LABEL 000000000PRINTER00175100CC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/TSDUMP←←

OBJECT /READ

SYMBOL/TSDUMP

```
COMMENT: * TITLE: B5500/B5700 MARK XIV SYSTEM RELEASE                    *     00000001
         * FILE ID: SYMBOL/TSDUMP TAPE ID: SYMBOL2/FILE000               *     00000002
         * THIS MATERIAL IS PROPRIETARY TO BURROUGHS CORPORATION         *     00000003
         * AND IS NOT TO BE REPRODUCED, USED, OR DISCLOSED               *     00000004
         * EXCEPT IN ACCORDANCE WITH PROGRAM LICENSE OR UPON             *     00000005
         * WRITTEN AUTHORIZATION OF THE PATENT DIVISION OF               *     00000006
         * BURROUGHS CORPORATION, DETROIT, MICHIGAN 48232                *     00000007
         *                                                               *     00000008
         * COPYRIGHT (C) 1971, 1972 BURROUGHS CORPORATION                *     00000009
         *  AA320206  AA332366  AA386657                                 *;    00000010
                     TSDUMP/ANALYZE                                             00000012
                     LAST PATCHED 12/11/73                                      00000013

    SEVERAL OPTIONS ARE AVAILABLE FOR CONTROLLING THE ANALYSIS                  00000015
OF A MEMORY DUMP.  THEY MAY BE CLASSIFIED INTO  FOUR CATEGORIES:                00000020
                                                                                00000025
I.  THOSE WHICH SPECIFY THE FORMAT OF THE PRINTOUT OF MEMORY:                   00000030
SPLASH DUMP,OCTAL DUMP,ALPHA/OCTAL DUMP,OR ALPHA DUMP;                          00000040
EITHER SINGLY OR DOUBLY SPACED.                                                 00000050
                                                                                00000055
II.  THOSE WHICH INCLUDE OR EXCLUDE AREAS OF MEMORY NOT OTHERWISE               00000060
INCLUDED OR EXCLUDED:                                                           00000070
1.  INCLUSION OF AVAILABLE AREAS:                                               00000080
A.  WHETHER ACTIVE OR INACTIVE.                                                 00000090
B.  ONLY IF ACTIVE.                                                             00000100
2.  EXCLUSION OF:                                                               00000110
A.  MCP CODE AREAS.                                                             00000120
B.  NORMAL STATE CODE AREAS.                                                    00000130
C.  NORMAL STATE AREAS OTHER THAN THOSE RELATING TO ONE                         00000140
SPCECIFIC MIX INDEX WHERE THE MIX INDEX IS SPECIFIED BY THE USER.               00000150
                                                                                00000155
III.  THOSE WHICH CAUSE OMISSION OF CERTAIN SECTIONS OF ANALYSIS:               00000160
1.  OMISSION OF THE PRINTOUT OF MCP PRT IDENTIFIERS, SORTED                     00000170
ALPHABETICALLY AND BY PRT LOCATION.                                             00000180
2.  OMISSION OF THE DUMPING OF ARRAYS USED BY THE LINE                          00000190
MAINTENANCE(DATACOM) PROCEDURES.                                                00000200
                                                                                00000205
IV.  THOSE WHICH CONTROL THE DUMPING OF STACKS:                                 00000210
1.  EXCLUSION OF ALL NORMAL STATE STACKS EXCEPTING THE ONE                      00000220
BELONGING TO THE SAME MIX INDEX AS THE ONE SPECIFIED BY THE USER.               00000230
2.  INCLUSION OF A USER SELECTED "STACK" AREA WHICH THE                         00000240
ANALYZER WOULD BE UNABLE TO LOCATE WITHOUT THE USERS ASSISTANCE.                00000250
                                                                                00000255
    THE MAJORITY OF OPTIONS MAY BE SET USING A COMMON CONTROL                   00000260
CARD; A FEW MAY BE SET ONLY VIA THE "IN" MESSAGE TO ALTER THE                   00000270
CONTENTS OF CERTAIN PRT CELLS WHICH HAVE BEEN RESERVED  FOR                     00000280
CONTROL FUNCTIONS.                                                              00000290
                                                                                00000295
    THE COMMON COTROL CARD IS ASSUMED TO CONTAIN AN EIGHT DIGIT                 00000300
NUMBER--IF LESS THAN EIGHT APPEAR, LEADING ZEROES ARE                           00000310
AUTOMATICALLY SUPPLIED. THE ANALYZER SEPARATES THE 8 DIGITS INTO                00000320
TWO GROUPS:                                                                     00000330
                                                                                00000335
A.  THE 5 LEFT-MOST DIGITS ARE EXPECTED TO BE *OCTAL* DIGITS                    00000340
WHICH REPRESENT THE TOP-OF-STACK ADDRESS OF THE USER SPECIFIED                  00000350
"STACK" AREA.  IF ANY DIGIT EXCEEDS "7" OR IF ALL FIVE ARE "0",                 00000360
ONLY STACKS LOCATED BY THE ANALYZER WILL BE DUMPED.  OTHERWISE,                 00000370
ONE ADDITIONAL "STACK", PROVIDING IT ALREADY HAS NOT BEEN DUMPED,               00000380
WILL BE DUMPED STARTING AT THIS ADDRESS AND CONTINUING UNTIL                    00000390
200(DECIMAL) WORDS BELOW IT HAVE BEEN PRINTED( THE STACK IS                     00000400
ASSUMED TO BE A CONTROL STATE STACK FOR PURPOSES OF STACK                       00000410
```

```
ANALYSIS).  SHOULD MORE OR LESS STACK BE DESIRED, THE AMOUNT                00000420
( IN DECIMAL) OF STACK TO ANALYZE CAN BE ENTERED INTO PRT CELL              00000430
27(OCTAL) USING THE "IN" MESSAGE.                                           00000435
IN MOST CASES, THE TOP-OF-STACK VALUE USED MAY BE THE SETTING               00000440
OF THE "S" REGISTER TAKEN FROM THE DISPLAY PANEL AT THE TIME                00000445
OF THE HANG.  IF NO TOP-OF-STACK IS SPECIFIED THE ANALYZER WILL OBTAIN      00000450
ONE FROM THE CONTROL STATE MSCW LOCATED IN CELL 7.  THE STACK               00000451
OBTAINED IN THIS MANNER IS GENERALLY OF INTEREST, ESPECIALLY WHEN           00000452
LINKS HAVE BEEN CLOBBERED.                                                  00000453
                                                                            00000455
B.  THE RIGHT-MOST 3 DIGITS ARE INTERPRETED AS DECIMAL DIGITS               00000460
AND SERVE TO SET THE MAJORITY OF ANALYZER OPTIONS.  GENERALLY               00000470
SPEAKING, THESE OPTIONS ARE INTEGRAL POWERS OF 2, ARE INDEPENDENT           00000480
OF ONE ANOTHER AND MAY THEREFORE BE USED ADDITIVELY.  THAT IS,              00000490
THE COMBINED FUNCTIONS OF COMMON VALUES 2,16,32, AND 64                     00000500
COULD BE INVOKED BY "COMMON=114"(2+16+32+64). THE EXCEPTIONS                00000510
TO THIS RULE WILL BE ELABORATED AFTER EACH COMMON OPTION IS                 00000520
DESCRIBED IN DETAIL.                                                        00000530
                        COMMON CARD OPTIONS                                 00000540
                                                                            00000550
NOTE: IN THE DESCRIPTION OF EACH OPTION, IT IS ASSUMED THAT A               00000560
STANDARD MEMORY DUMP ANALYSIS IS OBTAINED *EXCEPT* FOR THE                  00000570
DIFFERENCES GENERATED BY THAT PARTICULAR OPTION.  A "STANDARD"              00000580
ANALYSIS IS ONE OBTAINED WITH NO COMMON VALUE(I. E. COMMON=0).              00000590
                                                                            00000600
COMMON=0.                                                                   00000610
    YIELDS A STANDARD ANALYSIS.  A TSS/PRT FILE IS REQUIRED.                00000620
                                                                            00000625
COMMON=1.                                                                   00000630
    YIELDS A "SPLASH",UNANALYZED DUMP OF MEMORY.  NO TSS/PRT IS             00000640
REQUIRED.  A "COMMENT" ENTERED WHEN THE DUMP TAPE WAS CREATED               00000650
IS ALSO PRINTED.                                                            00000660
                                                                            00000665
COMMON=2.                                                                   00000670
    DUMPS ALL AVAILABLE AREAS BELOW THE FENCE BUT ONLY THOSE                00000680
ABOVE THE FENCE WHICH ARE CONTAINED WITHIN AN ACTIVE JOBS SET               00000690
OF CHUNKS.  A JOB IS CONSIDERED "ACTIVE" IF ITS MEMORY AREA IS              00000700
PRINTED AND ANALYZED BY THE ANALYZER.                                       00000710
                                                                            00000715
COMMON=3.                                                                   00000720
    DUMPS ALL AVAILABLE AREAS ABOVE AND BELOW THE FENCE.  THIS              00000730
IMPLIES THAT THE ENTIRE AREA ABOVE THE FENCE WILL BE DUMPED                 00000740
EVEN IF PART OF IT IS NOT "ACTIVE".  THE INACTIVE AREAS ARE                 00000750
PRINTED IN "SPLASH" FORMAT.                                                 00000760
                                                                            00000765
COMMON=4.                                                                   00000770
    EXCLUDES ALL NORMAL STATE OBJECT CODE, ABOVE AND BELOW THE              00000780
FENCE.  THIS INCLUDES ALL PROGRAM SEGMENTS ASSOCIATED WITH A                00000790
MIX INDEX AND ALL INTRINSIC SEGMENTS WHETHER IN RE-ENTRANT USE              00000800
BELOW THE FENCE OR EXCLUSIVE USE ABOVE THE FENCE.                           00000810
                                                                            00000815
COMMON=8.                                                                   00000820
    EXCLUDES ALL MCP OBJECT CODE EXCEPT THAT WHICH LIES IN THE              00000830
VERY FIRST AREA OF MEMORY.                                                  00000840
                                                                            00000845
COMMON=16.                                                                  00000850
    INHIBITS THE DUMPING OF THE DATACOM ARRAYS ASSOCIATED WITH              00000860
LINE MAINTENANCE.                                                           00000870
                                                                            00000875
COMMON=32.                                                                  00000880
```

INHIBITS THE DUMP OF THE MCP PRT IDENTIFIERS SORTED ALPHABETICALLY AND BY PRT LOCATION.

COMMON=64.
YIELDS AN OCTAL-ONLY DUMP OF MEMORY FORMATTED SIX WORDS/LINE.

COMMON=128.
YIELDS AN ALPHA/OCTAL DUMP OF MEMORY. EACH PRINTED LINE CONTAINS FOUR WORDS OF OCTAL FOLLOWED, ON THE SAME LINE, BY THEIR ALPHA EQUIVALENT.

COMMON=256.
YIELDS AN ALPHA DUMP OF MEMORY FORMATTED TWELVE WORDS OF ALPHA PER PRINTED LINE.

COMMON=384.
COMPLETELY INHIBITS THE PRINTOUT OF THE CONTENTS OF MEMORY. THIS DOES NOT IMPLY THAT MEMORY IS NOT ANALYZED: VERIFICATION OF MEMORY LINKS IS MADE, THE BED AND FORKQUE ARE CHECKED, STACKS ARE LOCATED, ETC.

COMMON=512
CAUSES THE CONTENTS OF THE "ARGH" ARRAY TO BE PRINTED UP TO THE LAST VALID WORD POINTED TO BY "YECH". THIS INFORMATION IS PRESENT ONLY WHEN AN MCP PATCH, THE "ROTO" PATCH, HAS BEEN COMPILED INTO THE MCP.

PRT CELLS WHICH EXERCISE CONTROL FUNCTIONS

NOTE: PRT LOCATIONS SPECIFIED REFER TO OCTAL PRT LOCATIONS FOR USE IN AN "IN" MESSAGE.

PRT 25.
THIS CELL IS NORMALLY SET FROM THE RIGHT-MOST 3 DIGITS WHICH APPEAR ON A COMMON CARD; HOWEVER, THEY MAY BE EXPLICITLY SET AND/OR MODIFIED DURING EXECUTION OF THE ANALYZER VIA THE "IN" MESSAGE.

PRT 26.
THIS CELL CAN BE USED TO ENABLE A DUMP-BY-MIX-INDEX. IF SET TO A NON-ZERO DECIMAL NUMBER EQUAL TO THE MIX INDEX OF A JOB IN THE MIX AT THE TIME THE DUMP IS TAKEN, IT WILL CAUSE MEMORY AREAS BELONGING TO OTHER MIX INDEXES TO BE EXCLUDED FROM THE ANALYSIS. NOTE THAT IF THE CONTENTS OF THIS CELL DOES NOT CORRESPOND TO A VALID MIX INDEX, *NO* NORMAL STATE STACKS WILL BE DUMPED NOR WILL ANY NORMAL STATE AREAS ASSOCIATED WITH ANY MIX INDEX BE PRINTED.

PRT 27.
THIS CELL MAY BE USED ONLY IN CONJUNCTION WITH THE PREVIOUSLY DESCRIBED COMMON CARD CONVENTION OF SPECIFYING AN OCTAL STACK ADDRESS AS THE FIRST 5 DIGITS OF THE COMMON VALUE. IF MADE NON-ZERO, THEN THE DECIMAL VALUE ENTERED IN CELL 27 WILL DETERMINE THE AMOUNT OF STACK DUMPED BELOW THE TOP-OF-STACK VALUE. IF LEFT ZERO, THE DEFAULT AMOUNT OF STACK DUMPED(AND ANALYZED) WILL BE 200(DECIMAL) WORDS.

