X          GET OUTPUT DA
004377  043232      4690            DAC     OUTDA
004400  103174      4700    BINL    JMS     GETW            GET A WORD OF PPT
004401  604410      4710            JMP     BDONE           DONE IF READER IS NOT READY
004402  060012      4720            DAC     CMDX,X          SAVE IN BUFFER
004403  443230      4730            ISZ     LEN             SEE IF WE HAVE MORE
004404  741000      4740            SKP
004405  604410      4750            JMP     BDONE           NO
004406  443217      4760            ISZ     BCOUNT          COUNT AMOUNT IN BUFFER
004407  604400      4770            JMP     BINL            GET ANOTHER WORD IF NOT FULL
004410  775611      4780    BDONE   LAW     -BUF+1          GET ORIGINAL POINTER
004411  340012      4790            TAD     CMDX            COMPUTE AMOUNT TO WRITE
004412  043244      4800            DAC     TPARAM+2
004413  203232      4810            LAC     OUTDA           GET OUTPUT DA
004414  043242      4820            DAC     TPARAM
004415  346070      4830            TAD     (BMAX/BLKLEN)
004416  043232      4840            DAC     OUTDA           RESTORE NEW POINTER
004417  763242      4850            LAW     TPARAM          POINT TO PARAMETERS
004420  103172      4860            JMS     TPOT            OUTPUT TAPE
004421  203230      4870            LAC     LEN
004422  740200      4880            SZA
004423  604400      4890            JMP     BINL            LOOP
```

M LOADER COMMANDS

```
                    4900            .EJECT
                    4910    *
                    4920    *
                    4930    *       GET THE TRANSFER CARD
                    4940    *
    004424          4950    TCD     MESS    <TDC?>,4
    004431          4960            LINE                    READ THE REPLY
004432  760003      4970            LAW     3
004433  343223      4980            TAD     CATP
004434  043240      4990            DAC     TEMP            MOVE THE CATALOG POINTER TO THE TRANSFER CARD SLOT
    004435          5000            NUM                     GET THE NUMBER
004436  604424      5010            JMP     TCD             VACUOUS LINE -- ASK AGAIN
004437  345140      5020            TAD     JMPW            ADD A JUMP INSTRUCTION
    004440          5030    TCD1    ...,
004440  563240      5040            SAD     TEMP,X          CHECK AGAINST THE CURRENT ONE
004441  741000      5050            SKP
004442  442156      5060            ISZ     D$BALT          WE CHANGED THE CATALOG
004443  063240      5070            DAC     TEMP,X          SAVE THE NEW POINTER
004444  603714      5080            JMP     NEXTL           EXIT
```

M LOADER COMMANDS

```
                    5090            .EJECT
                    5100    *
                    5110    *       LOADER FORMAST -- ABS
                    5120    *
                    5130    *       ABS LOADS A FILE IN ABSOLUTE ASSEMBLY FORMAT
                    5140    *
004445  143225      5150    ABSF    DZM     CKSUM           CLEAR RANDOM FLAGS
004446  143231      5160            DZM     NP
004447  143234      5170            DZM     P2
004450  143233      5180            DZM     OFFSET
004451  143224      5190            DZM     CBASE
004452  203222      5200            LAC     BOTM            GET CORE BOTM
004453  343230      5210            TAD     LEN             COMPUTE TOP
    004454          5220            NEG
004456  043241      5230            DAC     TOP
004457  143236      5240            DZM     PFLAG           CLEAR PPT FLAG
004460  103271      5250            JMS     C$DEVCV         GET THE DEVICE NAME AND CONVERT IT TO HANDLER DEVICE ADDRESS FORMAT
004461  604470      5260            JMP     PAPER           PAPER TAPE
004462  043227      5270            DAC     INDA            SET THE INPUT HANDLER DEVICE ADDRESS
    004463          5280            NUM                     GET THE BLOCK NUMBER
    004464          5290            FORMAT
004465  343227      5300            TAD     INDA            ADD IN THE DA
004466  043227      5310            DAC     INDA            SET THE INPUT FILE DEVICE ADDRESS
004467  604613      5320            JMP     ML01
                    5330    *
                    5340    *       PAPER TAPE
                    5350    *
    004470          5360    PAPER   ...
004470  143225      5370            DZM     CKSUM           CLEAR OUT OLD CHECKSUM
004471  443236      5380            ISZ     PFLAG           SET PAPER TAPE FLAG
    004472          5390            MESS    <MOUNT PAPER TAPE AND TYPE GO>,28.
    004507          5400            LINE
    004510          5410            CRLF                    MOVE TO A FRESH LINE
004511  103174      5420            JMS     GETW            GET THE FIRST WORD
004512  604470      5430            JMP     PAPER           RETRY TIMEOUTS
004513  505141      5440            AND     INSTM           MASK INSTRUCTION FIELD
004514  545137      5450            SAD     DACW            CHECK FOR DAC
004515  604533      5460            JMP     PL01            FOUND FIRST WORD
004516  700104      5470            RSA                     IGNORE LOADER
004517  700101      5480            RSF
004520  604517      5490            JMP     .-1
004521  700112      5500            RRB
004522  740200      5510            SZA
004523  604516      5520            JMP     .-5
004524  143225      5530            DZM     CKSUM           CLEAR CHECKSUM
004525  103174      5540    PL02    JMS     GETW            GET A WORD
004526  603764      5550            JMP     HARD            TIMEOUT
004527  505141      5560            AND     INSTM           MASK INSTRUCTION FIELD
004530  545137      5570            SAD     DACW            CHECK FOR DAC
004531  741000      5580            SKP
004532  604667      5590            JMP     LEND            END IF NOT
004533  203245      5600    PL01    LAC     WT              GET WORD
```

M LOADER COMMANDS

```
004534  343233  5610          TAD     OFFSET          OFFSET IT
004535  506101  5620          AND     (17777)         TRIM TO ADDRESS
004536  045142  5630          DAC     ADD             SET LOAD ADDRESS
004537  103174  5640          JMS     GETW            GET A WORD OF SOURCE
004540  603764  5650          JMP     HARD            TIMEOUT
    004541      5660          NEG                     FORM - COUNT
004543  043226  5670          DAC     COUNT
004544  103174  5680          JMS     GETW            READ CHECKSUM
004545  603764  5690          JMP     HARD            TIMEOUT
004546  203225  5700          LAC     CKSUM           CHECK IT
004547  740200  5710          SZA                     OK
004550  604560  5720          JMP     CKERR           CHECKSUM ERROR
004551  103174  5730          JMS     GETW            GET A WORD
004552  603764  5740          JMP     HARD            TIMEOUT
004553  103201  5750          JMS     PUTW            PUT IT
004554  445142  5760          ISZ     ADD             COUNT ADDRESS
004555  443226  5770          ISZ     COUNT           COUNT WORDS
004556  604551  5780          JMP     .-5             LOOP
004557  604525  5790          JMP     PL02            LOOP
    004560      5800  CKERR   MESS    <CHECKSUM ERROR>,14.
004571  604470  5810          JMP     PAPER           TRY AGAIN
                5820  *
                5830  *       SUBROUTINE TO PLACE A WORD OF SOURCE IN CORE
                5840  *
    004572      5850          ENTER   PUTW
                              .PMC    SAVE,ON
    003201                    PUTW    ...
004572  043245  5860          DAC     WT              SAVE WORD TO PUT
004573  205142  5870          LAC     ADD             GET ADDRESS
004574  343224  5880          TAD     CBASE           ADD BASE OF CORE
004575  741100  5890          SPA                     CHECK FOR SMALL ADDRESS
004576  623201  5900          RET     PUTW
004577  205142  5910          LAC     ADD             GET ADDRESS
004600  346102  5920          TAD     (-TBUF)         CHECK IF REASONABLE
004601  740100  5930          SMA
004602  604606  5940          JMP     PUTW1           NO -- CHECK IF WE REALLY NEED TO LOAD THIS
004603  203245  5950          LAC     WT
004604  065142  5960          DAC     ADD,X           SET WORD
004605  623201  5970          RET     PUTW            RETURN
004606  205142  5980  PUTW1   LAC     ADD             CHECK IF ADDRESS IS IN BOUNDS
004607  343241  5990          TAD     TOP
004610  741100  6000          SPA
004611  443231  6010          ISZ     NP
004612  623201  6020          RET     PUTW
004613  203227  6030  ML01    LAC     INDA            GET INPUT BLOCK NUMBER
004614  043214  6040          DAC     BLOCK
004615  103203  6050          JMS     BUFIN           READ FIRST BUFFERS
004616  604633  6060          JMP     NBLOCK          TREAT LIKE NEW BLOCK
004617  443226  6070  ML02    ISZ     COUNT           COUNT WORDS IN THIS STRING
004620  741000  6080          SKP
004621  604633  6090          JMP     NBLOCK          READ NEW ONE
004622  443221  6100  ML03    ISZ     BWS             COUNT BUFFER WORDS
```

M LOADER COMMANDS

```
004623  741000  6110            SKP
004624  103203  6120            JMS     BUFIN
004625  223220  6130            LAC     BUFA,X          GET NEXT WORD FROM BUFFER
004626  445142  6140            ISZ     ADD             INCREMENT ADDRESS
004627  740000  6150            NOP                     IN CASE OF - ADDRESSES
004630  443220  6160            ISZ     BUFA            AND BUFFER POINTER
004631  103201  6170            JMS     PUTW            OUTPUT IT
004632  604617  6180            JMP     ML02            LOOP
004633  203220  6190    NBLOCK  LAC     BUFA            GET BUFFER ADDRESS
004634  346103  6200            TAD     (377)           ROUND UP TO NEXT BLOCK
004635  506104  6210            AND     (177400)
004636  546105  6220            SAD     (BASE)          CHECK FOR END
004637  103203  6230            JMS     BUFIN           GET BUFFER IN IF SO
004640  203220  6240            LAC     BUFA            REPEAT THE PREVIOUS OPPERATION
004641  346103  6250            TAD     (377)
004642  506104  6260            AND     (177400)
004643  043220  6270            DAC     BUFA            SAVE IT
004644  773611  6280            LAW     -TBUF-1777
004645  343220  6290            TAD     BUFA            COMPUTE BUFFER COUNT
004646  043221  6300            DAC     BWS             SAVE
004647  223220  6310            LAC     BUFA,X          GET COUNT
004650  043226  6320            DAC     COUNT           SAVE AS SUCH
004651  443220  6330            ISZ     BUFA            INCREMENT COUNTER
004652  760100  6340            LAW     -17700          CHECK FOR VERY HIGH ADDRESS
004653  363220  6350            TAD     BUFA,X          WHICH SIGNNALS END OF BLOCK
004654  740100  6360            SMA
004655  604663  6370            JMP     MTCD            YES A TCD
004656  223220  6380            LAC     BUFA,X          GET ADDRESS AGAIN
004657  343233  6390            TAD     OFFSET
004660  045142  6400            DAC     ADD             SET IT
004661  443220  6410            ISZ     BUFA
004662  604617  6420            JMP     ML02            RETURN
004663  443220  6430    MTCD    ISZ     BUFA            INCREMENT BUFFER ADDRESS
004664  223220  6440            LAC     BUFA,X          GET WORD
004665  043245  6450            DAC     WT              SAVE
004666  604667  6460            JMP     LEND
                6470    *
                6480    *
                6490    *       END OF LOAD
                6500    *
    004667      6510    LEND    ...
004667  203234  6520            LAC     P2              CHECK IF PASS 2
004670  740200  6530            SZA
004671  604767  6540            JMP     P2L             FORCE BUFFER AND EXIT IF SO
004672  203223  6550            LAC     CATP            POINT TO CATALOG
004673  103172  6560            JMS     TPOT            WRITE IT OUT
004674  203231  6570            LAC     NP              CHECK FOR NEW PASS NEEDED
004675  741200  6580            SNA
004676  604771  6590            JMP     NXLT1           NO
004677  777700  6600            LAW     -100            SET CORE BASE
004700  043224  6610            DAC     CBASE
004701  766000  6620            LAW     -BASE+2000
```

M LOADER COMMANDS

```
004702  343222  6630          TAD     BOTM            CHECK FOR NOTHING LOADED AT ALL
004703  740100  6640          SMA
004704  604756  6650          JMP     LEND1
004705  203222  6660          LAC     BOTM            TRIM POINTER
004706  506103  6670          AND     (377)
004707  740200  6680          SZA
004710  346106  6690          TAD     (-400)
004711          6700          NEG
004713  346107  6710          TAD     (-TBUF+100)     FORM POINTER
004714  043233  6720          DAC     OFFSET
004715  203222  6730          LAC     BOTM
004716  740001  6740          CMA                     FORM PARAMETERS FOR FUTURE OUTPUT
004717  346110  6750          TAD     (TBUF+1)
004720  506104  6760          AND     (174000)        TRIM AGAIN
004721  043240  6770          DAC     TEMP
004722          6780          NEG
004724  343230  6790          TAD     LEN             DECREMENT LENGTH
004725  043244  6800          DAC     TPARAM+2
004726  203240  6810          LAC     TEMP            GET NUMBER OF BLOCKS PROCESSED
004727  660510  6820          LRSS    8.
004730  043240  6830          DAC     TEMP
004731  223223  6840          LAC     CATP,X          GET START BLOCK
004732  343240  6850          TAD     TEMP
004733  043242  6860          DAC     TPARAM          SET BLOCK NUMBER
004734  203236  6870  LEND2   LAC     PFLAG           SEE IF PPT
004735  443234  6880          ISZ     P2
004736  740200  6890          SZA
004737  604470  6900          JMP     PAPER
004740  604613  6910          JMP     MLO1
                6920
004741          6930          ENTER   BUFIN
                              .PMC    SAVE,ON
003203                BUFIN   ...
004741  764746  6940          LAW     .+5
004742  652000  6950          LMQ                     SET THE RESTART ADDRESS
004743  763214  6960          LAW     BPARAM          GET THE PARAMETER POINTER
004744  705003  6970          PREAD                   DO THE READ
004745  605223  6980          JMP     CSRCVR4         ERROR RETURN
004746  203214  6990          LAC     BLOCK           INCREMENT BLOCK NUMBER
004747  346075  7000          TAD     (4)
004750  043214  7010          DAC     BLOCK
004751  762170  7020          LAW     TBUF            GET POINTER TO BUFFER
004752  043220  7030          DAC     BUFA            SET BUFFER ADDRESS
004753  776000  7040          LAW     -2000           GET COUNT
004754  043221  7050          DAC     BWS             SET
004755  623203  7060          RET     BUFIN
004756  203222  7070  LEND1   LAC     BOTM            COMPUTE PROPER OFFSET FOR BIG LOWER BOUND
004757  740001  7080          CMA
004760  346111  7090          TAD     (101)
004761  043233  7100          DAC     OFFSET
004762  223223  7110          LAC     CATP,X
004763  043242  7120          DAC     TPARAM
```

M LOADER COMMANDS

```
004764  203230  7130          LAC     LEN
004765  043244  7140          DAC     TPARAM+2        SET IT
004766  604734  7150          JMP     LEND2           EXIT LOAD END PROPERLY
004767  763242  7160    P2L   LAW     TPARAM          FLUSH CORE
004770  103172  7170          JMS     TPOT
   004771       7180    NXLT1 ...
004771  760003  7190          LAW     3               GET THREE
004772  343223  7200          TAD     CATP
004773  043240  7210          DAC     TEMP
004774  203245  7220          LAC     WT              GET LAST THING PROCESSED
004775  505141  7230          AND     INSTM           GET INSTRUCTION
004776  545140  7240          SAD     JMPW            CHECK FOR JMP
004777  741000  7250          SKP
005000  604424  7260          JMP     TCD             GET TCD
005001  203245  7270          LAC     WT              GET WORD
005002  103273  7280          JMS     CSPAPER         CHECK FOR PAPER TAPE
005003  604470  7290          JMP     PAPER           YES
005004  604440  7300          JMP     TCD1            SET IT
```

M LOADER COMMANDS

```
                    7310            ,EJECT
                    7320    *       PUNCH <TREE NAME>
                    7330    *
                    7340    *       PUNCH PUNCHES A HARDWARE LOADABLE BINARY TAPE OF THE FILE NAME
                    7350    *
005005  103266      7360    PUN     JMS     CSGNAME         GET A NAME
005006              7370            FORMAT                  FORMAT ERROR -- PAPER TAPE NOT LEGAL FOR THIS OPERATION
005007  103254      7380            JMS     CSCATL          LOOK IT UP
005010  603775      7390            JMP     NSAVE           NOT SAVED
005011  143232      7400            DZM     OUTDA           SET NO FILE FLAG FOR COPY
005012  103210      7410            JMS     LEADER          GET SOME LEADER
005013  220011      7420            LAC     SCATX,X         GET THE DEVICE ADDRESS
005014  043227      7430            DAC     INDA            SAVE FOR COPY
005015  440011      7440            ISZ     SCATX           BYPASS STARTING LOCATION
005016  220011      7450            LAC     SCATX,X         GET LENGTH TO COPY
005017  043230      7460            DAC     LEN
005020  220011      7470            LAC     SCATX,X         LOAD THE TRANSFER CARD
005021  042000      7480            DAC     TEMPO           SAVE IT FOR NOW
005022  143205      7490            DZM     PUNF            SET  FORMAT TO NORMAL
005023  203230      7500    BUFN    LAC     LEN             GET LENGTH NOW
005024  741200      7510            SNA
005025  605044      7520            JMP     PUNX            END OF BUFFERS
005026              7530            NEG                     MAKE 2'S COMPLEMENT
005030  043226      7540            DAC     COUNT           SAVE IT
005031  103170      7550            JMS     COPY            COPY FILE
005032  203230      7560            LAC     LEN             GET NEW LENGTH
005033  343226      7570            TAD     COUNT           FORM COUNT
005034  043226      7580            DAC     COUNT
005035  762167      7590            LAW     BUF-1           POINT TO BUFFER ADDRESS
005036  040012      7600            DAC     CMDX            SAVE IT
005037  220012      7610    PUNL    LAC     CMDX,X          GET A WORD
005040  103206      7620            JMS     PNCH            PUNCH IT
005041              7630            LOOP    COUNT,PUNL      LOOP ON COUNTER
005043  605023      7640            JMP     BUFN            GET SOME MORE TO PUNCH
                    7650    *
                    7660    *       END OF PUNCHING
                    7670    *
005044  443205      7680    PUNX    ISZ     PUNF            SET FLAG FOR PUNCH ROUTINE
005045  202000      7690            LAC     TEMPO           RECOVER THE TRANSFER CARD
005046  103206      7700            JMS     PNCH            PUNCH AS LAST WORD
005047  103210      7710            JMS     LEADER
005050  603714      7720            JMP     NEXTL           GET NEXT LINE
                    7730    *
                    7740    *       SUBROUTINE TO PUNCH A WORD OF BINARY
                    7750    *
003205              7760            .USE    IMPURE
003205  740040      7770    PUNF    XX                      FORMAT FLAG
005051              7780            .USE    PURE
005051              7790            ENTER   PNCH
                                    .PMC    SAVE,ON
003206                      PNCH    ...
005051  652000      7800            LMQ                     PUT ARGUMENT AWAY
```

M LOADER COMMANDS

```
                    7810            ,DUP    4,2
005052  640606      7820            LLS     6,
005053  700244      7830            PSB
005054  700201      7840            PSF
005055  605054      7850            JMP     .-1
005056  640606                      LLS     6,
005057  700244                      PSB
005060  700201                      PSF
005061  605060                      JMP     .-1
005062  760002      7860            LAW     2               GET A 2
005063  243205      7870            XOR     PUNF            GET FORMAT
005064  640606      7880            LLS     6,              FORM CHARACTER
005065  700204      7890            PSA                     PUNCH A CHARACTER
005066  700201      7900            PSF
005067  605066      7910            JMP     .-1
005070  623206      7920            RET     PNCH            RETURN
                    7930    *
                    7940    *       SUBROUTINE TO PUNCH LEADER/TRAILER
                    7950    *
    005071          7960            ENTER   LEADER
                                    ,PMC    SAVE,ON
    003210                  LEADER  ,,,
005071  777500      7970            LAW     -300            2COUNT FOR LEADER
005072  043226      7980            DAC     COUNT
    005073          7990    LEAD1   ,,,
005073  700214      8000            PSA+10                  PUNCH
005074  700201      8010            PSF
005075  605074      8020            JMP     .-1
    005076          8030            LOOP    COUNT,LEAD1
005100  623210      8040            RET     LEADER
                    8050    *
                    8060    *
                    8070    *       WRITE OUT THE BUFFER IF THE ALTERED FLAG IS ON
                    8080    *
    005101          8090            ENTER   FORCE
                                    ,PMC    SAVE,ON
    003212                  FORCE   ,,,
005101  202156      8100            LAC     DSBALT          LOAD THE ALTERS FLAG
005102  142156      8110            DZM     DSBALT          CLEAR IT ON GENERAL PRINCIPLES
005103  741200      8120            SNA                     SKIP IF IT WAS SET
005104  623212      8130            RET     FORCE           ELSE EXIT
005105  103251      8140            JMS     C$RCOVR         SET UP THE ERROR RECOVERY
005106  203212      8150            LAC     FORCE           LOAD THE RETURN
005107  652000      8160            LMQ                     PABS IT TO THE EXEC
005110  762153      8170            LAW     DSBDA           LOAD A POINTER TO THE PARAMETERS
005111  705005      8180            PWRITE                  WRITE OUT THE BUFFER
005112  605223      8190            JMP     C$RCVR4         IN CASE OF ERROR
```

M STORAGE

```
                    8200            .STITL  STORAGE
                    8210    *
     005113         8220    COMTB   ...
005113  435445      8230            435445
005114  604116      8240            JMP     CLE
005115  655663      8250            655663
005116  604163      8260            JMP     UNS
005117  606562      8270            606562
005120  604176      8280            JMP     PUR
005121  544465      8290            544465
005122  604203      8300            JMP     LDU
005123  700000      8310            700000
005124  604174      8320            JMP     EXIT
005125  457051      8330            457051
005126  604174      8340            JMP     EXIT
005127  634166      8350            634166
005130  604256      8360            JMP     SAV
005131  624560      8370            624560
005132  604307      8380            JMP     REP
005133  606556      8390            606556
005134  605005      8400            JMP     PUN
005135  456262      8410            .ACI6   /ERR/
005136  603734      8420            JMP     ERROR
     005137         8430    COME    ...
005137  040000      8440    DACW    DAC     0
005140  600000      8450    JMPW    JMP     0
005141  740000      8460    INSTM   740000
005142  000000      8470    ADD     0
     003214         8480            .USE    IMPURE
     003214         8490    BLOCK   ...
003214  000000      8500    BPARAM  .DATA   0,TBUF,TBFL
003215  002170
003216  001000
003217  000000      8510    BCOUNT  0
003220  000000      8520    BUFA    0
003221  000000      8530    BWS     0
003222  000000      8540    BOTM    0
003223  000000      8550    CATP    0
003224  000000      8560    CBASE   0
003225  000000      8570    CKSUM   0
003226  000000      8580    COUNT   0
003227  000000      8590    INDA    0
003230  000000      8600    LEN     0
003231  000000      8610    NP      0
003232  000000      8620    OUTDA   0
003233  000000      8630    OFFSET  0
003234  000000      8640    P2      0
003235  000000      8650    PARW    0
003236  000000      8660    PFLAG   0
003237  000000      8670    PTMP    0
003240  000000      8680    TEMP    0
003241  000000      8690    TOP     0
```

```
                M                                   STORAGE

    003242  000000  8700    TPARAM  .DATA   0,BUFFER,0
    003243  002170
    003244  000000
    003245  000000  8710    WT      0
        005143        8720            .USE    PURE
                      8730            .HEAD
                      8740            .INSRT  MTSSCAT
```

DESCRIPTION OF THE GROWTH SYSTEM CATALOG STRUCTURE

```
100             .STITL  DESCRIPTION OF THE GROWTH SYSTEM CATALOG STRUCTURE
110             .HEAD   C
120
130
140     *
150     *       MAJOR REVISION -- JAN 21, 1972 BY ROBERT W. BLEAN
160     *
170     *       A GROWTH CATALOG FOR A FILE-ORIENTED DEVICE IS LOCATED IN THE 400 WORDS
180     *       OF LOGICAL BLOCK 1 OF THE LOGICAL DEVICE. THIS PERMITS DISK AND DECTAPE
190     *       TO BE USED INTERCHANGEABLY BY THE GROWTH SYSTEM PROGRAMS.
200     *
210     *       THE DEVICE ADDRESS OF A HANDLER IS THE HANDLER NUMBER IN BITS 0-2
220     *       AND THE TYPE (DISK (1) OR DECTAPE (0)) IN BIT 3.
230     *
240     *       THE DEVICE ADDRESS OF A FILE IS THE DEVICE ADDRESS OF THE HANDLER IT
250     *       IS ON PLUS IN BITS 8-17 ITS STARTING BLOCK NUMBER.
260     *
270     *       ALL DEVICE ADDRESSES IN A DECTAPE CATALOG ARE CORRECT FOR THE HANDLER
280     *       THE TAPE WAS MOUNTED ON THE LAST TIME IT WAS ALTERED.
290     *
300     *       THE FIRST FOUR WORDS OF THE CATALOG BLOCK ARE A HEADER:
310     *            1) THE DEVICE ADDRESS OF THE FIRST FREE BLOCK ON THE DEVICE
320     *            2) POINTER TO THE FIRST FREE WORD IN THE CATALOG MINUS ONE PLUS THE CATALOG'S CORE ADDRESS
330     *            3) TWOS COMPLEMENT COUNT OF THE NUMBER OF FILES CATALOGED
340     *            4) TWOS COMPLEMENT MAXIMUM BLOCK NUMBER ON THE DEVICE
350     *
360     *       THE REMAINDER OF THE CATALOG CONSISTS OF A SERIES OF FIVE WORD FILE-
370     *       CONTROL BLOCKS, THE FIRST FILE CONTROL BLOCK IS FOR THE CATALOG ITSELF.
380     *       THEN THERE IS ONE FILE CONTROL BLOCK FOR EACH FILE ON THE DEVICE.
390     *
400     *       FORMAT OF THE FILE CONTROL BLOCKS:
410     *            1) THE FIRST WORD IS THE SIXBIT ASCII (EIGHTBIT ASCII - 240)
420     *               FILENAME. THIS MEANS THE FILENAME IS RESTRICTED TO THREE
430     *               CHARACTERS, WITH NO EXTENSION OR PASSWORD.
440     *            2) THE DEVICE ADDRESS OF THE FILE.
450     *            3) THE FILE'S CORE ADDRESS
460     *            4) THE FILE'S LENGTH (IN WORDS)
470     *            5) THE PROGRAM START
480     *
490     *       THIS LEAVES TWO WORDS OF THE CATALOG BLOCK UNUSED. IT IS SUGGESTED THAT
500     *       THE SECOND OF THESE CONTAIN THE BLOCK NUMBER OF A CONTINUATION OF THE
510     *       CATALOG, SHOULD THIS EVER BE NECESSARY; IT WOULD BE ZERO IF THERE
520     *       IS NO CONTINUED CATALOG BLOCK.
```

```
C                                       GROWTH SYSTEM STANDARD CATALOG ROUTINES

                    530                 ,STITL  GROWTH SYSTEM STANDARD CATALOG ROUTINES
     003246         540                 .USE    IMPURE
                    550
003246  000000      560         CTEM1   ,DSA                    CATALOG ROUTINE'S PRIVATE TEMP
     005143         570                 .USE    PURE
                    580
                    590
                    600         *
                    610         *       RCAT -- THE BASIC CATALOG ROUTINE, IT READS IN CATALOGS AND UPDATES THEM
                    620         *       FOR THE CURRENT DEVICE AND (POSSIBLY NON-STANDARD) CATALOG LOCATION.
                    630         *
                    640         *
                    650         *       A HANDLER DEVICE ADDRESS IS PASSED IN THE AC. THE REQUESTED
                    660         *       CATALOG IS READ IN AND ALL OF THE DEVICE ADDRESSES ARE UPDATED.
                    670         *       AS A RESULT, THE CATALOG IN CORE ALWAYS HAS THE PROPER DEVICE ADDRESSES
                    680         *       FOR THE DEVICE IT WAS READ FROM.
                    690         *
                    700         *       RETURN IS +1 WHEN THE DESIRED CATALOG IS IN CORE.
                    710         *
                    720         *       IN THE EVENT OF UNRECOVERABLE ERROR, EXIT IS TO AN ERROR ROUTINE.
                    730         *
     005143         740                 ENTER   RCAT
                                        ,PMC    SAVE,ON
     003247                     RCAT    ...
005143  043246      750                 DAC     CTEM1           SAVE THE DEVICE ADDRESS OF THE DEVICE WHOSE CATALOG IS BEING REQUESTED
005144  103212      760                 JMS     M$FORCE         FORCE THE OLD BUFFER BEFORE READING A NEW ONE
                    770
     005145         780         RCAT1   ...
005145  203246      790                 LAC     CTEM1
005146  506112      800                 AND     (DVCMSK)        GET THE NEW HANDLER'S DEVICE ADDRESS
005147  042153      810                 DAC     D$BDA           SET THE HANDLER DEVICE ADDRESS
005150  506072      820                 AND     (040000)        RECOVER JUST THE DISK/DECTAPE BIT
005151  740200      830                 SZA                     SKIP FOR DECTAPE
005152  206113      840                 LAC     ($SYSBAS)       ELSE LOAD THE DISK SYSTEM BASE ADDRESS
005153  242153      850                 XOR     D$BDA           ADD THE BASE ADDRESS INTO THE HANDLER DEVICE ADDRESS
005154  246071      860                 XOR     ($CATBLK)       ADD IN THE CATALOG BLOCK NUMBER
005155  042153      870                 DAC     D$BDA           SAVE THE NEW CATALOG'S DEVICE ADDRESS
005156  762170      880                 LAW     BUFFER          LOAD A POINTER TO THE BUFFER
005157  042154      890                 DAC     D$BCA           SET IT AS THE CORE ADDRESS
005160  206074      900                 LAC     (400)           LOAD THE LENGTH
005161  042155      910                 DAC     D$BLEN          SET IT AS THE BUFFER LENGTH
005162  103251      920                 JMS     C$RCOVR         SET UP THE ERROR RECOVERY
005163  765170      930                 LAW     RCAT3
005164  652000      940                 LMQ                     LOAD THE RESTART ADDRESS
005165  762153      950                 LAW     D$BDA           GET A POINTER TO THE CATALOG PARAMETERS
005166  705003      960                 PREAD                   READ IN THE NEW CATALOG
005167  605223      970                 JMP     RCVR4           IN CASE OF ERROR
                    980         *
                    990         *       NOW UPDATE THE DEVICE ADDRESSES
                   1000         *
005170  203246     1010         RCAT3   LAC     CTEM1
005171  506112     1020                 AND     (DVCMSK)
```

```
                C                       GROWTH SYSTEM STANDARD CATALOG ROUTINES

005172  043246  1030            DAC     CTEM1           SET THE CURRENT DEVICE ADDRESS
                1040
005173  202170  1050            LAC     CATLOG
005174  506114  1060            AND     (BLKMSK)
005175  243246  1070            XOR     CTEM1
005176  042170  1080            DAC     CATLOG          UPDATE THE OLD DEVICE ADDRESS OF THE FIRST FREE BLOCK
                1090
005177  762175  1100            LAW     CATLOG+5
005200  043212  1110            DAC     MSFORCE
005201  043254  1120            DAC     CATL            SET POINTERS TO THE FIRST OLD DEVICE ADDRESS
005202  202172  1130            LAC     CATLOG+2
005203  043251  1140            DAC     RCOVR           SET THE COUNT OF FCB'S
                1150
005204  223212  1160    RCAT4   LAC     MSFORCE,X       LOAD THE NEXT OLD DEVICE ADDRESS
005205  506114  1170            AND     (BLKMSK)        RECOVER THE BLOCK NUMBER
005206  243246  1180            XOR     CTEM1           ADD IN THE CURRENT HANDLER DEVICE ADDRESS
005207  063254  1190            DAC     CATL,X          SAVE THE UPDATED FILE DEVICE ADDRESS
                1200
005210  443251  1210            ISZ     RCOVR           COUNT THE FILES DONE
005211  741000  1220            SKP
005212  623247  1230            JMP     RCAT,X          ALL DONE
                1240
005213  203212  1250            LAC     MSFORCE         LOAD THE FCB POINTER
005214  346115  1260            TAD     (FCBLEN)        ADVANCE IT TO THE NEXT FCB
005215  043212  1270            DAC     MSFORCE
005216  043254  1280            DAC     CATL            SAVE THE NEW POINTER
005217  605204  1290            JMP     RCAT4           LOOP
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
                    1300            .EJECT
                    1310
                    1320
                    1330
     005220         1340            ENTER   RCOVR           SUBROUTINE TO SET UP RECOVERY FROM HARDWARE ERRORS
                                    .PMC    SAVE,ON
     003251                 RCOVR    ...
005220   777776     1350            LAW     -2              SET FOR FIVE RETRIES BEFORE GIVING UP
005221   043253     1360            DAC     ERCNT
005222   623251     1370            JMP     RCOVR,X
                    1380
     005223         1390    RCVR4   MESS    <DEVICE ERROR>,12.
005233   443253     1400            ISZ     ERCNT           COUNT THE ERROR
005234   623251     1410            JMP     RCOVR,X
     005235         1420    RCVR5   MESS    <TYPE 'IGNORE' OR 'CONTINUE': >,29.
     005253         1430            LINE                    GET THE USER'S ANSWER TO WHAT HE WANTS TO DO ABOUT IT
     005254         1440            WORD                    READ HIS ANSWER
005255   605235     1450            JMP     RCVR5           NO INPUT IS ILLEGAL
005256   545263     1460            SAD     IGN
005257   603714     1470            JMP     $NEXTL          IGNORE THE LAST COMMAND
005260   545264     1480            SAD     CON
005261   603252     1490            JMP     RCOVR+1         SET UP TO TRY AGAIN
005262   605235     1500            JMP     RCVR5           ANY OTHER ANSWER IS ILLEGAL
     003253         1510            .USE    IMPURE
003253   000000     1520    ERCNT   .DSA
     005263         1530            .USE    PURE
005263   514756     1540    IGN     .ACI6   'IGN'
005264   435756     1550    CON     .ACI6   'CON'
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
                    1560            .EJECT
                    1570    *
                    1580    *       CATL
                    1590    *
                    1600    *       CATL SEARCHES THE CATALOG IN CORE FOR THE FILENAME
                    1610    *       PASSED IN THE AC
                    1620    *
                    1630    *       RETURN +2 WITH CATX POINTING TO THE FILE NAME IF SUCCESSFUL
                    1640    *
                    1650    *       RETURN +1 WITH CATX POINTING TO THE FIRST FREE SPACE -1 IN THE
                    1660    *       CATALOG IF THE FILE NAME IS NOT FOUND
                    1670    *
    005265          1680            ENTER   CATL
                                    .PMC    SAVE,ON
    003254                  CATL    ...
005265  043301      1690            DAC     T$WORDB         SAVE CATALOG NAME
                    1700    *
                    1710    *       FIRST CHECK WHETHER OR NOT THIS IS A SPECIAL FILE
                    1720    *
005266  203270      1730            LAC     CDFLG           LOAD THE CORE/DISK SPECIAL FILE FLAG
005267  741200      1740            SNA                     SKIP IF IT IS SET
005270  605312      1750            JMP     CATL1           NO -- THEREFORE IT IS A NORMAL FILE
                    1760    *
                    1770    *       FIND OUT WHICH KIND OF SPECIAL FILE WE ARE TALKING ABOUT
                    1780    *
    005271          1790            WORD1
005272  543256      1800            SAD     CORE
005273  605331      1810            JMP     CORE1           IT IS THE USER CORE FILE
005274  543262      1820            SAD     DISK
005275  605343      1830            JMP     DISK1           IT IS THE USER DISK FILE
    005276          1840            MESS    <ILLEGAL SPECIAL FILE>;20,
005311  603714      1850            JMP     MSNEXTL         GET THE NEXT COMMAND
                    1860    *
                    1870    *       NEXT CHECK FOR NORMAL FILES
                    1880    *
005312  762173      1890    CATL1   LAW     CATLOG+3
005313  040011      1900            DAC     $CATX           SET A POINTER TO THE FIRST FCB IN THE CATALOG AUTO-INDEX REGISTER
005314  202172      1910            LAC     CATLOG+2        GET CATALOG COUNT
005315  043246      1920            DAC     CTEM1           SAVE IT
    005316          1930    CATLL   WORD1                   RESTORE NAME TO SEARCH FOR
005317  560011      1940            SAD     $CATX,X         CHECK IT
005320  605327      1950            JMP     CATL9           FOUND IT
005321  200011      1960            LAC     $CATX
005322  346075      1970            TAD     (FCBLEN-1)      FAILED -- MOVE THE POINTER TO THE NEXT FILE CONTROL BLOCK
005323  040011      1980            DAC     $CATX
005324  443246      1990            ISZ     CTEM1           COUNT
005325  605316      2000            JMP     CATLL           LOOP
005326  623254      2010            JMP     CATL,X          EXHAUSTED, NO FILE FOUND -- BAD RETURN
005327  443254      2020    CATL9   ISZ     CATL            GOOD RETURN
005330  623254      2030            JMP     CATL,X
                    2040    *
                    2050    *       SPECIAL CATALOG AND ROUTINES  FOR THE USER CORE IMAGE
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
                    2060    *
    003256          2070            ,USE    IMPURE

003256  435762      2080    CORE    ,AC16   *COR*
003257  000000      2090    CORDA   ,DSA                    DISK ADDRESS OF THE USER CORE IMAGE
003260  002000      2100    CORCA   BOUNDARY                STARTING CORE ADDRESS OF THE USER CORE
003261  014000      2110    CORLN   CORMAX-BOUNDARY         LENGTH OF USER CORE
                    2120
    005331          2130            ,USE    PURE
    005331          2140    CORE1   ..,                     SET UP THE USER CORE IMAGE AS A FILE
    005331          2150            MPOFF
                                    ,PMC    SAVE,ON
005331  705000                      SPECIAL+0               TURN OFF MEMORY PROTECT
005332  201766      2160            LAC     $UCORE          LOAD THE USER CORE IMAGE DISK ADDRESS
005333  744000      2170            CLL                     PROTECT THE SHIFT
005334  640510      2180            LRS     8,              MAKE THE PHYSICAL ADDRESS INTO A BLOCK ADDRESS
005335  246072      2190            XOR     (040000)        SET THE DISK BIT ON
005336  043257      2200            DAC     CORDA           SET IT IN THE TEMPORARY CATALOG
005337  701742      2210            MPEU                    RE-ENABLE USER MODE
005340  763256      2220            LAW     CORE            LOAD A POINTER TO THE CATALOG
005341  040011      2230            DAC     $CATX           AND PASS IT TO THE CALLER
005342  605327      2240            JMP     CATL9           EXIT
                    2250    *
                    2260    *       SPECIAL CATALOG AND ROUTINES FOR  THE USER "PHYSICAL DISK"
                    2270    *
    003262          2280            ,USE    IMPURE
003262  445163      2290    DISK    ,AC16   *DIS*
003263  000000      2300    DISDA   ,DSA                    DISK ADDRESS OF THE USER "PHYSICAL DISK
003264  000000      2310    DISCA   0                       MINIMUM USER "PHYSICAL DISK" ADDRESS
003265  016000      2320    DISLN   $DKLEN                  LENGTH OF THE USER "PHYSICAL DISK"
                    2330
    005343          2340            ,USE    PURE
    005343          2350    DISK1   ..,
    005343          2360            MPOFF
                                    ,PMC    SAVE,ON
005343  705000                      SPECIAL+0               TURN OFF MEMORY PROTECT
005344  201767      2370            LAC     $UDISK          LOAD THE USER "PHYSICAL DISK" DISK ADDRESS
005345  744000      2380            CLL                     PROTECT THE SHIFT
005346  640510      2390            LRS     8,              MAKE THE PHYSICAL ADDRESS INTO A DISK ADDRESS
005347  246072      2400            XOR     (040000)        SET THE DISK BIT ON
005350  043263      2410            DAC     DISDA           AND SET IT IN THE TEMPORARY CATALOG
005351  701742      2420            MPEU                    RE-ENABLE USER MODE
005352  763262      2430            LAW     DISK            LOAD A POINTER TO THE CATALOG
005353  040011      2440            DAC     $CATX           PASS IT TO THE CALLER
005354  605327      2450            JMP     CATL9           EXIT
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
                    2460            .EJECT
                    2470    *
                    2480    *       GNAME
                    2490    *
                    2500    *       GNAME GETS A FILE NAME FROM THE TTY BUFFER
                    2510    *       AND READS IN THE CATALOG IF NECESSARY
                    2520    *
                    2530    *       RETURN IS +1 FOR PAPER TAPE DESIRED
                    2540    *       RETURN IS +2 FOR SUCCESS ON DISK OR DECTAPE
                    2550    *       OTHERWISE EXIT IS TO FORMAT ERROR
                    2560    *
                    2570    *       THE FILE NAME IS RETURNED IN T$WORDB AND IN THE AC.
                    2580    *
    005355          2590            ENTER   GNAME
                                    .PMC    SAVE,ON
    003266                  GNAME   ...
005355  143270      2600            DZM     CDFLG           INITIALIZE THE SPECIAL FILE FLAG
    005356          2610            WORD                    GET A WORD OF SIX BIT ASCII
005357  605412      2620            JMP     GNAM90          CHECK FOR A SPECIAL FILE IF A DELIMITER IS FIRST CHARACTER
    005360          2630            DELIM                   GET THE DELIMITER
005361  546116      2640            SAD     ($COLON)        CHECK FOR COLON
005362  605366      2650            JMP     GNAM2
005363  103273      2660    GNAM1   JMS     PAPER           CHECK FOR PAPER TAPE
005364  623266      2670            JMP     GNAME,X         YES -- PAPER TAPE
005365  605376      2680            JMP     GNAM5           NO -- SO USE CURRENT CATALOG
                    2690
005366  765372      2700    GNAM2   LAW     GNAM3
005367  043271      2710            DAC     DEVCV
    005370          2720            WORD1                   RELOAD THE CATALOG NAME
005371  605426      2730            JMP     DEVC3           CONVERT IT TO A DEVICE ADDRESS
005372  623266      2740    GNAM3   JMP     GNAME,X
005373  103247      2750            JMS     RCAT            READ IN THE CATALOG
    005374          2760            WORD                    GET ANOTHER WORD
005375  740000      2770            NOP
    005376          2780    GNAM5   DELIM                   GET THE DELIMITER
005377  546117      2790            SAD     ($SLASH)        CHECK FOR SLASH
005400  605405      2800            JMP     GNAME6          LOOK FOR OCTAL
    005401          2810            WORD1                   ELSE RECOVER THE SIXBIT NAME
005402  741200      2820            SNA                     CHECK FOR ALL SPACES
    005403          2830            FORMAT                  FORMAT ERROR -- ALL SPACES IS AN ILLEGAL NAME
005404  605410      2840            JMP     GNAM8
    005405          2850    GNAME6  NUM                     GET THE NUMBER
    005406          2860            FORMAT

