REFERENCE MANUAL FOR THE TIME-SHARING EXECUTIVE

by

L. Durham
M. Etherton

University of California, Berkeley

Document No. R-22
Revised November 21, 1968
Contract SD-185
Office of the Secretary of Defense
Advanced Research Projects Agency
Washington, D. C.   20325

TABLE OF CONTENTS

1.0 Introduction

The Project GENIE operating system is a medium scale multi-access computational system which implements a powerful and complex user machine. It is the role of the command language (here called the EXECUTIVE) to provide some tools to control this user machine, and to provide those services which users have come to expect of conversational systems.

This document describes the system command language; but before pursuing that topic, some items of a more general nature are in order.

2.0 Access to the Computing Facilities

The primary medium by which computer and user communicate is at present the teletype console, which allows input, user to computer, from a keyboard and output, computer to user, via a type-head. Paper tape and magnetic tape are also available but are generally more cumbersome and unsuitable for primary input. Cathode ray displays are also being developed and are likely to render manageable the input and output of graphical data.

Having performed the initial formalities of introducing himself to the project staff as a prospective user and persuading them of the particular suitability of their computing service to the work he has in mind, the new user will be presented with, or allowed access to, a teletype console. To establish a link between it and the computer, the power switch must be turned to ONLINE and the keyboard-tape selector switched to the keyboard, (K) position. It should be noted here that, when the power switch is in the ONLINE position, there is no direct link between keyboard and typehead; any such link as may appear to exist is established through intermediate software. The TSS software, of course, arranges that the typehead respond to keyboard input in a manner appropriate to the occasion--which usually involves echoing back to the typehead any characters input from the keyboard.

The necessary physical connection is now made between teletype and computer and the user may, by pressing the "rubout" key, signal the computer that he requires service on the chosen teletype. A printed response, (see Section 5.1), indicates that the request has been noted and the user has been connected to a built-in software package, called the Executive program or "Exec," which awaits further teletype input in the form of stylized English-language requests or commands. Via these commands the Exec will provide sufficient services to get the user into contact with all the other TSS facilities. It is itself primarily a medium for specifying machine requirements, handling user's files and establishing communication between different teletypes.

3.0 Exec Command Recognition

It is appropriate here to describe the way in which Exec commands are normally recognized, i.e., in the EXPERT mode (see below). As soon as sufficient characters have been typed in to distinguish that intended from all other permissible commands, the computer immediately responds by "taking wings" and typing back the remainder of the command. On the other hand, if the typed sequence of characters can form part of no permissible command, the computer gives a standard response, "? cr. lf.@". The Exec's anticipatory behavior is intended as a labor-saving device, but can prove the opposite to a beginner who habitually oversteps the required minimum input-thereby invariably corrupting his command and getting an error response. There are, however, commands designed to help the beginner in this respect. These are:

BEGINNER

NOVICE

EXPERT

> If the current mode is BEGINNER, each command must be typed in full before it will be recognized.
>
> If the current mode is NOVICE just so much of the command as is necessary to define it uniquely must be typed in. If there then remain three or less characters, these too must be typed. Four or more characters remaining will be output automatically. Characters will be ignored which are typed in while the Exec is typing back these four or more characters.
>
> If the current mode is EXPERT, commands are pre-recognized completely; that is to say, only so much of the command as is necessary to specify it uniquely need be typed, the rest is supplied by the Exec. Characters which are typed in when the Exec is not ready to receive them are stored in an input buffer and delivered in sequence when the Exec returns to the command recognizing mode.

Note that most Exec commands await a terminating "." -- which is the user's confirmation that the command typed is as intended. The command is not implemented until this "." is received and until then may be aborted by pressing, usually, any other character or, certainly, by pressing rubout.

In addition to the above modes, command recognition can be avoided by enclosing the command with single quote characters. The closing single quote may be typed anywhere after enough characters have been given to specify the command. Thus, the forms

```
'RECOVER FROM FILE'  <file>.
```

and

```
'RECOVER'  <file>.
```

are equivalent.