PRT 30.
THIS CELL, IF SET TO 1, WILL CAUSE THE PRINTOUT OF MEMORY

TO BE DOUBLE-SPACED. OTHERWISE, THE PRINTOUT IS SINGLE-SPACED. 00001360
00001370
PRECEDENCE OF OPTIONS AND SUGGESTIONS FOR COMBINING THEM 00001380
00001390
AS STATED EARLIER, COMMON OPTIONS MAY BE USED SEPARATELY 00001400
OR IN COMBINATION. ALTHOUGH MOST COMBINATIONS ARE ALLOWED, THERE 00001410
ARE IMPORTANT EXCEPTIONS. THE RULES ARE: 00001420
1. A COMMON VALUE OF 1 PRECLUDES THE USE OF ALL OTHERS. 00001430
2. A COMMON VALUE OF 384 PRECLUDES THE USE OF ALL OTHER 00001440
COMMON VALUES EXCEPT FOR 16 AND 32. NOTE THAT 384 IS THE 00001450
SUM OF 128+256. 00001460
3. EITHER A COMMON VALUE OF 2 OR A COMMON VALUE OF 3,NOT BOTH, 00001470
MAY BE USED. IT IS PERHAPS SIMPLEST TO THINK OF 3 AS AN ADDITIVE 00001480
OPTION JUST AS THOUGH IT WERE AN EVEN POWER OF 2. FOR EXAMPLE, 00001490
COMMON=10(8+2) OR COMMON=11(8+3) IS VALID BUT NOT COMMON=13 00001500
(8+5).(IN THIS LAST CASE, THE 5 WOULD BE ENTIRELY IGNORED RESULTING 00001510
IN ONLY THE EFFECTS OF THE 8). 00001520
4. COMMON VALUES 4,8,16, AND 32 ARE TOTALLY INDEPENDENT AND 00001530
ADDITIVE IN ALL POSSIBLE COMBINATIONS. 00001540
00001550
SOME SPECIFIC EXAMPLES OF COMMON OPTIONS 00001560
00001565
A. "COMMON=35721432". 00001570
THIS CAUSES A "STACK" AT 35721 TO BE DUMPED ALONG WITH 00001580
OTHER STACKS; NONE OF MEMORY IS PRINTED(384), THE MCP PRT 00001590
IDENTIFIERS ARE NOT PRINTED(32), AND THE DATACOM ARRAYS ARE 00001600
NOT DUMPED(16). THIS COMBINATION MIGHT BE USED TO PROVIDE A 00001610
MINIMUM OF OUTPUT AFTER A MEMORY DUMP HAS BEEN ONCE ANALYZED BUT 00001620
AFTERWARDS FOUND TO HAVE A MISSED A STACK(IN THIS CASE, 00001630
LOCATED NEAR ADDRESS 35721). 00001640
B. "COMMON=12". 00001650
THIS ELIMINATES ALL OBJECT CODE FROM THE PRINTOUT OF 00001660
MEMORY BELOW THE FENCE AND ALL NORMAL STATE CODE IN AREAS 00001670
ABOVE THE FENCE. 00001680
C. "COMMON=66". 00001690
THIS ENSURES THAT ACTIVE AVAILABLE AREAS ARE ALWAY PRINTED(2) 00001700
AND PROVIDES THAT MEMORY IS PRINTED IN OCTAL FORMAT ONLY(64). 00001710
D. "COMMON=24". 00001730
THIS TOGETHER WITH PRT 26 SET TO A VALID MIX INDEX REDUCES 00001740
THE AMOUNT OF OUTPUT OBTAINED DURING A MIX-ONLY DUMP. NO 00001750
MCP CODE IS PRINTED LEAVING ONLY NON-MCP CODE AREAS BESIDES 00001760
THOSE RELATING TO THE MIX INDEX BEING DUMPED. ALSO, THE 00001770
DATACOM TABLES ARE OMITTED(16). FINALLY, THE ONLY NORMAL 00001780
STATE STACK DUMPED IS THE ONE BELONGING TO THE PARTICULAR MIX 00001790
INDEX. 00001800
00001900
PROCESSING MULTIFILE DUMP TAPES 00003000
00003005
THE ANALYZER WILL AUTOMATICALLY PROCESS, IN A GIVEN RUN, ALL FILES 00003010
ON A "DPMT" TAPE IF LABEL EQUATION IS MADE TO ANY FILE BEYOND THE 00003020
FIRST ONE. THE ORDER OF ANALYSIS IS OPPOSITE THAT OF CREATION. FOR 00003030
EXAMPLE, IF LABEL EQUATION IS MADE TO THE FOURTH FILE, "FILE MDUMP= 00003040
MEMORY/DUMP004", THE FOURTH FILE IS FIRST ANALYZED, THE THIRD NEXT, 00003050
THE SECOND AFTER THAT ONE AND THE FIRST FILE LAST. AS PROCESSING OF A 00003060
NEW FILE IS BEGUN, A MESSAGE IS WRITTEN TO THE CONSOLE SO THAT THE RUN 00003070
MAY BE DISCONTINUED IF NO FURTHER FILES ARE TO BE ANALYZED. 00003080
00003090
A COMMON VALUE MAY BE INCLUDED IN THE COMMENT PORTION OF THE 00003100
REMARKS ENTERED AFTER A "DPMT" COMMAND OR AFTER KEYING IN THE UNIT 00003110
WHEN USING THE MEMORY DUMP DECK. THE SYNTAX IS: "COMMON="<COMMON VALUE> 00003120

```
, WHERE "COMMON" MAY BE ABBREVIATED AS "COM" AND MUST BEGIN NO LATER     00003130
THAN THE 140"TH CHARACTER OF THE MESSAGE.  A COMMON VALUE ENTERED IN     00003140
IN THIS MANNER OVERRIDES ANY WHICH MAY BE SUPPLIED BY LABEL EQUATION.    00003150
ONE EXAMPLE IS: "DPMT COMMON=36; ALL JOBS APPEAR ASLEEP."                00003160
                                                                         00003170
BEGIN                                                                    00003800
BOOLEAN COM; % MUST BE FIRST DECLARED                                    00003900
INTEGER PRT26; DEFINE ONEMIX=PRT26#; % FOR DUMPING ONLY ONE JOB          00004000
INTEGER PRT27; DEFINE MYSTACKSIZE=PRT27#; % IF ≠0,AMT BELOW MYSTACKADR   00005000
BOOLEAN PRT30; DEFINE DOUBLESPACE=PRT30#; % DBL-SPACE DUMP OF MEMORY     00006000
BOOLEAN PRT31;% RFE                                                      00007000
BOOLEAN PRT32; % USED FOR DEBUGGING                                      00008000
COMMENT DEFINES RELATING TO COMMON CARD OPTIONS;                         00009000
DEFINE COMMON=REAL(COM)=1#;% 1=SPLASH DUMP, NO ANALYSIS                  00010000
DEFINE DUMPAVAIL=COM.[46:1]#;% 2=DUMP ACTIVE AVAILABLE MEMORY            00011000
DEFINE DUMPALLAVAIL=REAL(COM).[46:2]=3#;% 3=DUMP ALL AVAILABLE MEMORY    00012000
DEFINE NONORMALCODE=COM.[45:1]#;% 4=DONT DUMP TYPE 1,7,OR 13 CODE        00013000
DEFINE NOMCPCODE=COM.[44:1]#;% 8=DONT DUMP MCP CODE(TYPE=1,MIX=0)        00014000
DEFINE NOTABLES=COM.[43:1]#;% 16=DONT DUMP SELECTED ARRAYS               00015000
DEFINE DONTDUMPRT=COM.[42:1]#;% 32=DONT DUMP PRT                         00016000
DEFINE DUMPOCTALONLY=COM.[41:1]#; % 64=UNCONDITIONAL OCTAL ONLY DUMP     00017000
DEFINE DUMPALPHAOCTAL=COM.[40:1]#;% 128=DUMP MEMORY IN ALPHA/OCTAL       00018000
DEFINE DUMPALPHAONLY=COM.[39:1]#;% 256=DUMP MEMORY IN ALPHA ONLY         00019000
DEFINE NODUMP=REAL(COM).[39:2]=3#;% 384=OMIT DUMP OF MEMORY              00020000
DEFINE DUMPCESSPOOLONLY=COM.[38:1]#;% 512=DUMP ONLY THE "ARGH" ARRAY     00020100
                                                                         00021000
FILE IN DISK DISK SERIAL "TSS" "PRT"(2,30);                              00021100
FILE IN MDUMP 2(2,533); % DISK=533, TAPE=513                             00021110
FILE SPO 11(1,10);                                                       00021120
FILE P 4(3,15);                                                          00021130
PROCEDURE PRINTCOMMONVALUES;                                             00021140
BEGIN                                                                    00021150
    INTEGER I;                                                           00021160
    SWITCH FORMAT COMINFO:=                                              00021170
    (///"THE COMMON VALUES AVAILABLE ARE:"/),                            00021180
    ("COMMON=0 YIELDS A STANDARD MEMORY DUMP AND ANALYSIS."),            00021190
    ("COMMON=1 YIELDS A SPLASH DUMP WITH NO ANALYSIS."),                 00021200
    ("COMMON=2 PRINTS AVAILABLE AREAS."),                                00021210
    ("COMMON=3 PRINTS AVAILABLE AREAS INCLUDING INACTIVE AREAS ABOVE ",  00021212
     "THE FENCE."),                                                      00021215
    ("COMMON=4 OMITS NORMAL STATE CODE SEGMENTS."),                      00021220
    ("COMMON=8 OMITS MCP CODE SEGMENTS."),                               00021230
    ("COMMON=16 OMITS DUMPING THE DATACOM ARRAYS ASSOCIATED WITH LINE ", 00021240
     "MAINTENANCE"),                                                     00021245
    ("COMMON=32 OMITS THE PRINTING OF THE SORTED MCP PRT IDENTIFIERS."), 00021250
    ("COMMON=64 CAUSES MEMORY TO BE PRINTED ENTIRELY IN OCTAL."),        00021260
    ("COMMON=128 CAUSES MEMORY TO BE PRINTED ENTIRELY IN ALPHA/OCTAL."), 00021270
    ("COMMON=256 CAUSES MEMORY TO BE PRINTED ENTIRELY IN ALPHA."),       00021280
    ("COMMON=384 CAUSES NONE OF THE CONTENTS OF MEMORY TO BE PRINTED."), 00021290
    ("COMMON=512 DISPLAYS THE CONTENTS OF THE ARGH ARRAY."/),            00021300
    ("FOR ADDITIONAL INFORMATION, CONSULT THE FIRST SEVERAL ",           00021310
     "PAGES OF THE FILE:SYMBOL/TSDUMP.");                                00021320
    FOR I:=0 STEP 1 UNTIL 14 DO WRITE(P,COMINFO[I]);                     00021330
END PRINTCOMMONVALUES;                                                   00021340
FORMAT FINI(48("*")," END OF TSDUMP ANALYZE ",48("*"));                  00023000
INTEGER MYSTACKADR;                                                      00023100
BOOLEAN AOTOG,MYSTACKDUMPED;                                             00023150
INTEGER MIXMAX;%ACTUAL MIXMAX IS OBTAINED FM MOTHER VECTOR               00024000
DEFINE DEFINEDMIXMAX=29#;                                                00025000
INTEGER PRTMAX,INTMAX;                                                   00026000
```

```
DEFINE PRTBASE=129#;%FIRST PRT CELL ALLOCATED BY ESPOL                          00027000
DEFINE ACTUALPRTBASE=112#;% FIRST PRT CELL AS PER MCP DEFINE                    00028000
DEFINE PRTSMAX=80#;% UPPER LIMIT OF PRTS ARRAY                                  00029000
DEFINE INFOMAX=4*PRTSMAX+3#;                                                    00029100
REAL LEVEL,SUBLEVEL,VERSION;                                                    00030000
INTEGER KLASS;% IDENTIFIER CLASS,AS DETERMINED BY ESPOL                         00031000
INTEGER NAMESIZE,NAMSSIZE;                                                      00032000
DEFINE NSNAME[NSNAME1]=NAME[NSNAME1].[8:10]#;                                   00033000
INTEGER INAMESIZE,INAMSSIZE;                                                    00034000
DEFINE ISNAME[ISNAME1]=INAME[ISNAME1].[8:10]#;                                  00035000
ARRAY SEGZERO[0:29];                                                            00036000
BOOLEAN  MCP;% TSFILL/PRT DETERMINED WHAT MODULES WERE INCLUDED                 00037000
STREAM PROCEDURE MOVE(S,D,W); VALUE W;                                          00038000
BEGIN SI:=S; DI:=D; DS:= W WDS ; END MOVE;                                      00039000
PROCEDURE BUSTCOMMON;                                                           00042500
BEGIN                                                                           00042510
    REAL MY,D;                                                                  00042520
    ARRAY T[0:0];                                                               00042530
    FORMAT F("%",I5,",",X1);                                                    00042540
    REAL STREAM PROCEDURE OCTDEC(C); VALUE C;                                   00042550
        BEGIN SI←LOC C; DI←LOC OCTDEC; DS←8DEC END;                             00042560
    BOOLEAN STREAM PROCEDURE NOT5OCTADES(C); VALUE C;                           00042570
        BEGIN SI←LOC C; 5(IF SC GTR "7" THEN BEGIN TALLY←1; JUMP OUT            00042575
        END ELSE SI←SI+1); NOT5OCTADES←TALLY END;                               00042580
    STREAM PROCEDURE DECOCT(D,O1,O2); VALUE D;                                  00042590
        BEGIN SI←LOC D; DI←O1;DS←5OCT;DI←O2;DS←3OCT END;                        00042600
%                                                                               00042610
    IF NOT5OCTADES(D←OCTDEC(COM))THEN D.[1:29]←0;                               00042620
    DECOCT(D,MY,COM);                                                           00042630
    IF MY=0 THEN MYSTACKADR←=1  ELSE                                            00042640
    BEGIN % CONVERT FIRST 5 DIGITS OF COM AS IF THEY WERE OCTAL DIGITS          00042650
        WRITE(T[*],F,MY);                                                       00042670
        READ(T[*],/,MYSTACKADR);                                                00042680
    END;                                                                        00042685
END BUSTCOMMON;                                                                 00042690
PROCEDURE READARRAY(WORDSIZE,ANAME,BASE);                                       00043000
VALUE WORDSIZE,BASE;                                                            00044000
INTEGER WORDSIZE,BASE;                                                          00045000
ARRAY ANAME[*];                                                                 00046000
BEGIN                                                                           00047000
    INTEGER I,SEGS,N;                                                           00048000
    ARRAY BUF[0:29];                                                            00049000
    N:=WORDSIZE MOD 30;                                                         00050000
    SEGS:=(WORDSIZE+29) DIV 30;                                                 00051000
    SEGS:=SEGS-1;                                                               00052000
    FOR I:=0 STEP 1 UNTIL SEGS DO                                               00053000
    BEGIN                                                                       00054000
        READ(DISK,30,BUF[*]);                                                   00055000
        MOVE(BUF,ANAME[BASE+30*I],IF I ≠SEGS OR N=0 THEN 30 ELSE N);            00056000
    END;                                                                        00057000
    IF PRT32 THEN BEGIN WRITE(P[PAGE]); WRITE(P[DBL],<"BASE=",                  00058000
    I5," SIZE=",I5>,BASE,WORDSIZE); FOR I:=BASE STEP 1 UNTIL                    00059000
    BASE+WORDSIZE -1 DO BEGIN MOVE(ANAME[ I],BUF[0],1);                         00060000
    WRITE(P,1,BUF[*]) END END;                                                  00061000
END OF READARRAY;                                                               00062000
BUSTCOMMON;                                                                     00063000
COMMENT:TO GET OPTIONS, USE "(MIX)IN25" AND/OR COMMON CONTROL CARD;             00063100
IF COMMON THEN NAMESIZE:=NAMSSIZE:=INAMESIZE:=INAMSSIZE:=1 ELSE                 00066000
BEGIN                                                                           00067000
READARRAY(30,SEGZERO[*],0);                                                     00068000
```

```
CLOSE(DISK);                                                                    00069000
LEVEL:="2"; % 1 DIGIT                                                           00070000
SUBLEVEL:="00"; % 2 DIGITS                                                      00070100
VERSION:="B"; %                                                                 00070200
COMMENT EACH TIME ANALYZER IS CHANGED INCREMENT ALPHABETICALLY;                 00070300
MCP:=BOOLEAN(SEGZERO[1]);% COMPILE-TIME OPTIONS                                 00071000
NAMESIZE:=SEGZERO[2]+1;% # OF MCP PRT CELLS(ABOVE @200)                         00072000
NAMSSIZE:=SEGZERO[3];% # OF WORDS THE MCP IDENTIFIERS OCCUPY                    00073000
INAMESIZE:=SEGZERO[4]+1;% # OF INTRINSICS                                       00074000
INAMSSIZE:=SEGZERO[5];% # OF WORDS THE INTRINSIC IDENTIFIERS OCCUPY             00075000
END SEGZEROSTUFF;                                                               00076000
MIXMAX:=DEFINEDMIXMAX;                                                          00077000
% % % % % % % % % % % % % INNER BLOCK BEGINS % % % % % % % % % % % % %         00082000
BEGIN                                                                           00083000
    LABEL EOPROG;                                                               00083050
    LABEL RESTART,XX;                                                           00083100
    BOOLEAN PRTOK,SGLTOG;                                                       00084000
    BOOLEAN NODUMPTOG;                                                          00084100
    INTEGER I;                                                                  00085000
    REAL R;                                                                     00086000
    REAL AUXTYPE;                                                               00086010
    BOOLEAN NOMD,MULTI,REPEATING,RELOAD;                                        00086020
    ARRAY PRTNAME[0:4];%                                                        00086100
    ARRAY TPNAME[0:2];                                                          00086110
    DEFINE TDMFID=TPNAME[0]#,                                                   00086120
           TDFID=TPNAME[1]#,                                                    00086130
           FILETYPE=TPNAME[2]#;                                                 00086140
    ARRAY PRTS[0:PRTSMAX];                                                      00087000
    ARRAY INFO[0:INFOMAX+1];                                                    00088000
    ARRAY NAME[PRTBASE:PRTBASE+NAMESIZE-1];                                     00089000
    ARRAY NAMS[0:NAMSSIZE-1];                                                   00090000
    ARRAY INAME[0:INAMESIZE-1];                                                 00091000
    ARRAY INAMS[0:INAMSSIZE-1];                                                 00092000
    ARRAY XNAME[ACTUALPRTBASE:201];                                             00093000
    ARRAY XNAMS[0:18];                                                          00094000
    DEFINE XSNAME[XSNAME1]=XNAME[XSNAME1].[8:10]#;                              00095000
    REAL VJOBNUM,VBED;                                                          00096000
    DEFINE STAXMAX=80#;%                                                        00097000
    ARRAY STAX[0:STAXMAX-1]; INTEGER MAXSTK,BEDSTK;                             00098000
COMMENT:THE DEFINES BELOW MUST CORRESPOND TO THOSE IN TSFILL/PRT;               00099000
    DEFINE CHECKLINK    = MCP.[47:1]#,                                          00100000
           DCP          = MCP.[46:1]#,                                          00101000
           DEBUGGING    = MCP.[45:1]#,                                          00102000
           DFX          = MCP.[44:1]#,                                          00103000
           DUMPP        = MCP.[43:1]#,                                          00104000
           SAVERESULTS  = MCP.[42:1]#,                                          00105000
           SHAREDISK    = MCP.[41:1]#,                                          00106000
           STATISTICS   = MCP.[40:1]#,                                          00107000
           TWXONLY      = MCP.[39:1]#,                                          00108000
           AUXMEMM      = MCP.[38:1]#,                                          00109000
           B6500LOAD    = MCP.[37:1]#,                                          00109100
           SEPTICTANK   = MCP.[36:1]#,                                          00109200
           PACKETS      = MCP.[35:1]#,                                          00109300
           MONITORR     = MCP.[34:1]#,                                          00109400
           MAXOPT       = 14#; % UPDATE AS OPTIONS ARE ADDED                    00110000
    ARRAY MEMORY[0:63,0:511];                                                   00111000
    DEFINE TYPMAX=22#;%                                                         00112000
    DEFINE M=MEMORY#, %                                                         00113000
      FF=[18:15]#,%                                                             00114000
      CF=[33:15]#,%                                                             00115000
```

```
        CTF=[18:33:15]#,%                                                    00116000
        CTC=[33:33:15]#,%                                                    00116100
            ROW=[33:6]#, %                                                   00117000
            COL=[39:9]#; %                                                   00118000
    DEFINE FENCE            = PRTS[00]#,                                     00119000
           FORKQUE          = PRTS[01]#,                                     00120000
           SQ               = PRTS[02]#,                                     00121000
           ESPBIT           = PRTS[03]#,                                     00122000
           BED              = PRTS[04]#,                                     00123000
           BED1             = PRTS[05]#,                                     00124000
           STATABLE         = PRTS[06]#,                                     00125000
           LINETABLE        = PRTS[07]#,                                     00126000
           TOGLE            = PRTS[08]#,                                     00127000
           DAATE            = PRTS[09]#,% DATE                               00127100
           CLOOCK           = PRTS[10]#,% CLOCK                              00127200
           XCLOOCK          = PRTS[11]#,% XCLOCK                             00127300
           PRT              = PRTS[12]#,                                     00128000
           JAR              = PRTS[13]#,                                     00129000
           INTRNSC          = PRTS[14]#,                                     00130000
           SHEET            = PRTS[15]#,                                     00131000
           MEMROW           = PRTS[16]#,                                     00132000
           TANKS            = PRTS[17]#,                                     00133000
           SEQARRAY         = PRTS[18]#,                                     00134000
           ISTACK           = PRTS[19]#,                                     00135000
           INPUTANK         = PRTS[20]#,                                     00136000
           UV               = PRTS[21]#,                                     00137000
           CHANNEL          = PRTS[22]#,                                     00138000
           FINALQUE         = PRTS[23]#,                                     00139000
           LOCATQUE         = PRTS[24]#,                                     00140000
           IOQUEAVAIL       = PRTS[25]#,                                     00141000
           IOQUE            = PRTS[26]#,                                     00142000
           UNIT             = PRTS[27]#,                                     00143000
           TINU             = PRTS[28]#,                                     00144000
           WAITQUE          = PRTS[29]#,                                     00145000
           NEXTWAIT         = PRTS[30]#,                                     00146000
           FIRSTWAIT        = PRTS[31]#,                                     00147000
           LABELTABLE       = PRTS[32]#,                                     00148000
           MULTITABLE       = PRTS[33]#,                                     00149000
           RDCTABLE         = PRTS[34]#,                                     00150000
           OPTION           = PRTS[35]#,                                     00151000
           MESSAGEHOLDER    = PRTS[36]#,                                     00152000
           PRNTABLE         = PRTS[37]#,                                     00153000
           INITIALIZE       = PRTS[38]#,                                     00154000
           P1MIX            = PRTS[39]#,                                     00155000
           P2MIX            = PRTS[40]#,                                     00156000
           NOTHINGTODO      = PRTS[41]#,                                     00157000
           STACKOVERFLOW    = PRTS[42]#,                                     00158000
           RETURN           = PRTS[43]#,                                     00159000
           OTHERCLOSE       = PRTS[44]#,                                     00160000
           SPACESTACK       = PRTS[45]#,                                     00161000
           STACKQ           = PRTS[46]#,                                     00162000
           LOGARRAY         = PRTS[47]#,                                     00163000
           SPACER           = PRTS[48]#,                                     00164000
           LOGLINE          = PRTS[49]#,                                     00165000
           INTRGATGTR       = PRTS[50]#,                                     00166000
           LMAX             = PRTS[51]#,                                     00167000
           STAMAX           = PRTS[52]#,                                     00168000
           TNAOG            = PRTS[53]#,                                     00169000
           LASTRESULT       = PRTS[54]#,                                     00170000
           RESULTHOLDER     = PRTS[55]#,                                     00171000
```

```
                WORKERSTACK     = PRTS[56]#,                                    00172000
                CT              = PRTS[57]#,                                    00173000
                READYEND        = PRTS[58]#,                                    00173010
                FORCEND         = PRTS[59]#,                                    00173020
                RDYRPTEND       = PRTS[60]#,                                    00173030
                SWAPEND         = PRTS[61]#,                                    00173040
                LOGLINE2        = PRTS[62]#,                                    00173050
                BIGUNS          = PRTS[63]#,                                    00173060
                CHUNKMAX        = PRTS[64]#,                                    00173070
                DIRECTORYFREE   = PRTS[65]#,                                    00173080
                ARGH            = PRTS[66]#,                                    00173082
                YECH            = PRTS[67]#,                                    00173084
                SYSNO           = PRTS[68]#,                                    00173086
                DISKOUNT        = PRTS[69]#,                                    00173088
                EUW             = PRTS[70]#,                                    00173090
                PUNTER          = PRTS[71]#,                                    00173092
                NOPROCESSTOG    = PRTS[72]#,                                    00173094
                CORE            = PRTS[73]#,                                    00173096
                EUQ             = PRTS[74]#,                                    00173098