005407  043301      2870            DAC     T$WORDB         TO BE COMPATABLE WITH SIXBIT INPUT
005410  443266      2880    GNAM8   ISZ     GNAME           GOOD RETURN
005411  623266      2890            JMP     GNAME,X
                    2900    *
                    2910    *
                    2920    *       CHECK FOR A SPECIAL FILE REQUEST (E.G. 'CORE' OR 'DISK')
                    2930    *
    003270          2940            .USE    IMPURE
003270  000000      2950    CDFLG   .DSA                    FLAG FOR PRESENCE OF SPECIAL FILE REQUEST
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
005412          2960            .USE    PURE
                2970
005412          2980    GNAM90  ...
005412          2990            DELIM                   RECOVER THE DELIMITER
005413  546116  3000            SAD     ($COLON)        CHECK FOR A VACUOUS COLON
005414  741000  3010            SKP                     YES -- IT IS A SPECIAL FILE
005415  605363  3020            JMP     GNAM1           NO -- RETURN TO NORMAL PROCESSING
                3030
005416  206120  3040            LAC     ($DK0)          LOAD THE IMPLIED SYSTEM DISK MNEMONIC
005417  043301  3050            DAC     TSWORDB         FAKE THAT IT WAS TYPED
005420  443270  3060            INX     CDFLG           FLAG THE SPECIAL FILE REQUEST
005421  605366  3070            JMP     GNAM2           RESUME NORMAL PROCESSING OF THE FAKED INPUT
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
                    3080            .EJECT
                    3090    *
                    3100    *       DEVCV -- READS THE NEXT WORD,
                    3110    *       RETURN IS +1 WITH THE NAME IN THE AC IF IT IS A PAPER TAPE CALL
                    3120    *
                    3130    *       OTHERWISE IT ATTEMPTS TO CONVERT THE NAME TO DEVICE ADDRESS FORMAT.
                    3140    *       IF SUCCESSFUL, IT RETURNS +2 WITH THE HANDLER NUMBER IN AC(0-2) AND
                    3150    *       THE DEVICE TYPE (DISK (1) OR DECTAPE (0)) IN AC(3). REMAINING BITS
                    3160    *       ARE ZEROED.
                    3170    *
                    3180    *       EXIT IS TO THE FORMAT ERROR MESSAGE IF THE DEVICE IS NEITHER PAPER TAPE
                    3190    *       NOR DISK NOR DECTAPE.
                    3200    *
    005422          3210            ENTER   DEVCV
                                    .PMC    SAVE,ON
    003271                  DEVCV   ...
    005422          3220            WORD                    GET THE DEVICE NAME
    005423          3230            FORMAT
005424  103273      3240            JMS     PAPER           CHECK FOR PAPER TAPE
005425  623271      3250            JMP     DEVCV,X         YES -- PAPER TAPE
005426  506121      3260    DEVC3   AND     (777700)        REMOVE DEVICE NUMBER
005427  546122      3270            SAD     ($TP,)          CHECK FOR DECTAPE
005430  605440      3280            JMP     DEVC1           YES
005431  546123      3290            SAD     ($DT,)          CHECK FOR DECTAPE
005432  605440      3300            JMP     DEVC1
005433  546124      3310            SAD     ($DK,)          CHECK FOR DISK
005434  605446      3320            JMP     DEVC4           IT IS DISK -- CKECK FOR VALIDATION
005435  603723      3330            JMP     M$MONX2         NO OTHERS -- MAYBE IT IS A COMMAND
005436  650004      3340    DEVC35  CLQ!CMQ                 FOR DISK PUT THE SIGN BIT ON IN THE MQ
005437  741000      3350            SKP
005440  650000      3360    DEVC1   CLQ                     CLEAR Q FOR TAPE
    005441          3370            WORD1                   RESTORE NAME
005442  640617      3380            LLS     18.-3           SHIFT TO POSITION
005443  506112      3390            AND     (DVCMSK)        CONVERT TO HANDLER DEVICE ADDRESS FORMAT
005444  443271      3400            ISZ     DEVCV           INCREMENT RETURN
005445  623271      3410            JMP     DEVCV,X         AND NOW RETURN
                    3420    *
                    3430    *
                    3440    *       DISK FILE OPERATIONS ARE PERMITTED ONLY FOR VALIDATED USERS
                    3450    *
    005446          3460    DEVC4   ...
    005446          3470            MPOFF
                                    .PMC    SAVE,ON
005446  705000                      SPECIAL+0               TURN OFF MEMORY PROTECT
005447  201770      3480            LAC     SVALID          LOAD THE USER'S VALIDATION WORD
005450  701742      3490            MPEU
005451  740200      3500            SZA                     SKIP IF THE USER IS NOT VALIDATED
005452  605436      3510            JMP     DEVC35          ELSE THE OPERATION CAN PROCEED
                    3520    *
                    3530    *       CHECK FOR A SPECIAL FILE OPERATION -- IF SO, IT IS ALLOWED
                    3540    *
005453  203270      3550            LAC     CDFLG           LOAD THE SPECIAL FILE FLAG
```

```
C                       GROWTH SYSTEM STANDARD CATALOG ROUTINES

005454  740200  3560            SZA                     SKIP IF NOT SET -- THEN THE OPERATION IS ILLEGAL
005455  605436  3570            JMP     DEVC35          IT IS A SPECIAL FILE OPERATION, SO ALLOW IT
                3580    *
                3590    *       DISK OPERATION IS ILLEGAL
                3600    *
    005456      3610            MESS    <DISK OPERATION IS FORBIDDEN>,27,
005473  603714  3620            JMP     MSNEXTL         GET THE NEXT COMMAND LINE
                3630    *
                3640    *       PAPER CHECKS THE AC FOR A PAPER TAPE MNEMONIC. IT RETURNS +1 IF IT
                3650    *       FINDS ONE, ELSE RETURNS +2. THE AC IS UNCHANGED.
                3660    *
    005474      3670            ENTER   PAPER
                                ,PMC    SAVE,ON
    003273              PAPER   ...
005474  546125  3680            SAD     ($PPT)
005475  623273  3690            JMP     PAPER,X
005476  546126  3700            SAD     ($PTR)
005477  623273  3710            JMP     PAPER,X
005500  546127  3720            SAD     ($PTP)
005501  623273  3730            JMP     PAPER,X
005502  443273  3740            ISZ     PAPER           NO PAPER TAPE MNEMONIC
005503  623273  3750            JMP     PAPER,X
```

C GROWTH SYSTEM STANDARD CATALOG ROUTINES

```
                    3760            ,EJECT
                    3770    *
                    3780    *       SAVE CHECKS THE CATALOG FOR THE NAME FOUND IN THE AC
                    3790    *
                    3800    *       RETURN IS +1 IF THE FILE IS ALREADY SAVED
                    3810    *       A CATALOG ENTRY IS CREATED FOR THE NAME AND RETURN IS +2 OTHERWISE
                    3820    *       EXITS TO AN ERROR MESSAGE IF THE CATALOG IS FULL
                    3830    *
                    3840    *       ON RETURN CATX POINTS TO THE FILE NAME IN THE CATALOG
                    3850    *
    005504          3860            ENTER   SAVE
                                    .PMC    SAVE,ON
    003275                  SAVE    ...
005504  103254      3870            JMS     CATL            LOOK UP NAME
005505  741000      3880            SKP
005506  623275      3890            JMP     SAVE,X          DON'T ALLOW DUPLICATES
005507  202172      3900            LAC     CATLOG+2        LOAD THE FCB COUNT
005510  546130      3910            SAD     (CATMAX)        CHECK FOR CATALOG ALREADY FULL
005511  605524      3920            JMP     CFULL           YES -- EXIT TO ERROR MESSAGE
005512  346131      3930            TAD     (-1)            COUNT THE NEW FILE
005513  042172      3940            DAC     CATLOG+2        UPDATE THE FCB COUNT
005514  202171      3950            LAC     CATLOG+1        GET FREE POINTER
005515  346115      3960            TAD     (FCBLEN)        ADD ONE FILE CONTROL BLOCK LENGTH FOR THE NEW ENTRY
005516  042171      3970            DAC     CATLOG+1
    005517          3980            WORD1                   RECOVER THE FILE NAME
005520  060011      3990            DAC     $CATX,X         SAVE IT
005521  442156      4000            ISZ     DSBALT          FLAG THE CATALOG HAS BEEN ALTERED
005522  443275      4010            ISZ     SAVE
005523  623275      4020            JMP     SAVE,X
                    4030
    005524          4040    CFULL   MESS    <CATALOG FULL>,12.
005534  603714      4050            JMP     $NEXTL
                    4060    *
                    4070    *       ALC RECEIVES A WORD COUNT IN THE AC AND CALCULATES THE LEAST INTEGER
                    4080    *       NUMBER OF BLOCKS THAT CAN HOLD THAT LENGTH. IT THEN ALLOCATES THE STORAGE
                    4090    *       IN THE CORE CATALOG HEADER AND RETURNS WITH THE DEVICE ADDRESS OF THE
                    4100    *       FIRST FREE BLOCK IN THE AC.
                    4110    *
                    4120    *       EXIT IS TO AN ERROR MESSAGE IF THIS ALLOCATION WOULD RESULT IN
                    4130    *       OVERFLOWING THE DEVICE. IN THIS CASE THE CATALOG IS UNALTERED.
                    4140    *
    005535          4150            ENTER   ALC
                                    .PMC    SAVE,ON
    003277                  ALC     ...
005535  346103      4160            TAD     (377)           ROUND UP TO A BLOCK
005536  660510      4170            LRSS    8.              AC = MINIMUM INTEGER NUMBER OF BLOCKS REQUIRED
005537  043246      4180            DAC     CTEM1           SAVE IN A GOOD RANDOM PLACE
005540  202170      4190            LAC     CATLOG          GET THE POINTER TO THE FIRST FREE BLOCK
005541  652000      4200            LMQ                     SAVE IT
005542  343246      4210            TAD     CTEM1           ADD THE REQUESTED NUMBER OF BLOCKS TO FORM A NEW POINTER
005543  043246      4220            DAC     CTEM1           SAVE THE NEW POINTER
005544  506114      4230            AND     (1777)          EXTRACT BLOCK NUMBER
```

```
C                                   GROWTH SYSTEM STANDARD CATALOG ROUTINES

005545  342173  4240            TAD     CATLOG+3        SEE IF WE OVERFLOWED THE DEVICE
005546  740100  4250            SMA                     NO IF SKP
005547  605554  4260            JMP     DFULL           FULL -- HELP*?!@
005550  203246  4270            LAC     CTEM1
005551  042170  4280            DAC     CATLOG          SET THE FREE FCB POINTER NOW WE KNOW IT WILL BE OK
005552  641002  4290            LACQ                    RESTORE THE DEVICE ADDRESS OF THE FIRST FREE BLOCK
005553  623277  4300            JMP     ALC,X
                4310
005554          4320    DFULL   MESS    <DEVICE FULL>,11.
005564  603714  4330            JMP     $NEXTL
                4340            .END
                8750            .HEAD
                8760            .INSRT  :DLIBRARY:PDP9LIB:TTYNON
                100             .INE    $DEBUG,1
                120             .IFE    $DEBUG,1
```

MTSS-PDP9 NON-INTERRUPTING TELETYPE HANDLER

```
130             .STITL  MTSS-PDP9 NON-INTERRUPTING TELETYPE HANDLER
140             .HEAD   T
150     *
160     *
170     *       PROGRAMMED BY ROBERT W. BLEAN
180     *
190     *
200     *       LAST REVISED 24 MARCH 1972
210     *
220     *
230     *       THIS HANDLER PERMITS NON-INTERRUPT DRIVEN INPUT FROM AND OUTPUT
240     *       TO THE CONSOLE TELETYPE ON THE PDP-9 COMPUTER.
250     *
260     *       THIS HANDLER ALTERS THE  AC, AND MQ. IT DOES NOT ALTER ANY CORE
270     *       MEMORY OUTSIDE OF ITSELF. IN PARTICULAR IT DOES NOT ALTER ANY AUTO-INDEX REGISTER.
280     *
290     *       DATA FORMATS:
300     *
310     *            1) OCTAL
320     *
330     *            2) SIXBIT -- SIXBIT IS 8-BIT ASCII MINUS 240. THIS MAPS THE PRINTING
340     *               CHARACTERS ONTO THE SET 0-77. ASCII VALUE 333 ([) IS USED FOR
350     *               CARRIAGE RETURN AND 335 (]) IS USED FOR LINEFEED. NOTE THAT NEITHER
360     *               333, 335, NOR ANY CONTROL CHARACTERS CAN BE RECOGNIZED IN SIXBIT.
370     *
380     *            3) ASCII -- ONE ASCII CHARACTER IS STORED PER WORD. LINE INPUT
390     *               IS STORED IN THIS FORMAT, SINCE THERE IS ONLY ONE LINE-BUFFER
400     *               THE EXTRA BUFFER LENGTH WASTES LESS SPACE THAN WOULD THE HANDLING
410     *               ROUTINES NECESSARY FOR OTHER FORMS OF PACKING CHHRACTERS.
```

```
T                                   (MTSS TELETYPE HANDLER) STORAGE AREA

                    420             .STITL  (MTSS TELETYPE HANDLER) STORAGE AREA
                    430             .IFE    PURCOD,1
          003301    440             .USE    IMPURE
                    450
                    460
          003301    470     WORDB   .BLOCK  2               ROOM TO ACCUMULATE TWO VALID WORDS
          000120    480     STD     .EQU    80.             STANDARD IS AN 80-CHARACTER LINE BUFFER
          003303    490     BUFFR   .BLOCK  STD
                    500     *
                    510     *
                    520     *       VARIABLES
                    530     *
 003423   003422    540     BEND    .-1                     END OF THE CHARACTER BUFFER
 003424   000000    550     BPTR    .DSA                    POINTER TO CURRENTLY ACTIVE WORD IN LINE BUFFER
 003425   000000    560     T1      .DSA                    TEMPORARY VARIABLE
 003426   000000    570     T2      .DSA                    TEMPORARY VARIABLE
 003427   000000    580     CHAR    .DSA                    STORES LATEST CHARACTER FROM FGET
 003430   000000    590     DLMTR   .DSA                    STORES LATEST DELIMITER THROUGH CHRID
 003431   000000    600     COUNT   .DSA
                    610             .IFE    PURCOD,1
          005565    620             .USE    PURE
```

```
T                       (MTSS TELETYPE HANDLER) LINE BUFFER INPUT

                  630           .STITL  (MTSS TELETYPE HANDLER) LINE BUFFER INPUT
                  640
                  650
                  660     *
                  670     *     THE PROGRAM IS PROTECTED AGAINST OVERFLOW OR UNDERFLOW OF THE LINE
                  680     *     BUFFER. UNDERFLOW (EXCESS DELETIONS) IS IGNORED, AND OVERFLOW CHARACTERS
                  690     *     ARE LOST, EXCEPT FOR THE LAST CHARACTER TYPED.
                  700     *
                  710
     005565       720           ENTER   INLIN           SUBROUTINE TO READ IN AND BUFFER A LINE FROM THE TELETYPE
                                .PMC    SAVE,ON
     003432                     INLIN   ...
005565  700312    730           KRB                     ONCE, ON ENTRANCE, CLEAN UP ANY PRIOR INPUT
005566  206132    740     INL   LAC     (BUFFR-1)       LOAD A POINTER TO START OF THE BUFFER MINUS ONE
005567  043424    750           DAC     BPTR            INITIALIZE THE BUFFER POINTER
005570  143431    760           DZM     COUNT           INITIALIZE THE WORD FETCHED COUNT
005571  143430    770           DZM     DLMTR           INITIALIZE THE LAST DELIMITER STORAGE
005572  700313    780     IN1   KSF!KRB                 GET THE NEXT INPUT CHARACTER
005573  605572    790           JMP     .-1
005574  546133    800           SAD     ($BKARR)
005575  605617    810           JMP     1CHAR           DELETE ONE CHARACTER IF IT WAS A BACKARROW
005576  546134    820           SAD     ($CONTX)
005577  605615    830           JMP     1LINE           DELETE THE ENTIRE LINE IF IT WAS A CONTROL X
005600  652000    840     IN4   LMQ                     SAVE THE CHARACTER
005601  203424    850           LAC     BPTR            LOAD THE CURRENT BUFFER POINTER
005602  543423    860           SAD     BEND            SKIP IF NO OVERFLOW
005603  741000    870           SKP                     AVOID DAMAGE DUE TO OVERFLOW
005604  443424    880           ISZ     BPTR            ADVANCE THE POINTER -- IT IS STILL WITHIN THE BUFFER
005605  641002    890           LACQ                    RELOAD THE CHARACTER
005606  063424    900           DAC     BPTR,X          AND PUT IT IN THE BUFFER
005607  546135    910           SAD     ($CR)
005610  741000    920           SKP                     EXIT WHEN A CARRIAGE RETURN IS FOUND
005611  605572    930           JMP     IN1             ELSE GET THE NEXT CHARACTER
005612  763302    940           LAW     BUFFR-1
005613  043424    950           DAC     BPTR            RESET THE BUFFER POINTER AT THE END OF THE LINE
005614  623432    960           JMP     INLIN,X         AND RETURN TO THE CALLER
                  970
005615  103460    980     1LINE JMS     CRLF            PRINT THE RESPONSE TO A LINE-DELETE
005616  605566    990           JMP     INL             REREAD THE LINE
005617  203424   1000     1CHAR LAC     BPTR            LOAD THE BUFFER POINTER
005620  545566   1010           SAD     INL             SKIP IF NO UNDERFLOW
005621  605572   1020           JMP     IN1             ELSE IGNORE THE COMMAND
005622  346131   1030           TAD     (-1)            DECREMENT THE BUFFER POINTER
005623  043424   1040           DAC     BPTR            AND SAVE IT
005624  605572   1050           JMP     IN1             GET THE NEXT CHARACTER
```

```
T                       (MTSS TELETYPE HANDLER) OCTAL WORD INPUT/OUTPUT

                1060            .STITL  (MTSS TELETYPE HANDLER) OCTAL WORD INPUT/OUTPUT
                1070
                1080    *
                1090    *       OPERATION RETURN L   AC     MQ    MEANING
                1100    *       --------- ------ - ------ ------ --------------------------------
                1110    *       INPUT       +1   0   X      X    FORMAT ERROR DISCOVERED
                1120    *                   +1   1 DELIM    X    FIRST NON-BLANK CHARACTER IS A DELIMITER
                1130    *                   +2   1 OCTAL  DELIM  SUCCESSFUL READ OF AN OCTAL NUMBER
                1140    *       OUTPUT      +1   X   X      X    SUCCESSFUL WRITE OF AN OCTAL NUMBER
                1150    *
                1160
    005625      1170            ENTER   NUMIN
                                .PMC    SAVE,ON
    003434              NUMIN   ...
005625  143426  1180            DZM     T2              INITIALIZE THE DECIMAL-DIGIT-RECEIVED FLAG
005626  103452  1190            JMS     INTIN           INITIALIZE THE INPUT STRING, ETC
005627  623434  1200            JMP     NUMIN,X         RETURN +1 FOR DELIMITER IS FIRST NON-BLANK CHARACTER
005630  103450  1210    NUM20   JMS     FGET            GET THE NEXT CHARACTER
005631  103454  1220            JMS     CHRID           IDENTIFY IT
005632  605654  1230            JMP     NUM26           IT IS A DELIMITER, SO EXIT
005633  623434  1240            JMP     NUMIN,X         IT IS A LETTER, SO EXIT +1 FOR A FORMAT ERROR
005634  741400  1250            SZL                     SKIP IF THE CHARACTER IS AN OCTAL DIGIT
005635  443426  1260            ISZ     T2              ELSE BE SURE THE DECIMAL-DIGIT-RECEIVED FLAG IS SET
005636  506136  1270            AND     (17)            RETAIN JUST THE DIGIT
005637  043425  1280            DAC     T1              AND SAVE IT FOR DECIMAL ACCUMULATION
                1290
005640  640503  1300            LRS     3               SAVE THE "OCTAL DIGIT"
005641  203301  1310            LAC     WORDB           LOAD THE PREVIOUSLY GATHERED "OCTAL NUMBER"
005642  640603  1320            LLS     3               CONCATENATE THE "OCTAL DIGITS"
005643  043301  1330            DAC     WORDB           AND SAVE THE RESULT
                1340
005644  203302  1350            LAC     WORDB+1         LOAD THE PREVIOUSLY GATHERED "DECIMAL NUMBER"
005645  744000  1360            CLL                     SET THE LINK FOR THE MULTIPLY
005646  653122  1370            MUL                     MULTIPLY THE PREVIOUS "DECIMAL VALUE"
005647  000012  1380            10.                     BY 10 FOR DECIMAL
005650  641002  1390            LACQ                    LOAD THE RESULT
005651  343425  1400            TAD     T1              ADD THE CURRENT "DECIMAL DIGIT"
005652  043302  1410            DAC     WORDB+1         AND SAVE THE TOTAL "DECIMAL NUMBER"
                1420
005653  605630  1430            JMP     NUM20           LOOP
                1440
                1450
005654  546137  1460    NUM26   SAD     ($POINT)        CHECK FOR A PERIOD
005655  605663  1470            JMP     NUM27           IF SO, PICK UP THE DECIMAL VALUE
005656  203426  1480            LAC     T2              ELSE LOAD THE DECIMAL-DIGITS-RECEIVED FLAG
005657  744200  1490            SZA!CLL                 AND SKIP IF THERE WERE NONE
005660  623434  1500            JMP     NUMIN,X         RETURN +1, LK=0 FOR A FORMAT ERROR; DECIMAL DIGITS, BUT NO PERIOD
005661  203301  1510            LAC     WORDB           LOAD THE OCTAL VALUE
005662  605672  1520            JMP     NUM29
005663  103450  1530    NUM27   JMS     FGET            GET THE NEXT CHARACTER
005664  103454  1540            JMS     CHRID           AND IDENTIFY IT
005665  605671  1550            JMP     NUM28           A DELIMITER IS LEGAL, SO EXIT
```

```
                T                       (MTSS TELETYPE HANDLER) OCTAL WORD INPUT/OUTPUT

005666  623434  1560            JMP     NUMIN,X         A LETTER -- EXIT +1 FOR A FORMAT ERROR
005667  744000  1570            CLL                     A NUMBER -- CLEAR THE LINK FOR A FORMAT ERROR
005670  623434  1580            JMP     NUMIN,X         AND EXIT +1
005671  203302  1590    NUM28   LAC     WORDB+1         LOAD THE DECIMAL VALUE
005672  043301  1600    NUM29   DAC     WORDB           SAVE THE CORRECT VALUE
005673  443434  1610            ISZ     NUMIN           BUMP TO A RETURN +2 FOR SUCCESSFUL
005674  623434  1620            JMP     NUMIN,X
                1630
                1640
                1650
    005675      1660            ENTER   OCTOT
                                .PMC    SAVE,ON
    003436              OCTOT   ...
005675  652000  1670    OCT42   LMQ                     SET THE VALUE TO BE OUTPUT
005676  741400  1680            SZL                     SKIP IF NO LEADING ZEROES ARE TO BE SUPPRESSED
005677  750201  1690            SZA!CLC                 SET A FLAG TO PRINT ONE CHARACTER, ANYWAY, IF THE AC IS ZERO
005700  777772  1700            LAW     -6              ELSE SET THE COUNT FOR THE STANDARD SIX CHARACTERS
005701  043425  1710            DAC     T1              SET THE NUMBER OF CHARACTERS TO BE OUTPUT
005702  641002  1720            LACQ                    RELOAD THE USER'S VALUE
005703  741200  1730            SNA                     SKIP FOR A NON-ZERO VALUE
005704  744000  1740            CLL                     ELSE FORCE A SINGLE ZERO TO PRINT
005705  641603  1750    OCT44   LLSC    3,              GET THE NEXT OCTAL DIGIT
005706  740200  1760            SZA                     IF IT IS ZERO, DON'T CHANGE PRINT-SUPPRESSION STATE
005707  744000  1770            CLL                     ELSE CLEAR THE PRINT INHIBIT AT THE FIRST NON-ZERO FOUND
005710  346140  1780            TAD     (260)           MAKE ASCII IN ANY CASE
005711  740400  1790            SNL                     BUT SKIP IF PRINT IS INHIBITED
005712  103456  1800            JMS     TTYOT           ELSE PRINT THE DIGIT
005713  443425  1810            ISZ     T1              DONE???
005714  605705  1820            JMP     OCT44           NO -- LOOP
005715  700401  1830            TSF
005716  605715  1840            JMP     .-1             WAIT FOR THE TELETYPE TO SETTLE
005717  623436  1850            JMP     OCTOT,X         YES -- EXIT
```

```
T                                   (MTSS TELETYPE HANDLER) SIXBIT WORD INPUT & SIXBIT BUFFER OUTPUT

                1860                ,STITL  (MTSS TELETYPE HANDLER) SIXBIT WORD INPUT & SIXBIT BUFFER OUTPUT
                1870
                1880
                1890    *
                1900    *       OPERATION RETURN L   AC     MQ   MEANING
                1910    *       --------- ------ - ------ ------ ------------------------------
                1920    *       INPUT       +1   1 DELIM    X    FIRST NON-BLANK CHARACTER IS A DELIMITER
                1930    *                   +2   1 SIXBIT DELIM  SUCCESSFUL READ OF A SIXBIT WORD
                1940    *       OUTPUT      +1   X   X      X    SUCCESSFUL WRITE OF A SIXBIT BUFFER
                1950    *
                1960
    005720      1970            ENTER   SIXIN
                                ,PMC    SAVE,ON
    003440              SIXIN   ...
005720  763301  1980            LAW     WORDB
005721  043425  1990            DAC     T1              INITIALIZE THE SIXBIT BUFFER POINTER
005722  103452  2000            JMS     INTIN           INITIALIZE THE INPUT
005723  623440  2010            JMP     SIXIN,X         RETURN +1 FOR DELIMITER IS FIRST NON-BLANK CHARACTER
005724  443440  2020            ISZ     SIXIN           ELSE BUMP TO A GOOD RETURN
005725  103442  2030    SIX2    JMS     SIX5            GET THE FIRST GOOD CHARACTER
005726  660714  2040            ALSS    12.             AND PUT IT IN THE FIRST CHARACTER POSITION
005727  063425  2050            DAC     T1,X            AND SAVE IT
005730  103442  2060            JMS     SIX5            GET THE SECOND CHARACTER
005731  660706  2070            ALSS    6.              PUT IT IN THE SECOND CHARACTER POSITION
005732  263425  2080            XOR     T1,X            CONCATENATE THE CHARACTERS
005733  063425  2090            DAC     T1,X            AND SAVE THE RESULT
005734  103442  2100            JMS     SIX5            GET THE THIRD CHARACTER
005735  263425  2110            XOR     T1,X            CONCATENATE THE CHARACTERS
005736  063425  2120            DAC     T1,X            AND SAVE THE RESULT
005737  443425  2130            ISZ     T1              BUMP THE STORAGE BUFFER POINTER
005740  605725  2140            JMP     SIX2            LOOP
                2150
005741  203301  2160    SIX9    LAC     WORDB           LOAD THE FIRST SIXBIT WORD
005742  623440  2170            JMP     SIXIN,X         EXIT
                2180
    005743      2190            ENTER   SIX5            SUBROUTINE TO GET THE NEXT CHARACTER, MAKE IT SIXBIT, EXIT IF A DELIMITER
                                ,PMC    SAVE,ON
    003442              SIX5    ...
005743  103450  2200            JMS     FGET            GET THE NEXT CHARACTER
005744  103454  2210            JMS     CHRID           IDENTIFY IT
005745  605741  2220            JMP     SIX9            EXIT IF IT IS A DELIMITER
005746  740000  2230            NOP                     PERMIT LETTERS
005747  346141  2240            TAD     (-240)          MAKE SIXBIT

005750  623442  2250            JMP     SIX5,X
                2260
                2270
    005751      2280            ENTER   SIXOT
                                ,PMC    SAVE,ON
    003444              SIXOT   ...
005751  043425  2290            DAC     T1              SET THE NEGATIVE CHARACTER COUNT
005752  223444  2300    SIX24   LAC     SIXOT,X         LOAD THE NEXT WORD OF OUTPUT
005753  652000  2310            LMQ                     SAVE IT FOR PRINTING
```

T (MTSS TELETYPE HANDLER) SIXBIT WORD INPUT & SIXBIT BUFFER OUTPUT

```
005754  443444  2320            ISZ     SIXOT           BUMP THE POINTER
005755  103446  2330            JMS     SIX26           OUTPUT THE FIRST CHARACTER
005756  103446  2340            JMS     SIX26           OUTPUT THE SECOND CHARACTER
005757  103446  2350            JMS     SIX26           OUTPUT THE THIRD CHARACTER
005760  605752  2360            JMP     SIX24           LOOP
                2370
    005761      2380            ENTER   SIX26
                                ,PMC    SAVE,ON
    003446              SIX26   ...
005761  641606  2390            LLSC    6,              GET THE NEXT SIXBIT CHARACTER
005762  346142  2400            TAD     (240)           MAKE IT ASCII
005763  546143  2410            SAD     (333)           CHECK FOR CARRIAGE RETURN MAPPING
005764  760215  2420            LAW     $CR
005765  546144  2430            SAD     (335)           CHECK FOR LINE FEED MAPPING
005766  760212  2440            LAW     $LF
005767  103456  2450            JMS     TTYOT           PRINT THE CHARACTER
005770  443425  2460            ISZ     T1              ALL CHARACTERS PRINTED?
005771  623446  2470            JMP     SIX26,X         NO -- LOOP
005772  700401  2480            TSF
005773  605772  2490            JMP     .-1             WAIT FOR THE TELETYPE TO SETTLE
005774  623444  2500            JMP     SIXOT,X         YES -- EXIT
                2510    *
                2520    *
```

```
T                                   (MTSS TELETYPE HANDLER) MISCELLANEOUS LINE BUFFER ROUTINES

                    2530            ,STITL  (MTSS TELETYPE HANDLER) MISCELLANEOUS LINE BUFFER ROUTINES
                    2540
                    2550
                    2560
                    2570
    005775          2580            ENTER   FGET            SUBROUTINE TO GET THE FIRST REMAINING CHARACTER FROM THE LINE BUFFER
                                    .PMC    SAVE,ON
    003450                  FGET    ...
005775  443424      2590            ISZ     BPTR            NO -- BUMP THE POINTER
005776  223424      2600            LAC     BPTR,X          LOAD THE NEXT CHARACTER
005777  043427      2610            DAC     CHAR            AND SAVE IT
006000  623450      2620    FGET9   JMP     FGET,X
                    2630
    006001          2640            ENTER   INTIN           INITIALIZE INPUT WORD-GETTING
                                    .PMC    SAVE,ON
    003452                  INTIN   ...
006001  443431      2650            ISZ     COUNT           COUNT THE WORD, SUCCESSFUL OR NOT
006002  143301      2660            DZM     WORDB           INITIALIZE THE TWO FIRST WORDS OF THE INPUT BUFFER
006003  143302      2670            DZM     WORDB+1
006004  103450      2680            JMS     FGET            GET THE NEXT CHARACTER
006005  546142      2690            SAD     ($SPACE)        CHECK IT FOR A SPACE
006006  606004      2700            JMP     .-2             THROW AWAY SPACES
006007  103454      2710            JMS     CHRID           IDENTIFY THE NON-SPACE
006010  623452      2720            JMP     INTIN,X         RETURN +1 FOR A DELIMITER
006011  740000      2730            NOP
006012  443452      2740            ISZ     INTIN           ELSE BUMP THE RETURN FOR A NUMBER OR A LETTER
006013  750001      2750            CLC
006014  343424      2760            TAD     BPTR            BACK UP THE POINTER TO POINT TO THE FIRST GOOD CHARACTER
006015  043424      2770            DAC     BPTR
006016  623452      2780            JMP     INTIN,X
```

```
T                       (MTSS TELETYPE HANDLER) MISCELLANEOUS CHARACTER-HANDLING SUBROUTINES

                2790            .STITL  (MTSS TELETYPE HANDLER) MISCELLANEOUS CHARACTER-HANDLING SUBROUTINES
                2800    *
                2810    *
                2820    *       CHRID -- SUBROUTINE TO CLASSIFY EIGHT-BIT ASCII CHARACTERS.
                2830    *       ENTER WITH THE CHARACTER IN THE AC; LEAVE WITH THE EIGHT-BIT CHARACTER
                2840    *       IN AC(0-17) AND THE LINK AS FOLLOWS:
                2850    *
                2860    *       RETURN LINK MEANING
                2870    *       ------ ---- ------------------------------------------------------
                2880    *         *1     1   THE CHARACTER IS A DELIMITER (I.E. NEITHER A DIGIT NOR A LETTER
                2890    *         *2     0   THE CHARACTER IS EITHER AN UPPER CASE OR A LOWER CASE LETTER
                2900    *         *3     0   THE CHARACTER IS AN OCTAL DIGIT
                2910    *         *3     1   THE CHARACTER IS A DECIMAL DIGIT (8 OR 9)
                2920    *
    006017      2930            ENTER   CHRID
                                ,PMC    SAVE,ON
    003454              CHRID   ...
006017  506103  2940            AND     (377)
006020  043456  2950            DAC     TTYOT           SAVE THE EIGHT-BIT ASCII CHARACTER
006021  346145  2960            TAD     (-260)          AC < 0 FOR DELIMITERS
006022  745102  2970            SPA!STL
006023  606041  2980            JMP     DLMR            CHARACTER IS A DELIMITER
006024  346146  2990            TAD     (-10)           AC < 0 FOR OCTAL DIGITS
006025  745100  3000            SPA!CLL
006026  606044  3010            JMP     DIGIT           CHARACTER IS AN OCTAL DIGIT
006027  346147  3020            TAD     (-2)            AC < 0 FOR DECIMAL DIGITS
006030  745102  3030            SPA!STL
006031  606044  3040            JMP     DIGIT           CHARACTER IS A DECIMAL DIGIT
006032  346150  3050            TAD     (-6)            AC <= 0 FOR DELIMITERS
006033  745302  3060            SNA!SPA!STL
006034  606041  3070            JMP     DLMR            CHARACTER IS A DELIMITER
006035  506151  3080            AND     (777737)        MAP LOWER CASE INTO UPPER CASE
006036  346152  3090            TAD     (-33)           AC < 0 FOR LETTERS -- L=1 FOR LETTERS; L=0 FOR DELIMITERS
006037  741102  3100            SPA!CML
006040  606045  3110            JMP     LETTR           THE CHARACTER IS A LETTER
                3120
006041  203456  3130    DLMR    LAC     TTYOT           LOAD THE DELIMITER
006042  043430  3140            DAC     DLMTR           SAVE IT
006043  623454  3150            JMP     CHRID,X
                3160
006044  443454  3170    DIGIT   ISZ     CHRID
006045  443454  3180    LETTR   ISZ     CHRID
006046  203456  3190            LAC     TTYOT           RELOAD THE CHARACTER
006047  623454  3200            JMP     CHRID,X
                3210
                3220
                3230
    006050      3240            ENTER   TTYOT
                                ,PMC    SAVE,ON
    003456              TTYOT   ...
006050  700401  3250            TSF
006051  606050  3260            JMP     .-1             WAIT FOR THE TELEPRINTER TO BE FREE
```

T (MTSS TELETYPE HANDLER) MISCELLANEOUS CHARACTER-HANDLING SUBROUTINES

```
006052  700301  3270            KSF                     KILL-THE-OUTPUT FEATURE
006053  700406  3280            TLS                     PRINT THE CHARACTER IN THE AC
006054  623456  3290            JMP     TTYOT,X
                3300
                3310
    006055      3320            ENTER   CRLF
                                ,PMC    SAVE,ON
    003460              CRLF    ...
006055  760215  3330            LAW     215
006056  103456  3340            JMS     TTYOT
006057  760215  3350            LAW     215
006060  103456  3360            JMS     TTYOT
006061  760212  3370            LAW     212
006062  103456  3380            JMS     TTYOT
006063  700401  3390            TSF
006064  606063  3400            JMP     .-1             WAIT FOR THE TTY TO SETTLE
006065  623460  3410            JMP     CRLF,X
                3420
                3430
                3440            ,HEAD                   TURN OFF THE INSERT'S HEAD SYMBOL
                3450            ,LIST   ON
                3460            ,END
                8770            ,HEAD
    003462      8780            ,USE    IMPURE
    003462      8790    CHECK   ,EQU    .
    003700      8800            ,LOC    PURSTR
    006066      8810            ,USE    PURE
                8820    RET     ,OPDEF  JMP+020000
006066  777000  8830            ,END    MSSTART
006067  001000
006070  000002
006071  000001
006072  040000
006073  436454
006074  000400
006075  000004
006076  414263
006077  425156
006100  476257
006101  017777
006102  775610
006103  000377
006104  017400
006105  014000
006106  777400
006107  775710
006110  002171
006111  000101
006112  740000
006113  001300
006114  001777
006115  000005
```

```
006116  000272
006117  000257
006120  445320
006121  777700
006122  646000
006123  446400
006124  445300
006125  606064
006126  606462
006127  606460
006130  777716
006131  777777
006132  003302
006133  000337
006134  000230
006135  000215
006136  000017
006137  000256
006140  000260
006141  777540
006142  000240
006143  000333
006144  000335
006145  777520
006146  777770
006147  777776
006150  777772
006151  777737
006152  777745