4.0 Files and File Naming

The contents of this section is now to be found in Chapter 12 of "The TSS Reference Manual" (Document R-21). It is recommended that the user read Sections 12.1 to 12.3 of this before trying to handle files.

5.0 The Executive Command Language

We now return to the theme of Section 2.0, wherein we left the reader poised on the brink of discovering the great potential of the Exec's command language.

The functioning of the Exec is described under the following headings:

1. Entering and leaving the system.
2. Manipulation of the user machine.
3. Subsystems
4. Communication with other users.
5. Control of files.
6. Control of the tape system.
7. Manipulation of the total system.

5.1 Entering and Leaving TSS

TSS 1.9's response on receiving the first rubout from a hitherto dead teletype is to produce an appropriate herald, followed by the message:

```
@ ENTER
```

Before the Exec will implement any of the possible commands which may be given to it, the prospective user must now make himself known by typing his name, terminated by a dot. This name must be one of which the system is aware, i.e., the name of an authorized user. If the user himself is not authorized, he must give some authorized user's name, with the permission of that user. If a password is associated with the declared user-name a carriage return is output followed by the word

```
PASSWORD-
```

The correct password must now be typed, the teletype-echo will be suppressed so that no characters are printed as the keys are pressed; the secrecy of the password is thus preserved. The password is terminated by a period, and, if it has been correctly delivered, "OK, carriage return, line feed" is output followed by the date and time. Commands may now be given to the system. A user may set his password with the command:

```
PASSWORD IS < password > .
```

< password > is an arbitrary string of characters terminated by a period.

PAUSE.

This command ignores all input (including rubout) until the user's password, terminated by a period, is correctly typed. In effect it enables the user to "lock" his teletype against interference by other users.

TIME USED

Produces the response

```
a:b:c  IN  x:y:z
```

where x:y:z is the total time elapsed (in hours, minutes, seconds) since the user entered the Exec, and a:b:c is the amount of time actually spent by the central processor in performing computations on the user's behalf.

DATE IS

Prints the time of day and date.

To sever his connection with TSS 1.9 the user must give the command

```
LOGOUT < user's name >.
```

This has the effect of causing TSS to forget everything that the user has in core memory and to delete all his scratch files. After printing the time-used statistics the system becomes "dead" to all input from the teletype keyboard, except rubout.

5.2 Manipulation of the User Machine

5.2.1 Description of the user machine

The Executive considers the active user machine to consist of three components: the Exec itself, a subsystem, and a program. A subsystem is invoked by typing the name of the subsystem desired. The available subsystems are outlined in Section (5.3). A program is invoked via a subsystem such as DDT or through commands such as GO TO FILE.

Note that the Exec does not consider open files to be a part of the user machine, and in fact ruthlessly closes all files whenever it gets control.

The Executive makes no distinction between a subsystem and the address space of the subsystem, so the terms 'subsystem' and 'subsystem memory' are synonomous in this document. The same is true for programs.

5.2.2 Commands

STATUS

Prints the current maps (the little numbers which describe address spaces--see R-21, Section 5) both for the user's program and for any subsystem the user has, followed respectively by the unused and total drum space and the unused and total core memory available to the user.

RESET.

Releases all program and subsystem memory.

MACHINE SIZE IS < n > K.

The decimal number n, must be terminated by K. n specifies to total size of the user machine address space. A memory trap will occur if the user's program subsequently try to make accesses to more than n/2 blocks of memory. The user cannot request that his machine size be greater than his total allowance or that it be smaller than the memory he currently has allocated to him.

UNUSED MEMORY IS

Prints out how many blocks of the user's total memory allocation remain unaccessed.

CHANGE DRUM ASG TO < m >.

Allows the user to increase his scratch file space to m blocks (of 256 words each).

For each user inflexible upper and lower bounds are set for m. Further, the drum assignment will only be increased until no blocks are left in the drum-space pool.

DRUM BLOCKS LEFT = < n > OUT OF < m >

Prints out the number, n, of unused drum blocks in a total assignment of m blocks for the user's drum files.

EXECUTIVITY -1.