                LQUE            = PRTS[75]#,                                    00173100
                LQAVAIL         = PRTS[76]#,                                    00173102
                TAR             = PRTS[77]#,                                    00173104
                IOQUESLOTS      = PRTS[78]#,                                    00173106
                DUMMY           = DUMMY#;%                                      00174000
        PROCEDURE FILLINFO;                                                     00175000
        FILL INFO[*] WITH %                                                     00176000
        "FENCE    ",0,0,22,                 % 00                               00177000
        "FORKQUE  ",0,0,26,                 % 01                               00178000
        "SQ       ",0,0,26,                 % 02                               00179000
        "ESPBIT   ",0,0,10,                 % 03                               00180000
        "BED      ",0,0,22,                 % 04                               00181000
        "BED1     ",0,0,26,                 % 05                               00182000
        "STATABLE",0,0,26,                  % 06                               00183000
        "LINETABL","E       ",0,26,         % 07                               00184000
        "TOGLE    ",0,0,22,                 % 08                               00185000
        "DATE     ",0,0,22,                 % 09                               00186000
        "CLOCK    ",0,0,22,                 % 10                               00187000
        "XCLOCK   ",0,0,22,                 % 11                               00188000
        "PRT      ",0,0,26,                 % 12                               00189000
        "JAR      ",0,0,26,                 % 13                               00190000
        "INTRNSC  ",0,0,26,                 % 14                               00191000
        "SHEET    ",0,0,26,                 % 15                               00192000
        "MEMROW   ",0,0,26,                 % 16                               00193000
        "TANKS    ",0,0,26,                 % 17                               00194000
        "SEQARRAY",0,0,26,                  % 18                               00195000
        "ISTACK   ",0,0,26,                 % 19                               00196000
        "INPUTANK",0,0,26,                  % 20                               00197000
        "UVROW    ",0,0,26,                 % 21                               00198000
        "CHANNEL  ",0,0,26,                 % 22                               00199000
        "FINALQUE",0,0,26,                  % 23                               00200000
        "LUCATQUE",0,0,26,                  % 24                               00201000
        "IOQUEAVA","IL      ",0,22,         % 25                               00202000
        "IOQUE    ",0,0,26,                 % 26                               00203000
        "UNIT     ",0,0,26,                 % 27                               00204000
        "TINU     ",0,0,26,                 % 28                               00205000
        "WAITQUE  ",0,0,26,                 % 29                               00206000
        "NEXTWAIT",0,0,22,                  % 30                               00207000
        "FIRSTWAI","T       ",0,22,         % 31                               00208000
        "LABELTAB","LE      ",0,26,         % 32                               00209000
        "MULTITAB","LE      ",0,26,         % 33                               00210000
```

```
"RDCTABLE",0,0,26,                          % 34                 00211000
"OPTION  ",0,0,22,                          % 35                 00212000
"MESSAGEH","OLDER   ",0,22,                 % 36                 00213000
"PRNTABLE",0,0,26,                          % 37                 00214000
"INITIALI","ZE      ",0,10,                 % 38                 00215000
"P1MIX   ",0,0,22,                          % 39                 00216000
"P2MIX   ",0,0,22,                          % 40                 00217000
"NOTHINGT","ODO     ",0,32,                 % 41                 00218000
"STACKOVE","RFLOW   ",0,32,                 % 42                 00219000
"RETURN  ",0,0,32,                          % 43                 00220000
"OTHERCLO","SE      ",0,10,                 % 44                 00221000
"SPACESTA","CK      ",0,22,                 % 45                 00222000
"STACKQ  ",0,0,22,                          % 46                 00223000
"LOGARRAY",0,0,26,                          % 47                 00224000
"SPACER  ",0,0,26,                          % 48                 00225000
"LOGLINE ",0,0,22,                          % 49                 00226000
"INTRGATC","TR      ",0,23,                 % 50                 00227000
"LMAX    ",0,0,22,                          % 51                 00228000
"STAMAX  ",0,0,22,                          % 52                 00229000
"TNAOG   ",0,0,26,                          % 53                 00230000
"LASTRESU","LT      ",0,23,                 % 54                 00231000
"RESULTHO","LDER    ",0,26,                 % 55                 00232000
"WORKERST","ACK     ",0,22,                 % 56                 00233000
"CT      ",0,0,26,                          % 57                 00234000
"READYEND",0,0,22,                          % 58                 00234010
"FORCEND ",0,0,22,                          % 59                 00234020
"RDYRPTEN","D       ",0,22,                 % 60                 00234030
"SWAPEND ",0,0,22,                          % 61                 00234040
"LOGLINE2",0,0,22,                          % 62                 00234050
"BIGUNS  ",0,0,22,                          % 63                 00234060
"CHUNKMAX",0,0,22,                          % 64                 00234070
"DIRECTOR","YFREE   ",0,22,                 % 65                 00234080
"ARGH    ",0,0,26,                          % 66                 00234082
"YECH    ",0,0,22,                          % 67                 00234084
"SYSNO   ",0,0,22,                          % 68                 00234086
"DISKOUNT",0,0,23,                          % 69                 00234088
"EUW     ",0,0,22,                          % 70                 00234090
"PUNTER  ",0,0,26,                          % 71                 00234092
"NOPROCES","STOG    ",0,22,                 % 72                 00234094
"CORE    ",0,0,22,                          % 73                 00234096
"EUQ     ",0,0,26,                          % 74                 00234098
"LQUE    ",0,0,26,                          % 75                 00234100
"LQAVAIL ",0,0,22,                          % 76                 00234102
"TAR     ",0,0,26,                          % 77                 00234104
"IOQUESLO","TS      ",0,22,                 % 78                 00234106
0;%                                                                 00235000
COMMENT*************************************************************** 00236000
    FOR EACH IDENTIFIER DEFINED IN THE ARRAY "PRTS", THERE MUST BE  00237000
    A CORRESPONDING 4 WORD ENTRY IN "INFO".  THE FIRST THREE WORDS  00238000
    ARE RESERVED FOR THE IDENTIFIER TEXT.  THE FOURTH MUST CONTAIN  00239000
    THE CLASS OF THE IDENTIFIER AS DETERMINED BY THE ESPOL COMPILER. 00240000
    THIS CLASS APPEARS IN THE FIRST FOUR COLUMNS OF THE STUFF CARD  00241000
    FOR THIS IDENTIFIER.  IF THE CLASS IS NOT KNOWN, A ZERO MAY BE  00242000
    USED IN THE "INFO" ENTRY FOR IT. HOWEVER USING A CLASS OF ZERO WILL 00243000
    INCREASE THE TIME NEEDED TO LOCATE THE PRT ADDRESS OF AN        00244000
    IDENTIFIER.  NOTE THAT ALTHOUGH ARRAY "PRTS" MAY CONTAIN GAPS   00245000
"INFO" MUST CONTAIN A DUMMY ENTRY COMPRISED OF ALL ZEROES.  .       00246000
    FOR CONVENIENCE, THE CLASS NUMBERS THAT THE ESPOL COMPILER MAY  00248000
    ASSIGN AND WHICH MAY APPEAR ON A STUFF CARD ARE LISTED BELOW:   00249000
        PROCID              =10#.                                   00250000
```

```
            STRPROCID           =12#,                                   00251000
            BOOSTRPROCID        =13#,                                   00252000
            REALSTRPROCID       =14#,                                   00253000
            INTSTRPROCID        =15#,                                   00254000
            BOOPROCID           =17#,                                   00255000
            REALPROCID          =18#,                                   00256000
            INTPROCID           =19#,                                   00257000
            BOOID               =21#,                                   00258000
            REALID              =22#,                                   00259000
            INTID               =23#,                                   00260000
            BOOARRAYID          =25#,                                   00261000
            REALARRAYID         =26#,                                   00262000
            INTARRAYID          =27#,                                   00263000
            NAMEID              =30#,                                   00264000
            ININAMEID           =31#,                                   00265000
            LABELID             =32#;                                   00266000
    ********************************************************************; 00267000
    PROCEDURE SETUPXNAMEANDXNAMS;                                       00268000
    BEGIN                                                               00269000
        STREAM PROCEDURE FILLXNAMS(XNAMS);                              00270000
        BEGIN                                                           00271000
            DI:=XNAMS;                                                  00272000
            DS:= 8LIT"NT1     ";                                        00273000
            DS:= 8LIT"NT2     ";                                        00274000
            DS:= 8LIT"NT3     ";                                        00275000
            DS:= 8LIT"NT4     ";                                        00276000
            DS:= 8LIT"NT5     ";                                        00277000
            DS:= 8LIT"NT6     ";                                        00278000
            DS:= 8LIT"NT7     ";                                        00279000
            DS:= 8LIT"DATE    ";                                        00280000
            DS:= 8LIT"CLOCK   ";                                        00281000
            DS:= 8LIT"XCLOCK  ";                                        00282000
            DS:= 8LIT"READY   ";                                        00283000
            DS:= 8LIT"--------";                                        00284000
            DS:= 8LIT"KLUMP   ";                                        00285000
            DS:=16LIT"FIRSTDECK       ";                               00286000
            DS:= 8LIT"LASTDECK";                                        00287000
            DS:= 8LIT"DIRDSK  ";                                        00288000
            DS:= 8LIT"MEMORY  ";                                        00289000
            DS:= 8LIT"RRRMECH ";                                        00290000
        END OF FILLXNAMS;                                               00291000
        ARRAY T[ACTUALPRTBASE:213];                                     00292000
        INTEGER I,J;                                                    00293000
        FILL T[*] WITH % XNAMES                                         00294000
            123,1,00,                                                   00295000
            120,1,01,                                                   00296000
            119,1,02,                                                   00297000
            127,1,03,                                                   00298000
            125,1,04,                                                   00299000
            124,1,05,                                                   00300000
            126,1,06,                                                   00301000
            128,1,07,                                                   00302000
            112,1,08,                                                   00303000
            113,1,09,                                                   00304000
            114,1,10,                                                   00305000
            115,1,11,                                                   00306000
            116,1,12,                                                   00307000
            117,2,13,                                                   00308000
            118,1,15,                                                   00309000
            122,1,16,                                                   00310000
```

```
        129,1,17,                                                       00311000
        121,1,18;%                                                      00312000
    FILLXNAMS(XNAMS);                                                   00313000
    J:=ACTUALPRTBASE-1;                                                 00314000
    FOR I:=ACTUALPRTBASE STEP 3 UNTIL 211 DO                            00315000
    XNAME[J:=J+1]:=0&T[I][8:38:10]&T[I+1][18:33:15]&T[I+2][33:33:15];   00316000
END SETTING UP XNAMEANDXNAMS;                                           00317000
REAL LINKTYPE;                                                          00317100
ARRAY AREATYPE[0:TYPMAX+1];                                             00317105
PROCEDURE FILLAREATYPE;                                                 00317110
COMMENT***AREATYPE[LINKTYPE], WHERE LINKTYPE IS THE TYPE OF MEMORY      00317120
LINK TO BE PRINTED, CONTAINS A CODE WHICH DETERMINES THE FORMAT FOR     00317130
PRINTING THE AREA : 1=OCTAL, 2=ALPHA/OCTAL, 3=ALPHA ONLY                00317144
DUMPMEMORYANDNOTESTACKS DETERMINES LINKTYPE FROM [3:6] OF A PRIMARY     00317146
LINK.  PRINTCORE THEN OBTAINS AREATYPE[LINKTYPE] AND PRINTS THE AREA    00317150
IN THE APPROPRIATE FORMAT.  ELEMENTS 0 THRU TYPMAX CORRESPOND TO        00317160
POSSIBLE LINK TYPES.  ELEMENT TYPMAX+1 IS USED TO CONTAIN  THE DEFAULT  00317170
FORMAT CODE USED IN PRINTING AN AREA WHICH DOES NOT BEGIN WITH A LINK   00317180
OR CONTAINS A CLOBBERED LINK.  THE CONTENTS OF AREATYPE ARE NOT         00317190
USED IF DUMPOCTALONLY, DUMPALPHOCTAL OR DUMPALPHAONLY ARE SET OR        00317200
IF AN AVAILABLE AREA IS BEING PRINTED WHOSE CORRESPONDING FORMAT        00317210
CODE IS NOT NEGATIVE.  FOR EXAMPLE, A NEGATIVE CODE SUCH AS -2 CAUSES  00317230
BOTH IN-USE AND AVAILABLE AREAS TO BE FORMATTED AS ALPHA/OCTAL,  A      00317240
CODE OF +2 WOULD CAUSE ONLY IN-USE AREAS TO BE PRINTED IN ALPHA/OCTAL-  00317250
AVAILABLE AREAS, IF PRINTED, WOULD BE PRINTED IN OCTAL ONLY.****;       00317260
FILL  AREATYPE[*] WITH % 1-OCTAL, 2=ALPHA/OCTAL,3=ALPHA                 00317500
2,% UNKNOWN(0)                                                          00317510
1,% CODE(1)                                                             00317520
2,% DATA(2)                                                             00317530
2,% IOBUF(3)                                                            00317540
1,% ALGFIB(4)                                                           00317550
2,% INQBUF(5)                                                           00317560
2,% COBFIB(6)                                                           00317570
1,% INTSEG(7)                                                           00317580
1,% HEADER(8)                                                           00317590
1,% (9)                                                                 00317600
1,% (10)                                                                00317610
1,% (11)                                                                00317620
2,% STACK(12)                                                           00317630
1,% (13)                                                                00317640
1,% (14)                                                                00317650
1,% (15)                                                                00317660
1,% (16)                                                                00317670
1,% (17)                                                                00317680
1,% (18)                                                                00317690
1,% (19)                                                                00317700
2,% CIDROW(20)                                                          00317710
1,% (21)                                                                00317720
1,% (22)                                                                00317730
1;%***DEFAULT IF UNABLE TO DETRMINE LINK TYPE(TYPMAX+1)                 00317740
    DEFINE NEXTPAGE=WRITE(P[PAGE])#;                                    00318000
    FORMAT STARS(20("*****")),                                          00319000
           STARZ(*("*"));                                               00320000
FORMAT MCPHDR("TSMCP.XV.",A1,".",A2," COMPILE-TIME OPTIONS:");          00321000
SWITCH FORMAT MCPOPT:=                                                  00322000
("CHECKLINK"),                                                          00323000
("DCP"),                                                                00324000
("DEBUGGING"),                                                          00325000
("DFX"),                                                                00326000
("DUMP"),                                                               00327000
```

```
        ("SAVERESULTS"),                                                    00328000
        ("SHAREDISK"),                                                      00329000
        ("STATISTICS"),                                                     00330000
        ("TWXONLY"),                                                        00331000
        ("AUXMEM"),                                                         00332000
        ("B6500LOAD"),                                                      00332100
        ("SEPTICTANK"),                                                     00332200
        ("PACKETS"),                                                        00332300
        ("MONITOR"),                                                        00332400
        (" ");                                                              00333000
        ALPHA ARRAY ID[0:2];%USED TO CONTAIN AN IDENT WHOSE PRT LOC IS SOUGHT 00334000
        PROCEDURE TABLELOOKUP(ID,LOC);ARRAY ID[0];INTEGER LOC;              00335000
        BEGIN                                                               00336000
        INTEGER I,N;                                                        00337000
        BOOLEAN STREAM PROCEDURE GOTIDENT(A,N,B);VALUE N;                   00338000
        BEGIN SI:=A;DI:=B;TALLY:=0;                                         00339000
            IF N SC = DC THEN TALLY:=1;                                     00340000
            GOTIDENT:=TALLY;                                                00341000
        END GOTIDENT;                                                       00342000
        KLASS:=KLASS-1;                                                     00343000
        FOR I:= 129 STEP 1 UNTIL PRTMAX DO                                  00344000
        IF N:=NAME[I].[18:30] NEQ 0 THEN                                    00345000
        IF N.FF=LOC OR LOC=0 THEN                                           00346000
        IF KLASS LEQ 0 OR KLASS=NAME[I].[3:5] THEN                          00347000
        IF GOTIDENT(NAMS[N.CF],N:=8*N.FF,ID[0]) THEN                        00348000
        BEGIN                                                               00349000
                LOC:=I;                                                     00350000
                I:=PRTMAX+2;%FORCE FALL THRU                                00351000
        END;                                                                00352000
         IF I GEQ PRTMAX +2 THEN ELSE LOC:=-1;                              00353000
        END OF TABLELOOKUP;                                                 00354000
        DEFINE GETIDLOC(GETIDLOC1,GETIDLOC2,GETIDLOC3,GETIDLOC4)=           00355000
                BEGIN                                                       00356000
                    FILL ID[*] WITH GETIDLOC1,GETIDLOC2,GETIDLOC3;          00357000
                    TABLELOOKUP(ID,GETIDLOC4);                              00358000
                END#;%                                                      00359000
        PROCEDURE FIXDEFINES;                                               00360000
        BEGIN COMMENT************************************************************ 00361000
            THIS PROCEDURE IS RESPONSIBLE FOR FINDING THE PRT LOCATION OF AN 00362000
            IDENTIFIER DEFINED IN "PRTS" FOR WHICH THERE IS A CORRESPONDING 00363000
            ENTRY IN "INFO".  IT DOES THIS BY CALLING "TABLELOOKUP" TO      00364000
            PERFORM A LINEAR SEARCH OF "NAMES" TO LOCATE AN IDENTIFIER IN   00365000
            "NAMS" AND COMPARE IT WITH THE ONE GIVEN IN "INFO".  IF AN      00366000
            IDENTIFIER CAN NOT BE LOCATED IN "NAMS", THE "PRTS" ELEMENT IS  00367000
            LEFT UNCHANGED--OTHERWISE, THE ADDRESS OBTAINED IS STORED       00368000
            IN "PRTS".*************************************************************; 00369000
            INTEGER I,LOC;                                                  00370000
            INTEGER STREAM PROCEDURE IDLENGTH(A);                           00371000
                BEGIN LOCAL L; SI:=LOC L; DI:=A;                            00372000
                  3(IF SC=DC THEN JUMP OUT ELSE BEGIN TALLY:=TALLY+1;       00373000
                    SI:=LOC L END);   IDLENGTH:=TALLY;                      00374000
                END IDLENGHTH;                                              00375000
            FILLINFO;                                                       00376000
            FOR I:=0 STEP 4 UNTIL INFOMAX DO                                00377000
            IF (LOC:=IDLENGTH(INFO[I]))=0 THEN ELSE BEGIN                   00378000
                MOVE(INFO[I],ID[0],3); KLASS:=INFO[I+3];                    00379000
                TABLELOOKUP(ID,LOC);                                        00380000
                PRTS[I/4]:=IF LOC GTR 0 THEN LOC ELSE PRTS[I/4] END;        00381000
            KLASS:=0;                                                       00382000
            IF PRT32.[46:1] THEN BEGIN FOR I:=0 STEP 1 UNTIL PRTSMAX DO     00383000
```

```
    WRITE(P,<"PRTS[",I3,"]= ",I4>,I,PRTS[I]);                          00384000
    WRITE(P[PAGE]) END;                                                00384100
END FIXING DEFINES;                                                    00385000
BOOLEAN STREAM PROCEDURE BITON(W,B);                                   00414000
    VALUE B;                                                           00415000
BEGIN                                                                  00416000
    SI:=W;SKIP B SB;                                                   00417000
    IF SB THEN TALLY:=1;                                               00418000
    BITON:=TALLY;                                                      00419000
END OF BITON;                                                          00420000
PROCEDURE NEXTITEM;                                                    00421000
    BEGIN                                                              00422000
        WRITE(P);                                                      00423000
        WRITE(P[DBL],STARS);                                           00424000
     END;                                                              00425000
PROCEDURE PRINTMCPOPTIONS;                                             00426000
BEGIN                                                                  00427000
     WRITE(P[DBL],MCPHDR,LEVEL,SUBLEVEL);                             00428000
    FOR I:=9,10,0,1,2,3,4,13,12,5,11,6,7,8 DO                          00429000
    IF BITON(MCP,47-I) THEN WRITE(P,MCPOPT[I]);                        00430000
    WRITE(P[DBL]);                                                     00431000
END OF PRINTMCPOPTIONS;                                                00432000
    REAL DATE,CLOCK,XCLOCK;                                            00433000
    BOOLEAN ARRAY MODON[0:7];                                          00436000
    INTEGER ARRAY ITD[0:9];                                            00437000
    ARRAY RTD[0:5];                                                    00438000
    DEFINE TIMEANALYZED          = ITD[0]#,                            00439000
           DATEANALYZED          = ITD[1]#,                            00440000
           TIMETAKEN             = ITD[2]#,                            00441000
           DATETAKEN             = ITD[3]#,                            00442000
           TIMELASTHL            = ITD[4]#,                            00443000
           SINCSLASTHL           = ITD[5]#,                            00444000
           MINUTES               = ITD[6]#,                            00445000
           SECONDS               = ITD[7]#,                            00446000
           DAYS                  = ITD[8]#,                            00447000
           YEARS                 = ITD[9]#;                            00448000