TRANSFER ADDRESS  603701
```

CROSS REFERENCE TABLE

1713	.0	4510	4520										
26	.310	3400											
27	.311	3410											
4453	.DK	770											
4464	.DT	750											
6460	.TP	760											
2023	10SAVE	1870	1880										
2024	11SAVE	1880	1920										
5	3AC	3370											
305	3REST	3870	3880										
51	3TEM0	3530	3540										
52	3TEM1	3540	3550										
53	3TEM2	3550	3560										
54	3TEM3	3560	3570										
55	3TEM4	3570	3580										
56	3TEM5	3580	3590										
57	3TEM6	3590	3600										
50	3TM20	3520	3530										
2	3TM21	3350											
3	3TM22	3360											
14000	7K	1030											
16000	8K	1020	910	1010	2640	2650							
6055	9MAPBU	3320	1320	1730	1880	2290	5850	6930	7790	7960	8090	740	1340	1680
			2590	3210	3670	3860	4150	720	1170	1660	1970	2190	2280	2380
			2580	2640	2930	3240	3320							
414263	ABS	850	4100										
1753	AC	4520	4530										
1756	ACS	4550	4560										
2015	ACSAVE	1810	1820										
2022	ACSW	1860	1870										
17737	ADRSS	430	4110										
246	AMPRSN	290											
300	AT	2950											
300	ATSGN	2900											
422030	BAS	420											
2151	BCNTRL	2330	2340										
3217	BCOUNT	8510	4610	4760									
425156	BIN	860	4120										
4366	BINPPT	4600	4490										
634	BIT0	3970	3980										
641	BIT17	4020	4030										
635	BIT36	3980	3990										
636	BIT5	3990	4000										
637	BIT6	4000	4010										
640	BIT7	4010	4020										
337	BKARR	240	800										
642	BL7	4030	4040										
643	BL8	4040	4050										
400	BLKLEN	630	1520	1640	4830								
1777	BLKMSK	610	1060	1170									
17740	BOOT	520	530	2580									
2000	BOUNDA	440	960	980	990	1000	1630	5040	5080	2100	2110		

CROSS REFERENCE TABLE

CROSS REFERENCE TABLE

3265	CDISLN	2320									
3253	CERCNT	1520	1360	1400							
5363	CGNAM1	2660	3020								
5366	CGNAM2	2700	2650	3070							
5372	CGNAM3	2740	2700								
5376	CGNAM5	2780	2680								
5410	CGNAM8	2880	2840								
3266	CGNAME	2590	2700	3660	3960	4250	7360	2670	2740	2880	2890
3462	CHECK	8790									
6	CHRMAX	3180	3200								
2	CHRPAK	3130	3200								
50	CLKMAX	2840	3180								
60	CLKSPD	3160	3170								
1757	CLOCK	4560	4570								
12	CMDX	510									
45	CMP1	3490	3500								
46	CMP2	3500	3510								
6	CNTRL	3380	3390								
272	COLON	370	2640	3000							
2053	COMFLG	2200	2210								
254	COMMA	320									
2150	COMSTO	2270	2280								
230	CONTX	230	820								
435762	COR	880									
16000	CORMAX	910	980	2110							
3273	CPAPER	3670	7280	2660	3240	3690	3710	3730	3740	3750	
2175	CPARAM	680	2310	2370	4230						
215	CR	220	910	2420							
5145	CRCAT1	780									
5170	CRCAT3	1010	930								
5204	CRCAT4	1160	1290								
3251	CRCOVR	1340	1540	1750	8140	920	1140	1210	1370	1410	1490
5223	CRCVR4	1390	1590	1800	6980	8190	970				
5235	CRCVR5	1420	1450	1500							
47	CSPL	3510	3520								
44	CSWP	3480	3490								
60	CTBFR	3600	3630	3640							
100	CTBIN	3640	3650	3670	4250						
2000	CTEMP0	1630									
2001	CTEMP1	1640									
2002	CTEMP2	1650									
2003	CTEMP3	1660									
2004	CTEMP4	1670									
2005	CTEMP5	1680									
2006	CTEMP6	1690									
2007	CTEMP7	1700									
2010	CTEMP8	1710									
2011	CTEMP9	1720									
102	CTFLG	3650	3660								
436454	CTL	810	2490								
104	CTNAM	3660									
2043	D PC	2120	2130								

CROSS REFERENCE TABLE

2154	D BCA	2370	2380	1350	2540	890							
2153	D BDA	2360	2370	1490	1570	1630	1660	2320	8170	810	850	870	950
2163	D FDA	2440	2450										
2042	D LOC	2110	2120										
2022	D ACSW	1860											
2156	D BALT	2390	2400	950	2600	2770	5060	8100	8110	4000			
2155	D BLEN	2380	2390	1470	2570	910							
2161	D BMAX	2420	2430										
2157	D BMIN	2400	2410										
2162	D BPTR	2430	2440										
2167	D FMAX	2480	2490										
2165	D FMIN	2460	2470										
2046	D MASK	2150	2160										
2164	D MFDA	2450	2460										
2036	DADRSW	2070	2080										
1762	DAP0	4590	4600										
1763	DAP1	4600	4610										
653	DBK	4120	4130										
24	DBKNUM	2220	2270										
2054	DBKTAB	2210	2270										
2035	DBSTOR	2050	2060										
422027	DDT	410											
12000	DDTST	5000											
2037	DDUMSW	2080	2090										
1	DEBUG	380	100	120	100	120	180	120					
5436	DEVC35	3340	3510	3570									
1761	DFLAG	4580	4590										
1764	DFN	4610	4620										
2151	DFTYPE	2340	2350										
2045	DHICOR	2140	2150										
2050	DINDIR	2170	2180										
445300	DK.	740	3310										
445320	DK0	780	3040										
127	DK1	4310											
156	DK2	4350											
37	DKCA	2750											
675	DKDON	4170	4180										
16000	DKLEN	2650	2660	2320									
34	DKLENB	2660											
266	DKLOK	3830	3840										
777601	DKMAX	650	2410										
672	DKOVR	4160	4170										
2	DKRD	2760											
36	DKWC	2740											
4	DKWRT	2770											
2041	DLIMIT	2100	2110										
2044	DLOCOR	2130	2140										
2160	DMBMIN	2410	2420										
2166	DMFMIN	2470	2480										
654	DO	4130	4140										
662	DO2	4140	4150										
663	DO3	4150	4160										

CROSS REFERENCE TABLE

2152	DOFTYP	2350	2360										
244	DOLLAR	280											
2032	DPACSW	1980											
2040	DPATSW	2090	2100										
2051	DPCMSK	2180	2190										
2052	DREGBR	2190	2200										
2035	DREGSW	2060	2070										
2047	DRELOC	2160	2170										
1765	DSTAT	4620	4630										
446400	DT.	720	3290										
2000	DTEMP0	1630											
2001	DTEMP1	1640											
2002	DTEMP2	1650											
2003	DTEMP3	1660											
2004	DTEMP4	1670											
2005	DTEMP5	1680											
2006	DTEMP6	1690											
2007	DTEMP7	1700											
2010	DTEMP8	1710											
2011	DTEMP9	1720											
776701	DTMAX	640	2350										
740000	DVCMSK	600	800	1020	3390								
275	EQUAL	2910											
241	EXCLAM	260											
5	FCBLEN	570	3530	1260	1970	3960							
602	FGET	3950	3960										
3745	FORMAT	1160	2190	2710	2920	3050	3080	3670	3730	3800	3830	3970	4090	4160
			4260	4520	5290	7370	2830	2860	3280				
1701	FRCA	4410	4420										
1700	FRDA	4400	4410										
1702	FRLEN	4420	4430										
1703	FRSTA	4430	4440										
2	FUDGE	3190	3200										
5412	GNAM90	2980	2620										
5405	GNAME6	2850	2800										
276	GREAT	2930											
476257	GRO	870	4140										
4	HDRLEN	580											
1700	IMPLEN	990											
3170	IMPSTR	530	560										
10	INDEX	490											
422020	INT	320											
513	IO.IN	3910	3920										
525	IO.OT	3920	3930										
300000	IOBLK	2830											
1760	IORS	4570	4580										
1002	IOTO	4900	4910										
652	JMP	4110	4120										
100	JTLEN	960											
1700	JTSTRT	950	940	960	1000	4400							
16	KBLEN	3610	3630	3640	3680	3690	3780	3740					
10	KBNUM	3620	3670	3720									

CROSS REFERENCE TABLE

76	L0LOK	3630											
107	L1BFR	3670	3680	3690									
127	L1BIN	3690	3700	3720	4290								
131	L1FLG	3700	3710										
125	L1LOK	3680											
133	L1NAM	3710											
136	L2BFR	3720	3730	3740									
156	L2BIN	3740	3750	4330									
160	L2FLG	3750	3760										
154	L2LOK	3730											
162	L2NAM	3760	3770										
422026	LDR	390											
2000	LDRST	5040											
3210	LEADER	7960	7410	7710	8040								
274	LESS	2920											
242	LF	210	2440										
3231	M NP	8610	5160	6010	6570								
3234	M P2	8640	5170	6520	6880								
3245	M WT	8710	2000	2030	4340	4550	5680	5860	5950	6450	7220	7270	
5142	M ADD	8470	5630	5760	5870	5910	5960	5980	6140	6400			
2170	M BUF	410	4620	4780	7590								
3221	M BWS	8530	6100	6300	7050								
4136	M CLE	2170	8240										
4107	M GW1	1990	1930										
4203	M LDU	3030	8300										
3230	M LEN	8600	1360	1420	1450	1460	3390	3450	3780	3810	3840	4060	4650	4670
			4730	4870	5210	6790	7130	7460	7500	7560				
4767	M P2L	7160	6540										
5005	M PUN	7360	8400										
4136	M PUR	2900	8280										
4307	M REP	3960	8380										
4256	M SAV	3660	8360										
4424	M TCD	4950	5010	7260									
3241	M TOP	8690	5230	5990									
4163	M UNS	2700	8260										
4445	M ABSF	5150	4110										
2022	M ACSW	1860											
14000	M BASE	400	6220	6620									
4355	M BINF	4480	4130										
4400	M BINL	4700	4770	4890									
1000	M BMAX	430	1390	1430	1520	1640	4680	4850					
3222	M BOTM	8540	3740	4040	5200	6630	6660	6730	7070				
3220	M BUFA	8520	6130	6160	6190	6240	6270	6290	6310	6330	6350	6380	6410	6430
			6440	7030										
5023	M BUFN	7500	7640											
3243	M CATP	8550	3710	3870	3880	4020	4380	4680	4980	6550	6840	7110	7200	
12	M CMDX	450	1070	1110	1120	2460	3290	3340	3350	3380	3470	3540	3550	4630
			4720	4790	7600	7610								
5137	M COME	8430	1080											
3730	M COML	1110	1130											
4025	M COPL	1360	1680											
3170	M COPY	1320	1380	1510	1620	3490	4480	7580						

CROSS REFERENCE TABLE

CROSS REFERENCE TABLE

4036	MCOPL2	1450	1410										
4040	MCOPL4	1470	1440										
3226	MCOUNT	8580	5670	5770	6070	6320	7540	7570	7580	7630	7980	8030	
4007	MDSAVE	1240	3690										
3734	MERROR	1140	8420										
3212	MFORCE	8090	1330	2170	2800	8130	8150	760	1110	1160	1250	1270	
10	MINBUF	3200	3610										
5141	MINSTM	8460	5440	5560	7230								
255	MINUS	330											
5073	MLEAD1	7990	8030										
4756	MLEND1	7070	6650										
4734	MLEND2	6870	7150										
3723	MMONX2	1060	3330										
3714	MMONXT	490											
3714	MNEXTL	980	470	490	1050	1150	1190	1210	1230	1250	2210	2780	3280	5080
			7720	1850	3620								
3775	MNSAVE	1220	2730	3990	4280	7390							
4771	MNXLT1	7180	4410	6590									
3232	MOUTDA	8620	1500	1600	1650	2940	3090	3180	3420	3500	4390	4690	4810	4840
			7400										
422023	MP1	350											
422024	MP2	360											
2032	MPACSW	1980											
4470	MPAPER	5360	5260	5430	5810	6900	7290						
4211	MPCOPY	3130	2950										
3236	MPFLAG	8660	5240	5380	6870								
1004	MPOPR	4920											
1000	MPST	4880	4890										
4606	MPUTW1	5980	5940										
1754	MQ	4530	4540										
2016	MQSAVE	1820	1830										
3701	MSTART	920	8830										
2000	MTEMP0	1630	7480	7690									
2001	MTEMP1	1640											
2002	MTEMP2	1650											
2003	MTEMP3	1660											
2004	MTEMP4	1670											
2005	MTEMP5	1680											
2006	MTEMP6	1690											
2007	MTEMP7	1700											
2010	MTEMP8	1710											
2011	MTEMP9	1720											
4072	MTPOT1	1780											
422025	MTR	370											
2000	MTRST	5080											
1772	NAME	4670	4680										
4633	NBLOCK	6190	6060	6090									
540	NEWBR	3930	3940										
3177	NEWHDR	2290	2200	2610	3190								
3714	NEXTL	470	1470	4050	4330								
1771	NUMBR	4660	4670										
243	NUMSGN	270											

CROSS REFERENCE TABLE

623	NXPTR	3960	3970				
702	OC0	4180	4190				
703	OC1	4190	4200				
704	OC2	4200	4210				
705	OC3	4210					
574646	OFF	2730					
3233	OFFSET	8630	5180	5610	6390	6720	7100
575600	ON	2720					
1773	OVER	4680	4690				
700	OVLEN	940					
1000	OVSTRT	930	920	940	4750	4880	4960
2033	P10SAV	1990	2000				
2034	P11SAV	2000	2050				
2025	PACSAV	1930	1940				
2032	PACSW	1980	1990				
241	PBFLAG	3810	3820				
2017	PCSAVE	1830	1840				
256	PERIOD	340	350				
227	PFLAG	3770	3780				
77	PH0	4260	4270				
126	PH1	4300	4310				
155	PH2	4340	4350				
1	PHANTO	2780					
2150	PHFLAG	2280	2330				
1700	PHLEN	2640					
2025	PHSTOR	1920	1930				
274	PIDN2	3850	3860				
270	PIDON	3840	3850				
1001	PINT	4890	4900				
303	PIOUT	3860	3870				
602026	PLDR	400					
253	PLUS	310					
2026	PMQSAV	1940	1950				
602025	PMTR	380					
256	PQINT	350	1460				
2027	PPCSAV	1950	1960				
606064	PPT	690	3680				
2031	PSCSAV	1970	1980				
2030	PSTSAV	1960	1970				
606460	PTP	710	3720				
606462	PTR	700	3700				
1	PURCOD	360	5140	5270	430	610	
12100	PURLEN	1010					
1775	PURNM	4700	4710				
3700	PURSTR	540	990	1010	2560	8800	
546	PUTIN	3940	3950				
34	RACS	3440					
6	RCNT	3390					
35	RCORE	3450					
1003	RDBLK	4910	4920				
32	RDT0	3420					
33	RDT1	3430					

CROSS REFERENCE TABLE

17505	RECOV	470											
422021	RES	330											
40	RESCAT	3470	3480										
1000	RESLEN	920											
234	RFLAG	3790	3800										
230	RPTP	3780	3790										
235	RPTR	3800	3810										
242	RSCO	3820	3830										
1776	RSTRT	4710											
1755	SC	4540	4550										
273	SCOLON	380											
640000	SCRSTR	2670											
2021	SCSAVE	1850	1860										
243	SHARP	2890											
257	SLASH	360	2790										
240	SPACE	250	2690										
377	SPCOD	5410											
422122	SPL	430											
1000	SPLST	4960											
777400	SPMSK	5390											
252	STAR	300											
2020	STSAVE	1840	1850										
335	SWAP	3880	3890										
336	SWAP1	3890	3900										
340	SWAP3	3900	3910										
1000	SWCAT	4750	4760										
1003	SWCLK	4780	4790										
1004	SWERR	4790	4800										
1007	SWMP1	4820	4830										
1010	SWMP2	4830	4840										
1002	SWMTR	4770	4780										
1011	SWOPR	4840											
422022	SWP	340											
1001	SWPPR	4760	4770										
40	SWPS	3460	3470										
1005	SWSPL	4800	4810										
1006	SXSPL	4810	4820										
1300	SYSBAS	2800	2810	840									
41300	SYSDA	2810											
17735	SYSDEV	530											
1777	SYSMAX	2820											
3425	T T1	560	1280 2460	1400	1710	1810	1990	2050	2080	2090	2110	2120	2130	2290
3426	T T2	570	1180	1260	1480									
5572	T IN1	780	930	1020	1050									
5600	T IN4	840												
5566	T INL	740	990	1010										
120	T STD	480	490											
3423	T BEND	540	860											
3424	T BPTR	550	750	850	880	900	950	1000	1040	2590	2600	2760	2770	
3427	T CHAR	580	2610											
3460	T CRLF	3320	960	5410	980	3410								

CROSS REFERENCE TABLE

CROSS REFERENCE TABLE

2001	TTEMP1	1640											
2002	TTEMP2	1650											
2003	TTEMP3	1660											
2004	TTEMP4	1670											
2005	TTEMP5	1680											
2006	TTEMP6	1690											
2007	TTEMP7	1700											
2010	TTEMP8	1710											
2011	TTEMP9	1720											
3456	TTTYOT	3240	1800	2450	2950	3130	3190	3290	3340	3360	3380		
6	TTYCLK	3170	3180										
3	TTYNUM	3140											
10	TTYSPD	3150	3170										
3301	TWORDB	470	1100	1690	1790	1930	2720	2810	2870	3050	3370	3980	1310	1330
			1350	1410	1510	1590	1600	1980	2160	2660	2670			
1774	TYPE	4690	4700										
1766	UCORE	4630	4640	2160									
1767	UDISK	4640	4650	2370									
336	UPARR	2940											
76	US0	4250	4260	4280									
125	US1	4290	4300	4320									
154	US2	4330	4340	4360									
0	USER	2790											
3	USERS	2850	3200										
14000	USLEN	980	2640										
2015	USTORE	1800	1810										
75	UT0	4280											
124	UT1	4320											
153	UT2	4360											
1704	UTEM0	4440	4450										
1705	UTEM1	4450	4460										
1706	UTEM2	4460	4470										
1707	UTEM3	4470	4480										
1710	UTEM4	4480	4490										
1711	UTEM5	4490	4500										
1712	UTEM6	4500	4510										
1770	VALID	4650	4660	3480									
17777	VFLAG	480											

UNDEFINED SYMBOLS

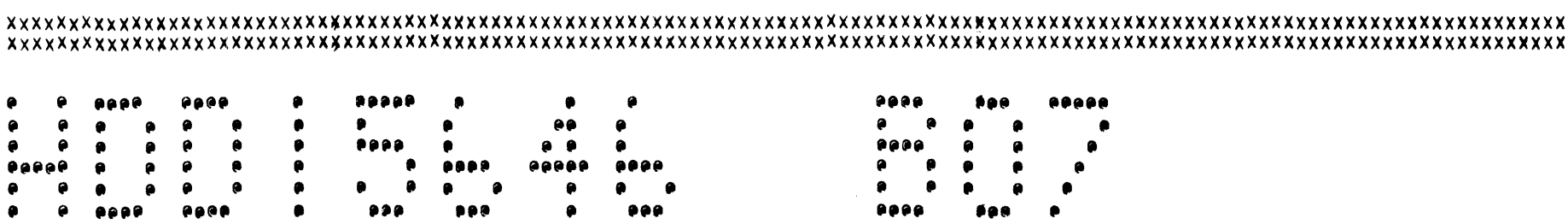
#1	5630											
#2	5640											
#3	5650											
#4	5660											
#5	5680											
DEFINS	100	100	120	120	580	580	2680	2680	3210	3210	5330	5330
M BAD	620											
MCHECK	610											
OCTZ	1790											

MACRO CROSS REFERENCE TABLE

CHAR	1380											
CHROT	1340											
COUNT	1460	1170										
CRLF	1300	960	5410									
DELIM	1420	2630	2780	2990								
EMESS	1750											
ENTER	770	1320	1730	1880	2290	5850	6930	7790	7960	8090	740	1340	1680
		2590	3210	3670	3860	4150	720	1170	1660	1970	2190	2280	2380
		2580	2640	2930	3240	3320							
FORMAT	1060	2190	2710	2920	3050	3080	3670	3730	3800	3830	3970	4090	4160
		4260	4520	5290	7370	2830	2860	3230					
LINE	1100	990	4960	5400	1430								
LOOP	960	1130	7630	8030									
MESS	1620	970	980	1140	1160	1200	1220	1240	4950	5390	5800	1390	1420
		1840	3610	4040	4320								
MESSR	1520	970	980	1140	1160	1200	1220	1240	4950	5390	5800	1390	1420
		1840	3610	4040	4320								
MPOFF	5430	2150	2360	3470									
NEG	1010	3770	5220	5660	6700	6780	7530						
NMESS	1670												
NUM	1260	3720	3790	4510	5000	5280	2850						
OCT	1890												
OCTZ	1840	1180											
START	1100												
SWAP	5610												
WORD	1140	1040	4080	1440	2610	2760	3220						
WORD1	1180	1100	1790	1930	2720	2810	3370	3980					
WORD2	1220												

USE CROSS REFERENCE TABLE

0	.!.!.!												
0	.REL.												
0	IMPURE	550	1320	1730	1880	2060	2290	5850	6930	7760	7790	7960	8090	8480
			540	740	1340	1510	1680	2070	2280	2590	2940	3210	3670	3860
			4150	440	720	1170	1660	1970	2190	2280	2380	2580	2640	2930
			3240	3320	8780									
3700	PURE	570	1320	1730	1880	2080	2290	5850	6930	7780	7790	7960	8090	8720
			570	740	1340	1530	1680	2130	2340	2590	2960	3210	3670	3860
			4150	620	720	1170	1660	1970	2190	2280	2380	2580	2640	2930
			3240	3320	8810									



05/31/72 01:58:43

```
**********************************
**********************************
**  PDP-9 MINI TIME-SHARING SYSTEM  **
**    NON-SWAPPING TSS DEBUGGER     **
**              TP2:DDT             **
**              DK0:B07             **
**********************************
**********************************
```

DDT INITIALIZATION

```
                    100                 ,STITL  DDT INITIALIZATION
                    110                 ,NAME   TSSDDT
                    120         ENTER   ,DEFIN
                    130         #1      HLT
                    140                 ,ENDM
                    150                 ,TITLE  PDP-9 TIME-SHARING SYSTEM DEBUGGER
                    160                 ,PMC    ON
                    170                 ,ABS
000010              180         INDEX   ,EQU    10
012000              190         BASE    ,EQU    12000
000000              200         DEBUG   ,EQU    0
                    210                 ,INSRT  :DLIBRARY:PDP9LIB:LIBMACRO
                    100                 ,INE    DEBUG,1
```

TELETYPE INPUT/OUTPUT MACROS

```
                1940            .LIST   ON
                1950            .END
                220             .INSRT  :DLIBRARY:PDP9LIB:GRODEFIN
                100             .INE    $DEBUG,1
                1250            .LIST   ON
                1260            .END
                230     RET     .OPDEF  620000
                240     TERMINATE .OPDEF 705001
                250
                260
        013116  270             .LOC    16000-2662      START SO AS TO PUT DDT AT THE END OF USER CORE
```

SUBROUTINES

```
                    280             ,STITL  SUBROUTINES
                    290             ,HEAD
                    300     *       EXP --
                    310     *
                    320     *       EXP EVALUATES A SYMBOLIC EXPRESSION FROM THE TTY
                    330     *
013116  700000      340             700000
013117  705001      350             TERMINAT                ;'X' IS ALSO AN EXIT
013120  000000      360     EXP     0
013121  155206      370             DZM     EVAL            CLEAR CURRENT VALUE
013122  215074      380             LAC     (JMP PLU)       FAKE A 0+ TO START WITH
013123  053174      390             DAC     OJMP            SAVE IN OPERATOR JMP
                    400     *
                    410     *       LOOP TO EVALUATE EXPRESSIONS
                    420     *
013124  115607      430     EVL     JMS     TSFGET          GET A CHARACTER
013125  555075      440             SAD     (SPACE)         CHECK FOR SPACE
013126  613124      450             JMP     EVL             TRY AGAIN IF SO
013127  555076      460             SAD     (POINT)         CHECK FOR .
013130  741000      470             SKP
013131  613134      480             JMP     EVL0
013132  213671      490             LAC     LOC             GET THE LOCATION COUNTER
013133  613173      500             JMP     EVX             GO TO END
    013134          510     EVL0    ...;
013134  115633      520             JMS     TSCHRID         CHECK FOR A NUMBER
013135  613175      530             JMP     EVD             INSTANT DELIMITER
013136  613145      540             JMP     EVL1            NO
013137  777777      550             LAW     -1
013140  355362      560             TAD     TSBPTR          BACK UP POINTER
013141  055362      570             DAC     TSBPTR          SAVE IT
    013142          580             NUM                     INPUT THE NUMBER
013142  115431                      JMS     TSNUMIN
013143  741000      590             SKP                     FORMAT ERROR ON NUMBER -- POSSIBLY A LETTER
013144  613173      600             JMP     EVX             AND SKIP SYMBOLIC
013145  777777      610     EVL1    LAW     -1              BACK UP CHARACTER POINTER
013146  355362      620             TAD     TSBPTR
013147  055362      630             DAC     TSBPTR
    013150          640             WORD                    GET A WORD
013150  115526                      JMS     TSSIXIN
013151  740000      650             NOP
    013152          660             DELIM                   GET THE DELIMITER
013152  215366                      LAC     TSDLMTR
013153  555077      670             SAD     (44)            CHECK FOR A $
013154  741000      680             SKP
013155  613164      690             JMP     NHD             NO EXTRANEOUS HEAD SYMBOL
    013156          700             WORD1                   GET THE HEAD SYMBOL
013156  215237                      LAC     TSWORDB
013157  055200      710             DAC     T1              SAVE IT
    013160          720             WORD                    GET MORE WORDS
013160  115526                      JMS     TSSIXIN
013161  740000      730             NOP                     IGNORE VACUOUS
013162  215200      740             LAC     T1              GET THE HEADSYMBOL
```

SUBROUTINES

```
013163  741000  750             SKP
013164  215210  760     NHD     LAC     HEAD            GET THE CURRENT HEADSYMBOL
013165  055201  770             DAC     T2              SAVE IT
013166  113337  780             JMS     RJUS            RIGHT JUSTIFY
013167  215201  790             LAC     T2              RESTORE
013170  255237  800             XOR     T$WORDB         TRY IT
013171  055237  810             DAC     T$WORDB         RESTORE IT
013172  113271  820             JMS     SYMB            LOOK IT UP IN THE SYMBOL TABLE
013173  055207  830     EVX     DAC     TVAL            SAVE TEMP VALUE
013174  613174  840     OJMP    JMP     .               FILLED IN TO LAST OPERATER
013175  750000  850     EVD     CLA                     VALUE IS ZERO FOR A VACUOUS WORD
013176  633174  860             JMP     OJMP,X          RETURN
                870     *
                880     *       PLUS
                890     *
013177  355206  900     PLU     TAD     EVAL
013200  613242  910             JMP     EVY             AND CONTINUE
                920     *
                930     *       MINUS
                940     *
   013201       950     MIN     NEG                     NEGATE AC
013201  740001                  CMA
013202  355100                  TAD     (1)
013203  613177  960             JMP     PLU             HANDLE LIKE PLUS
                970     *
                980     *       OR
                990     *
013204  740001 1000     OR      CMA                      GET MASK
013205  015206 1010             AND     EVAL            MASK OUT BITS
013206  255207 1020             XOR     TVAL            FIND NEW VALUE
013207  613242 1030             JMP     EVY             EXIT
               1040     *
               1050     *       XOR
               1060     *
013210  255206 1070     ERA     XOR     EVAL            DO THE XOR
013211  613242 1080             JMP     EVY             EXIT
               1090     *
               1100     *       AND
               1110     *
013212  015206 1120     AND     AND     EVAL            GET NEW VALUE
013213  613242 1130             JMP     EVY             AND EXIT
               1140     *
               1150     *       MPY
               1160     *
013214  000000 1170     OSET    0                       ROUTINE TO SET UP FOR EAE
013215  741100 1180             SPA
013216  355101 1190             TAD     (-1)            CONVERT TO UNARY
013217  664000 1200             GSM
013220  053230 1210             DAC     OST+1           SAVE ONE OP
013221  233214 1220             LAC     OSET,X          GET OP
013222  053227 1230             DAC     OST             SAVE IT
013223  215206 1240             LAC     EVAL            GET THE VALUE
```

SUBROUTINES

```
013224  740102  1250            CML!SMA                 CHECK
013225  353241  1260            TAD     ONE             ADJUST FOR 1'S COMPLEMENT
013226  355101  1270            TAD     (-1)
013227  740040  1280    OST     XX
013230  740040  1290            XX
013231  641002  1300            LACQ                    GET THE RESULT
013232  741100  1310            SPA
013233  353241  1320            TAD     ONE             ADJUST FOR 1'S COMPLEMENT
013234  613242  1330            JMP     EVY
                1340    *
                1350    *       MPY
                1360    *
013235  113214  1370    MPY     JMS     OSET
013236  657122  1380            MULS                    INSTRUCTION
                1390    *
                1400    *       DIV
                1410    *
013237  113214  1420    DIV     JMS     OSET            SET UP
013240  644323  1430            DIVS                    INSTRUCTION
013241  000001  1440    ONE     1
                1450    *
                1460    *       GET NEXT OPERATOR
                1470    *
013242  055206  1480    EVY     DAC     EVAL
    013243      1490            DELIM                   GET THE CHARACTER THAT STOPPED US
013243  215366                  LAC     TSDLMTR
                1500    OPA     .DEFIN
                1510            SAD     (#1             ) "CHECK FOR OPERATOR
                1520            LAW     #2              GET ADDRESS
                1530            .ENDM   OPA
                1540    *
    013244      1550            OPA     PLUS,PLU
013244  555102                  SAD     (PLUS)          CHECK FOR OPERATOR
013245  773177                  LAW     PLU             GET ADDRESS
    013246      1560            OPA     MINUS,MIN
013246  555103                  SAD     (MINUS)         CHECK FOR OPERATOR
013247  773201                  LAW     MIN             GET ADDRESS
    013250      1570            OPA     STAR,MPY
013250  555104                  SAD     (STAR)          CHECK FOR OPERATOR
013251  773235                  LAW     MPY             GET ADDRESS
    013252      1580            OPA     SLASH,DIV
013252  555105                  SAD     (SLASH)         CHECK FOR OPERATOR
013253  773237                  LAW     DIV             GET ADDRESS
    013254      1590            OPA     EXCLAM,OR
013254  555106                  SAD     (EXCLAM)        CHECK FOR OPERATOR
013255  773204                  LAW     OR              GET ADDRESS
    013256      1600            OPA     BSLASH,ERA
013256  555107                  SAD     (BSLASH)        CHECK FOR OPERATOR
013257  773210                  LAW     ERA             GET ADDRESS
    013260      1610            OPA     AMPRSND,AND
013260  555110                  SAD     (AMPRSND)       CHECK FOR OPERATOR
013261  773212                  LAW     AND             GET ADDRESS
```

SUBROUTINES

```
013262  740100  1620            SMA
013263  613267  1630            JMP     EVZ             EXIT IF DONE
013264  355111  1640            TAD     (JMP-LAW)       FORM POINTER
013265  053174  1650            DAC     OJMP            SAVE IT
013266  613124  1660            JMP     EVL             LOOP
    013267      1670    EVZ     ...                     DONE
013267  215206  1680            LAC     EVAL            GET THE VALUE
013270  633120  1690            RET     EXP             RETURN
                1700    *
    015200      1710            .USE    STOR
                1720    *
015200  000000  1730    T1      0
015201  000000  1740    T2      0
015202  000000  1750    T3      0
015203  000000  1760    T4      0
015204  000000  1770    T5      0
015205  000000  1780    T6      .DSA
015206  000000  1790    EVAL    0
015207  000000  1800    TVAL    0
015210  000000  1810    HEAD    0
    013271      1820            .USE
                1830    *
```

SUBROUTINES

```
                    1840            .EJECT
                    1850    *
                    1860    *       SYMBOL TABLE LOOKUP
                    1870    *
    000012          1880    SYMBX   .EQU    12              INDEX REGISTER FOR SYMBOL TABLE
                    1890    *
013271  000000      1900    SYMB    0
013272  213336      1910            LAC     SPTR            GET THE SYMBOL TABLE POINTER
013273  040012      1920            DAC     SYMBX
013274  777757      1930    SCOUNT  LAW     -SYMN           GET THE NUMBER OF SYMBOLS
013275  055202      1940            DAC     T3
    013276          1950    SYML    WORD1                   GET THE FIRST HALF
013276  215237                      LAC     T$WORDB
013277  560012      1960    SYML1   SAD     SYMBX,X         CHECK A SYMBOL
013300  613306      1970            JMP     SYMB2           FOUND HALF
013301  440012      1980            ISZ     SYMBX
013302  440012      1990            ISZ     SYMBX
    013303          2000            LOOP    T3,SYML1
013303  455202                      ISZ     T3
013304  613277                      JMP     SYML1
013305  613314      2010            JMP     NOS
    013306          2020    SYMB2   WORD2                   GET THE SECOND HALF
013306  215240                      LAC     T$WORDB+1
013307  560012      2030            SAD     SYMBX,X         CHECK IT
013310  613334      2040            JMP     SYMB3           TRY SOME MORE
013311  440012      2050            ISZ     SYMBX
    013312          2060            LOOP    T3,SYML         LOOK FOR COUNT
013312  455202                      ISZ     T3
013313  613276                      JMP     SYML
    013314          2070    NOS     EMESS   <UNDEFINED SYMBOL>,16,
                                    .CRSM   SAVE,ON
    013314                          MESS    <UNDEFINED SYMBOL WORD #>,16,-7
013314  700312                      KRB
    013315                          MESSR   <UNDEFINED SYMBOL WORD #>,16,-7
                                    .CRSM   SAVE,ON
013315  777747                      LAW     -16,-7-2
013316  115561                      JMS     T$$IXOT
                                    .PMC    RESTORE
                                    .CRSM   RESTORE

    013330                          COUNT
013330  215367                      LAC     T$COUNT
    013331                          OCTZ
013331  744002                      STL
013332  115502                      JMS     T$OCTOT
                                    .CRSM   RESTORE
013333  614007      2080            JMP     IDLE            RETURN TO IDLE LOOP
013334  220012      2090    SYMB3   LAC     SYMBX,X         GET THE VALUE OF THE SYMBOL
013335  033271      2100            RET     SYMB
013336  011673      2110    SPTR    SYMST-1                 POINTER TO SYMBOL TABLE
```

SUBROUTINES

```
                    2120            .EJECT
                    2130    *
                    2140    *       RJUS-
                    2150    *
                    2160    *       RJUS RIGHT JUSTIFIES THE CONTENTS OF TSWORDB
013337  000000      2170    RJUS    0
013340  155202      2180            DZM     T3              CLEAR WORD AREA
    013341          2190            WORD2                   GET THE SECOND WORD
013341  215240                      LAC     TSWORDB+1
013342  741200      2200            SNA                     CHECK FOR ALL ZEROS
013343  613353      2210            JMP     RJS1            HANDLE IT
    013344          2220    RJS2    ...
013344  652000      2230            LMQ                     MOVE CURRENT WORD TO MQ
013345  515112      2240            AND     (77)            EXTRACT LOW ORDER CHARACTER
013346  740200      2250            SZA
013347  613361      2260            JMP     RJS3            EXIT IF DONE
013350  660614      2270            LLSS    18.-6.
013351  455202      2280            ISZ     T3              INCREMENT COUNT
013352  613344      2290            JMP     RJS2
013353  215113      2300    RJS1    LAC     (3)             SET COUNT
013354  055202      2310            DAC     T3
    013355          2320            WORD1                   GET FIRST HALF
013355  215237                      LAC     TSWORDB
013356  741200      2330            SNA                     CHECK FOR 2 ZEROS
013357  633337      2340            RET     RJUS            OK IF SO
013360  613344      2350            JMP     RJS2            ENTER LOOP
013361  215202      2360    RJS3    LAC     T3              GET COUNTER
013362  744010      2370            CLLIRAL                 NOW 2X
013363  355202      2380            TAD     T3              NOW 3X
013364  740010      2390            RAL                     NOW 6X
013365  355114      2400            TAD     (LRS 0)         MAKE LRS INSTRUCTION
013366  053372      2410            DAC     RJS4            SAVE IT
    013367          2420            WORD2                   GET THE BUFFER WORD
013367  215240                      LAC     TSWORDB+1
013370  652000      2430            LMQ
    013371          2440            WORD1
013371  215237                      LAC     TSWORDB
013372  000000      2450    RJS4    0
013373  055237      2460            DAC     TSWORDB
013374  641002      2470            LACQ                    GET BACK TO AC
013375  055240      2480            DAC     TSWORDB+1
013376  633337      2490            RET     RJUS            RETURN
```

SUBROUTINES

```
                    2500            .EJECT
                    2510    *
                    2520    *       DUMP ADDRESS FORMATTING
                    2530    *
013377  000000      2540    DUMA    0
013400  613401      2550    AFORM   JMP     OCTF
                    2560    *
                    2570    *       OCTAL FORMAT
                    2580    *
    013401          2590    OCTF    ...
    013401          2600            OCT                     PRINT IN OCTAL
013401  744000                      CLL
013402  115502                      JMS     TSOCTOT
013403  633377      2610            RET     DUMA            RETURN
013404  661606      2620            EABCLA!LLSS 6           MOVE IN NEXT CHARACTER
                    2630    *
013405  113514      2640    DECF    JMS     PDEC            PRINT DECIMAL
013406  633377      2650            RET     DUMA            RETURN
                    2660    *
                    2670    *       SYMBOLIC
                    2680    *
013407  055201      2690    SYMF    DAC     T2              SAVE VALUE
013410  515115      2700            AND     (400000)        FIND PROPPER BOUNDARY
013411  055200      2710            DAC     T1              SAVE IT
013412  775200      2720            LAW     T1
013413  055203      2730            DAC     T4              SET POINTER TO A ZERO
013414  213336      2740            LAC     SPTR            GET SYMBOL TABLE POINTER
013415  355116      2750            TAD     (2)             MAKE IT SKIP THE NAMES
013416  040012      2760            DAC     SYMBX           SAVE IN XREGISTER
013417  213274      2770            LAC     SCOUNT          GET COUNT
013420  055204      2780            DAC     T5              SAVE IT
013421  220012      2790    SFL     LAC     SYMBX,X         GET THE VALUE
    013422          2800            NEG                     NEGATE
013422  740001                      CMA
013423  355100                      TAD     (1)
013424  055202      2810            DAC     T3              SAVE IT
013425  355201      2820            TAD     T2              ADD SYMBOL VALUE
013426  741100      2830            SPA
013427  613436      2840            JMP     SFX
013430  215202      2850            LAC     T3              RESTORE NEGATED SYMBOL
013431  375203      2860            TAD     T4,X            ADD VALUE
013432  740300      2870            SMA!SZA                 CHECK FOR RUNNOUT
013433  613436      2880            JMP     SFX             EXIT
013434  200012      2890            LAC     SYMBX           GET SYMBX POINTER

013435  055203      2900            DAC     T4              SAVE IT
013436  440012      2910    SFX     ISZ     SYMBX
013437  440012      2920            ISZ     SYMBX
    013440          2930            LOOP    T5,SFL          LOOP ON COUNTER
013440  455204                      ISZ     T5
013441  613421                      JMP     SFL
013442  235203      2940            LAC     T4,X            GET VALUE
013443  741200      2950            SNA                     CHECK FOR NOTHING THERE
```

SUBROUTINES

```
013444  613503  2960           JMP     SFD6            YES
    013445      2970           NEG
013445  740001                 CMA
013446  355100                 TAD     (1)
013447  355201  2980           TAD     T2              DECREMENT ORRIGNAL POINTER
013450  055201  2990           DAC     T2
013451  777775  3000           LAW     -3              BACK UP POINTER
013452  355203  3010           TAD     T4              ADD POINTER
013453  040012  3020           DAC     SYMBX           SAVE
013454  777776  3030           LAW     -2              GET COUNT OF WORDS
013455  055200  3040           DAC     T1
013456  155203  3050           DZM     T4              CLEAR FLAG
013457  777775  3060   SFD1    LAW     -3              GET CHARACTER COUNT
013460  055202  3070           DAC     T3              SAVE IT
013461  220012  3080           LAC     SYMBX,X
013462  652000  3090           LMQ                     LOAD THE MQ REG
    013463      3100   SFD5    ...
013463  750000  3110           CLA
013464  660606  3120           LLSS    6               SHIFT IN
013465  741200  3130           SNA                     CHECK FOR A ZERO
013466  613506  3140           JMP     SFD3            CHECK IT
013467  455203  3150           ISZ     T4              SET FLAG
    013470      3160   SFD4    ...
013470  355117  3170           TAD     (40)            FORM ASCII
013471  115665  3180           JMS     TSTTYOT         PRINT IT
    013472      3190   SFD4A   ...
    013472      3200           LOOP    T3,SFD5
013472  455202                 ISZ     T3
013473  613463                 JMP     SFD5
    013474      3210           LOOP    T1,SFD1
013474  455200                 ISZ     T1
013475  613457                 JMP     SFD1
013476  215201  3220           LAC     T2              GET VALUE
013477  741200  3230           SNA                     CHECK FOR NOTHING
013500  633377  3240           RET     DUMA            IF SO, DONE
013501  215120  3250           LAC     (53)            GET A +
013502  115665  3260           JMS     TSTTYOT         PRINT IT
013503  215201  3270   SFD6    LAC     T2              GET VALUE
013504  113711  3280           JMS     P6Z             PRINT IT
013505  633377  3290           RET     DUMA            AT LAST!
                3300   *
013506  215203  3310   SFD3    LAC     T4              CHECK FLAG
013507  741200  3320           SNA                     SKIP IF WE SHOULD PRINT
013510  613472  3330           JMP     SFD4A
013511  155203  3340           DZM     T4              SET FLAG AGAIN
013512  215121  3350           LAC     (4)             PRINT 'S'
013513  613470  3360           JMP     SFD4            PRINT IT
```

SUBROUTINES