Turns the user machine into a privileged user machine, i.e., able to execute without restraint all those calls to the system identified as "Exec-only" in R-21. This command is limited to a select few individuals who are assumed to know what they are doing.

EXECUTIVITY ∅.

Resets the privileged state described above.

< Subsystem >.

A subsystem is called by typing its name. The various subsystems are briefly described in Section 5.3. Most subsystems will let you escape back to the Executive by typing some command peculiar to the subsystem, such as %F, :(HL)', or ZAP, or by typing rubout enough times. Holding the repeat button down while typing rubout should always work.

CONTINUE < subsystem >.

Causes the user to re-enter the last subsystem he was using (if any), effectively at the point at which he left it. If no subsystem has been called or the subsystem memory has been released the command is automatically aborted.

< file name >.

Any file in the system may be invoked as a command. If the file is a "save" (type 1) file, it is run as a subsystem. If it is a "Dump" (type 4) file, delivery of its name is equivalent to RECOVER followed by CONTINUE. Any other type of file will produce a question mark.

SAVE CORE FROM < n > TO < m > ON FILE < output file name >.

or

ON FILE < output file name >, STARTING LOCATION < n >.

The contents of specified range of program memory, n to m, together with the starting location, if provided, are preserved on the named output file.

The output file name must be of a form accepted by BRS 16. If it satisfies the conditions for a no-skip return from BRS 16 the name is ignored and another name must be provided. The name may be terminated by a period, thus, terminating the command and causing it to be executed, or a comma, in which case a "starting address" (see also the "GO TO" command) must be typed in.

Each of the addresses, whether core range limit or starting address, is interpreted as an octal number. The starting address, n, must be terminated by a period. Delivery of any other non-octal digit character, except rubout, aborts the address-which must be retyped. The opcode field of the starting address word is used as follows:

bit	octal value	meaning	
8	100000	Do not release "program" memory when run as a subsystem.	
7	200000	Run the file as executive.	If both these bits are off, the file will be run as executive if and only if "executivity" is set.
6	400000	Do not run the file as executive	
5	1000000	When control returns to the exec after running the file, release all of its memory. Such files are run without affecting program or subsystem memory at all.	
4	2000000	File is a common subsystem, treat it as shared memory.	

PLACE < input file name >.

The contents of the named input file is transferred to the core addresses specified at the time of its creation (by BRS 93 or the 'SAVE CORE' command). It is transferred into the user's program memory which is extended, as necessary, to accommodate it.

The file name must be in the user's file directory. If it is not, ? is printed, the name is forgotten and must be delivered anew. The file name must be terminated by a period. The file must be a core-image (type 1) file. If any of these conditions is not satisfied, the command is aborted--as it is also if the attempted data transfer to core results in some transfer-error condition's arising.

GO TO < input file name >.

The action is initially as for the PLACE command. However, after transferring the file to core, instead of a return to the Exec. there is a branch of control into program memory at the starting address specified at the time of the file's creation. If a zero starting address or none at all was then given, the transfer is back to the Exec, as for the PLACE command.

BRANCH TO < address >.

A transfer of control is made to the specified address into the user's program memory. The address (an octal number) must be terminated by a period; any other character aborts the command.

DUMP ON FILE < output file name >.

The entire current status of the user machine (except open files) is preserved on the named output file, which is given the Exec. type number 4.

RECOVER FROM FILE < input file name >.

The named file must be an old, type 4 ("dump") file, i.e., it must have been created by the Exec command "DUMP ON". The RECOVER command restores the status of the user machine at the time of creating the Dump File.

5.2.3 Error Messages

Whenever a program or subsystem terminates, control is given to the Executive. If the termination was due to a catastrophe, the lucid message

```
I > > nnnnnnnn
          or
M > > nnnnnnnn
```

is given. The first message refers to an illegal instruction, the latter to a memory violation occuring somewhere in the user machine. nnnnnnnn is an eight-digit octal address of the offending instruction. As some parts of the user machine are invisible to the user, this address is occasionally confusing.