     DEFINE DATX                 = RTD[0]#,                            00449000
           XCLOCX                = RTD[1]#,                            00450000
           CLOCX                 = RTD[2]#,                            00451000
           TEMP                  = RTD[3]#,                            00452000
           KINDS                 = RTD[4]#,                            00453000
           MCPVERSION            = RTD[5]#;                            00454000
    FORMAT OUT FMXX("DATE ANALYZED ",I2,"/",I2,"/",I2/                 00455000
    "TIME ANALYZED ",I2,":",I2,":",I2),                                00456000
    X1("MCP VERSION NUMBER ",I2/                                       00457000
    "DATE TAKEN    ",I2,"/",I2,"/",I2/                                 00458000
    "TIME TAKEN    ",I2,":",I2,":",I2),                                00459000
    X1MARKXI("MCP VERSION NUMBER ",A1,A*/                              00460000
             "DATE TAKEN    ",I2,"/",I2,"/",I2 /                       00461000
             "TIME TAKEN    ",I2,":",I2,":",I2),                       00462000
    FMX2("TIME OF THE LAST HALT-LOAD ",I2,":",I2,":",I2),              00463000
  SYSID("DUMP TAKEN ON SYSTEM ",A1),                                   00463100
   FTAPDSK(A4," FILE CONTAINING DUMP IS ",A1,A6,"/",A1,A6),            00463110
   FCOMVAL("COMMON VALUE USED IS ",I5),                                00463120
    FANAL("ANALYZER VERSION=",A1),                                     00463150
  FBLANKS(X120),                                                       00463200
FMX3("TIME SINCE LAST HALT-LOAD  ",I2,":",I2,":",I2);                  00464000
     FORMAT BADBED("********* BAD BED ENTRY *********");               00465000
     FORMAT BADDATE ("BAD DATE TAKEN .....",                           00466000
            BADXCLOCK ("BAD TIME TAKEN .....",                         00467000
```

```
          BADCLOCK("BAD TIME OF H/L");                                          00468000
        FORMAT BADCELL3("WORD 3 HAS THE FLAG BIT ON.....");                     00469000
PROCEDURE TIMES (WHEN,HRS,MIN,SEC);                                            00470000
REAL WHEN;  INTEGER SEC,MIN,HRS;                                               00471000
BEGIN                                                                          00472000
INTEGER T;                                                                     00473000
IF WHEN LSS 0 OR WHEN GTR 5184000 THEN HRS:=MIN:=SEC:=100 %OVERFLOW            00474000
          ELSE BEGIN T:=WHEN;                                                  00475000
          HRS:=T DIV 216000;                                                   00476000
          MIN:=T DIV 3600 MOD 60;                                              00477000
          SEC:=T DIV 60 MOD 60;       END;                                     00478000
END OF TIMES PROCEDURE;                                                        00479000
PROCEDURE DATES(ADATE,MONTH,DAY,YEAR);                                         00480000
VALUE ADATE;                                                                   00481000
ALPHA ADATE;                                                                   00482000
INTEGER MONTH,DAY,YEAR ;                                                       00483000
   BEGIN                                                                       00484000
     REAL Y,D,M;                                                               00485000
     LABEL ON;                                                                 00486000
     ARRAY DAYTABLE [0:11];                                                    00487000
     STREAM PROCEDURE CONV (YEAR,DAY,DAT );                                    00488000
      VALUE DAT ;                                                              00489000
      BEGIN                                                                    00490000
SI:= LOC DAT;                                                                  00491000
SI  := SI +3;                                                                  00492000
DI  := YEAR; DS := 2 OCT;                                                      00493000
DI  := DAY; DS := 3 OCT;                                                       00494000
      END;                                                                     00495000
     FILL DAYTABLE [*] WITH                                                    00496000
      0,31,59,90,120,151,181,212,243,273,304,334;                              00497000
     CONV (Y,D,ADATE);                                                         00498000
     IF ((Y MOD 4)=0) AND (Y≠0) THEN                                           00499000
       BEGIN                                                                   00500000
         IF D =60 THEN                                                         00501000
           BEGIN                                                               00502000
M := 1; GO TO ON;                                                              00503000
           END;                                                                00504000
         IF D > 60 THEN D:=D-1;                                                00505000
       END;                                                                    00506000
FOR M := 0 STEP 1 UNTIL 11 DO                                                  00507000
       IF DAYTABLE [M] GEQ D THEN GO TO ON;                                    00508000
     ON:                                                                       00509000
MONTH := M;                                                                    00510000
 IF M=0 THEN D:=0 ELSE %GO AHEAD                                               00511000
DAY := D - DAYTABLE[M-1];                                                      00512000
YEAR :=Y;                                                                      00513000
   END OF PROCEDURE DATE;                                                      00514000
PROCEDURE GETIMES(XCLOCK,ITIMES,ATIMES);VALUE XCLOCK;                          00515000
REAL XCLOCK;INTEGER ARRAY ITIMES[0];ALPHA ARRAY ATIMES[0];                     00516000
BEGIN % RETURN H:M:S:60THS IN ELS 0-3 OF ITIMES(BINARY) & ATIMES(BCL)          00517000
DEFINE T=XCLOCK#;% TIME IN 60THS                                               00518000
STREAM PROCEDURE CONVERTIMES(ITIMES,ATIMES);                                   00519000
BEGIN SI:=ITIMES;DI:=ATIMES;4(DS:=8DEC)END;                                    00520000
%                                                                              00521000
ITIMES[0]:=T DIV 216000;                                                       00522000
ITIMES[1]:=T DIV 3600 MOD 60;                                                  00523000
ITIMES[2]:=T DIV 60 MOD 60;                                                    00524000
ITIMES[3]:=T MOD 60;                                                           00525000
CONVERTIMES(ITIMES[*],ATIMES[*]);                                              00526000
% NOTE IN CALL ON GETIMES, ARRAY ATIMES & ITIMES MAY BE THE SAME               00527000
```

```
END OF GETIMES;                                                          00528000
    INTEGER MAXMOD,MAXCOR;                                               00529000
    INTEGER TABLESLOC;                                                   00530000
    BOOLEAN LNKSOK,AVALNKOK,SOMOKF,SOMOKB;                               00531000
    % UTILITY PROCEDURES                                                 00561000
    REAL PROCEDURE OCTAL(N);%                                            00562000
        VALUE N; %                                                       00563000
        INTEGER N;%                                                      00564000
        % N.[1:23]=0 SO THAT IF N CONTAINS AT MOST A HALF-WORD THEN      00565000
        % OCTAL IF PRINTED USING O FORMAT, OR A FORMAT FOR FEWER OCTADES 00566000
        % WILL BE THE OCTAL REPRESENTATION OF N                          00567000
    OCTAL:=N.[45:3]&(IF N>7 THEN OCTAL(N.[24:21]) ELSE 0)[3:9:39];       00568000
    REAL STREAM PROCEDURE CHRS(AT,SKIPPING,MANY);                        00569000
        VALUE SKIPPING,MANY; %                                           00570000
        % RETURNING THE 7 OR LESS CHRS REQUIRED                          00571000
    BEGIN                                                                00572000
        SI:=AT; SI:=SI+SKIPPING;                                         00573000
        DI:=LOC CHRS; DS:=8 LIT"0"; DI:=DI-MANY;                         00574000
        DS:=MANY CHR;                                                    00575000
    END CHRS;                                                            00576000
    INTEGER PROCEDURE HIHALF(LOC);                                       00577000
        VALUE LOC;                                                       00578000
        INTEGER LOC; %                                                   00579000
    HIHALF:=CHRS(M[LOC.ROW,LOC.COL],0,4); %                              00580000
    INTEGER PROCEDURE LOHALF(LOC); %                                     00581000
        VALUE LOC;                                                       00582000
        INTEGER LOC; %                                                   00583000
    LOHALF:=CHRS(M[LOC.ROW,LOC.COL],4,4); %                              00584000
    BOOLEAN STREAM PROCEDURE FLGBIT(WORD);                               00585000
    BEGIN                                                                00586000
        SI:=WORD; %                                                      00587000
        IF SB THEN TALLY:=1; %                                           00588000
        FLGBIT:=TALLY; %                                                 00589000
    END FLGBIT; %                                                        00590000
    BOOLEAN PROCEDURE PDATADESC(AT,WHAT);                                00591000
        VALUE AT; INTEGER AT; REAL WHAT;          FORWARD;               00592000
ARRAY THISROW,LASTROW[0:11]; %                                           00593000
SAVE ARRAY ALINE[0:18];                                                  00594000
ARRAY BLINE[0:18];                                                       00594100
BOOLEAN NOTPRINTCALL;                                                    00594500
BOOLEAN AVALNK;                                                          00594600
BOOLEAN STREAM PROCEDURE COMPAREWORDS(S,D,W);                            00595000
    VALUE W;                                                             00596000
BEGIN                                                                    00597000
    LABEL XIT;                                                           00598000
    SI:=S;DI:=D;                                                         00599000
    W(IF 8 SC NEQ DC THEN JUMP OUT TO XIT);                              00600000
    TALLY:=1;                                                            00601000
    XIT:COMPAREWORDS:=TALLY;                                             00602000
END COMPARING WORDS;                                                     00603000
%                                                                        00603899
DEFINE PRINT(PRINT1,PRINT2)=IF NODUMP THEN ELSE                          00603900
        PRINTCORE(PRINT1,PRINT2)#;%                                      00603910
%                                                                        00603950
    PROCEDURE PRINTCORE(FROM,TOO);                                       00604000
        VALUE FROM,TOO; %                                                00605000
        INTEGER FROM,TOO; %                                              00606000
    BEGIN %                                                              00607000
        DEFINE FORI = FOR I := 0 STEP 1 UNTIL 21 DO#; %                  00608000
    DEFINE OCTADE = (DS:=3 RESET;3(IF SB THEN DS:=SET ELSE               00609000
```

```
                    DS:=RESET;SKIP SB))# ;                                  00610000
STREAM PROCEDURE BUILD(FROM,LINE,XA,NA,XB,NB,NC,AL,AO); %                   00611000
VALUE FROM,NA,NB,NC,AL,AO; %                                                00612000
1       BEGIN  DI:=LINE;SI:=LOC FROM;SI:=SI+5;SKIP 3SB;                     00613000
2       5 OCTADE; DS := LIT " "; AO( SI := XA; %                            00614000
3             NA(DS:=LIT" ";2(DS:=LIT" ";8 OCTADE));SI:=XB;                 00615000
4             NB(DS:=LIT" ";2(DS:=LIT" ";8 OCTADE));                        00616000
5       NC(DS:=19LIT" "); DS:=LIT" "); %                                    00617000
6     AL(SI:=XA;NA(DS:=LIT" ";DS:=8 CHR); %                                 00618000
7       SI:=XB;NB(DS:=LIT" ";DS:=8 CHR); %                                  00619000
8       NC(DS:=9LIT" "); DS:=6 LIT " "); END BUILD; %                       00620000
9   STREAM PROCEDURE MV(A,B); BEGIN SI:=A;DI:=B;DS:= WDS END MV;            00621000
10  STREAM PROCEDURE EXPAND(ALINE,BLINE,N); VALUE N;                        00621500
11      BEGIN DI:=BLINE; 18(DS:=8LIT" "); DI:=BLINE; DI:=DI+7;              00621510
12          SI:=ALINE; SI:=SI+7;                                            00621520
13          NC(2(8 OCTADE; DI:=DI+1); SI:=SI+1);                            00621530
14      END EXPAND;                                                         00621540
15      BOOLEAN MATCH,FIRST,LAST,STARD;                                     00622000
16      INTEGER I,R,STARCOUNT;                                              00623000
17       BOOLEAN AL,AO;     INTEGER Z,Z1; %                                 00624000
18      LABEL EDITLINE;                                                     00624450
19       IF DUMPALPHAONLY THEN AL:=TRUE ELSE                                00624500
20       IF DUMPALPHAOCTAL THEN AO:=TRUE ELSE                               00624600
21       IF NOT DUMPOCTALONLY THEN                                          00624700
22       IF(I:=AREATYPE[IF NOTPRINTCALL THEN LINKTYPE ELSE                  00625000
23            TYPMAX+1]) GTR 0 AND AVALNK THEN ELSE                         00626000
24       IF I:=ABS(I)=3 THEN AL:=TRUE ELSE                                  00627000
25       IF I=2 THEN AO:=TRUE;                                              00627500
26       Z1:=(Z:=IF AL THEN 12 ELSE IF AL:=AO THEN 4 ELSE 6)-1;             00628000
27       IF NOT (AL OR AO) THEN AO:=TRUE;                                   00629000
28      AO:=AO AND TRUE; AL:=AL AND TRUE; %                                 00630000
29      STARCOUNT:=IF Z=12 THEN 114 ELSE IF Z=6 THEN 120 ELSE 119;          00631000
30      FIRST:=TRUE;                                                        00632000
31      TOO:=TOO.[32:16]; FROM:=FROM.[32:16];                               00633000
32      IF TOO>FROM THEN %                                                  00634000
33      DO %                                                                00635000
34      BEGIN %                                                             00636000
35       IF NOT LAST := (TOO - FROM LEQ Z) THEN % NOT LAST LINE             00637000
36          BEGIN                                                           00638000
37              IF FIRST THEN                                               00639000
38              BEGIN                                                       00640000
39  IF FROM.ROW=(FROM+Z).ROW THEN MOVE(M[FROM.ROW,FROM.COL],LASTROW,Z)      00641000
40  ELSE %                                                                  00642000
41              FORI MV(M[(FROM+I).ROW,(FROM+I).COL],                       00643000
42                LASTROW[I]);                                              00644000
43              STARD:=FALSE;                                               00645000
44              IF Z=12 THEN GO EDITLINE;                                   00645500
45              FIRST:=FALSE;                                               00646000
46              END                                                         00647000
47              ELSE                                                        00648000
48              BEGIN                                                       00649000
49  IF FROM.ROW=(FROM+Z).ROW THEN MOVE(M[FROM.ROW,FROM.COL],THISROW,Z)      00650000
50  ELSE %                                                                  00651000
51              FORI MV(M[(FROM+I).ROW,(FROM+I).COL],                       00652000
52                      THISROW[I]);                                        00653000
53       IF (MATCH:=COMPAREWORDS(THISROW[0],LASTROW[0],Z)) %                00654000
54                      AND NOT STARD THEN                                  00655000
55              BEGIN                                                       00656000
56                  IF DOUBLESPACE THEN                                     00657000
57                  WRITE(P[DBL],STARZ,STARCOUNT) ELSE                      00658000
```

```
                    WRITE(P,STAR2,STARCOUNT);                                00659000
                   STARD:=TRUE;                                              00660000
               END;                                                          00661000
               IF NOT MATCH THEN                                             00662000
               BEGIN                                                         00663000
                    STARD:=FALSE;                                            00664000
                 MOVE(THISROW,LASTROW,2); %                                  00665000
               END;                                                          00666000
               END;                                                          00667000
          END;% IF NOT LAST                                                  00668000
          IF LAST    OR                                                      00669000
               NOT STARD OR                                                  00670000
               NOT MATCH THEN                                                00671000
          BEGIN                                                              00672000
EDITLINE:                                                                    00672500
          R := MIN(Z,TOO-FROM); %                                            00673000
          IF(FROM+R).ROW NEQ FROM.ROW THEN % CROSS ROW ROUND                 00674000
          BUILD(FROM.ALINE[*].M[FROM.ROW.FROM.COL],512-FROM.COL,             00675000
               M[(FROM+R).ROW,0],(FROM+R).COL,IF R LSS Z THEN %              00676000
          Z-R ELSE 0,AL,AO) ELSE %  STILL IN SAME ROW OF M ARRAY             00677000
          BUILD(FROM .ALINE[*],M[FROM.ROW,FROM.COL],R,                       00678000
               M[0,0],0,IF R LSS Z THEN Z-R ELSE 0,AL,AO); %                 00679000
          IF Z=12 THEN IF FIRST THEN                                         00679500
          BEGIN                                                              00679520
               FIRST:=FALSE;                                                 00679540
               EXPAND(ALINE,BLINE,2+REAL(AVALNK));                           00679550
               WRITE(P[DBL],18,BLINE[*]);                                    00679580
          END;                                                               00679620
          IF DOUBLESPACE THEN WRITE(P[DBL],18,ALINE[*]) ELSE                 00680000
          WRITE(P,18,ALINE[*]);                                              00681000
             END; %                                                          00682000
        END UNTIL (FROM := FROM + 2) GEQ TOO; %                              00683000
        WRITE(P); %                                                          00684000
    END PRINTCORE;                                                           00685000
%                                                                            00685001
    FORMAT ITEM(A5," = ",2(O,X1),A5);                                        00686000
    ARRAY LINE[0:14];                                                        00687000
    DEFINE DISPLAYDBL=DISPLAY#,                                              00687500
           DISPLAYSGL(DISPLAYSGL1,DISPLAYSGL2)=                              00687520
           BEGIN SGLTOG:=TRUE; DISPLAY(DISPLAYSGL1,DISPLAYSGL2);             00687540
                SGLTOG:=FALSE;                                               00687560
           END#;                                                             00687580
    PROCEDURE DISPLAYIT(WHAT,RANGE,ADR);                                     00688000
        VALUE WHAT,RANGE,ADR;                                                00688500
        INTEGER WHAT,ADR;                                                    00689000
        BOOLEAN RANGE;                                                       00689500
    BEGIN                                                                    00690000
        INTEGER H,L,T;                                                       00690500
        BOOLEAN PP;                                                          00691000
        T:=IF PP:=(ADR=0) THEN WHAT ELSE ADR;                                00691500
        IF NOT PP THEN IF PDATADESC(T,H) AND H.[8:10] NEQ 0 THEN ELSE        00692000
        RANGE:=FALSE;                                                        00692500
        WRITE(LINE[*],ITEM,0CTAL(T),                                         00693000
              OCTAL(H:=HIHALF(T)),                                           00693500
              OCTAL(L:=LOHALF(T)),                                           00694000
              IF RANGE THEN OCTAL(H.[32:10]+L.CF-1)                          00694500
              ELSE "      ");                                                00695000
        IF 129≤WHAT AND WHAT≤PRTMAX THEN                                     00695500
        MOVE(NAMS[NAME[WHAT].CF],LINE[4],                                    00696000
             NAME[WHAT].FF);                                                 00696500
```

```
        IF SGLTOG THEN WRITE(P,17,LINE[*]) ELSE                          00697000
            IF PP THEN                                                   00697500
        WRITE(P[DBL],17,LINE[*]);                                        00698000
        END DISPLAYIT;                                                   00698500
        DEFINE DISPLAY(DISPLAY1,DISPLAY2)=                               00699000
               DISPLAYIT(DISPLAY1,DISPLAY2,0)#;                          00699500
        BOOLEAN PROCEDURE OPERAND(AT,WHAT); VALUE AT;                    00703050
             INTEGER AT; REAL WHAT; FORWARD;                             00703075
       PROCEDURE DUMPRESULTHOLDER;                                       00703100
       BEGIN                                                             00703110
       BOOLEAN Y,BUNG;                                                   00703120
        REAL R,L,I,J,K,SIZ;                                              00703130
        ARRAY A[0:11];                                                   00703140
         LABEL FND;                                                      00703150
        FORMAT BADDY("THE RESULTHOLDER IS MESSED UP."),                  00703160
               HEAD(" I/O DESCRIPTOR ",X8," RES DESCRIPTOR ",X8,         00703170
                    " STATABLE[ST]",X12,"LINETABLE[S1]",X6,"TU/BUFF"),   00703180
               INTR("INTERRUPT");                                        00703190
        STREAM PROCEDURE BL(A);                                          00703200
         BEGIN                                                           00703210
          DI:=A; 12(DS:=8 LIT " ");                                      00703220
         END;                                                            00703230
        DEFINE BUMPI = I:=REAL(BOOLEAN(I+1) AND BOOLEAN(SIZ))#;          00703240
         STREAM PROCEDURE TUBU(A,B,C); VALUE A,B;                        00703250
         BEGIN                                                           00703260
              SI:=LOC A; DI:=C;                                          00703270
             DS:=3 LIT " ";DS:=2 DEC;A:=DI;DI:=DI-2;                     00703280
             DS:=1 FILL;DI:=A;DS:=LIT "/";                               00703290
             DS:=2 DEC;DI:=DI-2;DS:=1 FILL;                              00703300
         END;                                                            00703310
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%                                 00703320
        IF LASTRESULT≠0 AND RESULTHOLDER≠0 THEN                          00703330
        IF PDATADESC(RESULTHOLDER,R) AND OPERAND(LASTRESULT,L) THEN      00703340
         BEGIN                                                           00703350
          DISPLAY(RESULTHOLDER,TRUE);                                    00703370
          DISPLAY(LASTRESULT,FALSE);                                     00703380
           WRITE (P[DBL],HEAD);                                          00703390
          SIZ:=R.[8:10]-1;                                               00703400
          R:=R.CF;                                                       00703410
         FOR I:=0 STEP 1 UNTIL SIZ DO                                    00703420
        IF OPERAND(R+I,J) AND (J=REAL(NOT FALSE)) THEN BEGIN             00703430
        BUMPI; GO FND; END;                                              00703440