```
                    3370            ,EJECT
                    3380    *
                    3390    *       PDEC -
                    3400    *
                    3410    *       PRINT A NUMBER IN DECIMAL
                    3420    *
013514  000000      3430    PDEC    0
013515  055200      3440            DAC     T1              SAVE
013516  740100      3450            SMA
013517  613525      3460            JMP     PDE1
    013520          3470            NEG                     MAKE PLUS
013520  740001                      CMA
013521  355100                      TAD     (1)
013522  055200      3480            DAC     T1              SAVE FOR NOW
013523  215122      3490            LAC     (55)            GET A -
013524  115665      3500            JMS     TSTTYOT         PRINT CHARACTER
013525  775210      3510    PDE1    LAW     DECB-1          POINT TO DECIMAL OUTPUT BUFFER
013526  040012      3520            DAC     SYMBX           SAVE POINTER
013527  215200      3530            LAC     T1              GET THE OLD CHARACTER
013530  744000      3540    DECL    CLL
013531  653323      3550            IDIV                    DIVIDE
013532  000012      3560            12                      BY 10
013533  060012      3570            DAC     SYMBX,X         SAVE A DIGIT
013534  641002      3580            LACQ                    GET THE QUOTIENT
013535  740200      3590            SZA
013536  613530      3600            JMP     DECL            LOOP
013537  440012      3610            ISZ     SYMBX           ADVANCE POINTER
    013540          3620    DECL1   ...
013540  777776      3630            LAW     -2              GET AMOUNT TO BACKSPACE
013541  340012      3640            TAD     SYMBX           GET IT
013542  555123      3650            SAD     (LAW DECB-2     CHECK FOR DONE
013543  613551      3660            JMP     PDX             EXIT IF SO
013544  040012      3670            DAC     SYMBX           SAVE POINTER
013545  220012      3680            LAC     SYMBX,X         GET NEXT WORD
013546  355124      3690            TAD     (60)            MAKE NUMERIC
013547  115665      3700            JMS     TSTTYOT         PRINT IT
013550  613540      3710            JMP     DECL1           LOOP
013551  215076      3720    PDX     LAC     (POINT)         GET A DECIMAL POINT
013552  115665      3730            JMS     TSTTYOT         PRINT A CHARACTER
013553  633514      3740            RET     PDEC            RETURN
                    3750    *
    015211          3760            ,USE    STOR
    015211          3770    DECB    ,BLOCK  6,
    013554          3780            ,USE
```

SUBROUTINES

```
                    3790            .EJECT
                    3800    *
                    3810    *       DUMP SUBROUTINE
                    3820    *
                    3830    *       DUMPS FROM LOC LOC TO LOC  -N IN THE FORMAT DETERMINED
                    3840    *       BY FORMSW
                    3850    *
013554  000000      3860    DUMP    0
013555  113605      3870            JMS     EOLP            PRINT EOL
013556  213671      3880            LAC     LOC
013557  615125      3890            AND     (17777)         WRAP CORE IF NECESSARY
013560  113377      3900            JMS     DUMA            DUMP IT
013561  113614      3910            JMS     SPCOL           :
013562  233671      3920            LAC     LOC,X           GET WORD
013563  613566      3930    FORMSW  JMP     FORMO           OCTAL TO START
013564  113600      3940    DUMX    JMS     NEXT            END OF LINE
013565  613555      3950            JMP     DUMP+1          LOOP
                    3960    *
                    3970    *       OCTAL
                    3980    *
                    3990
    013566          4000    FORMO   ...
    013566          4010            OCT                     PRINT IT OCTAL
013566  744000                      CLL
013567  115502                      JMS     TSOCTOT
013570  113600      4020            JMS     NEXT            GET THE NEXT LOC
013571  213671      4030            LAC     LOC             GET THE LOCATION COUNTER
013572  615126      4040            AND     (7)             LOOK FOR NEEDING ANOTHER ADDRESS
013573  741200      4050            SNA                     SKIP IF ON THE SAME LINE
013574  613555      4060            JMP     DUMP+1          NO REPRINT ADD
013575  113610      4070            JMS     SPAC            PRINT SPACE
013576  233671      4080            LAC     LOC,X           GET WORD
013577  613566      4090            JMP     FORMO           LOOP
                    4100    *
                    4110    *       NEXT
                    4120    *
013600  000000      4130    NEXT    0                       GET THE NEXT WORD
013601  453671      4140            ISZ     LOC             BUMP LOC
013602  453672      4150            ISZ     N               COUNT
013603  633600      4160            RET     NEXT            RETURN IF OK
013604  633554      4170            RET     DUMP            RETURN FROM DUMP IF NOT
                    4180    *
                    4190    *       EOLP
                    4200    *
013605  000000      4210    EOLP    0
    013606          4220            CRLF                    PRINT A NULL MESSAGE
013606  115673                      JMS     TSCRLF
013607  633605      4230            RET     EOLP
                    4240    *
                    4250    *       SPACE
                    4260    *
013610  000000      4270    SPAC    0
```

SUBROUTINES

```
                    3790            .EJECT
                    3800    *
                    3810    *       DUMP SUBROUTINE
                    3820    *
                    3830    *       DUMPS FROM LOC LOC TO LOC  -N IN THE FORMAT DETERMINED
                    3840    *       BY FORMSW
                    3850    *
013554  000000      3860    DUMP    0
013555  113605      3870            JMS     EOLP            PRINT EOL
013556  213671      3880            LAC     LOC
013557  615125      3890            AND     (17777)         WRAP CORE IF NECESSARY
013560  113377      3900            JMS     DUMA            DUMP IT
013561  113614      3910            JMS     SPCOL           ;
013562  233671      3920            LAC     LOC,X           GET WORD
013563  613566      3930    FORMSW  JMP     FORMO           OCTAL TO START
013564  113600      3940    DUMX    JMS     NEXT            END OF LINE
013565  613555      3950            JMP     DUMP+1          LOOP
                    3960    *
                    3970    *       OCTAL
                    3980    *
                    3990
    013566          4000    FORMO   ...
    013566          4010            OCT                     PRINT IT OCTAL
013566  744000                      CLL
013567  115502                      JMS     TSOCTOT
013570  113600      4020            JMS     NEXT            GET THE NEXT LOC
013571  213671      4030            LAC     LOC             GET THE LOCATION COUNTER
013572  615126      4040            AND     (7)             LOOK FOR NEEDING ANOTHER ADDRESS
013573  741200      4050            SNA                     SKIP IF ON THE SAME LINE
013574  613555      4060            JMP     DUMP+1          NO REPRINT ADD
013575  113610      4070            JMS     SPAC            PRINT SPACE
013576  233671      4080            LAC     LOC,X           GET WORD
013577  613566      4090            JMP     FORMO           LOOP
                    4100    *
                    4110    *       NEXT
                    4120    *
013600  000000      4130    NEXT    0                       GET THE NEXT WORD
013601  453671      4140            ISZ     LOC             BUMP LOC
013602  453672      4150            ISZ     N               COUNT
013603  633600      4160            RET     NEXT            RETURN IF OK
013604  633554      4170            RET     DUMP            RETURN FROM DUMP IF NOT
                    4180    *
                    4190    *       EOLP
                    4200    *
013605  000000      4210    EOLP    0
    013606          4220            CRLF                    PRINT A NULL MESSAGE
013606  115673                      JMS     TSCRLF
013607  633605      4230            RET     EOLP
                    4240    *
                    4250    *       SPACE
                    4260    *
013610  000000      4270    SPAC    0
```

SUBROUTINES

```
013611  215117  4280            LAC     (40)            GET A SPACE
013612  115665  4290            JMS     T$TTYOT         PRINT IT
013613  633610  4300            RET     SPAC
                4310    *
                4320    *       PCOL
                4330    *
013614  000000  4340    SPCOL   0
013615  215127  4350            LAC     (72)            COLON
013616  115665  4360            JMS     T$TTYOT
013617  113610  4370            JMS     SPAC            SPACE IT OUT
013620  633614  4380            RET     SPCOL
                4390    *
                4400    *       P6
                4410    *
013621  000000  4420    P6      0
013622  652000  4430            LMQ                     LOAD WORD INTO MQ
                4440            .DUP    3,3
013623  641606  4450            EAECLA!LLS 6            SHIFT IN TO AC
013624  355117  4460            TAD     (40)            MAKE ASCII
013625  115665  4470            JMS     T$TTYOT
013626  641606                  EAECLA!LLS 6            SHIFT IN TO AC
013627  355117                  TAD     (40)            MAKE ASCII
013630  115665                  JMS     T$TTYOT
013631  641606                  EAECLA!LLS 6            SHIFT IN TO AC
013632  355117                  TAD     (40)            MAKE ASCII
013633  115665                  JMS     T$TTYOT
013634  633621  4480            RET     P6              RETURN
                4490    *
                4500    *       SYMBOLIC FORMAT
                4510    *
    013635      4520    FORMS   ...
013635  515130  4530            AND     (20000)         EXTRACT INDIRECT FLAG
013636  053673  4540            DAC     INDB            SAVE IT
013637  233671  4550            LAC     LOC,X           GET WORD AGAIN
013640  744000  4560            CLL                     CLEAR LINK
013641  640516  4570            LRS     18.-4.          MOVE OVER
013642  355131  4580            TAD     (-15)           CHECK FOR $OT EAE OR OPR
013643  740100  4590            SMA                     NO
013644  613655  4600            JMP     FORMS1          YES -- TROUBLE
013645  355132  4610            TAD     (OPTB+15)       POINT INTO TABLE
013646  055200  4620            DAC     T1              SAVE IT
013647  235200  4630            LAC     T1,X            GET OPCODE
013650  113621  4640            JMS     P6              PRINT IT
013651  113610  4650            JMS     SPAC
013652  233671  4660            LAC     LOC,X           GET THE WORD
013653  515125  4670            AND     (17777)         EXTRACT GOOD BITS
013654  613657  4680            JMP     FORMS2          SKIP OTHER CODE
013655  153673  4690    FORMS1  DZM     INDB            CLEAR INDIRECT FLAG
013656  233671  4700            LAC     LOC,X           GET THE WORD
013657  113377  4710    FORMS2  JMS     DUMA            PRINT IT
013660  213673  4720            LAC     INDB            GET INDIRECT FLAG
013661  741200  4730            SNA                     SKIP IF IT NEED BE PRINTED
```

SUBROUTINES

```
013662  613564  4740            JMP     DUMX
013663  215133  4750            LAC     (54)            GET A ,
013664  115665  4760            JMS     TSTTYOT
013665  215134  4770            LAC     (130)           GET A X
013666  115665  4780            JMS     TSTTYOT         PRINT IT
013667  613564  4790            JMP     DUMX
013670  613635  4800            JMP     FORMS           CONTINUE
                4810    *
                4820    *
013671  000000  4830    LOC     0
013672  000000  4840    N       0
013673  000000  4850    INDB    0
    013674      4860    OPTB    ,,,
013674  434154  4870            .DATA   434154,444143,525563,447255,544143,705762
013675  444143
013676  525563
013677  447255
013700  544143
013701  705762
013702  414444  4880            .ACI6   /ADDTADXCTISZANDSADJMP/
013703  644144
013704  704364
013705  516372
013706  415644
013707  634144
013710  525560
```

SUBROUTINES

```
                    4890            .EJECT
                    4900    *
                    4910    *       PRINT IN OCTAL WITH ZERO SUPPRESSION
                    4920    *
013711  000000      4930    P6Z     0
    013712          4940            OCTZ                    PRINT THE AC WITH LEADING ZEROES SUPPRESSED
013712  744002                      STL
013713  115502                      JMS     TSOCTOT
013714  633711      4950            RET     P6Z,X
                    4960
    013715          4970    P6OCT   ENTER
013715  740040                      XX
    013716          4980            OCT                     PRINT THE AC AS A SIX DIGIT OCTAL NUMBER
013716  744000                      CLL
013717  115502                      JMS     TSOCTOT
013720  633715      4990            RET     P6OCT,X
                    5000            .EOT    TSSDDT1
```

COMMANDS

```
                    100             .STITL  COMMANDS
                    110     *
                    120     *       COMMAND TABLE
                    130     *
        013721      140     COMTB   ...
013721  446555      150             446555
013722  614076      160             JMP     DUM
013723  604164      170             604164
013724  614155      180             JMP     PAT
013725  414444      190             414444
013726  614205      200             JMP     ADD
013727  637155      210             637155
013730  614171      220             JMP     SYM
013731  574364      230             574364
013732  614173      240             JMP     OCT
013733  444543      250             444543
013734  614175      260             JMP     DEC
013735  635170      270             635170
013736  614177      280             JMP     SIX
013737  634166      290             634166
013740  614232      300             JMP     SAVE
013741  457051      310             457051
013742  705001      320             TERMINAT                TSS DDT EXIT IS A HLT, WHICH RETURNS TO THE MONITOR
013743  700000      330             700000
013744  705001      340             TERMINAT                'X' ALSO EXITS
013745  624547      350             624547
013746  614271      360             JMP     REG
013747  415464      370             415464
013750  614310      380             JMP     ALT
013751  544463      390             544463
013752  614365      400             JMP     LDS
013753  414463      410             414463
013754  614371      420             JMP     ADS
013755  504541      430             504541
013756  614417      440             JMP     HEA
013757  426245      450             426245
013760  614440      460             JMP     BRE
013761  655642      470             655642
013762  614470      480             JMP     UNB
013763  525560      490             525560
013764  614617      500             JMP     TRA
013765  435756      510             435756
013766  614623      520             JMP     CON
013767  545741      530             545741
013770  614734      540             JMP     LOAD
013771  554163      550             554163
013772  615025      560             JMP     MAS
013773  634541      570             634541
013774  615030      580             JMP     SEA
        013775      590     COME    ...
        000014      600     CMDX    .EQU    14              COMMAND REGISTER
                    610     *
```

COMMANDS

```
                    620     *       STARTUP
                    630     *
013775  703302      640     START   CAF                     CLEAR Up
013776  700002      650             IOF
013777  700416      660             TLS+10                  PRINT A GRITCH
    014000          670             MESS    <DDT HERE>,8.
014000  700312                      KRB
    014001                          MESSR   <DDT HERE>,8.
                                    ,CRSM   SAVE,ON
014001  777766                      LAW     -8,-2
014002  115561                      JMS     T$SIXOT
                                    ,PMC    RESTORE
                                    ,CRSM   RESTORE
                    680     *
                    690     *       IDLE LOOP
                    700     *
    014007          710     IDLE    ...
    014007          720             MESS    <?>,1
014007  700312                      KRB
    014010                          MESSR   <?>,1
                                    ,CRSM   SAVE,ON
014010  777775                      LAW     -1-2
014011  115561                      JMS     T$SIXOT
                                    ,PMC    RESTORE
                                    ,CRSM   RESTORE
    014013          730             LINE                    READ A LINE
014013  115370                      JMS     T$INLIN
    014014          740     COMA    ...
014014  215135      750             LAC     (JMP COMTB-1)   GET POINTER
014015  040014      760             DAC     CMDX            SAVE IN REGISTER
014016  777724      770             LAW     COMTB-COME      GET COUNT
014017  055200      780             DAC     T1              SAVE IT
    014020          790             WORD                    GET A WORD OF INPUT
014020  115526                      JMS     T$SIXIN
014021  614007      800             JMP     IDLE            INPUT LINE WAS VACUOUS
014022  660014      810     COM0    SAD     CMDX,X          CHECK FOR A COMMAND
014023  620014      820             JMP     CMDX,X          GO TO IT
    014024          830             LOOP    T1,COM0         LOOP ON COUNTER
014024  455200                      ISZ     T1
014025  614022                      JMP     COM0
    014026          840     COM1    EMESS   <COMMAND ERROR>,13.
                                    ,CRSM   SAVE,ON
    014026                          MESS    <COMMAND ERROR WORD #>,13.-7
014026  700312                      KRB
    014027                          MESSR   <COMMAND ERROR WORD #>,13.-7
                                    ,CRSM   SAVE,ON
014027  777752                      LAW     -13.-7-2
014030  115561                      JMS     T$SIXOT
                                    ,PMC    RESTORE
                                    ,CRSM   RESTORE
    014041                          COUNT
014041  215367                      LAC     T$COUNT
```

COMMANDS

```
    014042                          OCTZ
014042  744002                      STL
014043  115502                      JMS     TSOCTOT
                                    ,CRSM   RESTORE
014044  614007  850                 JMP     IDLE
              860   *
              870   *       FORMAT ERROR
              880   *
    014045    890   FORMAT  EMESS   <FORMAT ERROR>,12.
                              ,CRSM   SAVE,ON
    014045                  MESS    <FORMAT ERROR WORD #>,12,-7
014045  700312                      KRB
    014046                  MESSR   <FORMAT ERROR WORD #>,12,-7
                              ,CRSM   SAVE,ON
014046  777753                      LAW     -12.,-7-2
014047  115561                      JMS     TSSIXOT
                                    ,PMC    RESTORE
                                    ,CRSM   RESTORE
    014057                          COUNT
014057  215367                      LAC     TSCOUNT
    014060                          OCTZ
014060  744002                      STL
014061  115502                      JMS     TSOCTOT
                                    ,CRSM   RESTORE
014062  614007  900                 JMP     IDLE
              910   *
    014063    920   COMX    DELIM                   GET THE LAST DELIMITER
014063  215366                      LAC     TSDLMTR
014064  555136  930                 SAD     ($COLON)        CHECK FOR A SEMI-COLON
014065  614074  940                 JMP     COMY            YES -- TROUBLE
014066  555137  950                 SAD     (CR)            CHECK FOR A CARRIAGE RETURN
014067  614007  960                 JMP     IDLE            YES
014070  555140  970                 SAD     (NUMSGN)        CHECK FOR #
014071  741000  980                 SKP
    014072      990                 FORMAT                  FORMAT ERROR IF NOT
014072  614045                      JMP     FORMAT
014073  614014  1000                JMP     COMA            ENTER COMMAND SCAN
014074  400014  1010    COMY        XCT     CMDX            GET THE POINTER
014075  000000  1020    CMDN        0                       COMMAND NUMBER
```

COMMANDS

```
                    1030            ,EJECT
                    1040    *
                    1050    *       DUMP
                    1060    *
                    1070    *       DUM ADD1;ADD2;ADD3              OR
                    1080    *       DUM ADD1 L;ADD2 L;ADD3 L        OR
                    1090    *       DUM ADD1L,ADD1U;ADD2L,ADD2U
                    1100    *
    014076          1110    DUM     ...
014076  113120      1120            JMS     EXP             GET AN EXPRESSION
014077  053671      1130            DAC     LOC             SAVE IT
    014100          1140            DELIM                   GET THE DELIMITER
014100  215366                      LAC     T$DLMTR
014101  555075      1150            SAD     (SPACE)         CHECK FOR A SPACE
014102  614107      1160            JMP     DUM1
014103  555141      1170            SAD     (COMMA)         CHECK FOR A COMMA
014104  614117      1180            JMP     DUM2
014105  777777      1190            LAW     -1              ELSE DUMP ONLY ONE WORD
014106  614114      1200            JMP     DUM3
014107  113120      1210    DUM1    JMS     EXP             GET ANOTHER EXPRESSION
    014110          1220            NEG                     FORM 2'S COMPLEMENT
014110  740001                      CMA
014111  355100                      TAD     (1)
014112  740100      1230            SMA                     CHECK FOR PROPER LENGTH
    014113          1240            FORMAT
014113  614045                      JMP     FORMAT
014114  053672      1250    DUM3    DAC     N               SAVE FOR DUMP ROUTINE
014115  113554      1260            JMS     DUMP            DUMP IT
014116  614063      1270            JMP     COMX
014117  113120      1280    DUM2    JMS     EXP             GET ANOTHER EXPRESSION
014120  740001      1290            CMA                     NEGATE
014121  353671      1300            TAD     LOC             FORM DIFFERENCE
014122  740100      1310            SMA                     CHECK FOR OK
014123  777777      1320            LAW     -1              GET -1 IF NOT
014124  614114      1330            JMP     DUM3            CONTINUE
```

COMMANDS

```
                    1340            .EJECT
                    1350    *
                    1360    *
                    1370    *       GET WORD ROUTINE
                    1380    *
 014125  000000     1390    GWORD   0
 014126  113120     1400            JMS     EXP             GET AN EXPRESSION
 014127  055204     1410            DAC     T5              SAVE FOR NOW
 014130  215365     1420            LAC     T$CHAR          GET THE CHARACTER THAT STOPPED US
 014131  555075     1430            SAD     (SPACE)         CHECK FOR SPACE
 014132  614135     1440            JMP     GWD1
    014133          1450    GWD0    ...
 014133  215204     1460            LAC     T5              GET THE VALUE
 014134  634125     1470            JMP     GWORD,X         RETURN
                    1480    *
                    1490    *       INSTRUCTION FOUND
                    1500    *
 014135  113120     1510    GWD1    JMS     EXP             GET ANOTHER EXPRESSION
 014136  515125     1520            AND     (ADRSS)         EXTRACT ADDRESS FIELD
 014137  355204     1530            TAD     T5              ADD TO CURRENT ONE
 014140  055204     1540            DAC     T5              SAVE IT
 014141  215365     1550            LAC     T$CHAR          GET THE CHARACTER
 014142  555141     1560            SAD     (COMMA)         CHECK FOR ,
 014143  741000     1570            SKP
 014144  614133     1580            JMP     GWD0            NO
 014145  115607     1590            JMS     T$FGET          GET A CHARACTER
 014146  555134     1600            SAD     (130)           CHECK FOR 'X'
 014147  741000     1610            SKP
    014150          1620            FORMAT                  FORMAT ERROR IF NOT
 014150  614045                     JMP     FORMAT
 014151  215204     1630            LAC     T5              GET VALUE
 014152  515142     1640            AND     (757777)        TRIM
 014153  355130     1650            TAD     (20000)         MAKE INDIRECT
 014154  634125     1660            RET     GWORD
```

COMMANDS

```
                    1670            .EJECT
                    1680    *
                    1690    *       PATCH
                    1700    *
                    1710    *       PATCH ADD;LOC LOC LOC;LOC LCO LOC
                    1720    *
    014155          1730    PAT     ...;
014155  113120      1740            JMS     EXP             GET THE ADDRESS
014156  053671      1750            DAC     LOC             SAVE IT
    014157          1760    PAT0    DELIM                   GET A CHARACTER
014157  215366                      LAC     TSDLMTR
014160  555075      1770            SAD     (SPACE)         CHECK FOR SPACE
014161  614165      1780            JMP     PAT1
014162  555136      1790            SAD     (SCOLON)        CHECK FOR ;
014163  741000      1800            SKP
014164  614063      1810            JMP     COMX            DONE IF NOT
014165  114125      1820    PAT1    JMS     GWORD           GET A WORD
014166  073671      1830            DAC     LOC,X           SAVE IT
014167  453671      1840            ISZ     LOC             INDEX
014170  614157      1850            JMP     PAT0            LOOP
                    1860    *
                    1870    *       SYM
                    1880    *
014171  215143      1890    SYM     LAC     (JMP FORMS)     GET THE PROPER JUMP
014172  614200      1900            JMP     FORM
                    1910    *
                    1920    *       OCT
                    1930    *
014173  215144      1940    OCT     LAC     (JMP FORMO)     GET THE PROPER JUMP
014174  614200      1950            JMP     FORM            DO IT
                    1960    *
                    1970    *       DEC
                    1980    *
014175  215145      1990    DEC     LAC     (JMS PDEC)      DECIMAL FORMAT
014176  614200      2000            JMP     FORM            DO IT
                    2010    *
                    2020    *       SIX
                    2030    *
014177  215146      2040    SIX     LAC     (JMS P6         GET FORMAT
014200  053563      2050    FORM    DAC     FORMSW          SAVE IT
014201  215365      2060            LAC     TSCHAR
014202  555075      2070            SAD     (SPACE)         CHECK FOR SPACE
014203  614014      2080            JMP     COMA            ANOTHER COMMAND IF SO
014204  614063      2090            JMP     COMX            TRY IF NOT
                    2100    *
                    2110    *       ADD
                    2120    *
014205  774223      2130    ADD     LAW     ADDTB-1         GET TABLE START
014206  040010      2140            DAC     INDEX           SAVE IT
014207  777775      2150            LAW     -ADDTL/2
014210  055200      2160            DAC     T1
    014211          2170            WORD                    GET A WORD
```

COMMANDS

```
014211  115526           2170          JMS     TSSIXIN
014212  740000  2180             NOP
014213  560010  2190     ADDL    SAD     INDEX,X         CHECK IT
014214  614221  2200             JMP     ADD1
014215  440010  2210             ISZ     INDEX
    014216       2220             LOOP    T1,ADDL         LOOP
014216  455200                   ISZ     T1
014217  614213                   JMP     ADDL
    014220       2230             FORMAT                  FORMAT ERROR
014220  614045                   JMP     FORMAT
014221  220010  2240     ADD1    LAC     INDEX,X         GET FORMAT
014222  053400  2250             DAC     AFORM           SAVE IT
014223  614063  2260             JMP     COMX            RETURN
                2270     *
                2280     *       ADDRESS FORMAT TABLE
                2290     *
    014224       2300     ADDTB   ...
014224  574364  2310             574364
014225  613401  2320             JMP     OCTF
014226  444543  2330             444543
014227  613405  2340             JMP     DECF
014230  637155  2350             637155
014231  613407  2360             JMP     SYMF
    000006       2370     ADDTL   .EQU    .-ADDTB
```

COMMANDS

```
                    2380            .EJECT
014232  215121      2390    SAVE    LAC     (D$WRITE)       LOAD THE WRITE COMMAND
014233  115707      2400            JMS     D$DO            GO DO THE DISK WRITE
014234  774236      2410            LAW     SAV4            POINTER TO THE DISK FILE PARAMETERS
014235  614063      2420            JMP     COMX            NEXT??
                    2430
014236  000000      2440    SAV4    0                       STARTING IN DISK BLOCK ZERO
014237  002000      2450            BOUNDARY                CORE STARTING ADDRESS
014240  010000      2460            $BASE-BOUNDARY          LENGTH
```

COMMANDS

```
                    2470            .EJECT
                    2480    *
                    2490    *       REGF
                    2500    *
                    2510    *       FIND A REGISTER
                    2520    *
014241  000000      2530    REGF    0
014242  774317      2540            LAW     REGT-1          GET POINTER TO TABLE
014243  040010      2550            DAC     INDEX           SAVE IT
014244  777772      2560            LAW     -REGN/3         GET NUMBER OF REGISTERS
014245  055200      2570            DAC     T1              SAVE IT
    014246          2580            WORD                    GET A WORD
014246  115526                      JMS     TSSIXIN
014247  740000      2590            NOP
014250  515147      2600            AND     (770000)        GET THE FIRST CHARACTER
014251  555150      2610            SAD     (700000)        CHECK FOR 'X'
014252  614263      2620            JMP     REG2            DO IT
014253  215237      2630            LAC     TSWORDB         RESTORE BUFFER
014254  560010      2640    REGL    SAD     INDEX,X         CHECK IT
014255  634241      2650            RET     REGF            RETURN IF WE FIND IT
014256  440010      2660            ISZ     INDEX
014257  440010      2670            ISZ     INDEX
    014260          2680            LOOP    T1,REGL         LOOP ON COUNTER
014260  455200                      ISZ     T1
014261  614254                      JMP     REGL
    014262          2690            FORMAT                  FORMAT ERROR
014262  614045                      JMP     FORMAT
    014263          2700    REG2    ...
014263  215237      2710            LAC     TSWORDB         GET THE BUFFER
014264  640506      2720            LRS     6               MOVE OVER
014265  515126      2730            AND     (7)             TRIM
014266  555151      2740            TAD     (REGS-1)        POINT
014267  040010      2750            DAC     INDEX           SAVE POINTER
014270  634241      2760            RET     REGF            RETURN
                    2770    *
                    2780    *       REGISTER COMMAND
                    2790    *
014271  114241      2800    REG     JMS     REGF            FIND A REGISTER
014272  113605      2810            JMS     EOLP            PRINT EOL
014273  215237      2820            LAC     TSWORDB         GET THE REGISTER
014274  113621      2830            JMS     P6              PRINT IT
014275  113614      2840            JMS     SPCOL           !
014276  215237      2850            LAC     TSWORDB         CHECK FOR X REGISTERS
014277  515147      2860            AND     (770000)
014300  555150      2870            SAD     (700000)        CHECK FOR X
014301  614305      2880            JMP     REG1
014302  220010      2890            LAC     INDEX,X
014303  420010      2900            XCT     INDEX,X         DO SOMETHING
014304  614063      2910            JMP     COMX            EXIT
    014305          2920    REG1    ...
014305  220010      2930            LAC     INDEX,X         GET THE REGISTER
014306  113377      2940            JMS     DUMA            DUMP IT
```

COMMANDS

```
014307  614063  2950            JMP     COMX            EXIT
                2960    *
                2970    *       ALTER
                2980    *
014310  114241  2990    ALT     JMS     REGF            FIND THE REGISTER
014311  440010  3000            ISZ     INDEX           FORM POINTER
014312  200010  3010            LAC     INDEX
014313  054317  3020            DAC     REGI            SAVE IT
014314  114125  3030            JMS     GWORD           GET A WORD FOR IT
014315  074317  3040            DAC     REGI,X          SAVE IT
014316  614063  3050            JMP     COMX            EXIT
014317  000000  3060    REGI    0
                3070    *
                3080    *       REGISTER TABLE
                3090    *
    014320      3100    REGT    ...
014320  414300  3110            414300
014321  000000  3120    AC      0
014322  113377  3130            JMS     DUMA            DUMP
014323  556100  3140            556100
014324  000000  3150    MQ      0
014325  113377  3160            JMS     DUMA            DUMP IT
014326  604300  3170            604300
014327  000000  3180    PC      0
014330  113377  3190            JMS     DUMA
014331  545300  3200            545300
014332  000000  3210    LK      0
014333  113711  3220            JMS     P6Z
014334  634300  3230            634300
014335  000000  3240    SC      0
014336  113711  3250            JMS     P6Z
014337  636463  3260            636463
014340  000000  3270    STS     0
    014341      3280            OCT                     IN OCTAL ALWAYS
014341  744000                  CLL
014342  115502                  JMS     TSOCTOT
    000023      3290    REGN    .EQU    .-REGT          NUMBER OF REGISTERS
```

COMMANDS

```
                    3300            .EJECT
                    3310    *
                    3320    *       TAPE READER ROUTINES
                    3330    *
014343  000000      3340    GETW    0
014344  154363      3350            DZM     TIME            CLEAR TIMER
014345  700144      3360            RSB                     START THE READER
    014346          3370    GW1     ...
014346  700101      3380            RSF                     CHECK FOR CHARACTER IN
014347  741000      3390            SKP
014350  614354      3400            JMP     WIN             YES
014351  454363      3410            ISZ     TIME            CHECK FOR TIMEOUT
014352  614346      3420            JMP     GW1
014353  634343      3430            RET     GETW            RETURN TIMEOUT
014354  700112      3440    WIN     RRB                     GET THE WORD
014355  054363      3450            DAC     WT              SAVE IT
014356  254364      3460            XOR     CKSUM           FORM CHECKSUM
014357  054364      3470            DAC     CKSUM           SAVE IT
014360  214363      3480            LAC     WT              RESTORE WORD
014361  454343      3490            ISZ     GETW            GIVE GOOD RETURN
014362  634343      3500            RET     GETW            RETURN
    014363          3510    WT      ...
014363  000000      3520    TIME    0
014364  000000      3530    CKSUM   0
                    3540    *
                    3550    *       LOAD SYMBOLTABLE
                    3560    *
    014365          3570    LDS     ...
014365  771673      3580            LAW     SYMST-1         GET POINTER TO SYMBOL TABLE
014366  053336      3590            DAC     SPTR            SAVE POINTER
014367  777757      3600            LAW     -SYMN           GET NUMBER OF SYMBOLS
014370  053274      3610            DAC     SCOUNT          SAVE IT
                    3620    *
                    3630    *       ADDD SYMBOLTABLE
                    3640    *
    014371          3650    ADS     ...
014371  777775      3660            LAW     -3              DECREMENT POINTER
014372  055200      3670            DAC     T1              SAVE COUNT
014373  353336      3680            TAD     SPTR
014374  053336      3690            DAC     SPTR
014375  040012      3700            DAC     SYMBX           SAVE FOR OUT USE
014376  777777      3710            LAW     -1              BACK UP COUNT
014377  353274      3720            TAD     SCOUNT
014400  053274      3730            DAC     SCOUNT
014401  114343      3740    ADSL    JMS     GETW            GET A WORD
014402  614411      3750            JMP     ADSX            DONE
014403  060012      3760            DAC     SYMBX,X         SAVE IT
    014404          3770            LOOP    T1,ADSL         LOOP
014404  455200                      ISZ     T1
014405  614401                      JMP     ADSL
014406  114343      3780            JMS     GETW            IGNORE EXTRA WORD
014407  614411      3790            JMP     ADSX            EXIT IF NONE
```

COMMANDS

```
014410  614371  3800            JMP     ADS             ADD SOME MORE
014411  213336  3810    ADSX    LAC     SPTR            GET POINTER
014412  355113  3820            TAD     (3)             ADD 3
014413  053336  3830            DAC     SPTR            BACK UP TO LAST SYMBOL
014414  453274  3840            ISZ     SCOUNT          AND UNCOUNT IT
014415  700112  3850            RRB                     CLEAR FLAG IN STUBORN READER
014416  614063  3860            JMP     COMX            CONTINUE
                3870    *
                3880    *       HEAD
                3890    *
    014417      3900    HEA     WORD                    GET THE SYMBOL
014417  115526                  JMS     TSSIXIN
014420  055210  3910            DAC     HEAD            SAVE IT
014421  614063  3920            JMP     COMX            EXIT
                3930    *
                3940    *       BREAK
                3950    *
                3960    *       SUBROUTINE TO SEARCH BREAK POINT TABLE
                3970    *
014422  000000  3980    BLOOK   0
014423  055200  3990            DAC     T1              SAVE THING TO LOOK FOR
014424  215152  4000            LAC     (DAC BTBL-1)    GET POINTER
014425  040010  4010            DAC     INDEX           SAVE IT
014426  777770  4020            LAW     -10             GET THE NUMBER OF BREAK POINTES
014427  055201  4030            DAC     T2
014430  215200  4040            LAC     T1              RESTORE PATTERN
014431  560010  4050    BRL     SAD     INDEX,X         CHECK FOR ONE
014432  634422  4060            RET     BLOOK           RETURN IF WE FIND IT
014433  440010  4070            ISZ     INDEX
    014434      4080            LOOP    T2,BRL          LOOP ON COUNTER
014434  455201                  ISZ     T2
014435  614431                  JMP     BRL
014436  454422  4090            ISZ     BLOOK           INCREMENT RETURN
014437  634422  4100            RET     BLOOK
                4110    *
                4120    *       BREAK TABLE
                4130    *
    015217      4140            .USE    STOR
                4150            .DUP    1,10
015217  000000  4160    BTBL    .DATA   0,0
015220  000000
015221  000000                  .DATA   0,0
015222  000000
015223  000000                  .DATA   0,0
015224  000000
015225  000000                  .DATA   0,0
015226  000000
015227  000000                  .DATA   0,0
015230  000000
015231  000000                  .DATA   0,0
015232  000000
015233  000000                  .DATA   0,0
```

COMMANDS

```
015234  000000
015235  000000                  ,DATA   0,0
015236  000000
    014440      4170            ,USE
```

COMMANDS

```
                    4180            .EJECT
                    4190    *
                    4200    *
                    4210    *       BREAK
                    4220    *
    014440          4230    BRE     ...;
014440  113120      4240            JMS     EXP             GET THE LOCATION
014441  055202      4250            DAC     T3              SAVE IT
014442  114422      4260            JMS     BLOOK           IS IT ALREADY THERE
014443  614063      4270            JMP     COMX            YES -- DO NOTHING
014444  750000      4280            CLA
014445  114422      4290            JMS     BLOOK           LOOK FOR A HOLE
014446  614465      4300            JMP     BR1
    014447          4310            EMESS   <TABLE FULL>,10.
                                    ,CRSM   SAVE,ON
    014447                          MESS    <TABLE FULL WORD #>,10.,-7
014447  700312                      KRB
    014450                          MESSR   <TABLE FULL WORD #>,10.,-7
                                    ,CRSM   SAVE,ON
014450  777755                      LAW     -10.,-7-2
014451  115561                      JMS     TSSIXOT
                                    ,PMC    RESTORE
                                    ,CRSM   RESTORE
                                    COUNT
    014461
014461  215367                      LAC     TSCOUNT
    014462                          OCTZ
014462  744002                      STL
014463  115502                      JMS     TSOCTOT
                                    ,CRSM   RESTORE
014464  614063      4320            JMP     COMX            EXIT
014465  215202      4330    BR1     LAC     T3              GET THE LOCATION
014466  400010      4340            XCT     INDEX           SAVE IT
014467  614063      4350            JMP     COMX            EXIT
                    4360    *
                    4370    *       UNBREAK
                    4380    *
    014470          4390    UNB     ...;
014470  113120      4400            JMS     EXP             GET THE EXPRESSION
014471  114422      4410            JMS     BLOOK           LOOK FOR IT
014472  614510      4420            JMP     UN1             FOUND IT
    014473          4430            EMESS   <NO BREAK>,8.
                                    ,CRSM   SAVE,ON
    014473                          MESS    <NO BREAK WORD #>,8.,-7
014473  700312                      KRB
    014474                          MESSR   <NO BREAK WORD #>,8.,-7
                                    ,CRSM   SAVE,ON
014474  777757                      LAW     -8.,-7-2
014475  115561                      JMS     TSSIXOT
                                    ,PMC    RESTORE
                                    ,CRSM   RESTORE
                                    COUNT
    014504
014504  215367                      LAC     TSCOUNT
```

COMMANDS

```
    014505                          OCTZ
014505  744002                      STL
014506  115502                      JMS     TSOCTOT
                                    .CRSM   RESTORE
014507  614063  4440                JMP     COMX            EXIT
014510  750000  4450      UN1       CLA
014511  400010  4460                XCT     INDEX           CLEAR BREAK POINT
014512  614063  4470                JMP     COMX            EXIT
                4480      *
                4490      *         RSAVE "REGISTER SAVE AND RESTORE ROUTINE
                4500      *
014513  000000  4510      RSAVE     0
014514  055201  4520                DAC     T2              SAVE
014515  777770  4530                LAW     -10             GET COUNT
014516  055202  4540                DAC     T3              SAVE IT
    014517      4550      RS1       ...
014517  235200  4560                LAC     T1,X            GET WORD
014520  075201  4570                DAC     T2,X
014521  455200  4580                ISZ     T1
014522  455201  4590                ISZ     T2
    014523      4600                LOOP    T3,RS1
014523  455202                      ISZ     T3
014524  614517                      JMP     RS1
014525  634513  4610                RET     RSAVE
                4620      *
                4630      *         ENTRY FROM INTERRUPT
                4640      *
014526  000000  4650      INT0      0
014527  054321  4660                DAC     AC              SAVE ACCUMULATOR
014530  214526  4670                LAC     INT0            GET THE PC
014531  054327  4680                DAC     PC              SAVE AS PC
014532  154616  4690                DZM     WF              CLEAR FLAG
014533  641002  4700      INT1      LACQ                    GET THE MQ
014534  054324  4710                DAC     MQ
014535  750010  4720                CLAIRAL                 GET THE LINK
014536  054332  4730                DAC     LK              SAVE IT
014537  641001  4740                LACS                    GET THE SC
014540  054335  4750                DAC     SC              SAVE IT
014541  154363  4760                DZM     TIME            WAIT FOR SLOW INTERRUPTS
014542  454363  4770                ISZ     TIME
014543  614542  4780                JMP     .-1
014544  700314  4790                IORS                    GET THE STATUS
014545  054340  4800                DAC     STS             SAVE IT
014546  700002  4810                IOF                     TURN PI OFF
                4820      *         RPL "GET THE API STATUS
                4830      *         DAC API "SAVE IT
014547  700416  4840                TLS*10                  SEND
                4850      *         LAC (700010) "SET LEVEL 4
                4860      *         ISA
014550  700312  4870                KRB
014551  760010  4880                LAW     10              POINT TO X REGISTERS
014552  055200  4890                DAC     T1              SAVE IT
```

COMMANDS