In the case of a system-wide catastrophe, the message SYSTEM RESTARTED - PROCEED WITH CAUTION is sent, with a long delay. This usually leaves the user in exactly the same state as if he had typed 'rubout' enough times to get back to the Executive, and had then typed CONTINUE.

5.3 Subsystems

Any of a number of special purpose TSS software aids, called "subsystems" can be requested simply by typing the name of the subsystem as a command. A subsystem is created by either assembling its name and map into the operating system, or by creating a permanent save file for user SYSTEM. (See the description of save files.) To delete a common subsystem, first delete the file, then create a non-common subsystem with the same name, then delete that file.

For full details of any subsystem, the appropriate subsystem manual should be consulted. We give here only a brief description of the facilities offered by the respective subsystems. A reference to the corresponding TSS document is given after each subsystem name.

Any file can be run as a private subsystem by the simple device of typing the file name. If a dump file is encountered, this command is equivalent to RECOVER followed by CONTINUE.

QED (ARPA Doc. R-15)

This subsystem will absorb text input from any peripheral and transform it to a fileable, internal format. There is provision for as-you-type editing as well as extensive features for the modification of existing symbolic files.

ARPAS (ARPA Doc. R-26)

This is the machine-language assembly subsystem for the SDS 930. It will transform a symbolic file consisting of a 930 machine-language program into a binary file which can be loaded directly to 930 core (using the loader in the DDT subsystem). It includes a powerful mechanism for expanding user-defined macros as well as a large variety of options which include the listing, skipping and repeating of parts of the assembly.

ODDT (ARPA Doc. R-11)

ODDT is the loading and debugging subsystem for SDS 940 programs. Binary files prepared by the language processors can be loaded by ODDT and executed under its continuous supervision. Other facilities include interrogating and changing memory locations, scanning memory for the specified digit patterns, inserting patches and breakpoints and performing traces.

DDT

A newer version of ODDT, DDT is a somewhat extended and more efficient debugging tool. DDT has provision for interpretive program execution and conditional breakpoints. DDT will not load binaries produced by ARPAS or FTC.

KDF (ARPA Doc. R-35)

KDF is the subsystem which provides permanent long-term file storage on the disk. It basically allows files to be copied to and from KDF's private disk area to the general system storage.

MAIL

This is a subsystem which allows users to leave messages for each other. It explains itself if you type a question mark.

CCP (ARPA Doc. R-29)

CCP is a conditional command processor which provides a means for generating system commands and examining the resulting output. It has proven to be of great utility for certain repetitive jobs, such as assembling and loading a subsystem.

BACKGROUND

An implementation of CCP, BACKGROUND allows some jobs to be run in sequential batch mode without requiring the user to relinguish his teletype. CCP jobs may be run, and the line printer may be operated.

NARP (ARPA Doc. R-32)

An assembler basically compatible with ARPAS, NARP enjoys the speed advantage gained by a one-pass assembler. Most of the incompatibilities between NARP and ARPAS arise from the fact that ARPAS is two-pass.

QSPL (ARPA Doc. R-28)

QSPL is a growing new language intended for use in implementing systems. It is basically an algebraic language with features for manipulating strings and paged data structures. QSPL produces DDT-loadable binaries and a listing similar to the source language for ARPAS or NARP.

LISP

Is a general-purpose List Processing System. It can, with greater or lesser efficiency, perform user-defined operations on any set of entities capable of being represented as lists either of other lists or, ultimately, of a finite number of distinguishable elements. Problems which are recursive by nature, i.e., in which the definition of a computable entity involves the entity itself, are particularly susceptible to attach using LISP.

SNOBOL (ARPA Doc. R-34)

SNOBOL is a version of Bell Laboratories SNOBOL IV, a powerful string-handling language. SNOBOL also includes the CCP functions described above.

CAL (ARPA Doc. R-23)

The "Conversational Algebraic Language" allows numerical computations to be performed interactively. That is to say, mathematical calculations can proceed under the continuous supervision of the user. Facilities are available for compiling and running complete programs delivered in a stylized semi-conversational form, as well as for carrying forward computations in short steps with printouts of intermediate results.