         I:=L;                                                           00703450
           DO BUMPI UNTIL                                                00703460
          OPERAND(I+R,J) AND REAL(BOOLEAN(J) EQV                         00703470
         BOOLEAN("×0000"&"4"[3:45:3]))=REAL(NOT FALSE) OR I=L;           00703480
    FND:                                                                 00703490
          BL(A);                                                         00703500
          DO BEGIN                                                       00703510
              BUNG:=I=L;                                                 00703520
              IF OPERAND(K:=I+R,J) AND J.[1:5]="D" THEN                  00703530
              WRITE(P[DBL],INTR) ELSE                                    00703540
              IF Y THEN                                                  00703550
               BEGIN                                                     00703560
                Y:=FALSE;                                                00703570
                A[3]:=OCTAL(HIHALF(K));                                  00703580
                A[4]:=OCTAL(LOHALF(K));                                  00703590
            TUBU(J.[9:4],J.[14:4],A[11]);                                00703600
                IF BUNG THEN ELSE                                        00703610
               BEGIN                                                     00703620
```

```
            BUMPI;  BUNG:=I=L;                                              00703630
                  A[6]:=OCTAL(HIHALF(K:=I+R));                              00703640
                  A[7]:=OCTAL(LOHALF(K));                                   00703650
            IF BUNG THEN ELSE                                               00703660
            BEGIN                                                           00703670
            BUMPI;BUNG:=I=L;                                                00703680
             A[9]:=OCTAL(HIHALF(K:=I+R));                                   00703690
             A[10]:=OCTAL(LOHALF(K));                                       00703700
                  WRITE(P[DBL],12,A[*]);                                    00703710
                  BL(A);                                                    00703720
         END                                                                00703730
                END                                                         00703740
              END ELSE                                                      00703750
              BEGIN                                                         00703760
               Y:=TRUE;                                                     00703770
               A[0]:=OCTAL(HIHALF(K));                                      00703780
               A[1]:=OCTAL(LOHALF(K));                                      00703790
              END;                                                          00703800
              BUMPI;                                                        00703810
            END UNTIL BUNG;                                                 00703820
          WRITE(P[DBL],12,A[*]);                                            00703830
         END ELSE WRITE(P,BADDY);                                           00703840
      END;                                                                  00703850
PROCEDURE DUMPARRAY(PRTLOC); VALUE PRTLOC; INTEGER PRTLOC;                  00704000
BEGIN % PROVIDES OCTAL DUMP OF AN ARRAY                                     00705000
    INTEGER LOC,SIZE;                                                       00706000
    FORMAT                                                                  00707000
          F(X2,I4,X2,2(O,X1));                                              00708000
    DISPLAY(PRTLOC,TRUE);                                                   00708500
    IF PRTLOC GEQ 129 AND PRTLOC LEQ PRIMAX THEN                            00709000
    IF PDATADESC(PRTLOC,LOC) THEN                                           00710000
    IF SIZE:=LOC.[8:10] NEQ 0 THEN                                          00711000
    IF LOC:=LOC.CF GTR 0 THEN                                               00712000
    BEGIN                                                                   00713000
         FOR I:=0 STEP 1 UNTIL SIZE-1 DO                                    00717000
         WRITE(P,F,I,OCTAL(HIHALF(LOC+I)),OCTAL(LOHALF(LOC+I)));            00718000
         WRITE(P[DBL]);                                                     00719000
    END;                                                                    00720000
END DUMPARRAY;                                                              00721000
PROCEDURE DUMPCESSPOOL;                                                     00721010
BEGIN                                                                       00721015
    FORMAT HDR("D25-ABN=ABNORMAL"/"D24-RR =READ READY"/                     00721090
                "D23-BFL=BUFFER FINAL LOCATION"/                            00721091
                "D21-WR =WRITE READY"/"D20-BSY=BUSY"/                       00721092
                "D18-NTR=DTCU NOT READY"/                                   00721093
                "NOTE:BSY AND NTR=ADAPTER/DTTU NOT READY"///);              00721094
    FORMAT WHATFILE("CORE PORTION OF SEPTIC /",A1,A6,//);                   00721095
    REAL I,B,FROM; BOOLEAN LO;                                              00721100
    FILE SEPTIC DISK SERIAL (2,60,MYUSE=INPUT);                             00721110
    DEFINE BUMPI = I:=I+1#;                                                 00721120
    DEFINE INTSP=INFO#;                                                     00721125
    ARRAY CC,NAM[0:8];                                                      00721130
    ARRAY LSTATUS[0:64];                                                    00721135
    LABEL LOOP,EOJ;                                                         00721140
    STREAM PROCEDURE PUTLSTATUS(STAT,LINE);                                 00721145
    BEGIN SI:=STAT; DI:=LINE; DS:=2 WDS END PUTLSTATUS;                     00721146
    STREAM PROCEDURE INIT(A);                                               00721150
    BEGIN DI:=A; DS:=32 LIT                                                 00721160
         "PASS INTACT. INTWRITE   READ    ";                               00721170
    END INIT;                                                               00721180
```

```
        STREAM PROCEDURE FLL(A,B,C,D,E); VALUE B,C,D;                          00721190
        BEGIN DI:=E; 15(DS:=8LIT" "); DI:=E; SI:=A;                            00721200
              DS:=WDS; SI:=LOC B; DI:=DI+1;                                    00721210
              2(DS:=2DEC; A:=DI;DI:=DI-2;DS:=FILL;DI:=A;DS:=LIT"/");           00721215
              DI:=DI-1; DS:=4LIT" ";                                           00721220
              SI:=SI+6; SKIP 2 SB; % START AT D25                              00721225
              IF SB THEN DS:=3LIT"ABN" ELSE DI:=DI+3; SKIP SB; %D25            00721230
              DI:=DI+3;                                                        00721231
              IF SB THEN DS:=3LIT"RR " ELSE DI:=DI+3; SKIP SB; %D24            00721235
              DI:=DI+3;                                                        00721236
              IF SB THEN DS:=3LIT"BFL" ELSE DI:=DI+3; SKIP SB; %D23            00721240
              DI:=DI+3;                                                        00721241
              SKIP SB;%IGNORE D22                                              00721245
              IF SB THEN DS:=3LIT"WR " ELSE DI:=DI+3; SKIP SB; %D21            00721250
              DI:=DI+3;                                                        00721251
              IF SB THEN DS:=3LIT"BSY" ELSE DI:=DI+3; SKIP SB; %D20            00721260
              DI:=DI+3;                                                        00721261
              SKIP SB; %IGNORE D19                                             00721270
              IF SB THEN DS:=3LIT"NTR" ELSE DI:=DI+3; %D18                     00721273
        END FLL;                                                               00721276
    PROCEDURE TIM(A,CC); VALUE A; REAL A; ARRAY CC[0];                         00721280
    BEGIN INTEGER I,J;                                                         00721290
        STREAM PROCEDURE FF(A,B);                                              00721300
        BEGIN SI:=A; DI:=B;                                                    00721310
            4(DS:=2 DEC; DS:=LIT ":");                                         00721320
        DI:=DI-1; DS:=53 LIT" ";                                               00721330
        END FF;                                                                00721340
        FOR I:=3 STEP -1 UNTIL 0 DO                                            00721350
        BEGIN CC[I]:=J:=A MOD 60;                                              00721360
            A:=A DIV 60;                                                       00721370
        END;                                                                   00721380
        FF(CC,LINE[7]);                                                        00721390
    END TIM;                                                                   00721400
        BOOLEAN STREAM PROCEDURE GLUNK(A);                                     00721410
        BEGIN SI:=A; IF SC=" " THEN TALLY:=1;                                  00721420
            GLUNK:=TALLY;                                                      00721430
        END GLUNK;                                                             00721440
        PROCEDURE MMM(B,I); VALUE B; REAL I,B;                                 00721450
        BEGIN REAL C;                                                          00721460
            STREAM PROCEDURE MOVE(A,B);                                        00721470
            BEGIN SI:=A; DI:=B;                                                00721480
                8(IF TOGGLE THEN DS:=LIT " " ELSE                              00721490
                IF SC="+" THEN DS:=CHR ELSE DS:=CHR);                          00721500
            END MOVE;                                                          00721510
            FOR C:=0 STEP 1 UNTIL B-1 DO                                       00721520
            MOVE(INTSB[BUMPI],LINE[C]);                                        00721530
            WRITE(P,B,LINE[*]);                                                00721540
        END;                                                                   00721545
        %%%%%%%%%%%%%%%%%%%%START                                              00721548
    FILL LSTATUS[*] WITH                                                       00721550
        "WRITING ","       ",                                                  00721551
        "POLLING ","       ",                                                  00721552
        "SELECTIN","G      ",                                                  00721553
        "ACKING  ","       ",                                                  00721554
        "NAKING  ","       ",                                                  00721555
        "ACKING-E","NG     ",                                                  00721556
        "NAKING-E","NG     ",                                                  00721557
        "WRITE-RE","ADY-BUSY",                                                 00721558
        "IDLE    ","       ",                                                  00721559
        "IDLE-POL","LING   ",                                                  00721560
```

```
        "WAITING ","          ",                                        00721561
        "WAITING-","ENG       ",                                        00721562
                0,          0,                                        00721563
                0,          0,                                        00721564
                0,          0,                                        00721565
                0,          0,                                        00721566
        "NORMAL  ","          ",                                        00721567
        "FIRST-TI","ME        ",                                        00721568
        "SELECT-A","NSWER     ",                                        00721569
        "ENQ-READ","          ",                                        00721570
        "BROKEN  ","          ",                                        00721571
        "POLL-TIM","E-OUT     ",                                        00721572
        "TIMED-OU","T         ",                                        00721573
        "EOT-READ","          ",                                        00721574
        "READ-REA","DY-ABN    ",                                        00721575
        "MESSAGE-","ANSWER    ",                                        00721576
                0,          0,                                        00721577
                0,          0,                                        00721578
                0,          0,                                        00721579
                0,          0,                                        00721580
                0,          0,                                        00721581
        "DISCONNE","CT        ";                                        00721582
        INIT(NAM);                                                      00721586
        IF OPERAND(TOGLE,I) AND BOOLEAN(I.[13:1]) THEN                  00721588
        IF PDATADESC(ARGH,FROM) AND FROM.[8:10]=128 THEN                00721590
        IF OPERAND((FROM:=FROM.CF)+127,I) THEN LO:=TRUE                 00721600
        ELSE GO TO EOJ ELSE GO TO EOJ ELSE GO TO EOJ;                   00721610
        WRITE(P[PAGE]);                                                 00721611
        WRITE(P,WHATFILE,I.[6:6],I);                                    00721612
        DISPLAY(ARGH,TRUE);                                             00721613
        WRITE(P,HDR);                                                   00721614
        GO EOJ; % VOID THIS CARD TO DUMP SEPTIC DISK FILE               00721615
        FILL SEPTIC WITH "SEPTIC ",1;                                   00721620
        WHILE LO DO                                                     00721630
        BEGIN READ(SEPTIC,60,INTSP[*])[EOJ]; I:=0;                      00721640
    LOOP:                                                               00721650
            IF (B:=INTSP[I]).[9:9]=0 THEN GO TO EOJ;                    00721660
            FLL(NAM[B.[7:2]],B.[9:4],B.[14:4],B.[21:12],LINE);          00721670
            TIM(INTSP[BUMPI].[18:30],CC);                               00721680
            PUTLSTATUS(LSTATUS[2×INTSP[I].[13:5]],LINE[10]);            00721685
            WRITE(P,12,LINE[*]);                                        00721690
            IF (B:=B.[33:15])≠0 THEN                                    00721700
            BEGIN WHILE B GTR 14 DO                                     00721710
                BEGIN MMM(14,I); B:=B-14; END;                          00721720
                MMM(B,I);                                               00721730
            END;                                                        00721740
            WRITE(P);                                                   00721750
            IF I≠59 THEN                                                00721760
            IF GLUNK(INTSP[BUMPI]) THEN ELSE GO TO LOOP;                00721770
        END;                                                            00721780
    EOJ:  IF LO THEN                                                    00721790
        BEGIN CLOSE(SEPTIC); LO:=FALSE;                                 00721800
            IF OPERAND(FROM+125,I) AND I NEQ 0 THEN                     00721810
            IF OPERAND(FROM+62,I) AND (I=0 OR I=64) THEN                00721820
            BEGIN FROM:=FROM+I;                                         00721830
                IF (B:=512-(R:=FROM.COL)) LSS (I:=60) THEN              00721840
                BEGIN MOVE(M[FROM.ROW,R],INTSP,B);                      00721850
                    R:=0; I:=60-B; FROM:=FROM+60;                       00721860
                END ELSE B:=0;                                          00721870
                MOVE(M[FROM.ROW,R],INTSP[B],I);                         00721880
```

```
                I:=0; GO TO LOOP;                                          00721890
END; END; END;                                                             00721900
PROCEDURE DUMPSELECTEDARRAYS;                                              00722000
BEGIN                                                                      00723000
    IF NOTABLES THEN ELSE                                                  00723100
    BEGIN                                                                  00723200
    WRITE(P[PAGE]);                                                        00724000
    DUMPARRAY(LINETABLE);                                                  00725000
    DUMPARRAY(STATABLE);                                                   00726000
    DUMPARRAY(SEQARRAY);                                                   00727000
    DUMPARRAY(INPUTANK);                                                   00728000
    DUMPARRAY(TANKS);                                                      00729000
    DUMPARRAY(TNAOG);                                                      00730000
    END;                                                                   00730100
    IF SAVERESULTS THEN                                                    00731000
    BEGIN                                                                  00732000
        DUMPRESULTHOLDER;                                                  00732100
    END;                                                                   00737000
    IF ARGH NEQ 0 THEN                                                     00737100
    BEGIN                                                                  00737200
        DUMPCESSPOOL;                                                      00737300
    END;                                                                   00737400
END OF DUMPING SELECTED ARRAYS;                                            00738000
    BOOLEAN PROCEDURE OPERAND(AT,WHAT); %                                  00739000
        VALUE AT; %                                                        00740000
        INTEGER AT; %                                                      00741000
        REAL WHAT; %                                                       00742000
    BEGIN %                                                                00743000
        INTEGER R,C; %                                                     00744000
        IF FLGBIT(M[R:=AT.ROW,C:=AT.COL]) THEN %                           00745000
        OPERAND:=FALSE %                                                   00746000
        ELSE %                                                             00747000
        BEGIN %                                                            00748000
            WHAT:=M[R,C]; %                                                00749000
            OPERAND:=TRUE; %                                               00750000
        END; %                                                             00751000
    END OPERAND; %                                                         00752000
    BOOLEAN PROCEDURE DESCRIPTOR(AT,WHAT,KIND);                            00753000
        VALUE AT;                                                          00754000
        INTEGER AT;                                                        00755000
        REAL WHAT,KIND;                                                    00756000
    BEGIN                                                                  00757000
        INTEGER R,C;                                                       00758000
        IF (KIND:=OCTAL(CHRS(M[R:=AT.ROW,C:=AT.COL],0,1)))≥"40" THEN       00759000
        BEGIN                                                              00760000
            DESCRIPTOR:=TRUE;                                              00761000
            WHAT:=CHRS(M[R,C],1,7);                                        00762000
        END;                                                               00763000
    END DESCRIPTOR;                                                        00764000
    BOOLEAN PROCEDURE PDATADESC(AT,WHAT);                                  00765000
        VALUE AT;                                                          00766000
        INTEGER AT;                                                        00767000
        REAL WHAT;                                                         00768000
    BEGIN                                                                  00769000
        INTEGER KIND;                                                      00770000
        IF DESCRIPTOR(AT,WHAT,KIND) AND                                    00771000
           KIND="50"      THEN                                             00772000
        PDATADESC:=TRUE;                                                   00773000
    END;                                                                   00774000
BOOLEAN PROCEDURE CONTRULDESC(AT,WHAT);                                    00775000
```

```
        VALUE AT;                                                       00776000
        INTEGER AT;                                                     00777000
        REAL WHAT;                                                      00778000
    BEGIN                                                               00779000
        INTEGER TYP;                                                    00780000
        CONTROLDESC:=DESCRIPTOR(AT,WHAT,TYP) AND                        00781000
                    TYP.[40:1]=1  AND                                   00782000
                    TYP.[45:1]=0;                                       00783000
    END CONTROLDESC;                                                    00784000
        BOOLEAN PROCEDURE MOMOK(MOMPRT,MOMLOC);                         00785000
            VALUE MOMPRT; INTEGER MOMPRT,MOMLOC;                        00786000
            MOMOK :=PDATADESC(MOMPRT,MOMLOC) AND (MOMLOC:=MOMLOC.CF)GTR 0; 00787000
        INTEGER MINLNK,MINBAD,MAXBAD,MAXLNK; %                          00788000
        BOOLEAN NEEDCHECKAVAILNKS;%FALSE IF DPMT DUMP                   00789000
         ARRAY COMMT[0:19]; BOOLEAN COMNT;                              00790000
              PROCEDURE MCPENTRIES;                                     00791000
                BEGIN                                                   00792000
                SGLTOG:=TRUE;                                           00792100
                DISPLAY(BED,FALSE);                                     00793000
                DISPLAY(PRT,FALSE);                                     00794000
                DISPLAY(JAR,FALSE);                                     00795000
                DISPLAY(FENCE  ,FALSE);                                 00796000
                DISPLAY(LOGLINE,FALSE);                                 00797000
                DISPLAY(LOGLINE2,FALSE);                                00797100
                DISPLAY(SPACESTACK,FALSE);                              00798000
                DISPLAY(STACKQ,FALSE);                                  00799000
                DISPLAY(ISTACK,FALSE);                                  00800000
                DISPLAY(LCGARRAY,FALSE);                                00801000
                DISPLAY(SPACER,FALSE);                                  00802000
                DISPLAY(BIGUNS,FALSE);                                  00802100
                DISPLAY(P1MIX,FALSE);                                   00803000
                DISPLAY(P2MIX,FALSE);                                   00804000
                SGLTOG:=FALSE;                                          00804999
                END;                                                    00805000
            PROCEDURE GETPRTENTRIES;                                    00806000
             BEGIN                                                      00807000
                REAL ADR,WC,PRTROW,L;                                   00808000
                REAL ADRR;                                              00809000
                INTEGER I,K;                                            00810000
                ARRAY MIX[0:40];                                        00811000
                FORMAT PRTOUT(X10,"PRT LOCATIONS:"/                     00812000
                      X20,"MIX",X20,"PRT"/),                            00813000
                   F1( X20, A2,X20, A5);                                00814000
                FORMAT BADPRT("********BAD PRT DESCRIPTOR*******");     00815000
                LABEL XIT;                                              00816000
                LABEL NEXT;                                             00817000
        IF NOT(PDATADESC(PRT,L) AND L.CF NEQ 0 AND L.[8:10] GTR 0       00818000
             AND L.[8:10] LSS 41) THEN BEGIN                            00819000
              MIXMAX:=DEFINEDMIXMAX;%                  FOR LINK CK      00820000
                WRITE (P,BADPRT); GO XIT END;                           00821000
             ADR:= L.[33:15];                                           00822000