```
014553  774724  4900            LAW     REGS            POINT TO SAVE AREA
014554  114513  4910            JMS     RSAVE           SAVE THEM
                4920    *
                4930    *       REMOVE BREAKPOINT CALS
                4940    *
014555  775216  4950            LAW     BTBL-1          POINT TO TABLE
014556  040014  4960            DAC     CMDX            SAVE POINTER
014557  777770  4970            LAW     -10             NUMBER OF BREAKPOINTS
014560  055200  4980            DAC     T1              SAVE IT
014561  200000  4990            LAC     0               PRESERVE LOCATION ZERO
014562  055202  5000            DAC     T3
014563  220014  5010    INT2    LAC     CMDX,X          GET A POINTER
014564  055201  5020            DAC     T2              SAVE IT
014565  220014  5030            LAC     CMDX,X          GET THE INSTRUCTION
014566  075201  5040            DAC     T2,X            RESTORE IT
    014567      5050            LOOP    T1,INT2         LOOP ON COUNTER
014567  455200                  ISZ     T1
014570  614563                  JMP     INT2
014571  215202  5060            LAC     T3
014572  040000  5070            DAC     0
    014573      5080            MESS    <BREAK - >8.
014573  700312                  KRB
    014574                      MESSH   <BREAK - >,8.
                                ,CRSM   SAVE,ON
014574  777766                  LAW     -8,-2
014575  115561                  JMS     TSSIXOT
                                ,PMC    RESTORE
                                ,CRSM   RESTORE
014602  777777  5090            LAW     -1              PRINT PROPPER PC
014603  354327  5100            TAD     PC
014604  055200  5110            DAC     T1              SAVE
014605  214616  5120            LAC     WF              CHECK FOR WAT
014606  740200  5130            SZA
014607  614612  5140            JMP     INT3            YES -- DON'T CHANGE INSTRUCTION
014610  235200  5150            LAC     T1,X            GET INSTRUCTION
014611  054711  5160            DAC     UINST           SAVE IT
014612  215200  5170    INT3    LAC     T1              RESTORE PC
014613  515125  5180            AND     (ADRSS)         TRIM
014614  113377  5190            JMS     DUMA            DUMP
014615  614007  5200            JMP     IDLE            AND LOOP IDLEY
014616  000000  5210    WF      0
```

COMMANDS

```
                    5220            .EJECT
                    5230    *
                    5240    *       TRANSFER
                    5250    *
014617  113120      5260    TRA     JMS     EXP             GET THE EXPRESSION
014620  515125      5270            AND     (ADRSS)         TRIM
014621  355153      5280            TAD     (JMP)           MAKE A JMP X INSTRUCTION
014622  054711      5290            DAC     UINST           SAVE IT
                    5300    *
                    5310    *       CONTINUE
                    5320    *
    014623          5330    CON     CRLF
014623  115673                      JMS     TSCRLF
014624  215154      5340            LAC     (BTBL-1)        GET INITIAL POINTER
014625  040014      5350            DAC     CMDX            SAVE IT
014626  777770      5360            LAW     -10             GET COUNT
014627  055200      5370            DAC     T1              SAVE IT
014630  220014      5380    CON1    LAC     CMDX,X          GET A POINTER
014631  741200      5390            SNA                     CHECK FOR SOMETHING
014632  614641      5400            JMP     CON2
014633  055201      5410            DAC     T2              SAVE IT
014634  235201      5420            LAC     T2,X            GET THE WORD
014635  060014      5430            DAC     CMDX,X          SAVE IT
014636  215155      5440            LAC     (JMS INTO
014637  075201      5450            DAC     T2,X            MAKE A CAL
014640  741000      5460            SKP
014641  440014      5470    CON2    ISZ     CMDX            SKIP POINTER
    014642          5480            LOOP    T1,CON1         LOOP COUNTER
014642  455200                      ISZ     T1
014643  614630                      JMP     CON1
014644  774724      5490            LAW     REGS            POINTER  TO X REGISTER 1
014645  055200      5500            DAC     T1              SAVE IT
014646  760010      5510            LAW     10              POINT TO REGISTERS
014647  114513      5520            JMS     RSAVE           RESTORE REGISTERS
014650  214335      5530            LAC     SC              GET THE STEP COUNTER
014651  255112      5540            XOR     (77)            COMPLEMENT
014652  355156      5550            TAD     (640402)        FORM RANDOM EAE
014653  515157      5560            AND     (640477
014654  054655      5570            DAC     .+1             SAVE IT
014655  740040      5580            XX
014656  214324      5590            LAC     MQ              RESTORE MQ
014657  652000      5600            LMQ
014660  214332      5610            LAC     LK              GET THE LINK
014661  740020      5620            RAR                     RESTORE IT
014662  700401      5630            TSF                     WAIT FOR LAST CHARACTER TO PRINT
014663  614662      5640            JMP     .-1
014664  214340      5650            LAC     STS             GET THE STATUS
014665  515130      5660            AND     (20000)         FIND TTY FLAG
014666  741200      5670            SNA                     SKIP IF SO
014667  700402      5680            TCF                     CLEAR FLAG
014670  703304      5690            DBK                     DEBREAK API
014671  214340      5700            LAC     STS             GET THE STATUS REGISTER
```

COMMANDS

```
014672  741100  5710            SPA                     CHECK FOR PI
014673  700042  5720            ION
                5730    *       LAC API "GET THE REGISTER
                5740    *       AND (400000) "JUST THE ENABLE BIT
                5750    *       ISA
014674  214711  5760            LAC     UINST           SPECIAL CHECK FOR JMS
014675  515160  5770            AND     (740000)
014676  555161  5780            SAD     (JMS)
014677  614715  5790            JMP     CON7            OOPS
014700  214711  5800            LAC     UINST           GET THE INSTRUCTION AGAIN
014701  515162  5810            AND     (740700         CHECK FOR WEIRD EAE
014702  555163  5820            SAD     (640100)        MUL
014703  614722  5830            JMP     EA
014704  555164  5840            SAD     (640300         DIV
014705  614722  5850            JMP     EA
014706  215165  5860            LAC     (SKP            SET UP INSTRUCTION
014707  054712  5870    CON6    DAC     UINST+1         SAVE IT
014710  214321  5880            LAC     AC              RESTORE THE ACC
014711  740040  5890    UINST   XX
014712  741000  5900            SKP
014713  454327  5910            ISZ     PC              ESTORE THE PC
014714  634327  5920            RET     PC
014715  214327  5930    CON7    LAC     PC              GET THE PC
014716  074711  5940            DAC     UINST,X         SAVE IT
014717  454711  5950            ISZ     UINST           INCREMENT INSTRUCTION
014720  214321  5960            LAC     AC              RESTORE THE AC
014721  634711  5970            RET     UINST           RETURN
014722  234327  5980    EA      LAC     PC,X            GET MPY OR DIV CONSTANT
014723  614707  5990            JMP     CON6            GO TO IT
                6000    *
014724          6010    REGS    .BLOCK  10
```

COMMANDS

```
                    6020            .EJECT
                    6030
    014734          6040    LOAD    WORD                    GET THE DEVICE NAME
014734  115526                      JMS     TSSIXIN
014735  614745      6050            JMP     DLOAD           VACUOUS ASSUMES LOAD FROM THE DISK
014736  555166      6060            SAD     (PPT)
014737  614751      6070            JMP     PPR
014740  555167      6080            SAD     (PTR)
014741  614751      6090            JMP     PPR
014742  555170      6100            SAD     (DK0)
014743  741000      6110            SKP
    014744          6120            FORMAT                  FORMAT ERROR
014744  614045                      JMP     FORMAT
014745  215116      6130    DLOAD   LAC     (DSREAD)        LOAD THE READ COMMAND
014746  115707      6140            JMS     DSDO            DO THE DISK READ
014747  774236      6150            LAW     SAV4            POINTER TO THE DISK READ PARAMETERS
014750  614063      6160            JMP     COMX            NEXT COMMAND??
                    6170    *
                    6180    *       LOAD PAPER TAPE
                    6190    *
014751  154364      6200    PPR     DZM     CKSUM           CLEAR CHECKSUM
014752  114343      6210            JMS     GETW            GET A WORD
    014753          6220            FORMAT
014753  614045                      JMP     FORMAT
014754  515160      6230            AND     (740000)        CHECK FOR LOADER
014755  555171      6240            SAD     (DAC 0)
014756  614774      6250            JMP     PPR4            GO USE IT
014757  700104      6260            RSA
014760  700101      6270            RSF
014761  614760      6280            JMP     .-1
014762  700112      6290            RRB
014763  740200      6300            SZA
014764  614757      6310            JMP     .-5
014765  614751      6320            JMP     PPR             LOOP
014766  114343      6330    PPR1    JMS     GETW            GET A WORD
    014767          6340            FORMAT                  FORMAT ERROR
014767  614045                      JMP     FORMAT
014770  515160      6350            AND     (740000)        CHECK FOR DAC
014771  555171      6360            SAD     (DAC 0)         CHECK FOR DAC
014772  741000      6370            SKP
014773  614063      6380            JMP     COMX
014774  214363      6390    PPR4    LAC     WT              GET WORD BACK
014775  053671      6400            DAC     LOC             SAVE IT
014776  114343      6410            JMS     GETW            GET LENGTH
    014777          6420            FORMAT
014777  614045                      JMP     FORMAT
    015000          6430            NEG                     FORM COUNT
015000  740001                      CMA
015001  355100                      TAD     (1)
015002  053672      6440            DAC     N
015003  114343      6450            JMS     GETW            TRY FOR CHECKSUM
    015004          6460            FORMAT
```

COMMANDS

```
015004  614045                  JMP     FORMAT
015005  214364  6470            LAC     CKSUM           CHECK CHECKSUM
015006  740200  6480            SZA
    015007      6490            FORMAT                  NOPE
015007  614045                  JMP     FORMAT
015010  114343  6500    PPRL    JMS     GETW            GET A WORD
    015011      6510            FORMAT                  FORMAT ERROR
015011  614045                  JMP     FORMAT
015012  073671  6520            DAC     LOC,X           SAVE IT
015013  453671  6530            ISZ     LOC             INDEX LOCATION
    015014      6540            LOOP    N,PPRL          LOOP
015014  453672                  ISZ     N
015015  615010                  JMP     PPRL
015016  614766  6550            JMP     PPR1            GET ANOTHER BLOCK
015017  000000  6560    TPR     .DATA   0,T$WORDB,2
015020  015237
015021  000002
015022  000000  6570    TPR1    .DATA   0,30,2          SPECIAL READ PARAMETER BLOCK
015023  000030
015024  000002
```

COMMANDS

```
                    6580            .EJECT
                    6590    *
                    6600    *       MASK
                    6610    *
015025  114125      6620    MAS     JMS     GWORD           GET THE MASK
015026  055072      6630            DAC     MASK            SAVE IT
015027  614063      6640            JMP     COMX            RETURN
015030  114125      6650    SEA     JMS     GWORD           GET SEARCH PATTERN
015031  055073      6660            DAC     ST              SAVE IT
    015032          6670            DELIM                   GET THE DELIMITER
015032  215366                      LAC     TSDLMTR
015033  555137      6680            SAD     (CR)
015034  615044      6690            JMP     SEA3            USE DEFAULT CONDITIONS WHEN NO ARGUMENT
015035  555140      6700            SAD     (NUMSGN)
015036  615044      6710            JMP     SEA3            USE DEFAULT CONDITIONS WHEN NO ARGUMENT
015037  113120      6720            JMS     EXP             2GET STARTING LOC
015040  053671      6730            DAC     LOC             SAVE
015041  355172      6740            TAD     (-BOUNDARY)
015042  740100      6750            SMA                     SKIP IF ATTEMPTING TO SEARCH PROTECTED MEMORY
015043  615050      6760            JMP     SEA1            ELSE CONTINUE
015044  762000      6770    SEA3    LAW     BOUNDARY        START SEARCH AT THE BOUNDARY
015045  053671      6780            DAC     LOC
015046  770000      6790            LAW     BOUNDARY-BASE   DEFAULT LENGTH IS ALL USER CORE EXCEPT DDT
015047  615053      6800            JMP     SEA2
015050  113120      6810    SEA1    JMS     EXP             GET ENDING LOC
015051  740001      6820            CMA                     NEGATE
015052  353671      6830            TAD     LOC             FORM LENGTH
015053  055071      6840    SEA2    DAC     N1              SAVE
015054  233671      6850    SEAL    LAC     LOC,X           CHECK A LOC
015055  255073      6860            XOR     ST              CHECK
015056  515072      6870            AND     MASK            CHECK IT
015057  740200      6880            SZA                     CHECK FOR MASK
015060  615065      6890            JMP     SEAX            DONE
015061  777777      6900            LAW     -1              SET COUNT
015062  053672      6910            DAC     N               SAVE IT
015063  113554      6920            JMS     DUMP            DUMP
015064  741000      6930            SKP                     SKIP ISZ
015065  453671      6940    SEAX    ISZ     LOC             BUMP LOC
    015066          6950            LOOP    N1,SEAL         LOOP
015066  455071                      ISZ     N1
015067  615054                      JMP     SEAL
015070  614063      6960            JMP     COMX            EXIT
015071  000000      6970    N1      0
015072  777777      6980    MASK    777777                  INITIALIZED TO -1
015073  000000      6990    ST      0
```

SYMBOL TABLE

```
                    7000            ,STITL  SYMBOL TABLE
                    7010    *
                    7020    *       SYSTEM SYMBOLS
                    7030    *
        000021      7040    SYMN    .EQU    21              NUMBER OF SYMBOLS
        015074      7050    HERE    ...
        011674      7060            .LOC    -4+SYMN+BASE
                    7070    DEF     .DEFIN  (OCTAL          SYMBOL,VALUE
                    7080            .DATA   0,#1,#2
                    7090            .ENDM   DEF
        011674      7100    SYMST   ...
        011674      7110            DEF     0,0
011674  000000                      .DATA   0,0,0
011675  000000
011676  000000
        011677      7120            DEF     434154,0        CAL
011677  000000                      .DATA   0,434154,0
011700  434154
011701  000000
        011702      7130            DEF     444143,40000    DAC
011702  000000                      .DATA   0,444143,40000
011703  444143
011704  040000
        011705      7140            DEF     525563,100000   JMS
011705  000000                      .DATA   0,525563,100000
011706  525563
011707  100000
        011710      7150            DEF     447255,140000   DZM
011710  000000                      .DATA   0,447255,140000
011711  447255
011712  140000
        011713      7160            DEF     544143,200000   LAC
011713  000000                      .DATA   0,544143,200000
011714  544143
011715  200000
        011716      7170            DEF     705762,240000   XOR
011716  000000                      .DATA   0,705762,240000
011717  705762
011720  240000
        011721      7180            DEF     414444,300000   ADD
011721  000000                      .DATA   0,414444,300000
011722  414444
011723  300000
        011724      7190            DEF     644144,340000   TAD
011724  000000                      .DATA   0,644144,340000
011725  644144
011726  340000
        011727      7200            DEF     704364,400000   XCT
011727  000000                      .DATA   0,704364,400000
011730  704364
011731  400000
        011732      7210            DEF     516372,440000   ISZ
```

SYMBOL TABLE

```
011732  000000                  ,DATA   0,516372,440000
011733  516372
011734  440000
    011735      7220            DEF     415644,500000   AND
011735  000000                  ,DATA   0,415644,500000
011736  415644
011737  500000
    011740      7230            DEF     634144,540000   SAD
011740  000000                  ,DATA   0,634144,540000
011741  634144
011742  540000
    011743      7240            DEF     525560,600000   JMP
011743  000000                  ,DATA   0,525560,600000
011744  525560
011745  600000
    011746      7250            DEF     454145,640000   EAE
011746  000000                  ,DATA   0,454145,640000
011747  454145
011750  640000
    011751      7260            DEF     515764,700000   IOT
011751  000000                  ,DATA   0,515764,700000
011752  515764
011753  700000
    011754      7270            DEF     576062,740000   OPT
011754  000000                  ,DATA   0,576062,740000
011755  576062
011756  740000
    015074      7280            ,LOC    HERE
015074  613177  7290            ,LIT
015075  000240
015076  000256
015077  000044
015100  000001
015101  777777
015102  000253
015103  000255
015104  000252
015105  000257
015106  000241
015107  000334
015110  000246
015111  620000
015112  000077
015113  000003
015114  640500
015115  400000
015116  000002
015117  000040
015120  000053
015121  000004
015122  000055
015123  775207
```

SYMBOL TABLE

```
015124  000060
015125  017777
015126  000007
015127  000072
015130  020000
015131  777763
015132  013711
015133  000054
015134  000130
015135  613720
015136  000273
015137  000215
015140  000243
015141  000254
015142  757777
015143  613635
015144  613566
015145  113514
015146  113621
015147  770000
015150  700000
015151  014723
015152  055216
015153  600000
015154  015216
015155  114526
015156  640402
015157  640477
015160  740000
015161  100000
015162  740700
015163  640100
015164  640300
015165  741000
015166  606064
015167  606462
015170  445320
015171  040000
015172  776000
015173  000000
015174  000000
    015237      7300            .USE    STOR
    000001      7310    DEBUG   .EQU    1               TURN ON THE TTY LISTING
                7320            .INSRT  :DLIBRARY:PDP9LIB:TTYNON
                100             .INE    $DEBUG,1
                120             .IFE    $DEBUG,1
```

MTSS-PDP9 NON-INTERRUPTING TELETYPE HANDLER

```
130             ,STITL  MTSS-PDP9 NON-INTERRUPTING TELETYPE HANDLER
140             ,HEAD   T
150     *
160     *
170     *       PROGRAMMED BY ROBERT W. BLEAN
180     *
190     *
200     *       LAST REVISED 24 MARCH 1972
210     *
220     *
230     *       THIS HANDLER PERMITS NON-INTERRUPT DRIVEN INPUT FROM AND OUTPUT
240     *       TO THE CONSOLE TELETYPE ON THE PDP-9 COMPUTER.
250     *
260     *       THIS HANDLER ALTERS THE  AC, AND MQ. IT DOES NOT ALTER ANY CORE
270     *       MEMORY OUTSIDE OF ITSELF, IN PARTICULAR IT DOES NOT ALTER ANY AUTO-INDEX REGISTER.
280     *
290     *       DATA FORMATS:
300     *
310     *            1) OCTAL
320     *
330     *            2) SIXBIT -- SIXBIT IS 8-BIT ASCII MINUS 240. THIS MAPS THE PRINTING
340     *               CHARACTERS ONTO THE SET 0-77. ASCII VALUE 333 (|) IS USED FOR
350     *               CARRIAGE RETURN AND 335 (|) IS USED FOR LINEFEED. NOTE THAT NEITHER
360     *               333, 335, NOR ANY CONTROL CHARACTERS CAN BE RECOGNIZED IN SIXBIT.
370     *
380     *            3) ASCII -- ONE ASCII CHARACTER IS STORED PER WORD. LINE INPUT
390     *               IS STORED IN THIS FORMAT, SINCE THERE IS ONLY ONE LINE-BUFFER
400     *               THE EXTRA BUFFER LENGTH WASTES LESS SPACE THAN WOULD THE HANDLING
410     *               ROUTINES NECESSARY FOR OTHER FORMS OF PACKING CHHRACTERS.
```

```
                T                       (MTSS TELETYPE HANDLER) STORAGE AREA

                        420             ,STITL  (MTSS TELETYPE HANDLER) STORAGE AREA
                        430             ,IFE    PURCOD,1
                        450
                        460
     015237             470     WORDB   ,BLOCK  2               ROOM TO ACCUMULATE TWO VALID WORDS
     000120             480     STD     ,EQU    80.             STANDARD IS AN 80-CHARACTER LINE BUFFER
     015241             490     BUFFR   ,BLOCK  STD
                        500     *
                        510     *
                        520     *       VARIABLES
                        530     *
015361  015360          540     BEND    ,-1                     END OF THE CHARACTER BUFFER
015362  000000          550     BPTR    ,DSA                    POINTER TO CURRENTLY ACTIVE WORD IN LINE BUFFER
015363  000000          560     T1      ,DSA                    TEMPORARY VARIABLE
015364  000000          570     T2      ,DSA                    TEMPORARY VARIABLE
015365  000000          580     CHAR    ,DSA                    STORES LATEST CHARACTER FROM FGET
015366  000000          590     DLMTR   ,DSA                    STORES LATEST DELIMITER THROUGH CHRID
015367  000000          600     COUNT   ,DSA
                        610             ,IFE    PURCOD,1
```

```
T                                   (MTSS TELETYPE HANDLER) LINE BUFFER INPUT

                    630             ,STITL  (MTSS TELETYPE HANDLER) LINE BUFFER INPUT
                    640
                    650
                    660     *
                    670     *       THE PROGRAM IS PROTECTED AGAINST OVERFLOW OR UNDERFLOW OF THE LINE
                    680     *       BUFFER. UNDERFLOW (EXCESS DELETIONS) IS IGNORED, AND OVERFLOW CHARACTERS
                    690     *       ARE LOST, EXCEPT FOR THE LAST CHARACTER TYPED.
                    700     *
                    710
    015370          720             ENTER   INLIN           SUBROUTINE TO READ IN AND BUFFER A LINE FROM THE TELETYPE
015370  740040              INLIN   XX
015371  700312      730             KRB                     ONCE, ON ENTRANCE, CLEAN UP ANY PRIOR INPUT
015372  215743      740     INL     LAC     (BUFFR-1)       LOAD A POINTER TO START OF THE BUFFER MINUS ONE
015373  055362      750             DAC     BPTR            INITIALIZE THE BUFFER POINTER
015374  155367      760             DZM     COUNT           INITIALIZE THE WORD FETCHED COUNT
015375  155366      770             DZM     DLMTR           INITIALIZE THE LAST DELIMITER STORAGE
015376  700313      780     IN1     KSF!KRB                 GET THE NEXT INPUT CHARACTER
015377  615376      790             JMP     .-1
015400  555744      800             SAD     ($BKARR)
015401  615423      810             JMP     1CHAR           DELETE ONE CHARACTER IF IT WAS A BACKARROW
015402  555745      820             SAD     ($CONTX)
015403  615421      830             JMP     1LINE           DELETE THE ENTIRE LINE IF IT WAS A CONTROL X
015404  652000      840     IN4     LMQ                     SAVE THE CHARACTER
015405  215362      850             LAC     BPTR            LOAD THE CURRENT BUFFER POINTER
015406  555361      860             SAD     BEND            SKIP IF NO OVERFLOW
015407  741000      870             SKP                     AVOID DAMAGE DUE TO OVERFLOW
015410  455362      880             ISZ     BPTR            ADVANCE THE POINTER -- IT IS STILL WITHIN THE BUFFER
015411  641002      890             LACQ                    RELOAD THE CHARACTER
015412  075362      900             DAC     BPTR,X          AND PUT IT IN THE BUFFER
015413  555746      910             SAD     ($CR)
015414  741000      920             SKP                     EXIT WHEN A CARRIAGE RETURN IS FOUND
015415  615376      930             JMP     IN1             ELSE GET THE NEXT CHARACTER
015416  775240      940             LAW     BUFFR-1
015417  055362      950             DAC     BPTR            RESET THE BUFFER POINTER AT THE END OF THE LINE
015420  635370      960             JMP     INLIN,X         AND RETURN TO THE CALLER
                    970
015421  115673      980     1LINE   JMS     CRLF            PRINT THE RESPONSE TO A LINE-DELETE
015422  615372      990             JMP     INL             REREAD THE LINE
015423  215362     1000     1CHAR   LAC     BPTR            LOAD THE BUFFER POINTER
015424  555372     1010             SAD     INL             SKIP IF NO UNDERFLOW
015425  615376     1020             JMP     IN1             ELSE IGNORE THE COMMAND
015426  355747     1030             TAD     (-1)            DECREMENT THE BUFFER POINTER
015427  055362     1040             DAC     BPTR            AND SAVE IT
015430  615376     1050             JMP     IN1             GET THE NEXT CHARACTER
```

T (MTSS TELETYPE HANDLER) OCTAL WORD INPUT/OUTPUT

```
                1060            .STITL  (MTSS TELETYPE HANDLER) OCTAL WORD INPUT/OUTPUT
                1070
                1080    *
                1090    *       OPERATION RETURN L    AC      MQ    MEANING
                1100    *       --------- ------ - ------ ------ ------------------------------
                1110    *       INPUT       +1   0   X      X     FORMAT ERROR DISCOVERED
                1120    *                   +1   1 DELIM    X     FIRST NON-BLANK CHARACTER IS A DELIMITER
                1130    *                   +2   1 OCTAL  DELIM   SUCCESSFUL READ OF AN OCTAL NUMBER
                1140    *       OUTPUT      +1   X   X      X     SUCCESSFUL WRITE OF AN OCTAL NUMBER
                1150    *
                1160
    015431      1170            ENTER   NUMIN
015431  740040          NUMIN   XX
015432  155364  1180            DZM     T2              INITIALIZE THE DECIMAL-DIGIT-RECEIVED FLAG
015433  115614  1190            JMS     INTIN           INITIALIZE THE INPUT STRING, ETC
015434  635431  1200            JMP     NUMIN,X         RETURN +1 FOR DELIMITER IS FIRST NON-BLANK CHARACTER
015435  115607  1210    NUM20   JMS     FGET            GET THE NEXT CHARACTER
015436  115633  1220            JMS     CHRID           IDENTIFY IT
015437  615461  1230            JMP     NUM26           IT IS A DELIMITER, SO EXIT
015440  635431  1240            JMP     NUMIN,X         IT IS A LETTER, SO EXIT +1 FOR A FORMAT ERROR
015441  741400  1250            SZL                     SKIP IF THE CHARACTER IS AN OCTAL DIGIT
015442  455364  1260            ISZ     T2              ELSE BE SURE THE DECIMAL-DIGIT-RECEIVED FLAG IS SET
015443  515750  1270            AND     (17)            RETAIN JUST THE DIGIT
015444  055363  1280            DAC     T1              AND SAVE IT FOR DECIMAL ACCUMULATION
                1290
015445  640503  1300            LRS     3               SAVE THE "OCTAL DIGIT"
015446  215237  1310            LAC     WORDB           LOAD THE PREVIOUSLY GATHERED "OCTAL NUMBER"
015447  640603  1320            LLS     3               CONCATENATE THE "OCTAL DIGITS"
015450  055237  1330            DAC     WORDB           AND SAVE THE RESULT
                1340
015451  215240  1350            LAC     WORDB+1         LOAD THE PREVIOUSLY GATHERED "DECIMAL NUMBER"
015452  744000  1360            CLL                     SET THE LINK FOR THE MULTIPLY
015453  653122  1370            MUL                     MULTIPLY THE PREVIOUS "DECIMAL VALUE"
015454  000012  1380            10.                     BY 10 FOR DECIMAL
015455  641002  1390            LACQ                    LOAD THE RESULT
015456  355363  1400            TAD     T1              ADD THE CURRENT "DECIMAL DIGIT"
015457  055240  1410            DAC     WORDB+1         AND SAVE THE TOTAL "DECIMAL NUMBER"
                1420
015460  615435  1430            JMP     NUM20           LOOP
                1440
                1450
015461  555751  1460    NUM26   SAD     (SPOINT)        CHECK FOR A PERIOD
015462  615470  1470            JMP     NUM27           IF SO, PICK UP THE DECIMAL VALUE
015463  215364  1480            LAC     T2              ELSE LOAD THE DECIMAL-DIGITS-RECEIVED FLAG
015464  744200  1490            SZA!CLL                 AND SKIP IF THERE WERE NONE
015465  635431  1500            JMP     NUMIN,X         RETURN +1, LK=0 FOR A FORMAT ERROR; DECIMAL DIGITS, BUT NO PERIOD
015466  215237  1510            LAC     WORDB           LOAD THE OCTAL VALUE
015467  615477  1520            JMP     NUM29
015470  115607  1530    NUM27   JMS     FGET            GET THE NEXT CHARACTER
015471  115633  1540            JMS     CHRID           AND IDENTIFY IT
015472  615476  1550            JMP     NUM28           A DELIMITER IS LEGAL, SO EXIT
015473  635431  1560            JMP     NUMIN,X         A LETTER -- EXIT +1 FOR A FORMAT ERROR
```

T (MTSS TELETYPE HANDLER) OCTAL WORD INPUT/OUTPUT

```
015474  744000  1570            CLL                     A NUMBER -- CLEAR THE LINK FOR A FORMAT ERROR
015475  635431  1580            JMP     NUMIN,X         AND EXIT +1
015476  215240  1590    NUM28   LAC     WORDB+1         LOAD THE DECIMAL VALUE
015477  055237  1600    NUM29   DAC     WORDB           SAVE THE CORRECT VALUE
015500  455431  1610            ISZ     NUMIN           BUMP TO A RETURN +2 FOR SUCCESSFUL
015501  635431  1620            JMP     NUMIN,X
                1630
                1640
                1650
    015502      1660            ENTER   OCTOT
015502  740040          OCTOT   XX
015503  652000  1670    OCT42   LMQ                     SET THE VALUE TO BE OUTPUT
015504  741400  1680            SZL                     SKIP IF NO LEADING ZEROES ARE TO BE SUPPRESSED
015505  750201  1690            SZA!CLC                 SET A FLAG TO PRINT ONE CHARACTER, ANYWAY, IF THE AC IS ZERO
015506  777772  1700            LAW     -6              ELSE SET THE COUNT FOR THE STANDARD SIX CHARACTERS
015507  055363  1710            DAC     T1              SET THE NUMBER OF CHARACTERS TO BE OUTPUT
015510  641002  1720            LACQ                    RELOAD THE USER'S VALUE
015511  741200  1730            SNA                     SKIP FOR A NON-ZERO VALUE
015512  744000  1740            CLL                     ELSE FORCE A SINGLE ZERO TO PRINT
015513  641603  1750    OCT44   LLSC    3.              GET THE NEXT OCTAL DIGIT
015514  740200  1760            SZA                     IF IT IS ZERO, DON'T CHANGE PRINT-SUPPRESSION STATE
015515  744000  1770            CLL                     ELSE CLEAR THE PRINT INHIBIT AT THE FIRST NON-ZERO FOUND
015516  355752  1780            TAD     (260)           MAKE ASCII IN ANY CASE
015517  740400  1790            SNL                     BUT SKIP IF PRINT IS INHIBITED
015520  115665  1800            JMS     TTYOT           ELSE PRINT THE DIGIT
015521  455363  1810            ISZ     T1              DONE???
015522  615513  1820            JMP     OCT44           NO -- LOOP
015523  700401  1830            TSF
015524  615523  1840            JMP     .-1             WAIT FOR THE TELETYPE TO SETTLE
015525  635502  1850            JMP     OCTOT,X         YES -- EXIT
```

T (MTSS TELETYPE HANDLER) SIXBIT WORD INPUT & SIXBIT BUFFER OUTPUT

```
                1860            .STITL  (MTSS TELETYPE HANDLER) SIXBIT WORD INPUT & SIXBIT BUFFER OUTPUT
                1870
                1880
                1890    *
                1900    *       OPERATION RETURN L   AC      MQ    MEANING
                1910    *       --------- ------ - ------ ------- ------------------------------
                1920    *       INPUT       +1   1 DELIM    X     FIRST NON-BLANK CHARACTER IS A DELIMITER
                1930    *                   +2   1 SIXBIT DELIM   SUCCESSFUL READ OF A SIXBIT WORD
                1940    *       OUTPUT      +1   X   X      X     SUCCESSFUL WRITE OF A SIXBIT BUFFER
                1950    *
                1960
    015526      1970            ENTER   SIXIN
015526  740040          SIXIN   XX
015527  775237  1980            LAW     WORDB
015530  055363  1990            DAC     T1              INITIALIZE THE SIXBIT BUFFER POINTER
015531  115614  2000            JMS     INTIN           INITIALIZE THE INPUT
015532  635526  2010            JMP     SIXIN,X         RETURN +1 FOR DELIMITER IS FIRST NON-BLANK CHARACTER
015533  455526  2020            ISZ     SIXIN           ELSE BUMP TO A GOOD RETURN
015534  115552  2030    SIX2    JMS     SIX5            GET THE FIRST GOOD CHARACTER
015535  660714  2040            ALSS    12.             AND PUT IT IN THE FIRST CHARACTER POSITION
015536  075363  2050            DAC     T1,X            AND SAVE IT
015537  115552  2060            JMS     SIX5            GET THE SECOND CHARACTER
015540  660706  2070            ALSS    6.              PUT IT IN THE SECOND CHARACTER POSITION
015541  275363  2080            XOR     T1,X            CONCATENATE THE CHARACTERS
015542  075363  2090            DAC     T1,X            AND SAVE THE RESULT
015543  115552  2100            JMS     SIX5            GET THE THIRD CHARACTER
015544  275363  2110            XOR     T1,X            CONCATENATE THE CHARACTERS
015545  075363  2120            DAC     T1,X            AND SAVE THE RESULT
015546  455363  2130            ISZ     T1              BUMP THE STORAGE BUFFER POINTER
015547  615534  2140            JMP     SIX2            LOOP
                2150
015550  215237  2160    SIX9    LAC     WORDB           LOAD THE FIRST SIXBIT WORD
015551  635526  2170            JMP     SIXIN,X         EXIT
                2180
    015552      2190            ENTER   SIX5            SUBROUTINE TO GET THE NEXT CHARACTER, MAKE IT SIXBIT, EXIT IF A DELIMITER
015552  740040          SIX5    XX
015553  115607  2200            JMS     FGET            GET THE NEXT CHARACTER
015554  115633  2210            JMS     CHRID           IDENTIFY IT
015555  615550  2220            JMP     SIX9            EXIT IF IT IS A DELIMITER
015556  740000  2230            NOP                     PERMIT LETTERS
015557  355753  2240            TAD     (-240)          MAKE SIXBIT
015560  635552  2250            JMP     SIX5,X
                2260
                2270
    015561      2280            ENTER   SIXOT
015561  740040          SIXOT   XX
015562  055363  2290            DAC     T1              SET THE NEGATIVE CHARACTER COUNT
015563  235561  2300    SIX24   LAC     SIXOT,X         LOAD THE NEXT WORD OF OUTPUT
015564  652000  2310            LMQ                     SAVE IT FOR PRINTING
015565  455561  2320            ISZ     SIXOT           BUMP THE POINTER
015566  115572  2330            JMS     SIX26           OUTPUT THE FIRST CHARACTER
015567  115572  2340            JMS     SIX26           OUTPUT THE SECOND CHARACTER
```

T (MTSS TELETYPE HANDLER) SIXBIT WORD INPUT & SIXBIT BUFFER OUTPUT

```
015570  115572  2350            JMS     SIX26           OUTPUT THE THIRD CHARACTER
015571  615563  2360            JMP     SIX24           LOOP
                2370
    015572      2380            ENTER   SIX26
015572  740040          SIX26   XX
015573  641606  2390            LLSC    6.              GET THE NEXT SIXBIT CHARACTER
015574  355754  2400            TAD     (240)           MAKE IT ASCII
015575  555755  2410            SAD     (333)           CHECK FOR CARRIAGE RETURN MAPPING
015576  760215  2420            LAW     $CR
015577  555756  2430            SAD     (335)           CHECK FOR LINE FEED MAPPING
015600  760212  2440            LAW     $LF
015601  115665  2450            JMS     TTYOT           PRINT THE CHARACTER
015602  455363  2460            ISZ     T1              ALL CHARACTERS PRINTED?
015603  635572  2470            JMP     SIX26,X         NO -- LOOP
015604  700401  2480            TSF
015605  615604  2490            JMP     .-1             WAIT FOR THE TELETYPE TO SETTLE
015606  635561  2500            JMP     SIXOT,X         YES -- EXIT
                2510    *
                2520    *
```

```
T                                   (MTSS TELETYPE HANDLER) MISCELLANEOUS LINE BUFFER ROUTINES

                2530            .STITL  (MTSS TELETYPE HANDLER) MISCELLANEOUS LINE BUFFER ROUTINES
                2540
                2550
                2560
                2570
    015607      2580            ENTER   FGET            SUBROUTINE TO GET THE FIRST REMAINING CHARACTER FROM THE LINE BUFFER
015607  740040          FGET    XX
015610  455362  2590            ISZ     BPTR            NO -- BUMP THE POINTER
015611  235362  2600            LAC     BPTR,X          LOAD THE NEXT CHARACTER
015612  055365  2610            DAC     CHAR            AND SAVE IT
015613  635607  2620    FGET9   JMP     FGET,X
                2630
    015614      2640            ENTER   INTIN           INITIALIZE INPUT WORD-GETTING
015614  740040          INTIN   XX
015615  455367  2650            ISZ     COUNT           COUNT THE WORD, SUCCESSFUL OR NOT
015616  155237  2660            DZM     WORDB           INITIALIZE THE TWO FIRST WORDS OF THE INPUT BUFFER
015617  155240  2670            DZM     WORDB+1
015620  115607  2680            JMS     FGET            GET THE NEXT CHARACTER

015621  555754  2690            SAD     ($SPACE)        CHECK IT FOR A SPACE
015622  615620  2700            JMP     .-2             THROW AWAY SPACES
015623  115633  2710            JMS     CHRID           IDENTIFY THE NON-SPACE
015624  635614  2720            JMP     INTIN,X         RETURN +1 FOR A DELIMITER
015625  740000  2730            NOP
015626  455614  2740            ISZ     INTIN           ELSE BUMP THE RETURN FOR A NUMBER OR A LETTER
015627  750001  2750            CLC
015630  355362  2760            TAD     BPTR            BACK UP THE POINTER TO POINT TO THE FIRST GOOD CHARACTER
015631  055362  2770            DAC     BPTR
015632  635614  2780            JMP     INTIN,X
```

```
T                       (MTSS TELETYPE HANDLER) MISCELLANEOUS CHARACTER-HANDLING SUBROUTINES

                2790            .STITL  (MTSS TELETYPE HANDLER) MISCELLANEOUS CHARACTER-HANDLING SUBROUTINES
                2800    *
                2810    *
                2820    *       CHRID -- SUBROUTINE TO CLASSIFY EIGHT-BIT ASCII CHARACTERS.
                2830    *       ENTER WITH THE CHARACTER IN THE AC; LEAVE WITH THE EIGHT-BIT CHARACTER
                2840    *       IN AC(0-17) AND THE LINK AS FOLLOWS:
                2850    *
                2860    *       RETURN LINK MEANING
                2870    *       ------ ---- ------------------------------------------------------
                2880    *         *1     1   THE CHARACTER IS A DELIMITER (I.E. NEITHER A DIGIT NOR A LETTER
                2890    *         *2     0   THE CHARACTER IS EITHER AN UPPER CASE OR A LOWER CASE LETTER
                2900    *         *3     0   THE CHARACTER IS AN OCTAL DIGIT
                2910    *         *3     1   THE CHARACTER IS A DECIMAL DIGIT (8 OR 9)
                2920    *
    015633      2930            ENTER   CHRID
015633  740040          CHRID   XX
015634  515757  2940            AND     (377)
015635  055665  2950            DAC     TTYOT           SAVE THE EIGHT-BIT ASCII CHARACTER
015636  355760  2960            TAD     (-260)          AC < 0 FOR DELIMITERS
015637  745102  2970            SPA!STL
015640  615656  2980            JMP     DLMR            CHARACTER IS A DELIMITER
015641  355761  2990            TAD     (-10)           AC < 0 FOR OCTAL DIGITS
015642  745100  3000            SPA!CLL
015643  615661  3010            JMP     DIGIT           CHARACTER IS AN OCTAL DIGIT
015644  355762  3020            TAD     (-2)            AC < 0 FOR DECIMAL DIGITS
015645  745102  3030            SPA!STL
015646  615661  3040            JMP     DIGIT           CHARACTER IS A DECIMAL DIGIT
015647  355763  3050            TAD     (-6)            AC <= 0 FOR DELIMITERS
015650  745302  3060            SNA!SPA!STL
015651  615656  3070            JMP     DLMR            CHARACTER IS A DELIMITER
015652  515764  3080            AND     (777737)        MAP LOWER CASE INTO UPPER CASE
015653  355765  3090            TAD     (-33)           AC < 0 FOR LETTERS -- L=1 FOR LETTERS; L=0 FOR DELIMITERS
015654  741102  3100            SPA!CML
015655  615662  3110            JMP     LETTR           THE CHARACTER IS A LETTER
                3120
015656  215665  3130    DLMR    LAC     TTYOT           LOAD THE DELIMITER
015657  055366  3140            DAC     DLMTR           SAVE IT
015660  635633  3150            JMP     CHRID,X
                3160
015661  455633  3170    DIGIT   ISZ     CHRID
015662  455633  3180    LETTR   ISZ     CHRID
015663  215665  3190            LAC     TTYOT           RELOAD THE CHARACTER
015664  635633  3200            JMP     CHRID,X
                3210
                3220
                3230
    015665      3240            ENTER   TTYOT
015665  740040          TTYOT   XX
015666  700401  3250            TSF
015667  615666  3260            JMP     .-1             WAIT FOR THE TELEPRINTER TO BE FREE
015670  700301  3270            KSF                     KILL-THE-OUTPUT FEATURE
015671  700406  3280            TLS                     PRINT THE CHARACTER IN THE AC
```

T (MTSS TELETYPE HANDLER) MISCELLANEOUS CHARACTER-HANDLING SUBROUTINES