5.4 Communication with Other Users

Any two teletypes may be "linked" together in one of the following senses: if teletype A establishes an "input-link" with teletype B, then any characters sent to A's input buffer simultaneously go to B's. Thus, a program, ostensibly taking characters from B, will equally accept them from A or B.

If teletype A establishes an "output-link" with teletype B, then any characters sent to A's output buffer--this includes echoes to characters input from teletype A--simultaneously go to B's, and hence normally are typed out both on A and B.

"

The delivery of the command " causes the Exec to echo but otherwise ignore all input characters up to the next EOT or rubout, except that carriage return is echoed by a line feed. It is used for typing comments and messages, as, for example, when linked to another user's teletype.

ACCEPT MESSAGES.
ACCEPT INPUT.
REFUSE MESSAGES.
REFUSE INPUT.

Each of these four commands can be given separately. The ACCEPT commands cause the teletype on which they are delivered (the command teletype) to be made open to any attempt to link another teletype to it--for output (MESSAGES) or input (INPUT) respectively.

The REFUSE commands, on the other hand, block any attempts to make output (MESSAGES) or input (INPUT) links to the command teletype, except such as are made by the user's own or TSS programs. Unless otherwise directed, the Executive assumes that ACCEPT is prevailing.

LINK TO < name or no. >, < name or no. >, ..., < name or no. >.

Output links are set up, in both directions, between each accessible named teletype and the teletype on which the command is given. All of the named teletypes which are set to refuse messages are listed as inaccessible and no link is established either to or from them. Up to 8 teletypes may be specified in the command either directly, by number or indirectly, by the name of the user currently logged in thereon. Delivery of a number which does not correspond to an existing teletype or a name not belonging to a recognized user elicits a question mark, '?'; the name or number is ignored. Delivery of the name of a user not currently entered on any teletype elicits a message; the name is ignored. Each name or number must be terminated by a comma, carriage return or, to terminate the list, a dot; any other terminator aborts the last-delivered name or number. If 8 teletypes are specified, a comma or carriage return after the last is treated as a dot. More generally, a dot delivered immediately after any comma or carriage return will serve to terminate the command.

ADVISE < name or no. >.

An output link is set up in both directions between the command and named teletypes. The named teletype may be specified via its teletype number or the name of the user currently logged in on it. The following three control characters are used:

(a) Control E - enters or leaves comment mode (types < and >). In comment mode no characters are input to the user being advised. In command mode characters typed by the advisor are input to the advisee's program.
(b) Control ← simulates a rubout on the advisee's teletype.
(c) Control V is used to send the next character typed to the advisee without interpretation. (The advise command is terminated by rubout.)

BREAK LINKS.

All links for input and output to and from the command teletype are broken.

WHERE IS < user's name >

A list of the teletype numbers on which the named user is currently entered is printed. A message is printed if the user is not currently entered. An illegal name aborts the command.

WHO IS ON < teletype no. >.

or

WHO IS ON?

The first form of the command prints the name of the user currently logged in on the specified teletype. The second form (terminated by a question mark) produces a listing of all teletypes in use, each with the name of the user currently entered on it.

5.5 Handling File Names

FILE DIRECTORY

Produces a listing of information about the user's files. The extent of the listing is optional and must be specified after typing the command. The options are:

a. FOR FILE: < file name >

produces, for the name file, the following information:

name
whether random (R) or sequential (S)
type

for drum files:

address of first index block

for tape files:

file length
tape system number and file position
date last opened for O-P

RO (read only) or RW (read write) to indicate the files's accessibility to the owner

- (inaccessible), PRO or PRW to indicate its accessibility to the public

G < n > if it belongs to a special group, where n is the group number

RO or RW to indicate the group accessibility

b. LONG:

Lists the above information for every file belonging to the user.

c. BRIEF:

Prints a list of all the user's file names.

d. DRUM FILES:

Prints a list of user's drum file names.

e. TAPE FILES:

Prints a list of user's tape file names.

f. GROUPS:

Produces a table showing the correspondence between group names and numbers.

g. PSEUDONYMS:

Lists all the user's pseudonyms, showing also the strings for which they stand.