             WC:= L.[8:10];                                             00823000
             K:=-1;                                                     00824000
             MIXMAX:=WC-1;                                              00825000
            FOR I:= 1 STEP 1 UNTIL (WC-1) DO BEGIN                      00826000
             IF PDATADESC(ADR+I,PRTROW) THEN ELSE PRTROW:=0;            00827000
                IF PRTROW =0 THEN                                       00828000
                   GO TO NEXT;                                          00829000
                K:=K+1;                                                 00830000
                MIX[K]:= I; % SAVE MIX NUMBER                           00831000
```

```
            MIX[K:=K+1]:= PRTROW;                                                00832000
            NEXT:                                                                00833000
            END; % OF FINDING PRTS & VALID MIXES;                                00834000
          WRITE(P,PRTOUT);                                                       00835000
          WRITE(P,F1,FOR I:=0 STEP 1 UNTIL K DO OCTAL(MIX[I],[33:15]));          00836000
           XIT:                                                                  00837000
         END OF GETPRTENTRIES;                                                   00838000
    INTEGER STREAM PROCEDURE MCPCNT(A);  VALUE A;                                00839000
    BEGIN SI:=LOC A; SI:=SI+1;                                                   00840000
        7(IF TOGGLE THEN TALLY:=TALLY+1 ELSE IF SC NEQ "0" THEN                  00841000
          TALLY:=TALLY+1;SI:=SI+1);    MCPCNT:=TALLY;                            00842000
    END OF MCPCNT;%                                                              00843000
    INTEGER PRTCODE; % SET TO 0,1 OR 2 BY CHECKPRTFILE                           00843100
    PROCEDURE DATIME;                                                            00844000
    BEGIN INTEGER I; BOOLEAN B;                                                  00845000
        STREAM PROCEDURE PRTFILEMESS(P,L,C);    VALUE C;                         00845100
        BEGIN LABEL MAYBE,BAD,FIX,XIT;                                           00845105
            SI:=P; DI:=L; DS:=4 WDS;                                             00845110
            CI:=CI+C;                                                            00845115
            GO XIT; % 0                                                          00845120
            GO MAYBE; % 1                                                        00845125
            GO BAD; % 2                                                          00845130
            MAYBE:DS:=9LIT"(APPEARS "; GO FIX;                                   00845135
            BAD:  DS:=12LIT"(DEFINITELY "; GO FIX;                               00845140
            FIX   :DS:=10LIT"INCORRECT)";                                        00845145
            XIT   :DS:=32LIT" ";                                                 00845150
        END PRTFILEMESS;                                                         00845155
          IF NOT COMMON THEN BEGIN                                               00846000
           IF NOT OPERAND(3,MCPVERSION) THEN WRITE(P,BADCELL3);                  00847000
            IF FLGBIT(M[DATE.ROW,DATE.COL]) THEN                                 00848000
             WRITE(P,BADDATE) ELSE                                               00849000
               BEGIN                                                             00850000
                  DATX := M[DATE.ROW,DATE.COL];                                  00851000
                  DATES(DATX,DATETAKEN,DAYS,YEARS);                              00852000
               END;  % OF DATE                                                   00853000
          IF FLGBIT(M[XCLOCK.ROW,XCLOCK.COL]) THEN                               00854000
       WRITE (P,BADXCLOCK) ELSE                                                  00855000
               BEGIN                                                             00856000
                  XCLOCX := M[XCLOCK.ROW,XCLOCK.COL];                            00857000
                  TIMES(XCLOCX,TIMETAKEN,MINUTES,SECONDS);                       00858000
               END;    %     OF XCLOCK                                           00859000
           PRTFILEMESS(PRTNAME,LINE,PRTCODE);                                    00859050
            WRITE(P,8,LINE[*]);                                                  00859100
            WRITE(P,FANAL,VERSION);                                              00859200
           WRITE(P,FTAPDSK,FILETYPE,TDMFID.[6:6],TDMFID,                         00859500
                                    TDFID.[6:6],TDFID);                          00859510
         WRITE(P,FCOMVAL,REAL(COM));                                             00859520
      WRITE(P,X1MARKXI,IF B:=(I:=MCPCNT(MCPVERSION) GTR 6)                       00860000
            THEN MCPVERSION.[6:6] ELSE " ", IF B THEN 6 ELSE IF I=0              00861000
            THEN 1 ELSE I,MCPVERSION,DATETAKEN,DAYS,YEARS,                       00862000
            TIMETAKEN,MINUTES,SECONDS);                                          00863000
       END;                                                                      00864000
         IF SHAREDISK THEN                                                       00864100
         IF OPERAND(SYSNO,I) AND I GEQ 0 AND I LEQ 3 THEN                        00864110
         WRITE(P,SYSID,I+17);                                                    00864120
        DATES(TIME(0),DATEANALYZED,DAYS,YEARS);                                  00865000
         TIMES(TIME(1),TIMEANALYZED,MINUTES,SECONDS);                            00866000
   WRITE(P,FMXX,DATEANALYZED,DAYS,YEARS,TIMEANALYZED,MINUTES,SECONDS);           00867000
         IF COMMON THEN WRITE(P);                                                00868000
         IF NOT COMMON THEN BEGIN                                                00869000
```

```
          IF FLGBIT(M[CLOCK.ROW,CLOCK.COL]) THEN                          00870000
            WRITE(P,BADCLOCK) ELSE                                        00871000
              BEGIN                                                       00872000
1                 CLOCX := M[CLOCK.ROW,CLOCK.COL];                        00873000
2               IF XCLOCX LSS CLOCX THEN XCLOCX:=XCLOCX+5184000;          00874000
3               TIMES(XCLOCX-CLOCX,TIMELASTHL,MINUTES,SECONDS);           00875000
4                 WRITE(P,FMX2,TIMELASTHL,MINUTES,SECONDS);               00876000
5                 TIMES(CLOCX,SINCSLASTHL,MINUTES,SECONDS);               00877000
6                 WRITE(P[DBL],FMX3,SINCSLASTHL,MINUTES,SECONDS);         00878000
7             END ; %OF CLOCK                                             00879000
8           END;                                                          00880000
9     END;% OF DATIME                                                     00881000
10    PROCEDURE PRINTARRAYROW(PRTLOC,INX); VALUE PRTLOC,INX;              00881100
11                                         INTEGER PRTLOC,INX;            00881105
12    BEGIN                                                               00881110
13        INTEGER LOC,SIZE;                                               00881115
14        BOOLEAN TOG,COMSAVE,DBLSAVE;                                    00881120
15        LABEL AGAIN;                                                    00881125
16        STREAM PROCEDURE FIXLINE(LINE,INX); VALUE INX;                  00881130
17        BEGIN LOCAL A,B;                                                00881135
18            SI:=LINE;4(SI:=SI+B);                                       00881140
19            32(IF SC=" " THEN JUMP OUT ELSE SI:=SI+1);                  00881145
20            A:=SI; DI:=A; DS:=LIT"["; A:=DI;                            00881150
21            SI:=LOC INX; DS:=4 DEC; B:=DI;                              00881155
22            DI:=A; DS:=3 FILL; DI:=B;                                   00881160
23            DS:=3LIT",*]";                                              00881165
24        END FIXLINE;                                                    00881170
25        IF TOG:=BOOLEAN(INX,[1:1]) THEN                                 00881175
26        DISPLAY(PRTLOC,TRUE);                                           00881180
27        INX:=ABS(INX);                                                  00881185
28        COMSAVE:=COM;                                                   00881190
29        COM:=FALSE;                                                     00881195
30        DBLSAVE:=DOUBLESPACE;                                           00881197
31        DOUBLESPACE:=TRUE;                                              00881200
32        IF PRTLOC GEQ 129 AND PRTLOC LEQ PRTMAX THEN                    00881205
33    AGAIN:                                                              00881210
34        IF PDATADESC(PRTLOC,LOC) THEN                                   00881215
35        IF (SIZE:=LOC.[8:10]) NEQ 0 THEN                                00881220
36        IF TOG THEN PRINT(LOC:=LOC.CF,MIN(MAXCOR,LOC+SIZE)) ELSE        00881225
37        BEGIN                                                           00881230
38            DISPLAYIT(PRTLOC,TRUE,PRTLOC:=LOC.CF+INX);                  00881235
39            FIXLINE(LINE,INX);                                          00881240
40            WRITE(P[DBL],15,LINE[*]);                                   00881245
41            TOG:=TRUE;                                                  00881250
42            GO AGAIN;                                                   00881255
43        END;                                                            00881260
44        COM:=COMSAVE;                                                   00881265
45        DOUBLESPACE:=DBLSAVE;                                           00881267
46    END PRINTARRAYROW;                                                  00881270
47    DEFINE PRINTARRAY(PRINTARRAY1)=                                     00881275
48        PRINTARRAYROW(PRINTARRAY1,-1)#;                                 00881280
49    DEFINE PA=PRINTARRAY#,PAROW=PRINTARRAYROW#;                         00881285
50    BOOLEAN BADCOMMENT;                                                 00882000
51    FORMAT COMNTPAR("---PARITY ERROR OCCURRED IN COMMENTS BLOCK..."); 00883000
52        PROCEDURE LOAD;                                                 00884000
53        BEGIN                                                           00885000
54            LABEL EOT;                                                  00886005
55            LABEL EF,PR,IGPAR;                                          00886010
56            LABEL FLAG,PAR,BADTM,IGNOREPAR;%                            00887000
57            FORMAT BADEOF("---INVALID EOF AFTER BLOCK # ",I2),          00888000
```

```
        PRTFILE("PRT FILE USED IS ",A1,A6,"/",A1,A6,X8),                00888100
        PARERR ("---IRRECOVERABLE PARITY IN BLOCK # ",I2),              00889000
        FLAGERR("---FLAG BIT IN WORD ZERO OF BLOCK # ",I2);             00890000
ARRAY A[0:532];                                                         00891000
STREAM PROCEDURE MOVER(S,D);                                            00892000
BEGIN                                                                   00893000
    SI:=S; DI:=D;                                                       00894000
    16(DS:=32 WDS);                                                     00895000
END MOVER;                                                              00896000
STREAM PROCEDURE GETCOMMON(COMMT,N);                                    00896010
BEGIN LOCAL X,Y; LABEL XIT,HIT,GETNUM;                                  00896020
DI:=N; DS:="8LIT"+"0000001"; SI:=COMMT;                                 00896030
7(20(IF SC="C" THEN BEGIN SI:=SI+1;                                     00896040
     IF SC="C" THEN BEGIN SI:=SI+1;                                     00896050
     IF SC="M" THEN JUMP OUT 2 TO HIT ELSE SI:=SI+1                     00896060
     END ELSE SI:=SI+1 END ELSE SI:=SI+1));                             00896070
GO XIT;                                                                 00896080
HIT:15(IF SC="=" THEN JUMP OUT TO GETNUM ELSE SI:=SI+1); GO XIT;        00896090
GETNUM:SI:=SI+1;                                                        00896100
DI:=LOC X;                                                              00896110
10(IF SC=" " THEN SI:=SI+1 ELSE JUMP OUT);                              00896120
8(IF SC GEQ "0" THEN IF SC LEQ "9" THEN                                 00896130
BEGIN TALLY:=TALLY+1; DS:=CHR END ELSE JUMP OUT ELSE JUMP OUT);         00896140
Y:=TALLY;                                                               00896150
SI:=LOC X;                                                              00896160
DI:=N;                                                                  00896170
DS:=Y OCT;                                                              00896180
XIT: END GETCOMMON;                                                     00896190
INTEGER I;                                                              00897000
 ALPHA N1,N2;                                                           00897010
 STREAM PROCEDURE KLUDGE(XNAMS);                                        00897100
BEGIN DI:=XNAMS; 11(DS:="8LIT" ") END;                                  00897200
COMNT←BADCOMMENT←FALSE;                                                 00898000
IF RELOAD THEN                                                          00898110
FOR I:=0 STEP 1 UNTIL 63 DO UNLOCK(M[A[0].ROW,*]);                      00898120
IF MULTI THEN                                                           00898130
READ REVERSE(MDUMP,20,COMMTL*])[BADTM:IGPAR];                           00898140
IF FALSE THEN IGPAR:BADCOMMENT:=TRUE;                                   00898150
FOR I:=0 STEP 1 UNTIL 63 DO                                             00899000
BEGIN                                                                   00900000
    IF MULTI THEN READ REVERSE(MDUMP,533,A[*])[BADTM:PAR] ELSE          00900100
    READ(MDUMP,533,A[*])[BADTM:PAR];%                                   00901000
    IF FLGBIT(A) THEN GO FLAG ELSE                                      00902000
    IF MODON[A[0].[33:3]]:=NOT BOOLEAN(A[0].[1:1]) THEN                 00903000
    BEGIN                                                               00904000
        MOVER(A[1],M[A[0].ROW,0]);                                      00906000
        LOCK (M[A[0].ROW,*]);                                           00906500
    END ELSE IF A[0].[3:5]=0 THEN I:=I+7; %                             00907000
    COMMENT ONE BLOCK IS WRITTEN PER ABSENT MOD;                        00907100
END;                                                                    00908000
TDMFID:=MDUMP.MFID;                                                     00908100
TDFID:=MDUMP.FID;                                                       00908110
FILETYPE:=IF MDUMP.TYPE=2 THEN "TAPE" ELSE "DISK";                      00908120
IF NOT MULTI THEN                                                       00908130
IF FILETYPE="TAPE" AND TDMFID NEQ 0 THEN                                00908140
IF TDFID.[30:18]=1 THEN ELSE MULTI:=TRUE;                               00908150
IF REPEATING THEN IF TDFID.[30:18]=1 THEN CLOSE(MDUMP)                  00908160
ELSE CLOSE(MDUMP,*) ELSE                                                00908170
IF (AUXTYPE:=A[0].[3:5])=0 THEN                                         00908180
IF FILETYPE="TAPE" THEN                                                 00908190
```

```
 READ(MDUMP,2C,COMMT[*])[EOT:IGNOREPAR] ELSE % DISK                    00909000
 MOVE(A[513],COMMT,20);                                                00909100
IF FALSE THEN IGNOREPAR:BADCOMMENT←TRUE;                               00910000
COMNT := TRUE;                                                         00911000
 GETCOMMON(COMMT,N1);                                                  00911100
 IF N1 GEQ 0 THEN COM:=BOOLEAN(N1); N1:=0;                             00911110
  IF MULTI THEN NOMO:=TRUE ELSE                                        00911140
  READ(MDUMP[NC])[EOT:PAR];                                            00911150
  IF FALSE THEN                                                        00911160
  BEGIN                                                                00911170
  EOT: CLOSE(MDUMP);                                                   00912000
  NOMO:=TRUE;                                                          00912050
  END;                                                                 00912100
  IF COMMON OR RELOAD THEN                                             00913000
  ELSE                                                                 00914000
  BEGIN                                                                00915000
 IF PRT32 THEN READARRAY(30,SEGZERO[*],0) ELSE                         00917000
   SPACE(DISK,1);                                                      00918000
   READARRAY(NAMESIZE,NAME[*],PRTBASE);                                00919000
   READARRAY(NAMSSIZE,NAMS[*],0);                                      00920000
 READARRAY(INAMESIZE,INAME[*],0);                                      00921000
   READARRAY(INAMSSIZE,INAMS[*],0);                                    00922000
 CLOSE(DISK);                                                          00923000
  N1:=DISK.MFID; N2:=DISK.FID;                                         00923100
   WRITE(PRTNAME[*],PRTFILE,N1.[6:6],N1,N2.[6:6],N2);                  00923110
 END;                                                                  00924000
     IF RELOAD THEN ELSE                                               00924100
     BEGIN                                                             00924200
 MAXMOD:=7; %                                                          00925000
 WHILE NOT MODON[MAXMOD] DO MAXMOD:=MAXMOD-1;                          00926000
 MAXCOR:=4096×(MAXMOD+1)-1; %                                          00927000
IF COMMON THEN ELSE                                                    00928000
BEGIN                                                                  00929000
 PRTMAX:=PRTBASE+NAMESIZE-1;                                           00930000
 INTMAX:=INAMESIZE-1;                                                  00931000
 FOR I:=PRTBASE STEP 1 UNTIL PRTMAX DO NSNAME[I]:=                     00932000
 NSNAME[I]+PRTBASE;                                                    00933000
 FOR I:=0 STEP 1 UNTIL INTMAX DO ISNAME[I]:=                           00934000
 ISNAME[I]+PRTBASE;                                                    00935000
 FIXDEFINES;                                                           00936000
 SETUPXNAMEANDXNAMS;                                                   00937000
 IF DAATE=0 THEN % ASSUME PRT BEGINS AT @160                           00937005
 BEGIN                                                                 00937010
      DATE:=119;                                                       00937015
      CLOCK:=120;                                                      00937020
      XCLOCK:=121;                                                     00937025
 END ELSE                                                              00937030
 BEGIN % ASSUME PRT BEGINS AT @173                                     00937035
      DATE:=DAATE;                                                     00937040
      CLOCK:=CLOOCK;                                                   00937045
      XCLOCK:=XCLOOCK;                                                 00937050
      KLUDGE(XNAMS);                                                   00937055
 END;                                                                  00937060
 IF MYSTACKADR LSS 0 THEN % LETS USE IT...                             00937100
 IF CONTROLDESC(7,I) THEN % CHECK FOR MSCW IN CELL 7                   00937110
 IF I.CF=0 AND I:=I.FF GEQ 0 THEN % ASSUME VALID                       00937120
  MYSTACKADR:=I+30; % MAY NEED MORE THAN 30                            00937130
END;                                                                   00938000
     END;                                                              00938100
 IF FALSE THEN BEGIN                                                   00939000
```

```
      BADTM:WRITE(SPO,BADEOF,I); GO EOPROG;                                    00940000
      PAR:   WRITE(SPO,PARERR,I+1); GO EOPROG;                                 00941000
      FLAG:  WRITE(SPO,FLAGERR,I+1); GO EOPROG;  END;                          00942000
    END LOAD;                                                                  00943000
PROCEDURE CHECKPRTFILE;%                                                       00944000
% SIGNALS USER IF HIS TSS/PRT FILE *MIGHT* BE THE WRONG ONE                    00945000
BEGIN                                                                          00946000
REAL LOC,TYP,I;                                                                00947000
REAL MYTYPE;                                                                   00948000
LABEL BAD,MAYBE;                                                               00949000
DEFINE CHECKFORLABELDESCRIPTOR=IF LOC LSS 0 THEN GO BAD ELSE                   00950000
       IF DESCRIPTOR(LOC,1,TYP) AND TYP=MYTYPE THEN ELSE GO MAYBE#;            00951000
DEFINE CHECKFORPROGRAMDESCRIPTOR=CHECKFORLABELDESCRIPTOR#;                     00952000
FORMAT BADPRT("---YOUR TSS/PRT FILE IS WRONG..."),                             00953000
INCOMPAT("---TSS/PRT FILE NOT COMPATIBLE WITH MCPS PRT IN MEMORY...");         00954000
DEFINE ERRMESS(ERRMESS1)=BEGIN WRITE(SPO,ERRMESS1);                            00955000
              WRITE(P[PAGE],ERRMESS1) END#;%                                   00956000
MYTYPE:="76";                                                                  00957000
IF (LOC:=NOTHINGTODO) NEQ 0 THEN                                               00958000
CHECKFORLABELDESCRIPTOR;                                                       00959000
IF (LOC:=STACKOVERFLOW) NEQ 0 THEN                                             00960000
CHECKFORLABELDESCRIPTOR;                                                       00961000
IF (LOC:=RETURN) NEQ 0 THEN                                                    00962000
CHECKFORLABELDESCRIPTOR;                                                       00963000
MYTYPE:="75";                                                                  00964000
IF (LOC:=OTHERCLOSE)=0 THEN LOC:=-1;                                           00965000
CHECKFORPROGRAMDESCRIPTOR;                                                     00966000
IF FALSE THEN MAYBE:BEGIN ERRMESS(INCOMPAT); PRTCODE:=1 END;                   00967000
IF FALSE THEN BAD:BEGIN ERRMESS(BADPRT); PRTCODE:=2 END;                       00968000
END OF CHECKPRTFILE;                                                           00969000
PROCEDURE DUMPMCPSPRT;                                                         00970000
BEGIN                                                                          00971000
FORMAT HDR(X42,"T S M C P S  P R T"/                                           00972000
           X42,"= = = = = =  = = ="///,                                        00973000
           2("PRT",X5,"CONTENTS",X16,"NAME",X28)/),                            00974000
             DASHES(X42,9("= ")),                                              00975000
        F(X8,"MEMORY MODS: ",8I1),                                             00976000
        PRTITEM(2(A5," = ",0,X1,0,X39));                                       00977000
INTEGER LOC,N;                                                                 00978000
LABEL EXIT;                                                                    00978050
IF DUMPCESSPOOLONLY THEN DONTDUMPRT:=TRUE;                                     00978100
IF NOT COMMON THEN                                                             00979000
BEGIN                                                                          00980000
CHECKPRTFILE;                                                                  00981000
IF NOT DONTDUMPRT THEN BEGIN DATIME; WRITE(P[DBL]) END;                        00982000
PRINTMCPOPTIONS;                                                               00983000
IF DONTDUMPRT THEN ELSE                                                        00984000
BEGIN                                                                          00985000
   WRITE(P[DBL],HDR);                                                          00986000
   FOR LOC:=ACTUALPRTBASE STEP 1 UNTIL PRTBASE DO                              00987000
   BEGIN                                                                       00988000
      WRITE(LINE[*],PRTITEM,OCTAL(LOC),OCTAL(HIHALF(LOC)),                     00989000
        OCTAL(LOHALF(LOC)),OCTAL(N:=XSNAME[LOC]),                              00990000
        OCTAL(HIHALF(N)),OCTAL(LOHALF(N)));                                    00991000
      MOVE(XNAMS[XNAME[LOC].CF],LINE[4],XNAME[LOC].FF);                        00992000
      MOVE(XNAMS[XNAME[N].CF],LINE[12],XNAME[N].FF);                           00993000
      WRITE(P,15,LINE[*]);                                                     00994000
      IF DONTDUMPRT THEN GO EXIT;                                              00994100