```
015672  635665  3290            JMP     TTYOT,X
                3300
                3310
    015673      3320            ENTER   CRLF
015673  740040          CRLF    XX
015674  760215  3330            LAW     215
015675  115665  3340            JMS     TTYOT
015676  760215  3350            LAW     215
015677  115665  3360            JMS     TTYOT
015700  760212  3370            LAW     212
015701  115665  3380            JMS     TTYOT
015702  700401  3390            TSF
015703  615702  3400            JMP     .-1             WAIT FOR THE TTY TO SETTLE
015704  635673  3410            JMP     CRLF,X
                3420
                3430
                3440            .HEAD                   TURN OFF THE INSERT'S HEAD SYMBOL
                3450            .LIST   ON
                3460            .END
```

SHORT DISK ROUTINE

```
                        7330            ,STITL  SHORT DISK ROUTINE
                        7340            ,HEAD   D
    000036              7350    DSKWC   ,EQU    36
    000037              7360    DSKCA   ,EQU    37
    000002              7370    READ    ,EQU    2
    000004              7380    WRITE   ,EQU    4
015705  000000          7390    CMND    ,DSA
    001777              7400    BLKMSK  ,EQU    1777
015706  000000          7410    PNTR    ,DSA
                        7420
                        7430
    015707              7440    DO      ENTER
015707  740040                          XX
015710  055705          7450            DAC     CMND            SAVE THE DISK COMMAND
015711  235707          7460            LAC     DO,X            GET THE POINTER TO THE PARAMETERS
015712  055706          7470            DAC     PNTR
015713  455707          7480            ISZ     DO              CORRECT THE RETURN
                        7490    *
015714  707074          7500            DLAH+10                 ONLY DISK 0 AVAILABLE
                        7510    *
015715  235706          7520            LAC     PNTR,X          GET THE BLOCK NUMBER
015716  301777          7530            AND     BLKMSK          AND OUT THE DISK NUMBER
015717  660710          7540            ALSS    8.              MULTIPLY BY 400 TO MAKE IT A PHYSICAL DISK ADDRESS
015720  707024          7550            DLAL                    PLACE THE PHYSICAL ADDRESS INTO THE DISK ADDRESS REGISTER
                        7560    *
015721  455706          7570            ISZ     PNTR            MOVE THE POINTER TO THE PARAMETERS
015722  777777          7580            LAW     -1              GET A MINUS 1 TO FORM THE CORE POINTER
015723  375706          7590            TAD     PNTR,X
015724  040037          7600            DAC     DSKCA           PLACE THE CORRECTED POINTER INTO THE DATA CHANNEL CORE ADDRESS
                        7610    *
015725  455706          7620            ISZ     PNTR            MOVE THE POINTER TO THE LAST PARAMETER
015726  777777          7630    NCHEK   LAW     -1              PREPARE TO COMPLEMENT A NUMBER
015727  375706          7640            TAD     PNTR,X
015730  740001          7650            CMA
015731  040036          7660            DAC     DSKWC           PLACE THE TWO'S COMPLEMENT WORD COUNT IN THE DATA CHANNEL WORD COUNT
                        7670    *
015732  215705          7680            LAC     CMND            GET THE COMMAND
015733  707047          7690            DSCF!DSFX!DSCN          ISSUE THE OPERATION
015734  707001          7700            DSSF                    SEE IF THE OPERATION IS DONE
015735  615734          7710            JMP     .-1             NO -- WAIT
                        7720    *
                        7730    *       CHECK THE OPERATION AND EXIT
                        7740    *
015736  707272          7750            DSRS+10                 LOAD THE AC WITH THE STATUS OF THE OPERATION
015737  707242          7760            DSCD                    LOAD THE FLAGS
015740  741100          7770            SPA                     SKIP IF OK
015741  740040          7780    ERROR   HLT                     BAD STATUS
015742  635707          7790            RET     DO              EXIT
                        7800    *
015743  015240          7810            .LIT
015744  000337
015745  000230
```

D SHORT DISK ROUTINE

```
015746  000215
015747  777777
015750  000017
015751  000256
015752  000260
015753  777540
015754  000240
015755  000333
015756  000335
015757  000377
015760  777520
015761  777770
015762  777776
015763  777772
015764  777737
015765  777745
    015770      7820            .USE    END
                7830            .DUP    1,10
015770  613775  7840            JMP     $START
015771  613775                  JMP     $START
015772  613775                  JMP     $START
015773  613775                  JMP     $START
015774  613775                  JMP     $START
015775  613775                  JMP     $START
015776  613775                  JMP     $START
015777  613775                  JMP     $START
    016000      7850            .END    $START

TRANSFER ADDRESS  613775
```

D CROSS REFERENCE TABLE

13317	.000.	2070											
14003	.001.	670											
14012	.002.	720											
14031	.003.	840											
14050	.004.	890											
14452	.005.	4310											
14476	.006.	4430											
14576	.007.	5080											
4453	.DK	770											
4464	.DT	750											
6460	.TP	760											
414263	ABS	850											
14321	AC	3120	4660	5880	5960								
14205	ADD	2130	200										
14221	ADD1	2240	2200										
14213	ADDL	2190	2220										
14224	ADDTB	2300	2130	2370									
6	ADDTL	2370	2150										
17737	ADR$S	430	1520	5180	5270								
14371	ADS	3650	420	3800									
14401	ADSL	3740	3770										
14411	ADSX	3810	3750	3790									
13400	AFORM	2550	2250										
14310	ALT	2990	380										
246	AMPRSN	290	1610										
13212	AND	1120	1610										
12000	BASE	190	2460	6790	7060								
425156	BIN	860											
337	BKARR	240	800										
400	BLKLEN	630											
1777	BLKMSK	7400	7530										
14462	BLOOK	3980	4060	4090	4100	4260	4290	4410					
17760	BOOT	520	530										
2000	BOUNDA	440	2450	2460	6740	6770	6790						
14465	BR1	4330	4300										
14440	BRE	4230	460										
14431	BRL	4050	4080										
334	BSLASH	390	1600										
15217	BTBL	4160	4000	4950	5340								
1	CATBLK	550											
400	CATLEN	560											
17000	CATLOG	540	590										
777716	CATMAX	620											
11	CATX	500											
14364	CKSUM	3530	3460	3470	6200	6470							
14075	CMDN	1020											
14	CMDX	600	760	810	820	1010	4960	5010	5030	5350	5380	5430	5470
272	COLON	370											
14022	COM0	810	830										
14026	COM1	840											
14014	COMA	740	1000	2080									
13775	COME	590	770										

D

CROSS REFERENCE TABLE

254	COMMA	320	1170	1560									
13721	COMTB	140	750	770									
14063	COMX	920	1270	1810	2090	2260	2420	2910	2950	3050	3860	3920	4270	4320
			4350	4440	4470	6160	6380	6640	6960					
14074	COMY	1010	940											
14623	CON	5330	520											
14640	CON1	5380	5480											
14641	CON2	5470	5400											
14707	CON6	5870	5990											
14715	CON7	5930	5790											
230	CONTX	230	820											
435762	COR	880												
17005	CPARAM	590												
215	CR	220	950	6680	910	2420								
436454	CTL	810												
15707	D DO	7440	2400	6140	7460	7480	7790							
15705	D CMND	7390	7450	7680										
15706	D PNTR	7410	7470	7520	7570	7590	7620	7640						
2	D READ	7370	6130											
37	DDSKCA	7360	7600											
36	DDSKWC	7350	7660											
1	DEBUG	7310	100	120	100	120	100	120						
14175	DEC	1990	260											
15211	DECB	3770	3510	3650										
13405	DECF	2640	2340											
13530	DECL	3540	3600											
13540	DECL1	3620	3710											
15741	DERROR	7780												
13237	DIV	1420	1580											
445300	DK.	740												
445320	DKO	780	6100											
777601	DKMAX	650												
14745	DLOAD	6130	6050											
15726	DNCHEK	7630												
284	DOLLAR	280												
446400	DT.	720												
776701	DTMAX	640												
14076	DUM	1110	160											
14107	DUM1	1210	1160											
14137	DUM2	1280	1180											
14114	DUM3	1250	1200	1330										
13377	DUMA	2540	2610	2650	3240	3290	3980	4710	2940	3130	3160	3190	5190	
13554	DUMP	3860	3950	4060	4170	1260	6920							
13564	DUMX	3940	4740	4790										
740000	DVCMSK	600												
4	DWRITE	7380	2390											
14722	EA	5980	5830	5850										
13605	EQLP	4210	3870	4230	2810									
13210	ERA	1070	1600											
15206	EVAL	1790	370	900	1010	1070	1120	1240	1480	1680				
13175	EVD	850	530											
13124	EVL	430	450	1660										

D CROSS REFERENCE TABLE

13134	EVL0	510	480											
13145	EVL1	610	540											
13173	EVX	830	500	600										
13242	EVY	1480	910	1030	1080	1130	1330							
13267	EVZ	1670	1630											
241	EXCLAM	260	1590											
13120	EXP	360	1690	1120	1210	1280	1400	1510	1740	4240	4400	5260	6720	6810
5	FCBLEN	570												
14200	FORM	2050	1900	1950	2000									
14045	FORMAT	890	990	1240	1620	2230	2690	6120	6220	6340	6420	6460	6490	6510
13566	FORMO	4000	3930	4090	1940									
13635	FORMS	4520	4800	1890										
13655	FORMS1	4690	4600											
13657	FORMS2	4710	4680											
13563	FORMSW	3930	2050											
14343	GETW	3340	3430	3490	3500	3740	3780	6210	6330	6410	6450	6500		
476257	GRO	870												
14346	GW1	3370	3420											
14133	GWDO	1450	1580											
14135	GWD1	1510	1440											
14125	GWORD	1390	1470	1660	1820	3030	6620	6650						
4	HDRLEN	580												
14417	HEA	3900	440											
15210	HEAD	1810	760	3910										
15074	HERE	7050	7280											
14007	IDLE	710	2080	800	850	900	960	5200						
13673	INDB	4850	4540	4690	4720									
10	INDEX	490	2140	2190	2210	2240	2550	2640	2660	2670	2750	2890	2900	2930
			3000	3010	4010	4050	4070	4340	4460					
14526	INT0	4650	4670	5440										
14533	INT1	4700												
14563	INT2	5010	5050											
14612	INT3	5170	5140											
14365	LDS	3570	400											
212	LF	210	2440											
14332	LK	3210	4730	5610										
14734	LOAD	6040	540											
13671	LOC	4830	490	3880	3920	4030	4080	4140	4550	4660	4700	1130	1300	1750
			1830	1840	6400	6520	6530	6730	6780	6830	6850	6940		
15025	MAS	6620	560											
15072	MASK	6980	6630	6870										
13201	MIN	950	1560											
255	MINUS	330	1560											
13235	MPY	1370	1570											
14324	MQ	3150	4710	5590										
13672	N	4840	4150	1250	6440	6540	6910							
15071	N1	6970	6840	6950										
13600	NEXT	4130	3940	4020	4160									
13164	NHD	760	690											
13314	NOS	2070	2010											
243	NUMSGN	270	970	6700										
14173	OCT	1940	240											

D CROSS REFERENCE TABLE

13401	OCTF	2590	2550	2320				
13174	OJMP	840	390	860	1650			
13241	ONE	1440	1260	1320				
13674	OPTB	4860	4610					
13204	OR	1000	1590					
13214	OSET	1170	1220	1370	1420			
13227	OST	1280	1210	1230				
13621	P6	4420	4480	4640	2040	2830		
13715	P6OCT	4970	4990					
13711	P6Z	4930	3280	4950	3220	3250		
14155	PAT	1730	180					
14157	PAT0	1760	1850					
14165	PAT1	1820	1780					
14327	PC	3180	4680	5100	5910	5920	5930	5980
13525	PDE1	3510	3460					
13514	PDEC	3430	2640	3740	1990			
13551	PDX	3720	3660					
256	PERIOD	340	350					
13177	PLU	900	380	960	1550			
253	PLUS	310	1550					
256	POINT	350	460	3720	1460			
14751	PPR	6200	6070	6090	6320			
14766	PPR1	6330	6550					
14774	PPR4	6390	6250					
15010	PPRL	6500	6540					
606064	PPT	690	6060					
606460	PTP	710						
606462	PTR	700	6080					
17505	RECOV	470						
14271	REG	2800	360					
14305	REG1	2920	2880					
14263	REG2	2700	2620					
14241	REGF	2530	2650	2760	2800	2990		
14317	REGI	3060	3020	3040				
14254	REGL	2640	2680					
23	REGN	3290	2560					
14724	REGS	6010	2740	4900	5490			
14320	REGT	3100	2540	3290				
13353	RJS1	2300	2210					
13344	RJS2	2220	2290	2350				
13361	RJS3	2360	2260					
13372	RJS4	2450	2410					
13337	RJUS	2170	780	2340	2490			
14517	RS1	4550	4600					
14513	RSAVE	4510	4610	4910	5520			
14236	SAV4	2440	2410	6150				
14232	SAVE	2390	300					
14335	SC	3240	4750	5530				
273	SCOLON	380	930	1790				
13274	SCOUNT	1930	2770	3610	3720	3730	3840	
15030	SEA	6650	580					
15050	SEA1	6810	6760					

D CROSS REFERENCE TABLE

D

CROSS REFERENCE TABLE

			830	2160	2220	2570	2680	3670	3770	3990	4040	4560	4580	4890
			4980	5050	5110	5150	5170	5370	5480	5500				
15443	T1CHAR	1000	810											
15421	T1LINE	980	830											
15201	T2	1740	770	790	2690	2820	2980	2990	3220	3270	4030	4080	4520	4570
			4590	5020	5040	5410	5420	5450						
15202	T3	1750	1940	2000	2060	2180	2280	2310	2360	2380	2810	2850	3070	3200
			4250	4330	4540	4600	5000	5060						
15203	T4	1760	2730	2860	2900	2940	3010	3050	3150	3310	3340			
15204	T5	1770	2780	2930	1410	1460	1530	1540	1630					
15205	T6	1780												
17500	TAPIN	450												
17502	TAPOT	460												
15241	TBUFFR	490	740	940										
15633	TCHRID	2930	520	1220	1540	2210	2710	3150	3170	3180	3200			
15367	TCOUNT	600	2070	840	890	4310	4430	760	2650					
15661	TDIGIT	3170	3010	3040										
15366	TDLMTR	590	660	1490	920	1140	1760	6670	770	3140				
15613	TFGET9	2620												
14363	TIME	3520	3350	3410	4760	4770								
15370	TINLIN	720	730	960										
15614	TINTIN	2640	1190	2000	2720	2740	2780							
15662	TLETTR	3180	3110											
15435	TNUM20	1210	1430											
15461	TNUM26	1460	1230											
15470	TNUM27	1530	1470											
15476	TNUM28	1590	1550											
15477	TNUM29	1600	1520											
15431	TNUMIN	1170	580	1200	1240	1500	1560	1580	1610	1620				
15503	TOCT42	1670												
15513	TOCT44	1750	1820											
15502	TOCTOT	1660	2070	2600	4010	4940	4980	840	890	3280	4310	4430	1850	
646000	TP.	730												
15017	TPR	6560												
15022	TPR1	6570												
14617	TRA	5260	500											
15563	TSIX24	2300	2360											
15572	TSIX26	2380	2330	2340	2350	2470								
15526	TSIXIN	1970	640	720	790	2170	2580	3900	6040	2010	2020	2170		
15561	TSIXOT	2280	2070	670	720	840	890	4310	4430	5080	2300	2320	2500	
15665	TTTYOT	3240	3180	3260	3500	3700	3730	4290	4360	4470	4470	4470	4760	4780
			1800	2450	2950	3130	3190	3290	3340	3360	3380			
15207	TVAL	1800	830	1020										
15237	TWORDB	470	700	800	810	1950	2020	2190	2320	2420	2440	2460	2480	2630
			2710	2820	2850	6560	1310	1330	1350	1410	1510	1590	1600	1980
			2160	2660	2670									
14711	UINST	5890	5160	5290	5760	5800	5870	5940	5950	5970				
14510	UN1	4450	4420											
14470	UNB	4390	480											
17777	VFLAG	480												
14616	WF	5210	4690	5120										
14354	WIN	3440	3400											

D

CROSS REFERENCE TABLE

14363 WT 3510 3450 3480 6390

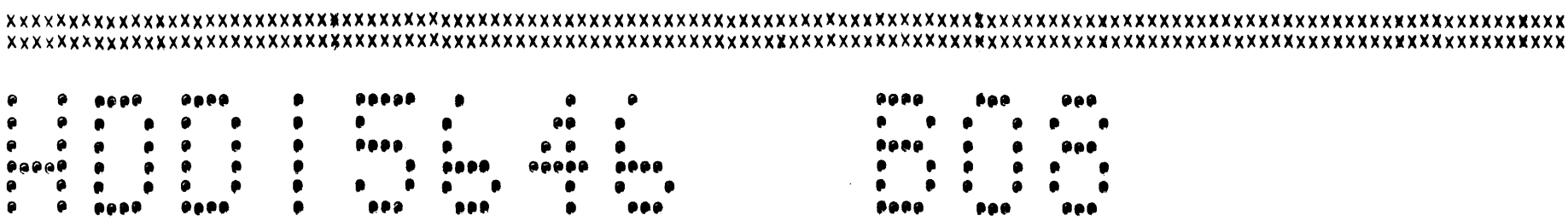
D UNDEFINED SYMBOLS

OCTZ	1790	
PURCOD	430	610

D

MACRO CROSS REFERENCE TABLE

CHAR	1380											
CHROT	1340											
COUNT	1460	2070	840	890	4310	4430						
CRLF	1300	4220	5330									
DEF	7070	7110	7120	7130	7140	7150	7160	7170	7180	7190	7200	7210	7220
		7230	7240	7250	7260	7270						
DELIM	1420	660	1490	920	1140	1760	6670					
EMESS	1750	2070	840	890	4310	4430						
ENTER	920	4970	720	1170	1660	1970	2190	2280	2380	2580	2640	2930	3240
		3320	7440									
FORMAT	1060	990	1240	1620	2230	2690	6120	6220	6340	6420	6460	6490	6510
LINE	1100	730										
LOOP	960	2000	2060	2930	3200	3210	830	2220	2680	3770	4080	4600	5050
		5480	6540	6950								
MESS	1620	2070	670	720	840	890	4310	4430	5080				
MESSR	1520	2070	670	720	840	890	4310	4430	5080				
NEG	1010	950	2800	2970	3470	1220	6430						
NMESS	1670												
NUM	1260	580											
OCT	1890	2600	4010	4980	3280								
OCTZ	1840	2070	4940	840	890	4310	4430						
OPA	1500	1550	1560	1570	1580	1590	1600	1610					
START	1100												
WORD	1140	640	720	790	2170	2580	3900	6040					
WORD1	1180	700	1950	2320	2440								
WORD2	1220	2020	2190	2420									



05/31/72 01:53:07

```
**********************************
**********************************
**                              **
**  PDP-9 MINI TIME-SHARING SYSTEM  **
**            DTSS:BASQ         **
**            MTSS:B06          **
**                              **
**********************************
**********************************
```

```
100             .TITLE  PDP-9 BASIC INTERPRETER
110             .NAME   BAS0
```

DEFINITIONS

```
                    120             ,STITL  DEFINITIONS
                    130     *
                    140     *       DEFINITION -- PSEUDO OPS
                    150     *
                    160     RET     ,OPDEF  JMP+20000       RETURN -- JMP **,X
                    170     *
                    180     *       DEFINITION -- PROGRAM CONSTANTS
                    190     *
                    200             ,HEAD   S               S FOR STORAGE
000001              210     UNIT    ,EQU    1               ARITHMETIC PRECISION
                    220     *
                    230     *       DEFINITION -- AUTO INDEX REGISTERS
                    240     *
                    250             ,HEAD   T               T FOR TTY
000010              260     TTYX    ,EQU    10              XR-0 USED FOR TTY I/O
000011              270     CHRX    ,EQU    11              XR-1 GLOBAL INPUT POINTER
000017              280     SBUF    ,EQU    17              XR-7 USED FOR SOURCE BUFFER POINTER
                    290             ,HEAD   0,P,I,T
000012              300     X       ,EQU    12              LOCAL
000013              310     Y       ,EQU    13              LOCAL
000014              320     Z       ,EQU    14              LOCAL
                    330     *
                    340     *       DEFINITION -- USEFUL MACROS
                    350     *
                    360     SOS     ,DEFINT                 SUBTRACT ONE FROM STORAGE
                    370             CLC                     C(AC) := 777777
                    380             TAD     #1              C(AC) := C(MEM)-1
                    390             DAC     #1              C(MEM) := C(AC)
                    400             ,ENDM   SOS
                    410     *
                    420     *       PROGRAM ORIGIN
                    430     *
                    440     *        .LOC 100
015777              450             ,LOC    16000-1
015777 615716       460             JMP     SUP             A CONVENIENT RESTART LOCATION
011551              470             ,LOC    16000-4227
                    480             ,HEAD   T               T FOR TTY
```

T TTY PRINT ROUTINE -- PRINT

```
                    490             .STITL  TTY PRINT ROUTINE -- PRINT
                    500     *
                    510     *       THIS ROUTINE PRINTS A SINGLE CHARACTER TO THE TTY.
                    520     *
                    530     *       ENTER WITH CHARACTER IN R-AC.
                    540     *
                    550     *       DEFINITION -- 'PRINT' -- PRINT CHARACTER MACRO
                    560     *
                    570     PRINT   .DEFIN
                    580             JMS     TSPRINT         CALL THE PRINT ROUTINE
                    590             .ENDM   PRINT
                    600     *
                    610     *       TSPRINT
                    620     *
011551  000000      630     PRINT   0
011552  700401      640             TSF                     ***
011553  611552      650             JMP     .-1             * PRINT THE CHARACTER WHEN THE TTY IS READY
011554  700406      660             TLS                     ***
011555  631551      670             RET     PRINT           RETURN
```

T MESSAGE PRINTING ROUTINE -- MESS

```
                    680             .STITL  MESSAGE PRINTING ROUTINE -- MESS
                    690     *
                    700     *       THIS ROUTINE IS CALLED TO PRINT A MESSAGE ON THE TTY.
                    710     *
                    720     *       ENTER WITH THE MESSAGE FOLLOWING THE CALL.  THE END OF THE
                    730     *       MESSAGE IS FLAGGED WITH A WORD OF -1.
                    740     *
                    750     *       DEFINITION -- 'MESS' -- MESSAGE MACRO
                    760     *
                    770     MESS    .DEFIN                  <MESSAGE>
                    780             JMS     TSMESS          CALL THE MESSAGE ROUTINE
                    790             .TEXT   \#1\
                    800             .IFE    '#2','CRLF',1
                    810             .DATA   15,12           GIVE CARRIAGE RETURN/LINE-FEED IF WANTED
                    820             777777                  FLAG THE END OF THE MESSAGE
                    830             .ENDM   MESS
                    840     *
                    850     *       TSMESS
                    860     *
011556  000000      870     MESS    0
011557  211556      880             LAC     MESS            ***
011560  040010      890             DAC     TTYX            * SET UP POINTER TO MESSAGE
011561  231556      900             LAC     MESS,X          ***
011562              910     MSA     ...
011562  751717      920             SAD     MONE            CHECK FOR END OF MESSAGE (EOM) FLAG
011563  611567      930             JMP     MSX             EOM -- RETURN
011564              940             PRINT                   PRINT THE CHARACTER
011565  220010      950             LAC     TTYX,X          FETCH THE NEXT CHARACTER
011566  611562      960             JMP     MSA             AND LOOP
011567              970     MSX     ...
011567  211713      980             LAC     FILL            SEND A FILL
011570              990             PRINT                   TO DELAY A LITTLE
011571  620010     1000             RET     TTYX,X          THEN RETURN
```

T TTY READ ROUTINE -- READ

```
                    1010          .STITL  TTY READ ROUTINE -- READ
                    1020    *
                    1030    *     THIS ROUTINE IS CALLED TO READ A LINE FROM THE TTY.
                    1040    *
                    1050    *     IT PLACES THE LINE ONE CHARACTER TO A WORD IN TSIBUF, AFTER
                    1060    *     PERFORMING SEVERAL ELEMENTARY EDITING OPERATIONS:
                    1070    *
                    1080    *     CTRL-X    DELETE THE LINE
                    1090    *     BACKARROW DELETE THE PRECEEDING CHARACTER
                    1100    *
011572  000000      1110    READ  0
011573  211714      1120          LAC     IBLN            GET THE BUFFER LENGTH
011574  051716      1130          DAC     IBCNT           SAVE FOR COUNTING
011575  211715      1140          LAC     IBFPT           GET INITIAL POINTER
011576  040011      1150          DAC     CHRX            SAVE IN LOCAL AUTO INDEX REGISTER
                    1160    *
                    1170    *     READ THE ACTUAL CHARACTER
                    1180    *
  011577            1190    RDA   ...
011577  700301      1200          KSF                     ***
011600  011577      1210          JMP     .-1             * READ THE CHARACTER
011601  700312      1220          KRB                     ***
                    1230    *
                    1240    *     CHECK FOR AN EDITING CHARACTER
                    1250    *
011602  011713      1260          AND     FILL            MASK OFF PARITY BIT
011603  555750      1270          SAD     (30)            CHECK FOR CTRL-X
011604  011643      1280          JMP     CTRLX           YES -- DELETE THE LINE
011605  555751      1290          SAD     (137)           CHECK FOR BACKARROW
011606  011660      1300          JMP     BACK            YES -- BACK UP
011607  555752      1310          SAD     (5)             CHECK FOR CTRL-E
011610  011701      1320          JMP     WRU             TELL HIM WHO WE ARE
011611  060011      1330          DAC     CHRX,X          SAVE THE CHARACTER IN THE BUFFER
011612  555753      1340          SAD     (15)            SEE IF THE END OF LINE
011613  011671      1350          JMP     EOL             YES -- CLEAN US UP
011614  451716      1360          ISZ     IBCNT           COUNT THE NEW CHARACTER
011615  011577      1370          JMP     RDA             IF MORE ROOM -- GET ANOTHER CHARACTER
```

T TTY READ ROUTINE -- READ

```
                1380            .EJECT
                1390    *
                1400    *       LINE TOO LONG -- GIVE MESSAGE AND DELETE IT
                1410    *
011616  111556  1420            JMS     TSMESS          CAN'T USE THE MESSAGE MACRO BECAUSE OF CR/LF
011617  000015  1430            .DATA   15,12           CR/LF
011620  000012
011621  000314  1440            .TEXT   \LINE TOO LONG -- \
011622  000311
011623  000316
011624  000305
011625  000240
011626  000324
011627  000317
011630  000317
011631  000240
011632  000314
011633  000317
011634  000316
011635  000307
011636  000240
011637  000255
011640  000255
011641  000240
011642  777777  1450            777777                  END FLAG
```

T TTY READ ROUTINE -- READ

```
                    1460            .EJECT
                    1470    *
                    1480    *       DELETE LINE ROUTINE
                    1490    *
    011643          1500    CTRLX   ...
    011643          1510            MESS    < DELETED>,CRLF
011657  611573      1520            JMP     READ+1          TRY AGAIN
                    1530    *
                    1540    *       BACKARROW ROUTINE
                    1550    *
    011660          1560    BACK    ...
011660  211717      1570            LAC     MONE            GET -1
011661  340011      1580            TAD     CHRX            AND GET THE POINTER LESS ONE IN R-AC
011662  555754      1590            SAD     (IBUF-2)        SEE IF TOO FAR
011663  611643      1600            JMP     CTRLX           IF SO -- SAY DELETED
011664  040011      1610            DAC     CHRX            RESTORE POINTER
011665  211717      1620            LAC     MONE            ***
011666  351716      1630            TAD     IBCNT           * DECREMENT THE COUNTER BY ONE
011667  051716      1640            DAC     IBCNT           ***
011670  611577      1650            JMP     RDA             AND READ ANOTHER CHARACTER
                    1660    *
                    1670    *       END OF LINE ROUTINE
                    1680    *
    011671          1690    EOL     ...
011671  215755      1700            LAC     (12)            SEND A LINE-FEED
    011672          1710            PRINT                   FOR GOOD MEASURE
011673  060011      1720            DAC     CHRX,X          SAVE IN BUFFER
011674  211717      1730            LAC     MONE            GET AN EOM FLAG
011675  060011      1740            DAC     CHRX,X          SAVE IN BUFFER
011676  211715      1750            LAC     IBFPT           SET UP A POINTER
011677  040011      1760            DAC     CHRX            FOR THE CALLER
011700  631572      1770            RET     READ            AND RETURN
                    1780    *
                    1790    *       WHO ARE YOU? ROUTINE
                    1800    *
    011701          1810    WRU     ...
    011701          1820            MESS    <BASIC>,CRLF
011712  611577      1830            JMP     RDA
```

T TTY READ ROUTINE -- READ

```
                    1840            ,EJECT
                    1850    *
                    1860    *       INPUT BUFFER AND STORAGE
                    1870    *
011713  000177      1880    FILL    177                     ASCII FILL/PARITY MASK
    000100          1890    IBN     ,EQU    100             LENGTH OF VISIBLE BUFFER
011714  777677      1900    IBLN    ,DATA   -IBN-1          LENGTH TO INITIALIZE COUNTER
011715  011717      1910    IBFPT   ,DATA   IBUF-1          INITIAL POINTER TO THE BUFFER
    011716          1920    IBCNT   ,BLOCK  1               COUNTER
011717  777777      1930    MONE    777777                  A MINUS ONE
    011720          1940    IBUF    ,BLOCK  IBN+2           INPUT BUFFER
```

T TTY READ -- CHAR

```
                    1950            .STITL  TTY READ -- CHAR
                    1960    *
                    1970    *       THIS ROUTINE IS CALLED TO PICK UP A CHARACTER FROM
                    1980    *       THE SOURCE LINE.
                    1990    *
                    2000    *       DEFINITION -- 'CHAR' -- GET CHARACTER MACRO
                    2010    *
                    2020    CHAR    .DEFIN
                    2030            JMS     TSCHAR          CALL THE SUBROUTINE
                    2040            .ENDM   CHAR
                    2050    *
                    2060    *       TSCHAR
                    2070    *
012022  000000      2080    CHAR    0
012023  220011      2090            LAC     CHRX,X          GET THE NEXT CHARACTER
012024  551717      2100            SAD     MONE            SEE IF EOM CHARACTER
012025  615551      2110            JMP     ESPARSE         SHOULDN'T GET THERE
012026  555756      2120            SAD     (040)           SEE IF BLANK
012027  612023      2130            JMP     CHAR+1          IGNORE IF SO
012030  052032      2140            DAC     LCHAR           SAVE IN CASE WE LOOK AGAIN
012031  632022      2150            RET     CHAR            AND RETURN
                    2160    *
                    2170    *       DEFINITION -- 'LCHAR' -- GET PREVIOUS CHARACTER MACRO
                    2180    *
                    2190    LCHAR   .DEFIN
                    2200            LAC     TSLCHAR         GET THE CHARACTER
                    2210            .ENDM   LCHAR
012032              2220    LCHAR   .BLOCK  1               LAST CHARACTER READ
```

```
T                                   TTY READ -- GET LINE NUMBER

                2230                ,STITL  TTY READ -- GET LINE NUMBER
                2240    *
                2250    *           THIS SUBROUTINE IS CALLED BY THE FILE BUILDING
                2260    *           ROUTINE TO ASSEMBLE A LINE NUMBER
                2270    *
012033  000000  2280    GNUM        0
    012034      2290                CHAR                    GET A CHARACTER
012035  052053  2300                DAC     TEM             SAVE IT
012036  777720  2310                LAW     -60             ***
012037  352053  2320                TAD     TEM             * SEE IF LESS THAN A NUMBER
012040  741100  2330                SPA                     ***
012041  632033  2340                RET     GNUM            IF SO -- RETURN
012042  052053  2350                DAC     TEM             SAVE IT
012043  777766  2360                LAW     -12             ***
012044  352053  2370                TAD     TEM             * SEE IF GREATER THAN A DIGIT
012045  740100  2380                SMA                     ***
012046  632033  2390                RET     GNUM            IF SO -- RETURN
012047  212033  2400                LAC     GNUM            FIX UP THE RETURN
012050  040012  2410                DAC     X               TO RETURN TO CALLER+2
012051  212053  2420                LAC     TEM             GET THE DIGIT
012052  620012  2430                RET     X               AND RETURN
    012053      2440    TEM         ,BLOCK  1
    012054      2450    TEM1        ,BLOCK  1
```

```
T                                TTY READ -- SOURCE FILE BUILDING ROUTINE

                    2460                 .STITL  TTY READ -- SOURCE FILE BUILDING ROUTINE
                    2470     *
                    2480     *           THIS ROUTINE IS USED TO BUILD THE PACKED SOURCE FILE
                    2490     *           FROM INPUT FROM THE TERMINAL
                    2500     *
012055  000000      2510     LINE        0
012056  111572      2520                 JMS       READ              READ IN THE LINE
012057  112033      2530                 JMS       GNUM              GET THE FIRST DIGIT OF LINE NUMBER
012060  612126      2540                 JMP       LINX              END IF DIDN'T BEGIN WITH DIGIT
012061  052054      2550                 DAC       TEM1              SAVE THE FIRST DIGIT
    012062          2560     LINA        ...
012062  112033      2570                 JMS       GNUM              GET THE NEXT DIGIT
012063  612074      2580                 JMP       LINB              NOT A DIGIT -- WE HAVE THE NUMBER
012064  744000      2590                 CLL                         ***
012065  212054      2600                 LAC       TEM1              * MULTIPLY PREVIOUS NUMBER
012066  653122      2610                 MUL                         * BY 10
012067  000012      2620                 10.                         ***
012070  641002      2630                 LACQ                        ***
012071  312053      2640                 ADD       TEM               * AND ADD THE NEW DIGIT
012072  052054      2650                 DAC       TEM1              ***
012073  612062      2660                 JMP       LINA              LOOP
                    2670     *
                    2680     *           SET UP TO CONDENSE THE LINE
                    2690     *
    012074          2700     LINB        ...
012074  212054      2710                 LAC       TEM1              GET THE LINE NUMBER
012075  060017      2720                 DAC       SBUF,X            AND SAVE IN THE BUFFER
012076  220017      2730                 LAC       SBUF,X            MAKE A PLACE FOR THE COUNT
012077  200017      2740                 LAC       SBUF              GET POINTER TO COUNT
012100  052134      2750                 DAC       BSBF              SAVE IT
012101  172134      2760                 DZM       BSBF,X            ZERO IT OUT
012102  212032      2770                 LAC       LCHAR             GET THE LAST CHARACTER READ
012103  612105      2780                 JMP       LIND              AND ENTER LOOP
    012104          2790     LINC        ...
    012104          2800                 CHAR                        GET THE NEXT CHARACTER
    012105          2810     LIND        ...
012105  640711      2820                 ALS       11                PUT THE FIRST CHARACTER IN THE UPPERHALF
012106  052054      2830                 DAC       TEM1              SAVE IT
012107  555757      2840                 SAD       (015000)          CHECK FOR EOL
012110  612120      2850                 JMP       LINF+1            YES
    012111          2860                 CHAR                        GET THE NEXT CHARACTER
012112  555753      2870                 SAD       (15)              SEE IF EOL
012113  612117      2880                 JMP       LINF              YES
012114  312054      2890                 ADD       TEM1              ADD IN FIRST CHARACTER
012115  060017      2900                 DAC       SBUF,X            SAVE IT IN BUFFER
012116  612104      2910                 JMP       LINC              LOOP
```

T TTY READ -- SOURCE FILE BUILDING ROUTINE

```
                    2920            ,EJECT
                    2930    *
                    2940    *       CLEAN UP AT END OF LINE
                    2950    *
        012117      2960    LINF    ...,
012117  312054      2970            ADD     TEM1            ADD IN THE LAST CHARACTER
012120  060017      2980            DAC     SBUF,X          SAVE IN THE BUFFER
012121  212134      2990            LAC     BSBF            GET POINTER TO BEGINNING OF LINE
012122  740001      3000            CMA                     COMPUTE THE LENGTH OF THE LINE
012123  300017      3010            ADD     SBUF            FROM THE POINTERS
012124  072134      3020            DAC     BSBF,X          SAVE IT IN THE BUFFER
012125  612056      3030            JMP     LINE+1          AND LOOP
        012126      3040    LINX    ...,
012126  200017      3050            LAC     SBUF            ***
012127  555760      3060            SAD     (SBFR-1)        * SEE IF JUST A COMMAND
012130  632055      3070            RET     LINE            ***
012131  160017      3080            DZM     SBUF,X          ***
012132  160017      3090            DZM     SBUF,X          * SET UP A NULL LINE
012133  632055      3100            RET     LINE            ***
012134  000000      3110    BSBF    ,DATA   0               POINTER TO BEGINNING OF LINE
```

T TTY READ -- RE-EXPAND A LINE

```
                    3120            .STITL  TTY READ -- RE-EXPAND A LINE
                    3130    *
                    3140    *       THIS ROUTINE IS CALLED BY THE PARSER TO RE-EXPAND
                    3150    *       A SOURCE LINE FROM THE BUFFER.
                    3160    *
012135  000000      3170    NLINE   0
012136  750004      3180            LAS                     ***
012137  741100      3190            SPA                     * SEE IF THE SWITCHES INDICATE A CHANGE OF HEART
012140  612173      3200            JMP     MON             ***
012141  220017      3210            LAC     SBUF,X          GET THE LINE NUMBER
012142  052240      3220            DAC     LNUM            SAVE IT
012143  220017      3230            LAC     SBUF,X          IGNORE THE LINE SIZE
012144  741200      3240            SNA                     SEE IF THERE IS ONE
012145  632135      3250            RET     NLINE           NO
012146  211715      3260            LAC     IBFPT           INITIALIZE THE CHARACTER BUFFER POINTER
012147  040011      3270            DAC     CHRX            FOR THE EXPANSION
                    3280    *
                    3290    *       EXPAND A LINE FROM THE SOURCE BUFFER INTO THE CHARACTER BUFFER
                    3300    *
    012150          3310    NLNA    ...
012150  220017      3320            LAC     SBUF,X          ***
012151  640511      3330            LRS     11              * GET THE FIRST CHARACTER OF THE WORD
012152  511713      3340            AND     FILL            * WITHOUT WHAT WAS IN THE LINK
012153  060011      3350            DAC     CHRX,X          ***
012154  555753      3360            SAD     (15)            SEE IF EOL
012155  612164      3370            JMP     NLNX            YES
012156  640611      3380            LLS     11              GET THE SECOND CHARACTER
012157  511713      3390            AND     FILL            MASK TO JUST THE CHARACTER
012160  060011      3400            DAC     CHRX,X          SAVE IT
012161  555753      3410            SAD     (15)            SEE IF EOL
012162  612164      3420            JMP     NLNX            YES
012163  612150      3430            JMP     NLNA
    012164          3440    NLNX    ...
012164  777777      3450            LAW     -1              FLAG THE EOL
012165  060011      3460            DAC     CHRX,X          FOR ALL TO KNOW
012166  211715      3470            LAC     IBFPT           REINITIALIZE THE POINTER
012167  040011      3480            DAC     CHRX            FOR THE SCAN TO USE
012170  212135      3490            LAC     NLINE           ***
012171  040012      3500            DAC     X               *NORMAL RETURN
012172  620012      3510            RET     X               ***
```

T MONITOR

```
                    3520            .STITL  MONITOR
                    3530    *
                    3540    *       THIS ROUTINE FUNCTIONS AS THE MONITOR FOR THE BASIC
                    3550    *       INTERPRETER,  IT EITHER BEGINS THE BUILDING OF
                    3560    *       A FILE OR ACCEPTS A COMMAND.
                    3570    *
    012173          3580    MON     ...
    012173          3590            MESS    <ENTER FILE OR COMMAND>,CRLF
012224  215760      3600            LAC     (SBFR-1)        POINTER TO THE SOURCE BUFFER
012225  040017      3610            DAC     SBUF            SAVE FOR SCAN
012226  112055      3620            JMS     LINE            BUILD THE PROGRAM
012227  215760      3630            LAC     (SBFR-1)
012230  040017      3640            DAC     SBUF            REINITIALIZE THE BUFFER POINTER
012231  212032      3650            LAC     LCHAR           GET THE COMMAND
012232  555761      3660            SAD     (122)
012233  614220      3670            JMP     PSPARSE         RUN
012234  555762      3680            SAD     (114)
012235  613241      3690            JMP     LIST            LIST
012236  705001      3700            705001                  ELSE HALT FOR NOW
012237  612173      3710            JMP     MON             BUT BE ABLE TO START UP AGAIN
    012240          3720    LNUM    .BLOCK  1               LINE NUMBER
    012241          3730    SBFR    .BLOCK  1000            SOURCE BUFFER
```