SET MODES FOR FILE

NAME: < file name >

is the command whereby a file can be assigned to a read-in or special group. Its type and public, private or group accessibility can also be defined. Any of five subcommands can be typed:

```
TYPE - t
PUBLIC - NO or RO or RW
PRIVATE - RO or RW
GROUP - < name >, RO or RW
```

The type, t, must be a number in the range 0 to 7 and cannot be 4 or 5. The group name is any string of characters terminated by EOT, (which is echoed as a comma); the group accessibility must simultaneously be set, (RO or RW). The file may be removed from any group to which it belongs by typing ' - ' in place of the group name. Each subcommand may be terminated by a comma, indicating another to follow; by a dot, indicating a new file name to follow; or by D^c, to end the command.

DEFINE NAME '< new name >' AS < old name >.

< old name > must be an existing name in the user's file directory; note that < new name > must be enclosed in quotes. The command causes < new name > to be attached to the same file--so that either of < new name > or < old name > can now be used to refer to the given file.

If < new name > is already attached to some other file, a message is printed. If the command is nonetheless completed, < new name > will be detached from the old file before being attached to the file < old name >.

DELETE NAME < name >.

This command may be used to remove any entry from the file directory. The entry may be:

a. A group name, which must here be prefixed by the character >;

thus: > SPECIAL

All files belonging to the specified group are released from it and the associated group number becomes available for reassignment to another group name.

b. A Pseudonym; here the pseudonym itself is removed, not the name for which the pseudonym stands.

c. A File name; if this is the name of a peripheral, tape file or permanent drum file then some other name must have been attached to this file (by the DEFINE NAME command, or otherwise). In any other case both the name and the file information are lost.

USE '< name >' FOR < string > D^c

This command defines the name to be a pseudonym for the string. That is to say, the name can be used in place of the string as an old file name, or initial part thereof, in all the commands and system subroutines which take an old file name as an argument--except the DEFINE & DELETE NAME commands.

The usual file-name recognition rules apply equally to pseudonyms, which may be delivered with or without quotes.

Example: The command

```
USE   'S'   FOR   (SMITH,SPECIAL) FILE D^c
```

causes the names

```
(SMITH, SPECIAL) FILE 123.
S123.
'S'123.
all to be equivalent.
```

COPY FROM FILE < input file name > TO < output file name >.

The content of the named input file is copied on to the output file. The output file name may be 'old', i.e., attached to some already existing file--in which case the said previous file is lost--or 'new', in which case a new entry in the user's file directory is created for it.

The input and output file names must be of forms accepted by BRS 15, 16 respectively. The format is described in Chapter 12, Doc. R-21.

If the input (output) file name satisfies the conditions for a no-skip return from BRS 15 (BRS 16) a question mark is typed back, the name is forgotten and a new attempt may be made to type it.

The command is executed on delivery of a terminal dot. The file created will be random or sequential according as the input file is random or sequential--but see also Section 12.4 of the TSS 2.0 Reference Manual.

$(SYS)^c$ TRANSFER FILE < file name > TO USER < user name >.

The file directory entry for the specified file is transferred to the directory of the given user; it is deleted from the user's own directory. If the user is privileged, the file name may be augmented; and the command may, therefore, be used to transfer entries from someone elses' file directory to yet a third persons'. $(SYS)^c$ TRANSFER does not work for tape files. Use TDUP.

5.6 Control of the Tape System

In lieu of a large-capacity secondary memory, the medium for storing permanent files is magnetic tape. The physical tapes are multiplexed in the sense that one reel of tape generally holds hundreds of files belonging to different users. Each tape contains a directory (CFD) which describes its contents. This is transferred to the drum when a tape is available, and written back out on the tape (if it has changed) when the tape becomes unavailable.

MOUNT TAPE ON UNIT: n.