   END;                                                                        00995000
   WRITE(P,DASHES);                                                            00996000
```

```
      FOR LOC:=PRTBASE+1 STEP 1 UNTIL PRTMAX DO                          00997000
      BEGIN                                                              00998000
          WRITE(LINE[*],PRTITEM,OCTAL(LOC),OCTAL(HIHALF(LOC)),           00999000
             OCTAL(LOHALF(LOC)),OCTAL(N:=NSNAME[LOC]),                   01000000
             OCTAL(HIHALF(N)),OCTAL(LOHALF(N)));                         01001000
          MOVE(NAMS[NAME[LOC],CF],LINE[4],NAME[LOC],FF);                 01002000
          MOVE(NAMS[NAME[N],CF],LINE[12],NAME[N],FF);                    01003000
          WRITE(P,15,LINE[*]);                                           01004000
          IF DONTDUMPR1 THEN GO EXIT;                                    01004100
      END;                                                               01005000
    EXIT:                                                                01005100
      NEXTPAGE;                                                          01006000
    END;                                                                 01007000
    END;                                                                 01008000
    WRITE(P[DBL],F,FOR N:=0 STEP 1 UNTIL 7 DO                            01009000
         [IF MODON[N] THEN N ELSE 10]);                                  01010000
    DATIME;                                                              01011000
    IF DUMPCESSPOOLONLY THEN ELSE                                        01011100
    IF NOT COMMON THEN                                                   01012000
    BEGIN                                                                01013000
      MCPENTRIES;                                                        01014000
      GETPRTENTRIES; NEXTPAGE;                                           01015000
    END;                                                                 01016000
    END OF DUMP OF MCPS PRT;                                             01017000
      ARRAY MEMLOC[0:MIXMAX],LINKLOC[0:MIXMAX];                          01018000
      COMMENT .....  MEMLOC[I] HOLDS THE START CELL AND LAST             01019000
            CELL ADDRESSES FOR MEMORY AREA I ....                        01020000
        MEMLOC[I],[3:1] -- 0,TH LINK IN AREA I IS BAD,                   01021000
                ,[5:1] -- BAD LINK(S) IN AREA, LAST GOOD                 01022000
                          LINK IS IN LINKLOC[I],                         01023000
                ,[4:1] -- LAST LINK IN AREA I IS BAD,                    01024000
          ,[2:1] -- STATUS OF AVAILABLE LINKS,                           01025000
         ,[13:5] -- MIX INDEX FOR AREA I                                 01026000
                ,[18:15] -- ADDRESS OF LAST CELL OF AREA,                01027000
                ,[33:15] -- ADDRESS OF 0,TH LINK IN AREA,                01028000
        LINKLOC[I] HOLDS ADDRESSES OF LAST GOOD LINKS                    01029000
                    IN FORWARD SEARCH IN LINKLOC[I].[33:15],             01030000
                    IN BACK SEARCH IN LINKLOC[I].[18:15]  ;              01031000
    REAL MSTART,MEND,VFENCE,PRTO;                                        01032000
    COMMENT ... MSTART  IS THE ADDRESS OF THE                            01033000
          0,TH MEM LINK (ALSO THE FIRST CELL) OF AN AREA,                01034000
          MEND IS THE ADDRESS OF THE LAST CELL OF AN AREA,               01035000
          THESE VALUES ARE HELD IN MEMLOC. ...;                          01036000
    BOOLEAN PROCEDURE CHECKMEMORYLINKS;                                  01037000
      BEGIN %                                                            01038000
    COMMENT .... THIS PROCEDURE HAS BEEN CHANGED TO CHECK                01039000
          MEM LINKS IN ALL AREAS IN CORE, FIRST WE ACCESS                01040000
          SQ TO DETERMINE WHICH JOBS ARE ACTUALLY IN CORE                01041000
          WE THEN COMPUTE THE ADDRESS OF THE FIRST                       01042000
          CELL AND THE LAST CELL IN A MEMORY AREA AND                    01043000
          STORE THESE VALUES IN MEMLOC, THE MEMLOC                       01044000
          ENTRIES ARE THEN ARRANGED IN ASCENDING                         01045000
          SEQUENCE, WE THEN PERFORM THE NORMAL MEMORY                    01046000
          LINK CHECKING PROCEDURE MAKING ENTRIES IN                      01047000
          MEMLOC AND LINKLOC TO REFLECT VARIOUS PROBLEMS                 01048000
          IN THE LINKS FOR USE IN PRINTING OUT CORE.  ;                  01049000
        BOOLEAN ZEROK; REAL VO; %                                        01050000
        BOOLEAN MSTARTOK; REAL VMSTART; %                                01051000
        BOOLEAN NOWRAPAROUND;                                            01051100
        BOOLEAN NEWTYPE;                                                 01051200
```

```
        REAL LINK,VLINK,PREVLINK;                                          01052000
        INTEGER AVAILN,AVAILT,AVAILM;                                      01053000
        DEFINE TYPMAX=22#; %                                               01054000
        BOOLEAN PROCEDURE LINKOK(WORD); %                                  01055000
            VALUE WORD; %                                                  01056000
            REAL WORD; %                                                   01057000
        LINKOK:=WORD.[3:6]≤TYPMAX AND %                                    01058000
    WORD.[9:6] LEQ MIXMAX       AND                                        01059000
    WORD.[15:2]=0;                                                         01060000
    BOOLEAN PROCEDURE FENCEOK(F);                                          01061000
    VALUE F;  REAL F;                                                      01062000
      FENCEOK:=F.[38:10]=0 AND F≠0;                                        01063000
        FORMAT RANGE(X8,"LINKS FROM ",A5," TO ",A5," ARE OK"),             01064000
               FCORE(A5," = ",2(O,X1),X7,"CORE",X8,"FACTOR = ",F4.2,       01064100
              ". MAX CORE = ",A5,"(",I5,")","USING ",A5,"(",I5,")"),       01064110
               ONEOK(X8,"LINK AT ",A5," IS OK"),                           01064500
               TELLM(X5,"END OF MEMORY LINK CHECK FOR MIX = ",             01064600
                     I2,3A6),                                              01064700
               BAD(X8,A6," LINK AT ",A5," IS NOT OK"),                     01065000
               AVLBAD(X8,"AVAILABLE STORAGE TOTALS DO NOT CROSS CHECK"),   01066000
               AVL(X8,"AVAILABLE LINKS ARE OK, TOTALING",                  01067000
              I4,"(DECIMAL) AREAS OF ",A5,"(",I5,                          01068000
              ") WORDS UP TO ",A5,   "(",I5,") WORDS EACH");               01069000
        ARRAY A[0:3];                                                      01070000
        REAL SQPTR,DUMY,AVAIL,                                             01071000
        LEFTLIT,CNTR,SQPTR1;                                               01072000
        INTEGER N,T,M1,I,J;                                                01073000
         DEFINE S = SQPTR1.ROW,SQPTR1.COL#;                                01076000
      PROCEDURE PRINTIT(ADDR,NAME);                                        01077000
       VALUE ADDR; REAL ADDR,NAME;                                         01078000
       BEGIN                                                               01079000
        INTEGER DUMY,DUM1;                                                 01060000
        FORMAT ITEM1(A5," = ",2(O,X1),X5);                                 01081000
        WRITE(LINE[*],ITEM1,OCTAL(ADDR),                                   01082000
         OCTAL(DUMY:=HIHALF(ADDR)),OCTAL(DUM1:=LOHALF(ADDR)));             01083000
        MOVE(NAME,LINE[4],1);                                              01084000
        WRITE(P[DBL],7,LINE[*]);                                           01085000
                  END PRINTIT;                                             01086000
     FILL A[*] WITH " MSTART "," LEFTLIT"," AVAIL  ",                      01087000
                    " MEND   ";                                            01088000
      CNTR:=0;                                                             01089000
     IF NOT (ZEROK:=OPERAND(FENCE,VFENCE) AND FENCEOK(VFENCE))             01090000
    THEN IF NOT (ZEROK:=(OPERAND(2,VFENCE) AND                             01091000
       FENCEOK(VFENCE:=VFENCE.CF+2))) THEN                                 01092000
      ZEROK:=OPERAND(0,VFENCE) AND FENCEOK(VFENCE:=VFENCE,FF+3);           01093000
    CHECKMEMORYLINKS:=ZEROK;                                               01094000
    IF PDATADESC(SQ,SQPTR) THEN  BEGIN  SQPTR:=SQPTR,CF;                   01095000
            FOR I:=0 STEP 1 UNTIL MIXMAX DO BEGIN                          01096000
       SQPTR1:=SQPTR + I;                                                  01097000
     DUMY:=CHRS(M[S],3,1); IF OPERAND(SQPTR1,V0) AND V0≠0 AND              01098000
    BOOLEAN(V0.[1:1]) THEN IF DUMY=0 OR DUMY=1 OR DUMY=4 OR DUMY=5 OR      01099000
          DUMY=16 OR DUMY=32 THEN BEGIN                                    01100000
           MSTART:=VFENCE +1024×CHRS(M[S],6,1);                            01101000
            IF ( MEND:= VFENCE+1024+1024×CHRS(M[S],5,1)=3 )                01102000
                  GTR 1024+ VFENCE THEN                                    01103000
    %**** THE ABOVE TESTS FOR JOBS ABOVE THE FENCE;                         01104000
     MEMLOC[CNTR:=CNTR+1]:=MSTART&I[13:43:5]&MEND[18:33:15];               01105000
      END; END; END; MEMLOC[CNTR+1]:=-7;                                   01106000
      FOR I:=1 STEP 1 UNTIL CNTR-1 DO                                      01107000
        FOR J:=I+1 STEP 1 UNTIL CNTR DO                                    01108000
```

```
 IF MEMLOC[I].CF GTR MEMLOC[J].CF THEN                                  01109000
 BEGIN                                                                  01110000
 VO:=MEMLOC[I];MEMLOC[I]:=MEMLOC[J];                                    01111000
 MEMLOC[J]:=VO; END;                                                    01112000
FOR I :=0 STEP 1 UNTIL CNTR DO                                          01113000
 BEGIN                                                                  01114000
 AVAILN←AVAILT←AVAILM←N←T←M1←0;                                         01115000
  PREVLINK:=MSTART:= MEMLOC[1].CF; MEND:= MEMLOC[I].FF;                 01116000
  LEFTLIT:= MSTART+1; AVAIL:= MSTART+2;                                 01117000
 IF I=0 THEN                                                            01118000
 BEGIN                                                                  01118100
     J←(MEND←VFENCE-3) DIV 4096;                                        01118200
     WHILE NOT MODON[J] DO BEGIN MEND:=MEND-4096; J:=J-1 END;           01118300
     MEMLOC[0].FF:=MEND; % THE ABOVE ALLOWS FOR MISSING MODS            01119000
 END;                                                                   01119100
  IF NEWTYPE THEN ELSE                                                  01119200
  IF OPERAND(MEND+2,DUMY) THEN                                          01119210
  IF DUMY.[1:17]=0 AND DUMY.FF=MEND THEN                                01119220
  NEWTYPE:=TRUE;                                                        01119230
  PRINTIT(MSTART,A[0]);                                                 01120000
  PRINTIT(LEFTLIT,A[1]);                                                01121000
  PRINTIT(AVAIL,A[2]);                                                  01122000
  PRINTIT(MEND,A[3]);                                                   01123000
  IF I=0 THEN IF OPERAND(CORE,DUMY) AND DUMY.[1:3]=0 THEN               01123100
  BEGIN                                                                 01123110
       WRITE(LINE[*],FCORE,OCTAL(CORE),OCTAL(HALF(CORE)),               01123120
            OCTAL(LOHALF(CORE)),DUMY.[4:14]/100,                        01123130
            OCTAL((J:=DUMY.CF×64).CF),J,                                01123140
            OCTAL((J:=DUMY.FF×64).CF),J);                               01123150
       WRITE(P[DBL],15,LINE[*]),                                        01123170
     END ELSE                                                           01123180
     DISPLAY(CORE,FALSE);                                               01123190
     AVALNKOK:=TRUE;                                                    01123500
     IF LNKSOK:=MSTARTOK:=OPERAND(MSTART,VO) AND                        01124000
          VO.[1:2]=1                  AND                               01125000
   VO.[3:12] = MEMLOC[I].[13:5]       AND                               01126000
     (IF I=0 THEN(VO.CF NEQ 0 AND VO.CF LSS 4096) ELSE TRUE) AND        01126100
     VO.FF = (MAXLNK:=MEND ) THEN LINK:=VO.CF ELSE                      01127000
    MEMLOC[I]:=MEMLOC[I] & 1[3:47:1];                                   01128000
      IF I=0 THEN MINLNK:=LINK;                                         01128500
      WHILE LNKSOK AND MAXLNK>PREVLINK DO %                             01129000
      IF LNKSOK:=(OPERAND(LINK,VLINK) AND                               01130000
                  LINKOK(VLINK)         AND %                         01131000
                  VLINK.FF=PREVLINK     AND %                         01132000
                 (IF LINK=MAXLNK                                        01133000
                THEN VLINK.CF = MSTART %                                01134000
                  ELSE VLINK.CF>LINK))THEN                              01135000
     BEGIN %                                                            01136000
         PREVLINK:=LINK; %                                              01137000
         LINK:=VLINK.CF; %                                              01138000
         IF AVALNKOK THEN %                                             01139000
         IF BOOLEAN(VLINK.[1:1]) THEN %                                 01140000
         IF AVALNKOK:=MAXLNK>PREVLINK THEN %                            01141000
         IF AVALNKOK:=(OPERAND(PREVLINK+1,VLINK) AND %                  01142000
                      VLINK.[1:17]=0) THEN %                            01143000
         BEGIN %                                                        01144000
             AVAILN:=AVAILN+1; %                                        01145000
             AVAILT:=AVAILT+VLINK.FF; %                                 01146000
             AVAILM:=MAX(AVAILM,VLINK.FF); %                            01147000
         END; %                                                         01148000
```

```
  END; %                                                                01149000
IF LNKSOK THEN WRITE(P,RANGE,OCTAL(MSTART),OCTAL(MAXLNK)) ELSE          01150000
BEGIN                                                                   01151000
IF MSTARTOK THEN                                                        01152000
      BEGIN %                                                           01153000
         IF PREVLINK=MSTART THEN WRITE(P,ONEOK,OCTAL(MSTART))ELSE       01153500
          WRITE(P,RANGE,OCTAL(MSTART),OCTAL(PREVLINK));                 01154000
          WRITE(P,BAD,"MEMORY",OCTAL(LINK)); %                          01155000
      LINKLOC[I]:=0&PREVLINK CTC;                                       01156000
      END; %                                                            01157000
      MINBAD:=LINK; %                                                   01158000
      PREVLINK:=MSTART;                                                 01158500
      LINK:=MAXLNK; %                                                   01159000
      WHILE OPERAND(LINK,VLINK) AND %                                   01160000
          LINKOK(VLINK)          AND                                    01161000
          VLINK.CF=PREVLINK      AND %                                  01162000
          LINK>VLINK.FF          DO %                                   01163000
      BEGIN %                                                           01164000
          PREVLINK:=LINK; %                                             01165000
          LINK:=VLINK.FF; %                                             01166000
      END; %                                                            01167000
      IF PREVLINK=MSTART THEN                                           01168000
      BEGIN                                                             01168100
          MEMLOC[I]:=MEMLOC[I]&1[4:47:1];                               01168200
      END ELSE                                                          01168400
      BEGIN                                                             01169000
      IF(MAXBAD:=PREVLINK)=MAXLNK THEN                                  01169500
      WRITE(P,ONEOK,OCTAL(MAXLNK)) ELSE                                 01169600
          WRITE(P,RANGE,OCTAL(MAXLNK),OCTAL(MAXBAD));                   01170000
      WRITE(P,BAD,"MEMORY",OCTAL(LINK)); %                              01171000
      MEMLOC[I]:=MEMLOC[I]&1[5:47:1];                                   01172000
      LINKLOC[I]:=LINKLOC[I]&PREVLINK CTF;                              01173000
      END;                                                              01174000
END;                                                                    01175000
      NOWRAPAROUND:=TRUE;                                               01175100
  IF AVALNKOK:=(OPERAND(AVAIL,PREVLINK) AND                             01177000
      PREVLINK=MAXLNK+1        AND %                                    01178000
      OPERAND(PREVLINK,VLINK) AND %                                     01179000
      VLINK.[1:17]=0           AND %                                    01180000
      VLINK.FF=32767                                 OR %               01181000
      (OPERAND(PREVLINK:=MAXLNK+1,VLINK) AND %                          01181100
      VLINK.[1:17]=0 AND %                                              01181200
      VLINK.FF=32767)) THEN %                                           01181300
BEGIN                                                                   01182000
      WHILE AVALNKOK AND %                                              01183000
          NOWRAPAROUND AND %                                            01183100
            (IF LNKSOK THEN AVAILN>N ELSE TRUE) AND %                   01184000
            (IF LNKSOK THEN AVAILT >T ELSE TRUE) AND %                  01185000
 (IF LNKSOK THEN AVAILM ≥M1 ELSE TRUE) DO                               01186000
 IF NOWRAPAROUND:=(MEND>LINK:=VLINK.CF AND LINK>0) THEN %               01187000
      IF AVALNKOK:=(OPERAND(LINK-1,VLINK) AND %                         01188000
          LINKOK(VLINK)          AND %                                  01189000
          BOOLEAN(VLINK.[11:1])  AND %                                  01190000
          OPERAND(LINK+1,VLINK) AND %                                   01191000
          VLINK.FF=(IF NEWTYPE THEN LINK-1 ELSE 0) AND %                01191100
          VLINK.CF=PREVLINK+REAL(NEWTYPE) AND %                         01192000
          OPERAND(LINK,VLINK)    AND %                                  01193000
          VLINK.[1:17]=0         AND %                                  01194000
          VLINK.CF≠LINK)         THEN %                                 01195000
      BEGIN %                                                           01196000
```

```
                    N:=N+1;                                                    01197000
                    T:=T+VLINK.FF; %                                           01198000
                   M1:=MAX(M1,VLINK.FF);                                       01199000
                    PREVLINK:=LINK; %                                          01200000
                END; %                                                         01201000
                IF NOT AVALNKOK THEN WRITE(P,BAD,"AVALBL",OCTAL(LINK)) ELSE    01202000
                IF LNKSOK AND                                                  01202500
                NOT(AVALNKOK:=                                                 01203000
                    LNKSOK AND %                                               01203100
                    AVAILN=N AND %                                             01204000
                    AVAILT=T AND %                                             01205000
                   AVAILM=M1 AND                                               01206000
                    VLINK.CF=MAXLNK+1) THEN WRITE(P,AVLBAD) ELSE %             01207000
            BEGIN WRITE(P,AVL,N,OCTAL(T),T,OCTAL(M1),M1);                      01208000
                MEMLOC[I].[2:1]:=REAL(LNKSOK);                                 01208050
            END;                                                               01208200
            END ELSE WRITE(P,BAD,"AVALBL",OCTAL(MAXLNK+1));                    01209000
            WRITE(P[DBL]); %                                                   01210000
        WRITE(P,TELLM,MEMLOC[I].[13:5],IF I=0                                  01211000
                THEN "(BELOW" ELSE "(ABOVE"," THE F","ENCE) ");                01212000
        WRITE(P,STARS);                                                        01213000
        END LOOP;                                                              01214000
        END CHECKMEMORYLINKS;                                                  01215000
        ARRAY MIXSTK[0:MIXMAX];                                                01216000
     BOOLEAN PROCEDURE RCWMSCWSEARCH(TOS,BOS,ANS);                             01217000
     VALUE TOS,BOS; REAL TOS,BOS,ANS;                                          01218000
      BEGIN                                                                    01219000
          REAL ADR,B;  LABEL UP;  ADR:=TOS;                                    01220000
         UP:  IF CONTROLDESC(ADR,B) THEN                                       01221000
          IF (B:=B.FF)=0 THEN RCWMSCWSEARCH:=TRUE                              01222000
          ELSE IF ADR:=B>BOS AND TOS>B                                         01223000
          THEN GO TO UP;                                                       01224000
          ANS:=ADR;                                                            01225000
    END RCW MSCW SEARCH;                                                       01226000
    INTEGER SSTACK, SSTACKINX;                                                 01227000
    PROCEDURE SPACESTACKFIND(TOS);                                             01228000
     VALUE TOS;  REAL TOS;                                                     01229000
      BEGIN                                                                    01230000
       REAL BOS,B;                                                             01231000
      IF OPERAND(SPACESTACK,BOS) AND (BOS:=BOS.CF+1) LSS TOS                   01232000
     AND TOS LEQ BOS+128 THEN BEGIN                                            01233000
       STAX[MAXSTK].FF:=BOS;                                                   01234000
       IF CONTROLDESC(BOS,B) THEN STAX[0]:=B.FF;                               01235000
       SSTACKINX:=MAXSTK;                                                      01236000
      END;  END SPACE STACK FIND;                                              01237000
       PROCEDURE GETSTACKSFROMTHEBED;                                          01238000
       BEGIN                                                                   01239000
    COMMENT .... THIS PROCEDURE HAS BEEN REWRITTEN TO                          01240000
          REFLECT THE MANNER IN WHICH THE BED IS KEPT IN                       01241000
          THE TIMESHARING SYSTEM.  ESSENTIALLY WE LINK                         01242000
          THROUGH THE BED FORWARD, PICKING UP STACKS                           01243000
          UNTIL WE FIND A BAD LINK OF FINISH SEARCHING.                        01244000
          IF WE FIND A BAD LINK WE START LINKING BACKWARD                      01245000
          UNTIL WE REACH A BAD LINK.  THIS PERMITS US TO                       01246000
          GET THE MAX INFO OUT OF THE BED WHEN PART OF IT                      01247000
          IS CLOBBERED.  .......... FGB..........  ;                         01248000
    REAL BEDLNK,VBEDLNK,PREVLNK,VPREVLNK;                                      01249000
          INTEGER MIX;                                                         01250000
    LABEL FWDBLOWUP,BKWDBLOWUP;                                                01251000
    IF (OPERAND(BED,VBED) AND                                                  01252000
```

```
      (VBED.[2:1]=1) AND (VBED.[9:9]=511) AND                          01253000
      (BEDLNK:=VBED.CF)≠BED) THEN                                      01254000
    BEGIN                                                              01255000
1     PREVLNK:=BED;                                                    01256000
2     WHILE BEDLNK≠BED DO                                              01257000
3     IF ( OPERAND(BEDLNK≠VBEDLNK) AND                                 01258000
4        (VBEDLNK.[2:1]=1) AND (VBEDLNK.[9:9]=511) AND                 01259000
5       PDATADESC(BEDLNK+1,VPREVLNK) AND                               01260000