```
T                                       LIST A SOURCE FILE

                    3740                .STITL  LIST A SOURCE FILE
                    3750        *
                    3760        *       THIS ROUTINE IS USED TO LIST THE SOURCE FILE
                    3770        *
        013241      3780        LIST    ...
013241  112135      3790                JMS     NLINE           GET THE NEXT LINE
013242  612173      3800                JMP     MON             REACHED THE END
013243  212240      3810                LAC     LNUM            GET THE LINE NUMBER
013244  114375      3820                JMS     I$PRT           PRINT IT
013245  760040      3830                LAW     040             A SPACE
        013246      3840                PRINT                   PRINT IT
        013247      3850        LISA    ...
        013247      3860                CHAR                    GET THE NEXT CHARACTER
        013250      3870                PRINT                   PRINT IT
013251  555753      3880                SAD     (015)           SEE IF EOL
013252  613254      3890                JMP     LISX            YES
013253  613247      3900                JMP     LISA            NO -- LOOP
        013254      3910        LISX    ...
013254  760012      3920                LAW     012             GIVE THE LINE FEED
        013255      3930                PRINT                   TO CLEAN US UP
013256  613241      3940                JMP     LIST            AND LOOP
                    3950                .HEAD
```

STACK MANAGEMENT -- MACROS

```
3960            ,STITL  STACK MANAGEMENT -- MACROS
3970    *
3980    *       THIS MACRO IS USED TO PUSH THE CONTENTS OF
3990    *       R-AC ONTO ONE OF THE THREE STACKS IN THIS
4000    *       PROGRAM.  THE STACK ARE:
4010    *
4020    *       R     RECURSION CONTROL STACK
4030    *       O     OPERATOR STACK
4040    *       V     VARIABLE STACK
4050    *
4060    *       DEFINITION -- 'PUSH' -- PUSH MACRO
4070    *
4080    PUSH    ,DEFIN                          <LETTER OF STACK>
4090            JMS     #1$PUSH                 CALL THE PROPER SUBROUTINE
4100            ,ENDM   PUSH
4110    *
4120    *       THIS MACRO IS USED TO POP THE TOP DATUM FROM
4130    *       ONE OF THE THREE STACKS INTO R-AC.  IT USES
4140    *       THE SAME SYMBOLS AS 'PUSH'.
4150    *
4160    *       DEFINITION -- 'POP' -- POP MACRO
4170    *
4180    POP     ,DEFIN                          <LETTER OF STACK>
4190            JMS     #1$POP                  CALL THE PROPER SUBROUTINE
4200            ,ENDM   POP
4210    *
4220    *       DEFINITION -- 'POPV' -- POP VARIABLE AND DEREFERENCE
4230    *
4240    POPV    ,DEFIN
4250            POP     V                       GET THE POINTER TO THE DATUM
4260            DAC     TEM                     SAVE IT TEMPORARILY
4270            LAC     TEM,X                   GET THE DATUM
4280            ,ENDM   POPV
```

STACK MANAGEMENT -- PUSH

```
                    4290            ,STITL  STACK MANAGEMENT -- PUSH
                    4300    *
                    4310    *       THIS SUBROUTINE PUSHES THE DATUM SAVED IN $DATUM
                    4320    *       ONTO THE STACK WHOSE POINTERS ARE AT C(AC)+1 ON
                    4330    *       ENTRANCE.  THE FORMAT OF THIS POINTER BLOCK FOR
                    4340    *       EACH OF THE STACKS IS:
                    4350    *
                    4360    *       0     CURRENT STACK POINTER
                    4370    *       1     POINTER TO BOTTOM OF STACK
                    4380    *       2     -SIZE-1 (ONE'S COMPLEMENT OF SIZE)
                    4390    *
                    4400    *       $PUSH
                    4410    *
013257  000000      4420    PUSH    0
013260  040012      4430            DAC     X               SAVE POINTER TO STACK DESCRIPTION
013261  040013      4440            DAC     Y               ALSO FOR UPDATING POINTER
013262  220012      4450            LAC     X,X             ***
013263  040014      4460            DAC     Z               * SAVE STACK POINTER AND
013264  740001      4470            CMA                     * COMPUTE AMOUNT OF STACK IN USE
013265  360012      4480            TAD     X,X             ***
013266  560012      4490            SAD     X,X             COMPARE WITH THE SIZE
013267  615570      4500            JMP     ESSOVF          STACK OVERFLOW -- EXPRESSION TOO COMPLICATED
013270  213315      4510            LAC     DATUM           GET THE DATUM TO BE PUSHED
013271  060014      4520            DAC     Z,X             PUSH
013272  460013      4530            ISZ     Y,X             INCREMENT THE REAL POINTER
013273  633257      4540            RET     PUSH            RETURN
013274  740040      4550            HLT                     SHOULDN'T GET HERE
```

STACK MANAGEMENT -- POP

```
                    4560            ,STITL  STACK MANAGEMENT -- POP
                    4570    *
                    4580    *       THIS SUBROUTINE POPS THE DATUM FROM THE TOP OF THE
                    4590    *       STACK WHOSE DESCRIPTION BLOCK IS AT C(AC)+1 ON ENTRANCE.
                    4600    *       THIS DESCRIPTION BLOCK IS THE SAME AS THE ONE USED BY PUSH.
                    4610    *
                    4620    *       $POP
                    4630    *
013275  000000      4640    POP     0
013276  040012      4650            DAC     X               SAVE POINTER TO DESCRIPTION BLOCK
013277  040013      4660            DAC     Y               ALSO FOR UPDATING POINTER
013300  220012      4670            LAC     X,X             GET POINTER TO STACK
013301  053315      4680            DAC     DATUM           SAVE TEMPORARILY
013302  220012      4690            LAC     X,X             ***
013303  740001      4700            CMA                     * COMPUTE -AMOUNT IN USE
013304  353315      4710            TAD     DATUM           ***
013305  553314      4720            SAD     MTWO            SEE IF WE WERE AT THE BOTTOM
013306  615551      4730            JMP     ESPARSE         WE'VE BLOWN IT SOMEWHERE
013307  777777      4740            LAW     -1              ***
013310  353315      4750            TAD     DATUM           * DECREMENT REAL POINTER
013311  060013      4760            DAC     Y,X             ***
013312  233315      4770            LAC     DATUM,X         GET THE DATUM
013313  633275      4780            RET     POP             AND RETURN
                    4790    *
                    4800    *       STORAGE FOR $PUSH AND $POP
                    4810    *
013314  777776      4820    MTWO    ,DATA   -2              FOR CHECKING STACK UNDERFLOW
    013315          4830    DATUM   ,BLOCK  1               FOR DATUM OR POINTER TO DATUM
                    4840            ,EOT    BAS1
                    4000            ,HEAD   R               R FOR RECURSION
```

R STACK MANAGEMENT -- RECURSION CONTROL STACK

```
                    4010            ,STITL  STACK MANAGEMENT -- RECURSION CONTROL STACK
                    4020    *
                    4030    *       THIS ROUTINE IS CALLED TO PUSH THE DATUM IN R-AC
                    4040    *       ONTO THE RECURSION CONTROL STACK, R$STACK.
                    4050    *
013316  000000      4060    PUSH    0
013317  053315      4070            DAC     $DATUM          SAVE DATUM FOR $PUSH
013320  773327      4080            LAW     SDB-1           POINTER TO STACK DESCRIPTION BLOCK
013321  113257      4090            JMS     $PUSH           PUSH
013322  633316      4100            RET     PUSH            AND RETURN
                    4110    *
                    4120    *       THIS ROUTINE IS USED TO POP THE DATUM ON THE
                    4130    *       TOP OF THE RECURSION CONTROL STACK INTO R-AC.
                    4140    *
013323  000000      4150    POP     0
013324  773327      4160            LAW     SDB-1           POINTER TO STACK DESCRIPTION BLOCK
013325  113275      4170            JMS     $POP            POP
013326  633323      4180            RET     POP             AND RETURN
                    4190    *
                    4200    *       STACK DESCRIPTION BLOCK
                    4210    *
000400              4220    SIZE    ,EQU    400             STACK SIZE
013327  013332      4230    IPTR    ,DATA   STACK-1         INITIAL POINTER
013330              4240    SDB     ,,
013330  013332      4250            ,DATA   STACK-1         STACK POINTER
013331  013333      4260            ,DATA   STACK           BOTTOM POINTER
013332  777377      4270            ,DATA   -SIZE-1         ONE'S COMPLEMENT OF SIZE
013333              4280    STACK   ,BLOCK  SIZE            STACK
                    4290            ,HEAD   0               0 FOR OPERATOR
```

0 STACK MANAGEMENT -- OPERATOR STACK

```
                    4300            .STITL  STACK MANAGEMENT -- OPERATOR STACK
                    4310    *
                    4320    *       THIS ROUTINE IS CALLED TO PUSH THE DATUM IN R-AC
                    4330    *       ONTO THE OPERATOR STACK, OSSTACK.
                    4340    *
013733  000000      4350    PUSH    0
013734  053315      4360            DAC     $DATUM          SAVE DATUM FOR $PUSH
013735  773744      4370            LAW     SDB-1           POINTER TO STACK DESCRIPTION BLOCK
013736  113257      4380            JMS     $PUSH           PUSH
013737  633733      4390            RET     PUSH            AND RETURN
                    4400    *
                    4410    *       THIS ROUTINE IS USED TO POP THE DATUM ON THE
                    4420    *       TOP OF THE OPERATOR STACK INTO R-AC.
                    4430    *
013740  000000      4440    POP     0
013741  773744      4450            LAW     SDB-1           POINTER TO STACK DESCRIPTION BLOCK
013742  113275      4460            JMS     $POP            POP
013743  633740      4470            RET     POP             AND RETURN
                    4480    *
                    4490    *       STACK DESCRIPTION BLOCK
                    4500    *
000100              4510    SIZE    .EQU    100             STACK SIZE
013744  013747      4520    IPTR    .DATA   STACK-1         INITIAL POINTER
013745              4530    SDB     ...
013745  013747      4540            .DATA   STACK-1         STACK POINTER
013746  013750      4550            .DATA   STACK           BOTTOM POINTER
013747  777677      4560            .DATA   -SIZE-1         ONE'S COMPLEMENT OF SIZE
013750              4570    STACK   .BLOCK  SIZE            STACK
                    4580            .HEAD   V               V FOR VARIABLE
```

V STACK MANAGEMENT -- VARIABLE STACK

```
                    4590            .STITL  STACK MANAGEMENT -- VARIABLE STACK
                    4600    *
                    4610    *       THIS ROUTINE IS CALLED TO PUSH THE DATUM IN R-AC
                    4620    *       ONTO THE VARAIBLE STACK, VSSTACK,
                    4630    *
014050  000000      4640    PUSH    0
014051  053315      4650            DAC     $DATUM          SAVE DATUM FOR $PUSH
014052  774061      4660            LAW     SDB-1           POINTER TO STACK DESCRIPTION BLOCK
014053  113257      4670            JMS     $PUSH           PUSH
014054  634050      4680            RET     PUSH            AND RETURN
                    4690    *
                    4700    *       THIS ROUTINE IS USED TO POP THE DATUM ON THE
                    4710    *       TOP OF THE VARIABLE STACK INTO R-AC,
                    4720    *
014055  000000      4730    POP     0
014056  774061      4740            LAW     SDB-1           POINTER TO STACK DESCRIPTION BLOCK
014057  113275      4750            JMS     $POP            POP
014060  634055      4760            RET     POP             AND RETURN
                    4770    *
                    4780    *       STACK DESCRIPTION BLOCK
                    4790    *
000100              4800    SIZE    .EQU    100             STACK SIZE
014061  014064      4810    IPTR    .DATA   STACK-1         INITIAL POINTER
014062              4820    SDB     ...
014062  014064      4830            .DATA   STACK-1         STACK POINTER
014063  014065      4840            .DATA   STACK           BOTTOM POINTER
014064  777677      4850            .DATA   -SIZE-1         ONE'S COMPLEMENT OF SIZE
014065              4860    STACK   .BLOCK  SIZE            STACK
                    4870            .HEAD   S               S FOR STORAGE
```

S STORAGE MANAGEMENT -- TEMPORARY

```
                    4880            ,STITL  STORAGE MANAGEMENT -- TEMPORARY
                    4890    *
                    4900    *       THIS ROUTINE ALLOCATES A TEMPORARY STORAGE CELL FOR
                    4910    *       THE RESULT OF A COMPUTATION.  IT RETURNS A POINTER IN
                    4920    *       R-AC TO THE BEGINNING OF THE ALLOCATED STORAGE.  IT
                    4930    *       ALLOCATES THE CURRENT LOGICAL (ARITHMETIC) WORD SIZE.
                    4940    *
014165  000000      4950    TEMP    0
014166  777777      4960            LAW     -UNIT           DECREMENT THE CURRENT
014167  354176      4970            TAD     TCNT            TEMPORARY POINTER
014170  741100      4980            SPA                     IF WE'VE RUN OUT
014171  214175      4990            LAC     TSIZE           START OVER AGAIN
014172  054176      5000            DAC     TCNT            SAVE NEW COUNT
014173  354177      5010            TAD     TPRT            RELOCATE BY BASE OF AREA
014174  634165      5020            RET     TEMP            RETURN
                    5030    *
                    5040    *       TEMPORARY STORAGE AREA
                    5050    *
000020              5060    TN      ,EQU    20              NUMBER OF TEMPORARY CELLS
014175  000017      5070    TSIZE   ,DATA   TN*UNIT-UNIT    SIZE OF TEMPORARY AREA
014176  000000      5080    TCNT    ,DATA   0               CURRENT POINTER
014177  014200      5090    TPRT    ,DATA   .+1             POINTER TO TEMPORARY AREA BASE
014200              5100            ,BLOCK  TN*UNIT         TEMPORARY AREA
                    5110            ,HEAD   P               P FOR PARSE
```

P PARSER -- INITIALIZATION

```
                    5120            ,STITL  PARSER -- INITIALIZATION
                    5130    *
                    5140    *       THIS ROUTINE IS USED TO INITIALIZE THE SCAN OF
                    5150    *       A LINE.  IT:
                    5160    *
                    5170    *       READS THE LINE
                    5180    *       INITIALIZES THE STACKS
                    5190    *       ENTERS THE PARSER
                    5200    *
    014220          5210    PARSE   ,,;
014220  112135      5220            JMS     TSNLINE         GET THE NEXT LINE
014221  615644      5230            JMP     ESNEND          NO END STATEMENT
                    5240    *
                    5250    *       INITIALIZE THE STACKS
                    5260    *
014222  213327      5270            LAC     RSIPTR          RECURSION STACK
014223  053330      5280            DAC     RSSDB           SAME
014224  200011      5290            LAC     TSCHRX          GET THE CHARACTER POINTER
    014225          5300            PUSH    R               AND PUSH
014226  215763      5310            LAC     (SSSTATE)       ***
014227  054263      5320            DAC     ALT             * GET FIRST ALTERNATIVE
    014230          5330            PUSH    R               ***
014231  215764      5340            LAC     (SSSTATE+2)     ***
014232  054262      5350            DAC     PART            * GET FIRST PART
    014233          5360            PUSH    R               AND PUSH
014234  213744      5370            LAC     OSIPTR          ***
014235  053745      5380            DAC     OSSDB           * OPERATOR STACK
    014236          5390            PUSH    R               ***
014237  214061      5400            LAC     VSIPTR          ***
014240  054062      5410            DAC     VSSDB           * VARIABLE STACK
    014241          5420            PUSH    R               ***
014242  614266      5430            JMP     TEST            AND START IN THE MIDDLE
```

P PARSER -- PUSH ONTO THE CONTROL STACK

```
                    5440            .STITL  PARSER -- PUSH ONTO THE CONTROL STACK
                    5450    *
                    5460    *       PUSH THE CURRENT STATE ONTO THE STACK
                    5470    *
        014243      5480    PUSH    ...
    014243  200011  5490            LAC     TSCHRX          THE SOURCE POINTER
        014244      5500            PUSH    R               PUSH
    014245  214263  5510            LAC     ALT             GET THE CURRENT ALTERNATIVE
        014246      5520            PUSH    R               AND PUSH
    014247  214262  5530            LAC     PART            GET THE CURRENT PART
        014250      5540            PUSH    R               AND PUSH
    014251  213745  5550            LAC     OSSDB           OPERATOR STACK POINTER
        014252      5560            PUSH    R               PUSH
    014253  214062  5570            LAC     VSSDB           VARIABLE STACK POINTER
        014254      5580            PUSH    R               PUSH
    014255  234262  5590            LAC     PART,X          GET POINTER TO NEW TYPE
    014256  054263  5600            DAC     ALT             SAVE AS NEW ALTERNATIVE
    014257  355765  5610            TAD     (2)             AND MAKE POINTER TO NEW PART
    014260  054262  5620            DAC     PART            SAVE
    014261  614266  5630            JMP     TEST            AND TEST THE NEW ONE
                    5640    *
                    5650    *       ALTERNATIVE AND PART POINTERS FOR OUR USE
                    5660    *
        014262      5670    PART    .BLOCK  1               POINTER TO CURRENT PART
        014263      5680    ALT     .BLOCK  1               POINTER TO CURRENT ALTERNATIVE
        014264      5690    TEMP    .BLOCK  1               A LOCAL TEMPORARY
        014265      5700    LPART   .BLOCK  1               LAST PART RECOGNIZED
```

P PARSER -- TEST

```
                    5710            .STITL  PARSER -- TEST
                    5720    *
                    5730    *       THIS ROUTINE IS USED TO DECIDE WHETHER THE NEXT
                    5740    *       SYNTACTIC OBJECT IS A TYPE OR AN ATOM.
                    5750    *
                    5760    *       IF A TYPE -- PUSH THE CURRENT STATE ONTO THE
                    5770    *       RECURSION CONTROL STACK AND TRY THE NEW TYPE.
                    5780    *
                    5790    *       IF AN ATOM TRY TO MATCH -- ON SUCCESS GO TO TRY THE NEXT PART
                    5800    *       ON FAILURE -- GO TO PSFAIL
                    5810    *
      014266        5820    TEST    ...
014266  234262      5830            LAC     PART,X          GET THE NEW PART
014267  741100      5840            SPA                     SEE IF A LITERAL
014270  614272      5850            JMP     MLIT            YES -- TRY TO MATCH LITERAL
014271  614243      5860            JMP     PUSH            IS A TYPE -- PUSH AND KEEP GOING
                    5870    *
                    5880    *       NEXT OBJECT IS AN ATOM -- TRY TO MATCH
                    5890    *
      014272        5900    MLIT    ...
014272  555766      5910            SAD     (-1)            IF EMPTY
014273  614336      5920            JMP     TRUE            WE ALWAYS MATCH
014274  511713      5930            AND     TSFILL          MASK THE LITERAL FLAG
014275  054264      5940            DAC     TEMP            AND SAVE
      014276        5950            CHAR                    GET A CHARACTER FROM THE SOURCE STRING
014277  554264      5960            SAD     TEMP            SEE IF WE'VE MATCHED
014300  614336      5970            JMP     TRUE            YES -- SEE ABOUT THE NEXT THING
```

```
P                                   PARSER -- FAIL

                    5980                .STITL  PARSER -- FAIL
                    5990    *
                    6000    *       THIS ROUTINE TRYS TO GET THE NEXT ALTERNATIVE WHEN A MATCH
                    6010    *       FAILS.  IF THERE ARE NO MORE ALTERNATIVES TO THE CURRENT DEFINITION,
                    6020    *       IT POPS THE CONTROL STACK AND TRYS FOR MORE IN THAT ONE.
                    6030    *       IF WE GET BACK TO THE BOTTOM OF THE STACK, WE'VE FAILED UTTERLY.
                    6040    *
                    6050    *       FALL THROUGH FROM THE PREVIOUS PAGE OR CALLS ITSELF.
                    6060    *
        014301      6070    FAIL    ...
014301  234263      6080            LAC     ALT,X           GET THE NEXT ALTERNATIVE
014302  054263      6090            DAC     ALT             SAVE AS NEW ALTERNATIVE, MAYBE
014303  741200      6100            SNA                     SEE IF THERE WERE MORE
014304  614324      6110            JMP     FPOP            NO -- GO POP
                    6120    *
                    6130    *       BACK UP AND TRY AGAIN
                    6140    *
014305  777773      6150            LAW     -5              ***
014306  353330      6160            TAD     RSSDB           * FUDGE A POINTER INTO THE STACK
014307  040012      6170            DAC     X               ***
014310  220012      6180            LAC     X,X             GET THE OLD SOURCE POINTER
014311  040011      6190            DAC     T$CHRX          RESTORE IT
014312  220012      6200            LAC     X,X             IGNORE OLD ALTERNATIVE
014313  220012      6210            LAC     X,X             IGNORE OLD PART
014314  220012      6220            LAC     X,X             GET OLD OPERATOR STACK POINTER
014315  053745      6230            DAC     OSSDB           RESTORE IT
014316  220012      6240            LAC     X,X             GET OLD VARIABLE STACK POINTER
014317  054062      6250            DAC     VSSDB           RESTORE IT
014320  214263      6260            LAC     ALT             GET THE ONE WE JUST FIGURED OUT
014321  355765      6270            TAD     (2)             MAKE IT POINT TO THE FIRST PART
014322  054262      6280            DAC     PART            AND SAVE IN PART POINTER
014323  614266      6290            JMP     TEST            TRY AGAIN
                    6300    *
                    6310    *       TRY GETTING THE NEXT ALTERNATIVE FROM THE ONE ABOVE US
                    6320    *
        014324      6330    FPOP    ...
        014324      6340            POP     R               GET VARIABLE STACK POINTER
        014325      6350            POP     R               GET OPERATOR STACK POINTER
        014326      6360            POP     R               GET PART POINTER
        014327      6370            POP     R               GET ALTERNATIVE POINTER
014330  054263      6380            DAC     ALT             AND USE IT
        014331      6390            POP     R               GET CHARACTER POINTER
014332  213330      6400            LAC     RSSDB           SEE IF WE'RE DONE
014333  353327      6410            SAD     RSIPTR          SEE IF AT THE BEGINNING
014334  615625      6420            JMP     ESFAIL          WE BLEW IT
014335  614301      6430            JMP     FAIL            TRY AGAIN
```

P PARSER -- SUCCESSFUL MATCH

```
                    6440            .STITL  PARSER -- SUCCESSFUL MATCH
                    6450    *
                    6460    *       COME HERE ON A SUCCESSFUL MATCH
                    6470    *
    014336          6480    TRUE    ...
014336  214262      6490            LAC     PART            SAVE WHATEVER WE JUST MATCHED
014337  054265      6500            DAC     LPART           IN CASE WE SHOULD WANT TO REFER TO IT
    014340          6510    TRUA    ...
014340  454262      6520            ISZ     PART            INCREMENT PART POINTER
014341  234262      6530            LAC     PART,X          GET THE POINTER TO THE SUCCESS ROUTINE (MAYBE)
014342  054264      6540            DAC     TEMP            SAVE IT IN CASE IT IS
014343  214263      6550            LAC     ALT             ***
014344  040012      6560            DAC     X               * GET THE PART COUNTER
014345  220012      6570            LAC     X,X             ***
014346  554262      6580            SAD     PART            SEE IF WE'RE DONE WITH THIS TYPE
014347  634264      6590            JMP     TEMP,X          DO WHAT IT SAYS
014350  614266      6600            JMP     TEST            ELSE TRY TO MATCH THIS PART
                    6610    *
                    6620    *       COME HERE AFTER DOING WHATEVER ACTION WAS NECESSARY
                    6630    *
                    6640    *       POP THE STACK AND GET THE NEXT PART OF THE CALLER
                    6650    *
    014351          6660    OK      ...
    014351          6670            POP     R               POP THE VARIABLE STACK
    014352          6680            POP     R               AND THE OPERATOR STACK
    014353          6690            POP     R               AND THE PART
014354  054262      6700            DAC     PART            SAVE AS CURRENT PART
    014355          6710            POP     R               GET THE ALTERNATIVE
014356  054263      6720            DAC     ALT             AND SAVE AS ALTERNATIVE
    014357          6730            POP     R               THROW AWAY THE SOURCE POINTER
014360  213330      6740            LAC     R$SDB           ***
014361  553327      6750            SAD     R$IPTR          * SEE IF WE'RE DONE
014362  614220      6760            JMP     PARSE           ***
014363  614340      6770            JMP     TRUA            TRY TO POP AGAIN
                    6780            .HEAD   I               I FOR INTERPRETER
```

```
I                                   INTERPRETER -- PRINT

                    6790            ,STITL  INTERPRETER -- PRINT
                    6800    *
                    6810    *       THIS ROUTINE PRINTS THE VALUE OF THE THING
                    6820    *       POINTED TO BY THE TOP OF THE VARIABLE STACK
                    6830    *
          014364    6840    PRINT   ,,,
          014364    6850            POPV                    GET THE DATUM
    014367 114375   6860            JMS     PRT             CALL THE PRINT SUBROUTINE
    014370 111556   6870            JMS     T$MESS          PRINT CR/LF
    014371 000015   6880            ,DATA   015,012,-1
    014372 000012
    014373 777777
    014374 614351   6890            JMP     PSOK
```

I INTERPRETER -- PRINT SUBROUTINE

```
                    6900            ,STITL  INTERPRETER -- PRINT SUBROUTINE
                    6910    *
                    6920    *       THIS ROUTINE PRINTS THE VALUE OF R-AC ON ENTRANCE
                    6930    *
014375  000000      6940    PRT     0
014376  054442      6950            DAC     NTEM            SAVE IT
014377  741200      6960            SNA                     SEE IF TO PRINT ZERO
014400  614437      6970            JMP     PRT0            YES
014401  740100      6980            SMA
014402  614411      6990            JMP     PRTA            POSITIVE -- PRINT NO SIGN
                    7000    *
                    7010    *       HANDLE NEGATIVE NUMBERS
                    7020    *
014403  740001      7030            CMA                     MAKE THE NUMBER POSITIVE
014404  054442      7040            DAC     NTEM            SAVE IT BACK
014405  741200      7050            SNA                     CHECK FOR THE OTHER ZERO
014406  614437      7060            JMP     PRT0            PRINT A ZERO IF SO
014407  760055      7070            LAW     55              PRINT A '-'
    014410          7080            PRINT                   PRINT IT
                    7090    *
                    7100    *       DIVIDE BY 10 AND STACK REMAINDER
                    7110    *       UNTIL A QUOTIENT OF 0
                    7120    *
    014411          7130    PRTA    ...
014411  777772      7140            LAW     -6              SET UP A COUNTER
014412  054264      7150            DAC     PSTEMP          THE NUMBER OF DIGITS
014413  214442      7160            LAC     NTEM            GET THE NUMBER
    014414          7170    PRTB    ...
014414  741200      7180            SNA                     SEE IF MORE TO DO
014415  614425      7190            JMP     PRTC            NO
014416  744000      7200            CLL                     CLEAR THE LINK
014417  657323      7210            IDIVS                   DIVIDE
014420  000012      7220            10.                     BY 10
    014421          7230            PUSH    V               PUT THE CHARACTER ON THE VARIABLE
014422  641002      7240            LACQ                    GET THE QUOTIENT INTO AC
014423  454264      7250            ISZ     PSTEMP          COUNT THIS DIGIT
014424  614414      7260            JMP     PRTB            IF MORE ROOM -- LOOP
```

INTERPRETER -- PRINT SUBROUTINE

```
                    7270            ,EJECT
                    7280    *
                    7290    *       PRINT THE DIGITS WE STACKED
                    7300    *
        014425      7310    PRTC    ...
014425  214264      7320            LAC     PSTEMP          ***
014426  355752      7330            TAD     (5)             * SEE HOW MANY TO PRINT
014427  740001      7340            CMA                     ***
014430  054264      7350            DAC     PSTEMP          SAVE THE COUNT
        014431      7360    PRTD    ...
        014431      7370            POP     V               ***
014432  355767      7380            TAD     (60)            * PRINT THE DIGIT
        014433      7390            PRINT                   ***
014434  454264      7400            ISZ     PSTEMP          SEE IF DONE
014435  614431      7410            JMP     PRTD            NO -- GET ANOTHER
        014436      7420    PRTX    ...
014436  634375      7430            RET     PRT             AND RETURN
        014437      7440    PRTO    ...
014437  760060      7450            LAW     60              GET A ZERO
        014440      7460            PRINT                   PRINT IT
014441  614436      7470            JMP     PRTX            AND EXIT
        014442      7480    NTEM    .BLOCK  1               TEMPORARY FOR NUMBERS
        014443      7490    TEM     .BLOCK  1               TEMPORARY FOR DEREFERENCING VARIABLES
```

I INTERPRETER -- GOTO

```
                    7500            .STITL  INTERPRETER -- GOTO
                    7510    *
                    7520    *       THIS ROUTINE SETS UP A TRANSFER OF CONTROL TO
                    7530    *       THE LINE WHOSE NUMBER IS ON THE TOP OF THE VARIABLE STACK
                    7540    *
    014444          7550    GOTO    ...
    014444          7560            POPV                    GET THE LINE NUMBER
014447  054442      7580            DAC     NTEM            SAVE IT
    014450          7582    GOTE    ...
014450  215760      7590            LAC     (TSSBFR-1)      GET THE BUFFER POINTER
    014451          7600    GOTA    ...
014451  040012      7610            DAC     X               SAVE IT
014452  214442      7620            LAC     NTEM            GET THE LINE NUMBER
014453  560012      7630            SAD     X,X             SEE IF WE MATCH
014454  614462      7640            JMP     GOTX            YES -- EXIT
014455  220012      7650            LAC     X,X             GET THE LENGTH OF THIS LINE
014456  741200      7660            SNA                     SEE IF WE'RE PAST THE END
014457  615666      7670            JMP     ESUND           UNDEFINED LINE NUMBER
014460  300012      7680            ADD     X               AND A POINTER TO THE NEXT ONE
014461  614451      7690            JMP     GOTA            AND LOOP
    014462          7700    GOTX    ...
014462  777777      7710            LAW     -1              DECREMENT THE POINTER BY ONE
014463  340012      7720            TAD     X               SO THAT IT IS PROPER FOR THIS LINE
014464  040017      7730            DAC     TSSBUF          SET THE NEW LINE
014465  614351      7740            JMP     PSOK            AND EXIT
014466  000000      7742    ITEM0   0
014467  000000      7744    ITEM1   0
```

```
I                                   INTERPRETER -- IF

                    7750                ,STITL  INTERPRETER -- IF
                    7760    *
                    7770    *       THIS ROUTINE DETERMINES THE TRUTH OR FALSITY OF A
                    7780    *       CONDITIONAL
                    7790    *
    014470          7800    IF      ...;
    014470          7810            POPV                    GET THE LINE NUMBER
014473  054442      7820            DAC     NTEM            SAVE IT FOR THE TRANSFER ROUTINE
    014474          7830            POPV                    GET THE RIGHT RESULT
014477  054467      7840            DAC     ITEM1           SAVE IT
    014500          7850            POPV                    GET THE LEFT RESULT
014503  054466      7860            DAC     ITEM0           SAVE IT
    014504          7870            POP     0               ***
014505  054264      7880            DAC     PSTEMP          * BRANCH ON THE CONDITIONAL
014506  634264      7890            JMP     PSTEMP,X        ***
                    7920    *
                    7930    *       =
                    7940    *
    014507          7950    EQ      ...;
014507  214466      7960            LAC     ITEM0           GET THE LEFT HALF
014510  554467      7970            SAD     ITEM1           COMPARE
014511  614450      7980            JMP     GOTE            EQUAL
014512  614351      7990            JMP     PSOK            NOT EQUAL
                    8000    *
                    8010    *       <
                    8020    *
    014513          8030    LESS    ...;
014513  214467      8040            LAC     ITEM1           GET RIGHT RESULT
014514  740001      8050            CMA                     COMPLEMENT
014515  314466      8060            ADD     ITEM0           SUBTRACT FROM LEFT
014516  614522      8070            JMP     IFA             AND DO COMPARE
                    8080    *
                    8090    *       >
                    8100    *
    014517          8110    GTR     ...;
014517  214466      8120            LAC     ITEM0           GET LEFT RESULT
014520  740001      8130            CMA                     COMPLEMENT
014521  314467      8140            ADD     ITEM1           SUBTRACT FROM RIGHT
    014522          8150    IFA     ...;
014522  741100      8160            SPA                     COMPARE
014523  614450      8170            JMP     GOTE            TRUE
014524  614351      8180            JMP     PSOK            FALSE
                    8190    *
                    8200    *       <>
                    8210    *
    014525          8220    NEQ     ...;
014525  214467      8230            LAC     ITEM1           GET RIGHT RESULT
014526  540000      8240            SAD                     COMPARE
014527  614351      8250            JMP     PSOK            FALSE
014530  614450      8260            JMP     GOTE            TRUE
```

```
I                                   INTERPRETER -- STOP/END

                    8270            ,STITL  INTERPRETER -- STOP/END
                    8280    *
                    8290    *       THIS ROUTINE HANDLES A STOP OR END INSTRUCTION
                    8300    *
          014531    8310    END     ...
014531    612173    8320            JMP     TSMON           EXIT TO THE MONITOR
```

I INTERPRETER -- ASSIGN

```
                    8330            .STITL  INTERPRETER -- ASSIGN
                    8340    *
                    8350    *       THIS ROUTINE IS CALLED TO ASSIGN THE VALUE OF
                    8360    *       THE THING POINTED TO BY THE TOP ELEMENT OF THE
                    8370    *       VARIABLE STACK TO THE NEXT TO TOP ELEMENT OF THE STACK.
                    8380    *
     014532         8390    ASIGN   ..;
     014532         8400            POP     V               GET THE POINTER FROM WHICH TO ASSIGN
 014533  054264     8410            DAC     PSTEMP          SAVE IT
     014534         8420            POP     V               GET THE POINTER TO ASSIGN
 014535  054442     8430            DAC     NTEM            SAVE IT
 014536  234264     8440            LAC     PSTEMP,X        GET THE VALUE
 014537  074442     8450            DAC     NTEM,X          REPLACE IT
 014540  614351     8460            JMP     PSOK            AND EXIT
```

```
I                       INTERPRETER -- UNARY MINUS

                8470            .STITL  INTERPRETER -- UNARY MINUS
                8480    *
                8490    *       THIS ROUTINE NEGATES THE TOP DATUM OF THE VARIABLE
                8500    *       STACK.  IT ALSO MAPS -0 INTO PLUS 0
                8510    *
    014541      8520    UNMIN   ...
014541  114165  8530            JMS     SSTEMP          GET A TEMPORARY FOR THE RESULT
014542  054442  8540            DAC     NTEM            SAVE THE POINTER
    014543      8550            POPV                    POP THE VARIABLE
014546  740200  8560            SZA                     IF TWO'S ZERO -- DON'T NEGATE
014547  740001  8570            CMA                     NEGATE
014550  074442  8580            DAC     NTEM,X          SAVE THE NEW VALUE
014551  214442  8590            LAC     NTEM            GET THE REFERENCE
    014552      8600            PUSH    V               AND PUSH IT
014553  614351  8610            JMP     PSOK            EXIT
```

```
                I                       INTERPRETER -- ADD

                        8620            .STITL  INTERPRETER -- ADD
                        8630    *
                        8640    *       THIS ROUTINE FORMS THE SUM OF THE TOP 1 OR TWO OBJECTS
                        8650    *       ON THE VARIABLE STACK.  IT DECIDES HOW MANY AND
                        8660    *       WHETHER TO ADD OR SUBTRACT BASED ON THE TOP OF THE OPERATOR
                        8670    *       STACK.
                        8680    *
    014554              8690    ADD     ..;
014554  114755          8700            JMS     ITER            SEE IF WE'VE ITERATED
014555  015150          8710            .DATA   S$STAG,
014556  114165          8720            JMS     S$TEMP          GET A TEMPORARY FOR THE RESULT
014557  054442          8730            DAC     NTEM            SAVE THE POINTER
    014560              8740            POP     0               GET THE OPERATION
014561  741200          8750            SNA                     SEE IF + OR -
014562  614571          8760            JMP     AD1             PLUS
    014563              8770            POPV                    GET THE SECOND TERM
014566  740200          8780            SZA                     DON'T NEGATE IF ZERO
014567  740001          8790            CMA                     NEGATE IT
014570  614574          8800            JMP     AD2
    014571              8810    AD1     ..;
    014571              8820            POPV                    GET THE SECOND TERM
    014574              8830    AD2     ..;
014574  074442          8840            DAC     NTEM,X          SAVE IT
    014575              8850            POPV                    GET FIRST TERM
014600  334442          8860            ADD     NTEM,X          ADD THEM
    014601              8870    ARTHX   ..;
014601  074442          8880            DAC     NTEM,X          SAVE THE RESULT
014602  214442          8890            LAC     NTEM            GET POINTER
    014603              8900            PUSH    V               AND STACK IT
014604  614266          8910            JMP     PSTEST          AND EXIT
```

I INTERPRETER -- MULTIPLICATION

```
                    8920            .STITL  INTERPRETER -- MULTIPLICATION
                    8930    *
                    8940    *       THIS ROUTINE IS CALLED TO COMPUTE THE PRODUCT OF THE
                    8950    *       LAST N OBJECTS ON THE  VARIABLE STACK
                    8960    *
        014605      8970    MULT    ..,
014605  114755      8980            JMS     ITER            THIS MAY BE ITERATED
014606  015176      8990    MUL.    .DATA   SSPTAG.
014607  114165      9000            JMS     SSTEMP          ASSIGN A TEMPORARY
014610  054442      9010            DAC     NTEM            SAVE THE POINTER
        014611      9020            POP     0               GET  THE OPERATION
014612  315770      9030            ADD     (INST)          ***
014613  054264      9040            DAC     PSTEMP          * GET THE ACTUAL INSTRUCTION
014614  234264      9050            LAC     PSTEMP,X        ***
014615  054632      9060            DAC     MULY-1          SAVE TO EXECUTE
        014616      9070            POPV                    GET THE MULTIPLIER/DIVISOR
014621  054264      9080            DAC     PSTEMP          SAVE IT
        014622      9090            POPV                    GET THE MULTIPLICAND/DIVIDEND
014625  074442      9100            DAC     NTEM,X          SAVE IT
014626  214264      9110            LAC     PSTEMP          RESTORE MULTIPLIER/DIVISOR
014627  664000      9120            GSM                     GET ITS SIGN
014630  054633      9130            DAC     MULY            SAVE IT FOR THE OPERATION
014631  234442      9140            LAC     NTEM,X          RESTORE MULTIPLICAND/DIVIDENT
014632  657122      9150            MULS                    MULTIPLY/DIVIDE
014633  000000      9160    MULY    .DATA   0               MULTIPLIER/DIVISOR
014634  641002      9170            LACQ                    GET THE RESULT
014635  614601      9180            JMP     ARTHX           AND EXIT
014636  657122      9190    INST    MULS                    MULTIPLY INSTRUCTION
014637  657323      9200            IDIVS                   DIVIDE INSTRUCTION
```

```
                I                       INTERPRETER -- EXPONENTIATION

                     9210               .STITL  INTERPRETER -- EXPONENTIATION
                     9220       *
                     9230       *       THIS ROUTINE IS CALLED TO COMPUTE POWERS
                     9240       *
          014640     9250       EXP     ..,
014640    114755     9260               JMS     ITER            THIS MAY BE ITERATED
014641    015224     9270       EXP.    .DATA   SSFTAG,
```

I INTERPRETER -- STACK AN OPERATOR

```
                    9280            .STITL  INTERPRETER -- STACK AN OPERATOR
                    9290    *
                    9300    *       THIS ROUTINE IS CALLED TO STACK AN OPERATOR ONTO
                    9310    *       THE OPERATOR STACK.
                    9320    *
    014642          9330    INSTK   ...
014642  454647      9340            ISZ     OP              SET FLAG FOR AN INVERSE OPERATION
    014643          9350    OPSTK   ...
014643  214647      9360            LAC     OP              GET THE OPERATION CODE
    014644          9370            PUSH    0               PUSH IT
014645  154647      9380            DZM     OP              CLEAR THE CODE BACK TO 0
014646  614351      9390            JMP     PSOK            AND EXIT
014647  000000      9400    OP      .DATA   0               OPERATOR CODE
```

```
                I                       INTERPRETER -- STACK A RELATIONAL

                        9410            .STITL  INTERPRETER -- STACK A RELATIONAL
                        9420    *
                        9430    *       THIS ROUTINE IS USED TO STACK THE RELATIONAL OPERATORS
                        9440    *
014650  454660  9450    RNEQ    ISZ     ROP             NOT EQUAL
014651  454660  9460    RGTR    ISZ     ROP             GREATER (>)
014652  454660  9470    RLES    ISZ     ROP             LESS (<)
    014653      9480    REQ     ...
014653  234660  9490            LAC     ROP,X           GET THE OPERATOR
    014654      9500            PUSH    0               PUSH ON THE OPERATOR STACK
014655  215771  9510            LAC     (RTAB)          REINITIALIZE THE POINTER
014656  054660  9520            DAC     ROP             REPLACE
014657  614351  9530            JMP     PSOK            AND EXIT
014660  014661  9540    ROP     .DATA   .+1             RELATIONAL OPERATOR
    014661      9550    RTAB    ...
014661  014507  9560            .DATA   EQ              EQUAL
014662  014513  9570            .DATA   LESS            LESS
014663  014517  9580            .DATA   GTR             GREATER
014664  014525  9590            .DATA   NEQ             NOT EQUAL
```

I INTERPRETER -- STACK A VARIABLE