This privileged command serves to change tapes on unit n. The command will rewind the tape (if any) currently mounted on unit n and delete all references to it; if necessary, the tape file directory will be rewritten. When the status of the specified tape unit has been entirely reset, the message "MOUNT NEW TAPE" will be given. Until this time the tape must not be disturbed. Typing dot (.) tells the system that the new tape has been mounted. The file directory is then copied, as needed, onto the drum.

If the new tape is not the current tape for its system, a message will be typed; and it will be mounted as a backup tape. "Old" format tapes must be verified as such by typing "Yes" when asked; any other answer means that it is a "new" format tape. The reel number will also be requested when mounting "old" format tapes. Any tape made before March, 1967 (i.e., using a pre-1.9 system) is old format.

DISMOUNT TAPE ON UNIT: n.

This is identical to the first part of MOUNT TAPE. It returns to the executive just before typing the "MOUNT NEW TAPE" message.

(SYS)^{c} CSTCOR

This privileged command requests a system number, then a reel number, which becomes the current reel for that system. Reel 0 means that any reel may be mounted. Do not use this command for any system which is currently mounted.

RETRIEVE FILES FROM REEL n.

This command is provided for recovery of files from backup tapes which have already been mounted with the MOUNT TAPE command. The command requests a reel number (n) which indicates the source tape. This number can be read from the TAPE STATUS command. Specification of a reel which is not a backup tape is an error. Errors will be indicated, also, if the file directory for the tape is corrupted.

The command will respond with "COPY FILE" and wait for a tape file to be specified. User names may be given in parenthesis; however, no group names are allowed. Unfortunately, quotes may not be used to delimit tape file names. Once the tape file name has been accepted the system responds with "TO" and awaits specification of any TSS file as output. The copy sequence will continue until terminated by rubout.

TAPE STATUS

Prints, with a heading, the current values of various parameters associated with the system's handling of the magnetic tape units.

5.7 Manipulation of the Total System

The commands listed in this section are used for querying and manipulating the total system, and so are generally highly restricted in use. They are useful for investigating and repairing failures in the system, both hardware and software.

(SYS) PSPAR

Lists the time that the system has been running, the number of available drum blocks in the pool, and various statistics relating to system performance.

(SYS) DISCONNECT.

Forcibly disconnects the currently-connected magnetic tape channel from the computer's I-O buffer. It will rescue TSS 1.9 from most hang-ups arising from a malfunctioning tape unit.

(SYS) REWIND UNIT: < n >.

Causes the tape on logical unit n to be rewound. It has the side effect of reinitializing some of the system's records concerning the state of the tape unit.

(SYS) DEBUG <.n >.

Resets the user machine, then sets relabelling for the running monitor in the "program" map and enters DDT with the symbol table from band < n > of the drum. This band should, of course, have been set up with a symbol table in advance (see (SYS) SAVE SYMBOLS).

(SYS) DUMP PAGES < m > TO < n >, BAND $\left\{\begin{array}{l} < r >. \\ < r >, \text{ PAGE } < s >. \end{array}\right.$

$0 \leq m,n \leq 7$ refer to page numbers in the program relabelling, thus, Pages 5 to 6 refers to locations 24000 to 33777 inclusive. r is a drum band number (decimal) and s, if given, is a page number within the band ($0 \leq s \leq 3$).--it is taken to be zero if not given. The contents of pages m to n of the program relabelling are preserved on the drum, starting at band r, page s and using as many subsequent drum pages as are required to complete the transfer. The contents of core are unaffected.

(SYS) LOAD PAGES < m > to < n > BAND $\left\{\begin{array}{l} < r >. \\ < r >, \text{ PAGE } < s >. \end{array}\right.$

This reverses the action of (SYS) DUMP. Information is transferred from the drum to the specified pages of the program relabelling. Additional memory is assigned as needed.

(SYS) SAVE SYMBOLS ON < r >.

DDT should be the current subsystem when this command is given. Its action is to write DDT's symbol table on to drum band r. 8K of drum space is always taken, regardless of the size of the symbol table.

(SYS) GO TO < r >.

This is intended for transferring control to a new version of the system.