```
                9600            .STITL  INTERPRETER -- STACK A VARIABLE
                9610    *
                9620    *       THIS ROUTINE IS CALLED TO PLACE A VARIABLE ON THE
                9630    *       VARIABLE STACK, MAPPING IT APPROPRIATELY.
                9640    *
    014665      9650    VSTK    ...
    014665      9660            LCHAR                   GET THE VARIABLE NAME
914666  355772  9670            TAD     (VTAB-101)      MAKE POINT TO THE TABLE
    014667      9680            PUSH    V               PUSH ONTO THE STACK
014670  614351  9690            JMP     PSOK            AND EXIT
    014671      9700    VTAB    .BLOCK  26.             THE VARIABLE AREA FOR NOW
```

I INTERPRETER -- STACK A DIGIT

```
                    9710            .STITL  INTERPRETER -- STACK A DIGIT
                    9720    *
                    9730    *       THIS ROUTINE IS CALLED TO STACK A DIGIT ON THE VARIABLE
                    9740    *       STACK.  IT DIFFERS FROM VSTK ONLY IN THAT IT DOESN'T MAP
                    9750    *       THINGS
                    9760    *
          014723    9770    DSTK    ...
          014723    9780            LCHAR                   GET THE DIGIT
          014724    9790            PUSH    V               STACK IT
    014725  614351  9800            JMP     PSDK            AND EXIT
```

```
I                                   INTERPRETER -- EVALUATE A NUMBER

                    9810            ,STITL  INTERPRETER -- EVALUATE A NUMBER
                    9820    *
                    9830    *       THIS ROUTINE EVALUATES THE LAST N DIGITS ON THE
                    9840    *       VARIABLE STACK AS A NUMBER
                    9850    *
    014726          9860    EVAL3   ...;
014726  114165      9870            JMS     SSTEMP          ***
014727  054264      9880            DAC     PSTEMP          * GET A TEMPORARY CELL AND STACK POINTER
    014730          9890            PUSH    V               ***
014731  214442      9900            LAC     NTEM            ***
014732  074264      9910            DAC     PSTEMP,X        * SAVE THE VALUE IN THE TEMPORARY
014733  614351      9920            JMP     PSOK            ***
                    9930    *
                    9940    *       THIS ROUTINE IS FOR THE FIRST DIGIT
                    9950    *
    014734          9960    EVAL1   ...;
    014734          9970            POP     V               ***
014735  515773      9980            AND     (17)            * GET AND SAVE THE FIRST DIGIT
014736  054442      9990            DAC     NTEM            ***
014737  614351      10000           JMP     PSOK
                    10010   *
                    10020   *       THIS ROUTINE IS FOR ALL OTHER DIGITS
                    10030   *
    014740          10040   EVAL2   ...;
014740  114755      10050           JMS     ITER            SEE IF WE ARE ITERATING
014741  015305      10060           ,DATA   SSDTAG,
014742  744000      10070           CLL                     CLEAR THE LINK
014743  214442      10080           LAC     NTEM            ***
014744  653122      10090           MUL                     * SHIFT PREVIOUS TOTAL OVER FOR NEW ONE
014745  000012      10100           10.                     ***
    014746          10110           POP     V               GET THE DIGIT
014747  515773      10120           AND     (17)            MASK TO THE DIGIT
014750  054442      10130           DAC     NTEM            ***
014751  641002      10140           LACQ                    * ADD THE PREVIOUS TO THE NEW
014752  314442      10150           ADD     NTEM            ***
014753  054442      10160           DAC     NTEM            RESTORE
014754  614266      10170           JMP     PSTEST          AND TRY AGAIN
```

```
                I                       INTERPRETER -- SEE IF A FIELD IS ITERATED

                        10180           .STITL  INTERPRETER -- SEE IF A FIELD IS ITERATED
                        10190   *
                        10200   *       THIS ROUTINE CHECKS FOR AN ITERATED FIELD BY
                        10210   *       SEEING IF THE LAST MATCH WAS <EMPTY>
                        10220   *
    014755  000000      10230   ITER    0
    014756  214265      10240           LAC     PSLPART         GET THE LAST PART MATCHED
    014757  574755      10250           SAD     ITER,X          SEE IF WAS EMPTY OF THING WE TRIED FOR
    014760  614351      10260           JMP     PSOK            YES -- NO MORE
        014761          10270           SOS     PSPART          ELSE SUBTRACT ONE FROM PART
    014764  777774      10280           LAW     -4              ***
    014765  353330      10290           TAD     RSSDB           * GET  A POINTER TO THE SOURCE POINTER IN THE STACK
    014766  054264      10300           DAC     PSTEMP          ***
    014767  200011      10310           LAC     TSCHRX          GET THE CURRENT SOURCE POINTER
    014770  074264      10320           DAC     PSTEMP,X        FUDGE THE STACK
    014771  214755      10330           LAC     ITER            ***
    014772  040012      10340           DAC     X               * RETURN
    014773  620012      10350           RET     X               ***
```

```
                I                       INTERPRETER -- EXIT

                    10360               .STITL  INTERPRETER -- EXIT
                    10370   *
                    10380   *           THIS ROUTINE IS A TEMPORARY COMMON EXIT FOR
                    10390   *           THE INTERPRETER.
                    10400   *
          014774    10410   EXIT        .,
014774    111556    10420               JMS     T$MESS
014775    000015    10430               .DATA   15,12,-1
014776    000012
014777    777777
015000    614351    10440               JMP     PSOK            AND REALLY EXIT
                    10450               .HEAD   S               S FOR SYNTAX
```

```
S                                   DEFINITION -- SYNTAX TABLE

                    10460           ,STITL  DEFINITION -- SYNTAX TABLE
                    10470   *
                    10480   *       THIS TABLE IS DERIVED FROM A BNF SYNTAX FOR A SUBSET
                    10490   *       OF BASIC.  IT IS COMPOSED OF ENTRIES IN THE FOLLOWING
                    10500   *       FORMAT:
                    10510   *
                    10520   *       WORD 0   POINTER TO NEXT ALTERNATIVE DEFINITION
                    10530   *                0 INDICATES THAT THIS IS THE LAST
                    10540   *       WORD 1   NUMBER OF PARTS TO THIS ALTERNATIVE
                    10550   *       WORD 2
                    10560   *         ,      PARTS -- POINTERS TO OTHER DEFINITIONS OR LITERAL
                    10570   *         ,      CHARACTERS.  LITERALS FLAGGED NEGATIVE.
                    10580   *       WORD N
                    10590   *       WORD N+1 POINTER TO ROUTINE ON SUCCESSFUL RECOGNITION.
                    10600   *
                    10610   *       DEFINITION -- SPECIAL SYNTAX SYMBOLS
                    10620   *
    400015          10630   EOL     ,EQU    400015          END OF LINE -- LITERAL CR
    777777          10640   EMPTY   ,EQU    777777          EMPTY -- A DISTINCTIVE LITERAL
                    10650           ,EOT    BASSYN
    015001          9000    STATE   ,,,
015001  015005      9010            ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015002  015004      9020            ,DATA   .+2             NUMBER OF PARTS
015003  015025      9030            ,DATA   LETS
015004  014351      9040            ,DATA   PSOK
015005  015011      9050            ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015006  015010      9060            ,DATA   .+2             NUMBER OF PARTS
015007  015034      9070            ,DATA   PRINT
015010  014351      9080            ,DATA   PSOK
015011  015015      9090            ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015012  015014      9100            ,DATA   .+2             NUMBER OF PARTS
015013  015046      9110            ,DATA   GOTO
015014  014351      9120            ,DATA   PSOK
015015  015021      9130            ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015016  015020      9140            ,DATA   .+2             NUMBER OF PARTS
015017  015057      9150            ,DATA   IF
015020  014351      9160            ,DATA   PSOK
015021  000000      9170            ,DATA   0               LAST ALTERNATIVE
015022  015024      9180            ,DATA   .+2             NUMBER OF PARTS
015023  015116      9190            ,DATA   END
015024  014351      9200            ,DATA   PSOK
    015025          9210    LETS    ,,,
015025  000000      9220            ,DATA   0               LAST ALTERNATIVE
015026  015033      9230            ,DATA   .+5             NUMBER OF PARTS
015027  400114      9240            ,DATA   400114
015030  400105      9250            ,DATA   400105
015031  400124      9260            ,DATA   400124
015032  015125      9270            ,DATA   ASIGN
015033  014351      9280            ,DATA   PSOK
    015034          9290    PRINT   ,,,
015034  000000      9300            ,DATA   0               LAST ALTERNATIVE
015035  015045      9310            ,DATA   .+8             NUMBER OF PARTS
```

S DEFINITION -- SYNTAX TABLE

```
015036  400120  9320            ,DATA   400120
015037  400122  9330            ,DATA   400122
015040  400111  9340            ,DATA   400111
015041  400116  9350            ,DATA   400116
015042  400124  9360            ,DATA   400124
015043  015134  9370            ,DATA   SUM
015044  400015  9380            ,DATA   EOL
015045  014364  9390            ,DATA   ISPRINT
    015046      9400    GOTO    ...,
015046  000000  9410            ,DATA   0               LAST ALTERNATIVE
015047  015056  9420            ,DATA   .+7             NUMBER OF PARTS
015050  400107  9430            ,DATA   400107
015051  400117  9440            ,DATA   400117
015052  400124  9450            ,DATA   400124
015053  400117  9460            ,DATA   400117
015054  015266  9470            ,DATA   NUM
015055  400015  9480            ,DATA   EOL
015056  014444  9490            ,DATA   ISGOTO
    015057      9500    IF      ...,
015057  000000  9510            ,DATA   0               LAST ALTERNATIVE
015060  015074  9520            ,DATA   .+14            NUMBER OF PARTS
015061  400111  9530            ,DATA   400111
015062  400106  9540            ,DATA   400106
015063  015134  9550            ,DATA   SUM
015064  015075  9560            ,DATA   RELOP
015065  015134  9570            ,DATA   SUM
015066  400107  9580            ,DATA   400107
015067  400117  9590            ,DATA   400117
015070  400124  9600            ,DATA   400124
015071  400117  9610            ,DATA   400117
015072  015266  9620            ,DATA   NUM
015073  400015  9630            ,DATA   EOL
015074  014470  9640            ,DATA   ISIF
    015075      9650    RELOP   ...,
015075  015101  9660            ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015076  015100  9670            ,DATA   .+2             NUMBER OF PARTS
015077  400075  9680            ,DATA   400075
015100  014653  9690            ,DATA   ISREQ
015101  015106  9700            ,DATA   .+5             POINTER TO NEXT ALTERNATIVE
015102  015105  9710            ,DATA   .+3             NUMBER OF PARTS
015103  400074  9720            ,DATA   400074
015104  400076  9730            ,DATA   400076
015105  014650  9740            ,DATA   ISRNEQ
015106  015112  9750            ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015107  015111  9760            ,DATA   .+2             NUMBER OF PARTS
015110  400074  9770            ,DATA   400074
015111  014652  9780            ,DATA   ISRLES
015112  000000  9790            ,DATA   0               LAST ALTERNATIVE
015113  015115  9800            ,DATA   .+2             NUMBER OF PARTS
015114  400076  9810            ,DATA   400076
015115  014651  9820            ,DATA   ISRGTR
    015116      9830    END     ...
```

S DEFINITION -- SYNTAX TABLE

```
015116  000000  9840            ,DATA   0               LAST ALTERNATIVE
015117  015124  9850            ,DATA   .+5             NUMBER OF PARTS
015120  400105  9860            ,DATA   400105
015121  400116  9870            ,DATA   400116
015122  400104  9880            ,DATA   400104
015123  400015  9890            ,DATA   EOL
015124  014531  9900            ,DATA   ISEND
    015125      9910    ASIGN   ...
015125  000000  9920            ,DATA   0               LAST ALTERNATIVE
015126  015133  9930            ,DATA   .+5             NUMBER OF PARTS
015127  015262  9940            ,DATA   VAR
015130  400075  9950            ,DATA   400075
015131  015134  9960            ,DATA   SUM
015132  400015  9970            ,DATA   EOL
015133  014532  9980            ,DATA   ISASIGN
    015134      9990    SUM     ...
015134  000000  10000           ,DATA   0               LAST ALTERNATIVE
015135  015140  10010           ,DATA   .+3             NUMBER OF PARTS
015136  015162  10020           ,DATA   PROD
015137  015141  10030           ,DATA   STAG
015140  014554  10040           ,DATA   ISADD
    015141      10050   STAG    ...
015141  015146  10060           ,DATA   .+5             POINTER TO NEXT ALTERNATIVE
015142  015145  10070           ,DATA   .+3             NUMBER OF PARTS
015143  015152  10080           ,DATA   ADDOP
015144  015162  10090           ,DATA   PROD
015145  014351  10100           ,DATA   PSOK
015146  000000  10110           ,DATA   0               LAST ALTERNATIVE
015147  015151  10120           ,DATA   .+2             NUMBER OF PARTS
015150  777777  10130   STAG,   ,DATA   EMPTY
015151  014351  10140           ,DATA   PSOK
    015152      10150   ADDOP   ...
015152  015156  10160           ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015153  015155  10170           ,DATA   .+2             NUMBER OF PARTS
015154  400053  10180           ,DATA   400053
015155  014643  10190           ,DATA   ISOPSTK
015156  000000  10200           ,DATA   0               LAST ALTERNATIVE
015157  015161  10210           ,DATA   .+2             NUMBER OF PARTS
015160  400055  10220           ,DATA   400055
015161  014642  10230           ,DATA   ISINSTK
    015162      10240   PROD    ...
015162  000000  10250           ,DATA   0               LAST ALTERNATIVE
015163  015166  10260           ,DATA   .+3             NUMBER OF PARTS
015164  015210  10270           ,DATA   FACT
015165  015167  10280           ,DATA   PTAG
015166  014605  10290           ,DATA   ISMULT
    015167      10300   PTAG    ...
015167  015174  10310           ,DATA   .+5             POINTER TO NEXT ALTERNATIVE
015170  015173  10320           ,DATA   .+3             NUMBER OF PARTS
015171  015200  10330           ,DATA   MLOP
015172  015210  10340           ,DATA   FACT
015173  014351  10350           ,DATA   PSOK
```

S DEFINITION -- SYNTAX TABLE

```
015174  000000  10360           ,DATA   0               LAST ALTERNATIVE
015175  015177  10370           ,DATA   .+2             NUMBER OF PARTS
015176  777777  10380   PTAG,   ,DATA   EMPTY
015177  014351  10390           ,DATA   PSOK
    015200      10400   MLOP    ...
015200  015204  10410           ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015201  015203  10420           ,DATA   .+2             NUMBER OF PARTS
015202  400052  10430           ,DATA   400052
015203  014643  10440           ,DATA   ISOPSTK
015204  000000  10450           ,DATA   0               LAST ALTERNATIVE
015205  015207  10460           ,DATA   .+2             NUMBER OF PARTS
015206  400057  10470           ,DATA   400057
015207  014642  10480           ,DATA   ISINSTK
    015210      10490   FACT    ...
015210  000000  10500           ,DATA   0               LAST ALTERNATIVE
015211  015214  10510           ,DATA   .+3             NUMBER OF PARTS
015212  015232  10520           ,DATA   SAE
015213  015215  10530           ,DATA   FTAG
015214  014640  10540           ,DATA   ISEXP
    015215      10550   FTAG    ...
015215  015222  10560           ,DATA   .+5             POINTER TO NEXT ALTERNATIVE
015216  015221  10570           ,DATA   .+3             NUMBER OF PARTS
015217  015226  10580           ,DATA   EXPOP
015220  015232  10590           ,DATA   SAE
015221  014351  10600           ,DATA   PSOK
015222  000000  10610           ,DATA   0               LAST ALTERNATIVE
015223  015225  10620           ,DATA   .+2             NUMBER OF PARTS
015224  777777  10630   FTAG,   ,DATA   EMPTY
015225  014351  10640           ,DATA   PSOK
    015226      10650   EXPOP   ...
015226  000000  10660           ,DATA   0               LAST ALTERNATIVE
015227  015231  10670           ,DATA   .+2             NUMBER OF PARTS
015230  400136  10680           ,DATA   400136
015231  014643  10690           ,DATA   ISOPSTK
    015232      10700   SAE     ...
015232  015240  10710           ,DATA   .+6             POINTER TO NEXT ALTERNATIVE
015233  015237  10720           ,DATA   .+4             NUMBER OF PARTS
015234  400050  10730           ,DATA   400050
015235  015134  10740           ,DATA   SUM
015236  400051  10750           ,DATA   400051
015237  014351  10760           ,DATA   PSOK
015240  015244  10770           ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015241  015243  10780           ,DATA   .+2             NUMBER OF PARTS
015242  015262  10790           ,DATA   VAR
015243  014351  10800           ,DATA   PSOK
015244  015250  10810           ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015245  015247  10820           ,DATA   .+2             NUMBER OF PARTS
015246  015266  10830           ,DATA   NUM
015247  014351  10840           ,DATA   PSOK
015250  015255  10850           ,DATA   .+5             POINTER TO NEXT ALTERNATIVE
015251  015254  10860           ,DATA   .+3             NUMBER OF PARTS
015252  400055  10870           ,DATA   400055
```

S DEFINITION -- SYNTAX TABLE

```
015253  015232  10880           ,DATA   SAE
015254  014541  10890           ,DATA   ISUNMIN
015255  000000  10900           ,DATA   0               LAST ALTERNATIVE
015256  015261  10910           ,DATA   .+3             NUMBER OF PARTS
015257  400053  10920           ,DATA   400053
015260  015232  10930           ,DATA   SAE
015261  014351  10940           ,DATA   PSOK
    015262      10950   VAR     ...
015262  000000  10960           ,DATA   0               LAST ALTERNATIVE
015263  015265  10970           ,DATA   .+2             NUMBER OF PARTS
015264  015357  10980           ,DATA   LET
015265  014665  10990           ,DATA   ISVSTK
    015266      11000   NUM     ...
015266  000000  11010           ,DATA   0               LAST ALTERNATIVE
015267  015272  11020           ,DATA   .+3             NUMBER OF PARTS
015270  015273  11030           ,DATA   DHEAD
015271  015277  11040           ,DATA   DTAG
015272  014726  11050           ,DATA   ISEVAL3
    015273      11060   DHEAD   ...
015273  000000  11070           ,DATA   0               LAST ALTERNATIVE
015274  015276  11080           ,DATA   .+2             NUMBER OF PARTS
015275  015307  11090           ,DATA   DIGIT
015276  014734  11100           ,DATA   ISEVAL1
    015277      11110   DTAG    ...
015277  015303  11120           ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015300  015302  11130           ,DATA   .+2             NUMBER OF PARTS
015301  015307  11140           ,DATA   DIGIT
015302  014740  11150           ,DATA   ISEVAL2
015303  000000  11160           ,DATA   0               LAST ALTERNATIVE
015304  015306  11170           ,DATA   .+2             NUMBER OF PARTS
015305  777777  11180   DTAG,   ,DATA   EMPTY
015306  014351  11190           ,DATA   PSOK
    015307      11200   DIGIT   ...
015307  015313  11210           ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015310  015312  11220           ,DATA   .+2             NUMBER OF PARTS
015311  400060  11230           ,DATA   400060
015312  014723  11240           ,DATA   ISDSTK
015313  015317  11250           ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015314  015316  11260           ,DATA   .+2             NUMBER OF PARTS
015315  400061  11270           ,DATA   400061
015316  014723  11280           ,DATA   ISDSTK
015317  015323  11290           ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015320  015322  11300           ,DATA   .+2             NUMBER OF PARTS
015321  400062  11310           ,DATA   400062
015322  014723  11320           ,DATA   ISDSTK
015323  015327  11330           ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015324  015326  11340           ,DATA   .+2             NUMBER OF PARTS
015325  400063  11350           ,DATA   400063
015326  014723  11360           ,DATA   ISDSTK
015327  015333  11370           ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015330  015332  11380           ,DATA   .+2             NUMBER OF PARTS
015331  400064  11390           ,DATA   400064
```

S DEFINITION -- SYNTAX TABLE

```
015332  014723  11400          ,DATA   I$DSTK
015333  015337  11410          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015334  015336  11420          ,DATA   .+2             NUMBER OF PARTS
015335  400065  11430          ,DATA   400065
015336  014723  11440          ,DATA   I$DSTK
015337  015343  11450          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015340  015342  11460          ,DATA   .+2             NUMBER OF PARTS
015341  400066  11470          ,DATA   400066
015342  014723  11480          ,DATA   I$DSTK
015343  015347  11490          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015344  015346  11500          ,DATA   .+2             NUMBER OF PARTS
015345  400067  11510          ,DATA   400067
015346  014723  11520          ,DATA   I$DSTK
015347  015353  11530          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015350  015352  11540          ,DATA   .+2             NUMBER OF PARTS
015351  400070  11550          ,DATA   400070
015352  014723  11560          ,DATA   I$DSTK
015353  000000  11570          ,DATA   0               LAST ALTERNATIVE
015354  015356  11580          ,DATA   .+2             NUMBER OF PARTS
015355  400071  11590          ,DATA   400071
015356  014723  11600          ,DATA   I$DSTK
    015357      11610  LET     ...
015357  015363  11620          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015360  015362  11630          ,DATA   .+2             NUMBER OF PARTS
015361  400101  11640          ,DATA   400101
015362  014351  11650          ,DATA   P$OK
015363  015367  11660          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015364  015366  11670          ,DATA   .+2             NUMBER OF PARTS
015365  400102  11680          ,DATA   400102
015366  014351  11690          ,DATA   P$OK
015367  015373  11700          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015370  015372  11710          ,DATA   .+2             NUMBER OF PARTS
015371  400103  11720          ,DATA   400103
015372  014351  11730          ,DATA   P$OK
015373  015377  11740          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015374  015376  11750          ,DATA   .+2             NUMBER OF PARTS
015375  400104  11760          ,DATA   400104
015376  014351  11770          ,DATA   P$OK
015377  015403  11780          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015400  015402  11790          ,DATA   .+2             NUMBER OF PARTS
015401  400105  11800          ,DATA   400105
015402  014351  11810          ,DATA   P$OK
015403  015407  11820          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015404  015406  11830          ,DATA   .+2             NUMBER OF PARTS
015405  400106  11840          ,DATA   400106
015406  014351  11850          ,DATA   P$OK
015407  015413  11860          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015410  015412  11870          ,DATA   .+2             NUMBER OF PARTS
015411  400107  11880          ,DATA   400107
015412  014351  11890          ,DATA   P$OK
015413  015417  11900          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015414  015416  11910          ,DATA   .+2             NUMBER OF PARTS
```

S DEFINITION -- SYNTAX TABLE

```
015415  400110  11920          ,DATA   400110
015416  014351  11930          ,DATA   PSOK
015417  015423  11940          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015420  015422  11950          ,DATA   .+2             NUMBER OF PARTS
015421  400111  11960          ,DATA   400111
015422  014351  11970          ,DATA   PSOK
015423  015427  11980          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015424  015426  11990          ,DATA   .+2             NUMBER OF PARTS
015425  400112  12000          ,DATA   400112
015426  014351  12010          ,DATA   PSOK
015427  015433  12020          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015430  015432  12030          ,DATA   .+2             NUMBER OF PARTS
015431  400113  12040          ,DATA   400113
015432  014351  12050          ,DATA   PSOK
015433  015437  12060          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015434  015436  12070          ,DATA   .+2             NUMBER OF PARTS
015435  400114  12080          ,DATA   400114
015436  014351  12090          ,DATA   PSOK
015437  015443  12100          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015440  015442  12110          ,DATA   .+2             NUMBER OF PARTS
015441  400115  12120          ,DATA   400115
015442  014351  12130          ,DATA   PSOK
015443  015447  12140          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015444  015446  12150          ,DATA   .+2             NUMBER OF PARTS
015445  400116  12160          ,DATA   400116
015446  014351  12170          ,DATA   PSOK
015447  015453  12180          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015450  015452  12190          ,DATA   .+2             NUMBER OF PARTS
015451  400117  12200          ,DATA   400117
015452  014351  12210          ,DATA   PSOK
015453  015457  12220          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015454  015456  12230          ,DATA   .+2             NUMBER OF PARTS
015455  400120  12240          ,DATA   400120
015456  014351  12250          ,DATA   PSOK
015457  015463  12260          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015460  015462  12270          ,DATA   .+2             NUMBER OF PARTS
015461  400121  12280          ,DATA   400121
015462  014351  12290          ,DATA   PSOK
015463  015467  12300          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015464  015466  12310          ,DATA   .+2             NUMBER OF PARTS
015465  400122  12320          ,DATA   400122
015466  014351  12330          ,DATA   PSOK
015467  015473  12340          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015470  015472  12350          ,DATA   .+2             NUMBER OF PARTS
015471  400123  12360          ,DATA   400123
015472  014351  12370          ,DATA   PSOK
015473  015477  12380          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015474  015476  12390          ,DATA   .+2             NUMBER OF PARTS
015475  400124  12400          ,DATA   400124
015476  014351  12410          ,DATA   PSOK
015477  015503  12420          ,DATA   .+4             POINTER TO NEXT ALTERNATIVE
015500  015502  12430          ,DATA   .+2             NUMBER OF PARTS
```

S DEFINITION -- SYNTAX TABLE

```
015501  400125  12440          .DATA   400125
015502  014351  12450          .DATA   PSOK
015503  015507  12460          .DATA   .+4             POINTER TO NEXT ALTERNATIVE
015504  015506  12470          .DATA   .+2             NUMBER OF PARTS
015505  400126  12480          .DATA   400126
015506  014351  12490          .DATA   PSOK
015507  015513  12500          .DATA   .+4             POINTER TO NEXT ALTERNATIVE
015510  015512  12510          .DATA   .+2             NUMBER OF PARTS
015511  400127  12520          .DATA   400127
015512  014351  12530          .DATA   PSOK
015513  015517  12540          .DATA   .+4             POINTER TO NEXT ALTERNATIVE
015514  015516  12550          .DATA   .+2             NUMBER OF PARTS
015515  400130  12560          .DATA   400130
015516  014351  12570          .DATA   PSOK
015517  015523  12580          .DATA   .+4             POINTER TO NEXT ALTERNATIVE
015520  015522  12590          .DATA   .+2             NUMBER OF PARTS
015521  400131  12600          .DATA   400131
015522  014351  12610          .DATA   PSOK
015523  000000  12612          .DATA   0               LAST ALTERNATIVE
015524  015526  12614          .DATA   .+2             NUMBER OF PARTS
015525  400132  12616          .DATA   400132
015526  014351  12618          .DATA   PSOK
                12620          .EOT    BA$2
                12000          .HEAD   E               E FOR ERROR
                12010          .PMC    SAVE,OFF
```

```
E                                   ERROR MESSAGE ROUTINES

                    12020           .SITL  ERROR MESSAGE ROUTINES
                    12030   *
                    12040   *       THESE ROUTINES SEND ERROR MESSAGES FOR BOTH USER
                    12050   *       AND BASIC GENERATED ERRORS.
                    12060   *
    015527          12070   ERROR   ...
    015527          12071           MESS    < AT LINE >
015542  212240      12072           LAC     TSLNUM          GET THE LINE NUMBER
015543  114375      12073           JMS     ISPRT           PRINT IT
015544  111556      12080           JMS     TSMESS          ***
015545  000012      12090           .DATA   12,15           * SEND A CR/LF
015546  000015
015547  777777      12100           777777                  ***
015550  612173      12110           JMP     TSMON           AND EXIT TO THE MONITOR
                    12120   *
                    12130   *       PARSER ERROR
                    12140   *
    015551          12150   PARSE   ...
    015551          12160           MESS    <PARSER ERROR>
015567  615527      12170           JMP     ERROR           EXIT
                    12180   *
                    12190   *       STACK OVERFLOW
                    12200   *
    015570          12210   SOVF    ...
    015570          12220           MESS    <EXPRESSION TOO COMPLICATED>
015624  615527      12230           JMP     ERROR           EXIT
                    12240   *
                    12250   *       USER FAILED
                    12260   *
    015625          12270   FAIL    ...
    015625          12280           MESS    <SYNTAX ERROR>
015643  615527      12290           JMP     ERROR           EXIT
                    12300   *
                    12310   *       END IS NOT LAST
                    12320   *
    015644          12330   NEND    ...
    015644          12340           MESS    <END IS NOT LAST>
015665  615527      12350           JMP     ERROR           EXIT
                    12360   *
                    12370   *       UNDEFINED LINE NUMBER
                    12380   *
    015666          12390   UND     ...
    015666          12400           MESS    <UNDEFINED LINE NUMBER>
015715  615527      12410           JMP     ERROR
                    12420           .HEAD   0
```

RUN TIME INITIALIZATION

```
                    12430           .STITL  RUN TIME INITIALIZATION
                    12440   *
                    12450   *       THIS ROUTINE INITIALIZED THE INTERPRETER
                    12460   *
    015716          12470   UP      ...
015716  700416      12480           TLS     10              MAKE THE TTY FLAG COME UP
    015717          12482           MESS    <>,CRLF         GET A FRESH LINE
    015724          12484           MESS    <>,CRLF         AND ANOTHER
    015731          12490           MESS    <BASIC HERE>,CRLF
015747  612173      12500           JMP     TSMON           AND START UP THE PROGRAM
                    12510           .PMC    RESTORE
015750  000030      12520           .LIT
015751  000137
015752  000005
015753  000015
015754  011716
015755  000012
015756  000040
015757  015000
015760  012240
015761  000122
015762  000114
015763  015001
015764  015003
015765  000002
015766  777777
015767  000060
015770  014636
015771  014661
015772  014570
015773  000017
015774  000000
015775  000000
015776  000000
    015777          12530   THIS    .END    SUP

TRANSFER ADDRESS  615716
```

CROSS REFERENCE TABLE

13315	DATUM	4830	4510	4680	4710	4750	4770	4070	4360	4650			
15666	E UND	12390	7670										
15625	E FAIL	12270	6420										
15604	E NEND	12330	5230										
15570	E SOVF	12210	4500										
15527	EERROR	12070	12170	12230	12290	12350	12410						
15551	EPARSE	12150	2110	4730									
12	I X	300	7610	7630	7650	7680	7720	10340	10350				
13	I Y	310											
14	I Z	320											
14507	I EQ	7950	9560										
14470	I IF	7800	9640										
14647	I OP	9400	9340	9360	9380								
14571	I AD1	8810	8760										
14574	I AD2	8830	8800										
14554	I ADD	8690	10040										
14531	I END	8310	9900										
14640	I EXP	9250	10540										
14547	I GTR	8110	9580										
14522	I IFA	8150	8070										
14525	I NEG	8220	9590										
14375	I PRT	6940	3820	6860	7430	12073							
14653	I REQ	9480	9690										
14660	I ROP	9540	9450	9460	9470	9490	9520						
14443	I TEM	7490	6850	6850	7560	7560	7810	7810	7830	7830	7850	7850	8550	8550
			8770	8770	8820	8820	8850	8850	9070	9070	9090	9090		
14723	I DSTK	9770	11240	11280	11320	11360	11400	11440	11480	11520	11560	11600		
14774	I EXIT	10410												
14641	I EXP,	9270												
14451	I GOTA	7600	7690											
14450	I GOTE	7582	7980	8170	8260									
14444	I GOTO	7550	9490											
14462	I GOTX	7700	7640											
14636	I INST	9190	9030											
14755	I ITER	10230	8700	8980	9260	10050	10250	10330						
14513	I LESS	8030	9570											
14606	I MUL,	8990												
14605	I MULT	8970	10290											
14633	I MULY	9160	9060	9130										
14442	I NTEM	7480	6950	7040	7160	7580	7620	7820	8430	8450	8540	8580	8590	8730
			8840	8860	8880	8890	9010	9100	9140	9900	9990	10080	10130	10150
			10160											
14437	I PRTO	7440	6970	7060										
14431	I PRTA	7130	6990											
14414	I PRTB	7170	7260											
14425	I PRTC	7310	7190											
14431	I PRTD	7360	7410											
14436	I PRTX	7420	7470											
14651	I RGTR	9460	9820											
14652	I RLES	9470	9780											
14650	I RNEQ	9450	9740											
14661	I RTAB	9550	9510											

CROSS REFERENCE TABLE

14665	I VSTK	9650	10990										
14671	I VTAB	9700	9670										
14601	IARTHX	8870	9180										
14532	IASIGN	8390	9980										
14734	IEVAL1	9960	11100										
14740	IEVAL2	10040	11150										
14726	IEVAL3	9860	11050										
14642	IINSTK	9330	10230	10480									
14466	IITEM0	7742	7860	7960	8060	8120							
14467	IITEM1	7744	7840	7970	8040	8140	8230						
14643	IOPSTK	9350	10190	10440	10690								
14364	IPRINT	6840	9390										
14541	IUNMIN	8520	10890										
13314	MTWO	4820	4720										
13740	O POP	4440	4470	7870	8740	9020							
13745	O SDB	4530	4370	4450	5380	5550	6230						
13744	O IPTR	4520	5370										
13733	O PUSH	4350	4390	9370	9500								
100	O SIZE	4510	4560	4570									
13750	OSTACK	4570	4520	4540	4550								
12	P X	300	6170	6180	6200	6210	6220	6240	6580	6570			
13	P Y	310											
14	P Z	320											
14351	P OK	6660	6890	7740	7990	8180	8250	8460	8610	9390	9530	9690	9800	9920
			10000	10260	10440	9040	9080	9120	9180	9200	9280	10100	10140	10350
			10390	10600	10640	10760	10880	10840	10940	11190	11650	11690	11730	11770
			11810	11850	11890	11930	11970	12010	12050	12090	12130	12170	12210	12250
			12290	12330	12370	12410	12450	12490	12530	12570	12610	12618		
14263	P ALT	5680	5320	5510	5600	6080	6090	6260	6380	6550	6720			
14301	P FAIL	6070	6430											
14324	P FPOP	6330	6110											
14272	P MLIT	5900	5850											
14262	P PART	5670	5350	5530	5590	5620	5830	6280	6490	6520	6530	6580	6700	10270
			10270											
14243	P PUSH	5480	5860											
14264	P TEMP	5690	5940	5960	6540	6590	7150	7250	7320	7350	7400	7880	7890	8410
			8440	9040	9050	9080	9110	9880	9910	10300	10320			
14266	P TEST	5820	5430	5630	6290	6600	8910	10170						
14340	P TRUA	6510	6770											
14336	P TRUE	6480	5920	5970										
14265	PLPART	5700	6500	10240										
13275	POP	4640	4780	4170	4460	4750								
14220	PPARSE	5210	3670	6760										
13257	PUSH	4420	4540	4090	4380	4670								
13323	R POP	4150	4180	6340	6350	6360	6370	6390	6670	6680	6690	6710	6730	
13330	R SDB	4240	4080	4160	5280	6160	6400	6740	10290					
13327	R IPTR	4230	5270	6410	6750									
13316	R PUSH	4060	4100	5300	5330	5360	5390	5420	5500	5520	5540	5560	5580	
400	R SIZE	4220	4270	4280										
13333	RSTACK	4280	4230	4250	4260									
15057	S IF	9500	9150											
20	S TN	5060	5070	5100										

CROSS REFERENCE TABLE

15146	S END	9830	9190										
400045	S EOL	10630	9380	9480	9630	9890	9970						
15357	S LET	11610	10980										
15266	S NUM	11000	9470	9620	10830								
15232	S SAE	10700	10520	10590	10880	10930							
15134	S SUM	9990	9370	9550	9570	9960	10740						
15262	S VAR	10950	9940	10790									
15277	S DTAG	11110	11040										
15240	S FACT	10490	10270	10340									
15215	S FTAG	10550	10530										
15046	S GOTO	9400	9110										
15025	S LETS	9210	9030										
15200	S MLOP	10400	10330										
15162	S PROD	10240	10020	10090									
15167	S PTAG	10300	10280										
15141	S STAG	10050	10030										
14176	S TCNT	5080	4970	5000									
14165	S TEMP	4950	5020	8530	8720	9000	9870						
14177	S TPRT	5090	5010										
1	S UNIT	210	4960	5070	5070	5100							
15152	SADDOP	10150	10080										
15125	SASIGN	9910	9270										
15273	SDHEAD	11060	11030										
15307	SDIGIT	11200	11090	11140									
15305	SDTAG,	11180	10060										
777777	SEMPTY	10640	10130	10380	10630	11180							
15226	SEXPOP	10650	10580										
15224	SFTAG,	10630	9270										
15034	SPRINT	9290	9070										
15176	SPTAG,	10380	8990										
15075	SRELOP	9650	9560										
15150	SSTAG,	10130	8710										
15001	SSTATE	9000	5310	5340									
14175	STSIZE	5070	4990										
42	T X	300	2410	2430	3500	3510							
43	T Y	310											
44	T Z	320											
11671	T EOL	1690	1350										
100	T IBN	1890	1900	1940									
12143	T MON	3580	3200	3710	3800	8320	12110	12500					
11562	T MSA	910	960										
11567	T MSX	970	930										
11577	T RDA	1190	1370	1650	1830								
12053	T TEM	2440	2300	2320	2350	2370	2420	2640					
11701	T WRU	1810	1320										
11660	T BACK	1560	1300										
12134	T BSBF	3110	2750	2760	2990	3020							
12022	T CHAR	2080	2130	2150	2290	2800	2860	3860	5950				
11	T CHRX	270	1150	1330	1580	1610	1720	1740	1760	2090	3270	3350	3400	3460
			3480	5290	5490	6190	10310						
11713	T FILL	1880	980	1260	3340	3390	5930						
12033	T GNUM	2280	2340	2390	2400	2530	2570						

CROSS REFERENCE TABLE

11714	T IBLN	1900	1120										
11720	T IBUF	1940	1590	1910									
12062	T LINA	2560	2660										
12074	T LINB	2700	2580										
12104	T LINC	2790	2910										
12105	T LIND	2810	2780										
12055	T LINE	2510	3030	3070	3100	3620							
12117	T LINF	2960	2850	2880									
12126	T LINX	3040	2540										
13247	T LISA	3850	3900										
13241	T LIST	3780	3690	3940									
13254	T LISX	3910	3890										
12240	T LNUM	3720	3220	3810	12072								
11556	T MESS	870	880	900	1420	1510	1820	3590	6870	10420	12071	12080	12160	12220
			12280	12340	12400	12482	12484	12490						
11717	T MONE	1930	920	1570	1620	1730	2180							
12150	T NLNA	3310	3430											
12164	T NLNX	3440	3370	3420										
11572	T READ	1110	1520	1770	2520									
12241	T SBFR	3730	3060	3600	3630	7590								
17	T SBUF	280	2720	2730	2740	2900	2980	3010	3050	3080	3090	3210	3230	3320
			3610	3640	7730									
12054	T TEM1	2450	2550	2600	2650	2710	2830	2890	2970					
10	T TTYX	260	890	950	1000									
11643	TCTRLX	1500	1280	1600										
15777	THIS	12530												
11716	TIBCNT	1920	1130	1360	1630	1640								
11715	TIBFPT	1910	1140	1750	3260	3470								
12032	TLCHAR	2220	2140	2770	3650	9660	9780							
12135	TNLINE	3170	3250	3490	3790	5220								
11551	TPRINT	630	670	940	990	1710	3840	3870	3980	7080	7390	7460		
15716	UP	12470	460	12530										
14055	V POP	4730	4760	6850	7370	7560	7810	7830	7850	8400	8420	8550	8770	8820
			8850	9070	9090	9970	10110							
14062	V SDB	4820	4660	4740	5410	5570	6250							
14061	V IPTR	4810	5400											
14050	V PUSH	4640	4680	7230	8600	8900	9680	9790	9890					
100	V SIZE	4800	4850	4860										
14065	VSTACK	4860	4810	4830	4840									
12	X	300	4430	4450	4480	4490	4650	4670	4690					
13	Y	310	4440	4530	4660	4760								
14	Z	320	4460	4520										

MACRO CROSS REFERENCE TABLE

CHAR	2020	2290	2800	2860	3860	5950							
LCHAR	2190	9660	9780										
MESS	770	1510	1820	3590	12071	12160	12220	12280	12340	12400	12482	12484	12490
POP	4180	6340	6350	6360	6370	6390	6670	6680	6690	6710	6730	6850	7370
		7560	7810	7830	7850	7870	8400	8420	8550	8740	8770	8820	8850
		9020	9070	9090	9970	10110							
POPV	4240	6850	7560	7810	7830	7850	8550	8770	8820	8850	9070	9090	
PRINT	570	940	990	1710	3840	3870	3930	7080	7390	7460			
PUSH	4080	5300	5330	5360	5390	5420	5500	5520	5540	5560	5580	7230	8600
		8900	9370	9500	9680	9790	9890						
SOS	360	10270											