The contents of drum band r and the first two pages of its successor are transferred into the first 12K of real core and there is a branch to location 24B. The specified 1-1/2 drum bands should, of course, contain the required new version of the system.

(SYS) NUNAME "< file name >" AS < n_1 > < n_2 > < n_3 > < n_4 >.

Sets the file name in the file directory and attaches to it a 4-word description block, whose contents are the octal numbers n_1, n_2, n_3, n_4 respectively.

(SYS) LFDBT

Lists the bit table for file storage space on the drum.

(SYS) RFDBT

Invokes the file directory clean up routine.

This checks each entry in every user's file directory for meaningfulness and consistency. Unrecoverable entries are deleted, as is a totally unrecoverable file directory--which is replaced by a newly initialized, empty file directory.

The bit table for file storage space on the drum is recomputed.

(SYS) PUDIR.

Prints the entire user directory for all users.

(SYS) SUDIR.

Followed by a user's name. This allows each of the seven locations in the user's user-directory data block to be interrogated and changed if required.

(SYS) TUDIR

Prints the entire user directory contents for each user satisfying a specified condition, in particular, the command accepts three octal numbers, a, b, c, separated by commas and terminated by a dot. The first number ($0 \leq a \leq 6$) indexes a word, w(a), of interest in each user directory description block. The second number is used to mask both w(a) and the third number, c. The name and user directory is printed for each user for which $b \wedge w(a)$ matches $b \wedge c$.

(SYS) PPMT

Prints the user's private memory table. All assigned memory blocks in the user machines' memory are listed.

(SYS) SPMT

Set private memory table. This enables entries to be artificially set up in the users private memory table.

(SYS) PSMT

Print shared memory table.

(SYS) SSMT

Set shared memory table.

(SYS) STOD mm-dd-yy hhmm:ss

Sets the given date into the system clock.

6.0 List of Commands

The following is a list of commands accepted by the current (November, 1968) Executive. The commands marked with () require a password preceding the command.

Command	Described in Section
ACCEPT INPUT	5.4
ACCEPT MESSAGES	5.4
ADVISE	5.4
BEGINNER	3.0
BRANCH TO	5.2.2
BREAK LINKS	5.4
CHANGE DRUM ASSIGNMENT	5.2.2
CONTINUE	5.2.2
COPY FILE	5.5
()CSTCOR	5.6
DATE IS	5.1
()DEBUG	5.7
DEFINE NAME	5.5
DELETE NAME	5.5
()DISCONNECT	5.7
DISMOUNT TAPE	5.6
DRUM BLOCKS LEFT	5.2.2
()DUMP PAGES	5.7
DUMP ON FILE	5.2.2
ENTER	5.1
EXECUTIVITY	5.2.2
EXPERT	3.0
GO TO FILE	5.2.2
()GO TO	5.7

Command	Described in Section
KILL PROGRAM	5.2.2
()LFDBT	5.7
LINK TO	5.4
()LOAD PAGES	5.7
LOGOUT	5.1
MACHINE SIZE	5.2.2
MOUNT TAPE	5.6
NOVICE	3.∅
()NUNAME	5.7
PASSWORD IS	5.1
PAUSE	5.1
PLACE FILE	5.2.2
()PPMT	5.7
()PSMT	5.7
()SPMT	5.7
()SSMT	5.7
()PSPAR	5.7
()PUDIR	5.7
RECOVER FROM FILE	5.2.2
REFUSE INPUT	5.4
REFUSE MESSAGES	5.4
RESET	5.2.2
RETRIEVE FILES	5.6
()REWIND	5.7
()RFDBT	5.7
SAVE CORE	5.2.2
()SAVE SYMBOLS	5.7

Command	Described in Section
SET MODES FOR FILE	5.5
()STOD	5.7
()SUDIR	5.7
STATUS	5.2.2
TAPE STATUS	5.6
TIME USED	5.1
()TRANSFER FILE TO USER	5.7
()TUDIR	5.7
USE	5.5
UNUSED MEMORY	5.2.2
WHERE IS	5.4
WHO IS ON	5.4
"	5.4