6       (VPREVLNK:=VPREVLNK.CF)= PREVLNK AND                           01261000
7        OPERAND(BEDLNK+2≠MIX)) THEN                                   01262000
8     BEGIN                                                            01263000
9       MAXSTK:=MAXSTK+1;                                              01264000
10        IF MIX GTR MIXMAX THEN MIX:=0;                               01265000
11      IF MIX GTR 0 THEN MIXSTK[MIX]:=MAXSTK;                         01266000
12      STAX[MAXSTK]:= (BEDLNK+1) &(REAL(MIX GTR 0))[7:47:1];          01267000
13    PREVLNK:=BEDLNK; BEDLNK:=VBEDLNK.CF;                             01268000
14   SPACESTACKFIND(PREVLNK); END ELSE GO TO FWDBLOWUP;                01269000
15    END ELSE                                                         01270000
16    BEGIN                                                            01271000
17    FWDBLOWUP:                                                       01272000
18    IF (PDATADESC(BED1,VBED) AND                                     01273000
19       (VBED.[9:9]=511) AND (VBED.FF>63) AND                         01274000
20        (BEDLNK:=VBED.CF +1)≠BED1) THEN                              01275000
21      BEGIN                                                          01276000
22        PREVLNK:=BED;                                                01277000
23        WHILE BEDLNK≠BED DO                                          01278000
24      IF (PDATADESC(BEDLNK,VBEDLNK) AND                              01279000
25         (VBEDLNK.[9:9]=511) AND                                     01280000
26          OPERAND(BEDLNK-1,VPREVLNK) AND                             01281000
27         (VPREVLNK.[2:1]=1) AND (VPREVLNK.[9:9]=511) AND             01282000
28          (VPREVLNK:=VPREVLNK.CF)=PREVLNK AND                        01283000
29        OPERAND(BEDLNK+1≠MIX)) THEN                                  01284000
30        BEGIN                                                        01285000
31        MAXSTK:=MAXSTK+1;                                            01286000
32          IF MIX GTR MIXMAX THEN MIX:=0;                             01286500
33          IF MIX GTR 0 THEN MIXSTK[MIX]:=MAXSTK;                     01287000
34      STAX[MAXSTK]:=(BEDLNK)&(REAL(MIX GTR 0))[7:47:1];              01288000
35       PREVLNK:=BEDLNK-1;    BEDLNK:=VBEDLNK.CF+1;                   01289000
36     SPACESTACKFIND(PREVLNK); END ELSE GO TO BKWDBLOWUP;              01290000
37     END;END;                                                        01291000
38      BKWDBLOWUP: BEDSTK:=MAXSTK;                                    01292000
39       END GETSTACKSFROMTHEBED;                                      01293000
40       ARRAY INTCDE[0:MIXMAX,0:INAMESIZE-1];                         01293800
41       ARRAY NEXTINT[0:MIXMAX];                                      01293850
42       PROCEDURE DUMPMEMORYANDNOTESTACKS(FROM,TOO);                  01294000
43          VALUE FROM,TOO;                                            01295000
44          INTEGER FROM,TOO;                                          01296000
45       BEGIN                                                         01297000
46           DEFINE T=LINKTYPE#;                                       01297100
47          BOOLEAN BEDDED;                                            01298000
48           REAL L,MX;                                                01299000
49        FORMAT ITEM(X2,A3," =",2(X4,A5));                            01300000
50        REAL R,ADR,Q,LL; BOOLEAN INTR,QT;                            01301000
51        BOOLEAN IRSTACK,MCPSAVE;                                     01302000
52        BOOLEAN MXNOTO,TYP13;                                        01302100
53        DEFINE MXO=NOT MXNOTO#;                                      01302110
54        REAL V,W,X,AVSIZE;                                           01303000
55          INTEGER N,STK;                                             01304000
56        FORMAT AVAILABLE(X27,"**** AVAILABLE SIZE=",A5,"(",I5,")"),  01305000
57             MIXUSE(X27,"MIX=",I2,X1,A6),                            01306000
```

```
          MCPUSE(X27,"MCP--",A6),                                          01307000
          MCPSTACK(X27,"MCP FORKED STACK--"),                              01308000
          XXX(" ");                                                        01309000
 1   ARRAY ATP[0:TYPMAX+1];                                                01310000
 2  NOTPRINTCALL:=TRUE;                                                    01310500
 3  FILLAREATYPE;                                                          01310600
 4  INTR:=PDATADESC(INTRNSC,ADR) AND I=0;                                  01311000
 5  ADR:=ADR.CF;  FILL ATP[*] WITH                                         01312000
 6       "UNKNWN","CODE  ","DATA  ","IO-BUF",                              01313000
 7         "ALGFIB","INQBUF","COBFIB","INTSEG",                            01314000
 8   "HEADER","MNTLOG","    10","    11","STACK ",                         01315000
 9   "INTCDE","    14","    15","    16","    17",                         01316000
10   "    18","    19","CIDROW","    21","    22",                         01317000
11   "@@@@@@";%"@" DENOTES AN UNRECOGNIZED LINK TYPE                       01318000
12    WHILE FROM<TOC DO                                                    01319000
13    BEGIN                                                                01320000
14       N:=(L:=M[FROM,ROW,FROM.COL]).CF;                                  01322000
15       IF T:=L.[3:6] GTR TYPMAX THEN L.[3:6]:=T:=TYPMAX+1;%BADTYPE       01322100
16       MXNOTO:=(MX:=L.[9:6]) NEQ 0;                                      01322150
17       IRSTACK:=L.[1:8]=76;                                              01322200
18       IF BOOLEAN(L.[1:1]) THEN                                          01323000
19       IF NODUMP THEN ELSE                                               01323100
20      BEGIN                                                              01324000
21  AVSIZE:=                                                               01325000
22  IF OPERAND(FROM +1,X) AND X.[1:17]=0 THEN                              01326000
23  X.FF ELSE -10000;                                                      01327000
24  WRITE(P[DBL],AVAILABLE,IF AVSIZE LSS 0 THEN "*****"                    01328000
25  ELSE OCTAL(AVSIZE),AVSIZE);                                            01329000
26  AVALNK:=TRUE;                                                          01330000
27  PRINT(FROM,IF DUMPAVAIL OR                                             01331000
28            NEEDCHECKAVAILNKS THEN N ELSE FROM+3);                       01332000
29  AVALNK:=FALSE;                                                         01332500
30       END                                                               01333000
31         ELSE                                                            01334000
32         BEGIN                                                           01335000
33    QT:=OPERAND(FROM+1,Q);                                               01336000
34    MCPSAVE:=QT AND Q ≠0 AND L.[2:13]=4096 AND Q.FF≠0 AND Q.CF≥129       01336500
35           AND Q.CF ≤PRTMAX;                                             01336600
36    IF MXO AND((T=1 AND QT) OR MCPSAVE) THEN % MCP                       01339000
37       IF NODUMP THEN ELSE DISPLAY(Q.CF,FALSE) ELSE                      01339100
38    IF QT AND (T=7 OR T=13 OR (T=1 AND Q.CF LEQ 1023                     01340000
39                  AND Q.FF LEQ 1023 AND Q.[1:7]=0))                      01340050
40                  AND(LL:=Q.[8:10]) NEQ 0 AND                            01340075
41                  NOT BOOLEAN(L.[2:1]) THEN                              01340077
42         BEGIN % INTRINSIC                                               01340100
43           IF(R:=                                                        01340150
44              IF T=7 AND INTR AND MXO AND OPERAND(ADR+LL,V)              01340200
45                  AND V:=V.CF GTR 1023 THEN V ELSE                       01340250
46              IF (TYP13:=T=13 AND MXNOTO) OR(T=1 AND MXNOTO)             01340300
47                  THEN FROM+2 ELSE 0)                                    01340350
48              GTR 0 THEN                                                 01340400
49              BEGIN                                                      01340450
50                  IF T NEQ 7 THEN % SAVE OFF TYPE 1&13 INTS              01340500
51                  BEGIN                                                  01340510
52                  IF NEXTINT[MX]:=NEXTINT[MX]+1 GTR                      01340520
53                  INAMESIZE-2 THEN NEXTINT[MX]:=NEXTINT[MX]-1;           01340530
54                  INTCDE[MX,NEXTINT[MX]]:=                               01340550
55                  R & LL[8:38:10]  &                                     01340600
56                  (R+@.FF-1) CTF & REAL(TYP13)[1:47:1];                  01340650
57                  END;                                                   01340660
```

```
                    IF NODUMP THEN ELSE IF LL LEQ INTMAX THEN BEGIN 01340700
                    WRITE(LINE[*],ITEM,OCTAL(LL),OCTAL(R),          01340750
                                    OCTAL(R+0.FF-1));               01341000
                    MOVE(INAMS[INAME[LL].CF],LINE[4],               01342000
                                INAME[LL].FF);                      01343000
                    WRITE(P[DBL],7,LINE[*]) END                     01344000
                 END                                                01344100
   END ELSE                                                         01344200
IF NODUMP THEN ELSE                                                 01344300
IF MXNOTO THEN WRITE(P[DBL],MIXUSE,MX,ATP[T]) ELSE                  01344500
IF IRSTACK THEN                                                     01345000
   BEGIN                                                            01346000
        WRITE(LINE[*],MCPSTACK);                                    01347000
        IF OPERAND(FROM+2,W) AND                                    01348000
        V:=W.[8:10] GTR 129 AND V LEQ PRIMAX                        01349000
   THEN MOVE(NAMS[NAME[V].CF],LINE[6],NAME[V].FF);                  01350000
   WRITE(P[DBL],9,LINE[*]);                                         01351000
   END ELSE                                                         01352000
   WRITE(P[DBL],MCPUSE,ATP[T]);                                     01353000
     END IN USE LINK DECODING;                                      01353500
     BEDDED:=FALSE;                                                 01353800
   IF NOT BOOLEAN(L.[1:1]) THEN                                     01353900
   BEGIN STK:=-1;                                                   01354000
         WHILE MAXSTK≥STK:=STK+1 AND NOT BEDDED DO                  01355000
         IF BEDDED:=                                                01356000
         (LL:=STAX[STK].CF GTR FROM AND N GTR LL) THEN              01357000
         IF STAX[STK].FF=0 THEN STAX[STK].FF:=FROM +2;              01358000
         NOTPRINTCALL:=NOTPRINTCALL AND NOT MCPSAVE;                01358050
         PRINT(FROM,IF(ONEMIX≠0 AND ONEMIX≠MX AND MX≠0)             01358100
         OR(NONORMALCODE AND((T=7 OR T=13) OR                       01358200
                              (T=1 AND MX≠0)))                      01358300
            OR (NOMCPCODE AND(L.[3:12]=64 OR MCPSAVE))              01358400
                 THEN FROM+2 ELSE N);                               01358500
                MCPSAVE:=FALSE;                                     01358550
          END;                                                      01358600
%BELOW,WE PICK UP STACKS NOT ASLEEP FOR "DUMPSTACK" TO DUMP LATER   01359000
% [6:1] IN STAX[*] IS USED TO INDICATE SPECIAL TOP-OF-STACK TREATMENT 01360000
          IF NOT BEDDED THEN                                        01361000
          IF IRSTACK THEN                                           01362000
          STAX[MAXSTK:=MAXSTK+1]:=                                  01363000
          (N-1)&(FROM+2) CTF & 1[6:47:1];                           01364000
          FROM:=N;                                                  01376000
        END;                                                        01377000
     NOTPRINTCALL:=FALSE;                                           01377500
     SSTACK:=STAX[0];                                               01378000
     END DUMP MEMORY AND NOTE STACKS;                               01379000
     ARRAY MCPROG[0:255];                                           01380000
     INTEGER MAXMCPROG,ESP;                                         01381000
      ARRAY OUTERBLOCK[0:0];%                                       01382000
     PROCEDURE SEQUENCE(ARAY,LIM);                                  01383000
         VALUE LIM;                                                 01384000
         ARRAY ARAY[0];                                             01385000
         INTEGER LIM;                                               01386000
     BEGIN                                                          01387000
         INTEGER T,L;                                               01388000
         REAL V;                                                    01389000
         STREAM PROCEDURE MOVE(S,D,D32,M32);                        01390000
             VALUE D32,M32;                                         01391000
         BEGIN                                                      01392000
             SI:=S; DI:=D;                                          01393000
```

```
            D32(DS:=32 WDS);                                                    01394000
            DS:=M32 WDS;                                                        01395000
        END MOVE;                                                               01396000
        T:=LIM;                                                                 01397000
        WHILE (T:=T-1)≥0 DO                                                     01398000
        BEGIN                                                                   01399000
            I:=LIM;                                                             01400000
            L:=(V:=ARAY[T]).CF;                                                 01401000
            WHILE ARAY[I].CF>L DO                                               01402000
                I:=I-1;                                                         01403000
            IF (L:=I-T)>0 THEN                                                  01404000
            BEGIN                                                               01405000
                MOVE(ARAY[T+1],ARAY[T],L.[37:6],L.[43:5]);                      01406000
                ARAY[I]:=V;                                                     01407000
            END;                                                                01408000
        END;                                                                    01409000
    END SEQUENCE;                                                               01410000
    PROCEDURE GETSORTANDLISTMCPROG;                                             01411000
    BEGIN                                                                       01412000
        REAL R;                                                                 01413000
        INTEGER TYP;                                                            01414000
         FORMAT TOTALPROCS(///"### A TOTAL OF ",I3," MCP ",                     01415000
                "PROCEDURES WERE PRESENT IN MEMORY"),                           01416000
                F(X8,"PRESENT MCP PROCEDURES"//                                 01417000
                "PRT",X5,"DESCRIPTOR",X8,"THRU"//);                             01418000
        IF DESCRIPTOR(ESPBIT,R,TYP) AND TYP="75" THEN                           01419000
        MCPROG[MAXMCPROG]:=                                                     01420000
            (ESP:=R.CF)&                                                        01421000
            (ESP+R.[8:10]-1)[18:33:15]&                                         01422000
            ESPBIT[8:38:10]                                                     01423000
        ELSE MAXMCPROG:=-1;                                                     01424000
         OUTERBLOCK[0]:=(PRTMAX+1)&(ESP-1) CTF;                                 01425000
        FOR I:=129 STEP 1 UNTIL PRTMAX DO                                       01426000
        IF DESCRIPTOR(I,R,TYP) AND                                              01427000
             (TYP="75" OR TYP="77" OR TYP="74") AND                             01428000
           R.CF≠ESP          THEN                                               01429000
        MCPROG[MAXMCPROG:=MAXMCPROG+1]:=                                        01430000
            R.CF&                                                               01431000
            (R.CF+R.[8:10]-1)[18:33:15]&                                        01432000
            I[8:38:10];                                                         01433000
        SEQUENCE(MCPROG,MAXMCPROG);                                             01434000
        NEXTPAGE;                                                               01435000
        WRITE(P,F);                                                             01436000
         SGLTOG:=TRUE;                                                          01436100
        FOR I:=0 STEP 1 UNTIL MAXMCPROG DO                                      01437000
        DISPLAY(MCPROG[I].[8:10],TRUE);                                         01438000
         SGLTOG:=FALSE;                                                         01438100
     WRITE(P,TOTALPROCS,MAXMCPROG+1);                                           01439000
    END GET SORT AND LIST PRESENT MCP PROGRAM SEGMENTS;                         01440000
    ARRAY INTSP[0:2×INAMESIZE];                                                 01441000
    INTEGER INTSPMAX,EXTRAINTSPMAX;                                             01442000
    PROCEDURE GETSORTANDLISTINTRINSICS;                                         01443000
    BEGIN                                                                       01444000
        INTEGER IMAX,MIX,I;                                                     01445000
        REAL ADR,R,L,C,V;                                                       01446000
        FORMAT ITEM(X2,A3," =",2(X4,A5),X2,A2,X2),                              01447000
          F(X8,"PRESENT INTRINSICS"//                                           01448000
         "INDEX",X6,"FROM",X5,"THRU",X2,"TYP"/),                                01449000
          BADINTO("*** BAD INTRNSC[0]"),                                        01449050
          MIXINT(X8,"MIX = ",I2);                                               01449100
```

```
                FOR MIX:=1 STEP 1 UNTIL MIXMAX DO                                01449120
                BEGIN                                                            01449140
                    I:=0; V:=1;                                                  01449160
1               IF (Q:= INTCDE[MIX,I:=I+1]) NEQ 0 THEN                           01449170
2               DO IF Q LSS 0 THEN V:=-1 UNTIL Q:=INTCDE[MIX,I:=I+1]=0;          01449180
3               IF I GTR 1 THEN BEGIN R:=-1; INTCDE[MIX,0]:= V END;              01449200
4               END;                                                             01449220
5               IF R LSS 0 THEN INTCDE[0,0]:=1;                                  01449240
6               INTSPMAX:=-1;                                                    01450000
7             IF PDATADESC(INTRNSC,ADR) THEN                                     01451000
8               BEGIN                                                            01452000
9                   IMAX:=ADR.[8:10]-1;                                          01453000
10                  ADR:=ADR.CF;                                                 01454000
11                  IF OPERAND(ADR,R) AND R.[1:38]=0                             01454100
12                  AND R NEQ 0 AND R LEQ IMAX THEN IMAX:=R ELSE MIX:=-1;        01454200
13                   IF IMAX GTR 0 THEN                                          01455000
14                  FOR I:=1 STEP 1 UNTIL IMAX DO                                01456000
15                  IF OPERAND(ADR+I,R) AND R.CF≥1023 THEN                       01457000
16                  INTSP[INTSPMAX:=INTSPMAX+1]:=                                01458000
17                    IF I=17 THEN R.CF&(R.CF+3) CTF & I[8:38:10] ELSE           01458100
18                      R.CF&                                                    01459000
19                      (IF OPERAND(R.CF-1,L) AND                                01460000
20                          0<(L:=L.FF) AND                                      01461000
21                          L<1023 THEN                                          01462000
22                      R.CF+L-1 ELSE R.CF)[18:33:15]&                           01463000
23                      I[8:38:10];                                              01464000
24              END;                                                             01465000
25              IF INTSPMAX GEQ 0 OR INTCDE[0,0] NEQ 0 THEN                      01466000
26              BEGIN                                                            01467000
27                  NEXTPAGE;                                                    01468000
28                  DISPLAY(INTRNSC,TRUE);                                       01469000
29                  IF MIX LSS 0 THEN WRITE(P[DBL],BADINTO);                     01469100
30                  WRITE(P,F);                                                  01470000
31                  IF INTSPMAX GEQ 0 THEN BEGIN                                 01470500
32                  SEQUENCE(INTSP,INTSPMAX);                                    01471000
33                  WRITE(P[DBL],MIXINT,0);                                      01471100
34                  FOR I:=0 STEP 1 UNTIL INTSPMAX DO                            01472000
35                  BEGIN                                                        01473000
36                      WRITE(LINE[*],ITEM,OCTAL(L:=INTSP[I].[8:10]),            01474000
37                            OCTAL(INTSP[I].CF),                                01475000
38                            OCTAL(INTSP[I].FF),7);                             01476000
39                      IF L≤INTMAX THEN                                         01477000
40                      MOVE(INAMS[INAME[L].CF],LINE[4],                         01478000
41                                  INAME[L].FF);                                01479000
42                      WRITE(P,7,LINE[*]);                                      01480000
43                  END;                                                         01481000
44                    WRITE(P);                                                  01481500
45              END;                                                             01482000
46              IF INTCDE[0,0] NEQ 0 THEN                                        01482100
47              FOR MIX:=1 STEP 1  UNTIL MIXMAX DO                               01482120
48            IF (R:=INTCDE[MIX,0]) NEQ 0 THEN                                   01482140
49              BEGIN                                                            01482160
50                  I:=0;                                                        01482180
51                  WRITE(P[DBL],MIXINT,MIX);                                    01482200
52                  WHILE (R:=INTCDE[MIX,I:=I+1]) NEQ 0 DO BEGIN                 01482220
53                      WRITE(LINE[*],ITEM,OCTAL(L:=R.[8:10]),OCTAL(R.CF),       01482260
54                          OCTAL(R.FF),OCTAL(IF R LSS 0 THEN 13 ELSE 1));       01482300
55                      IF L LEQ INTMAX THEN                                     01482320
56                      MOVE(INAMS[INAME[L].CF],LINE[4],                         01482340
57                                  INAME[L].FF);                                01482360
```

```
                WRITE(P,7,LINE[*]);                                          01482380
                END;                                                         01482400
                WRITE(P);                                                    01482420
        END;                                                                 01482440
        END;                                                                 01482445
        COMMENT MOVE TYPE 1 & 13 INTRINSICS INTO INTSP FOR USE LATER         01482450
            IN STACK ANALYSIS;                                               01482460
        EXTRAINTSPMAX:=INTSPMAX; I:=0;                                       01482470
        IF INTCDE[0,0] NEQ 0 THEN                                            01482480
        FOR MIX:=1 STEP 1 UNTIL MIXMAX DO                                    01482490
        IF INTCDE[MIX,0]=0 THEN I:=0 ELSE                                    01482500
        WHILE(R:=INTCDE[MIX,I:=I+1]) NEQ 0 DO                                01482510
        INTSP[EXTRAINTSPMAX:=EXTRAINTSPMAX+1]:=R;                            01482520
    END GET SORT AND LIST PRESENT INTRINSICS;                                01483000
    STREAM PROCEDURE MOVC(S,SP,D,DP,W,C);                                    01484000
        VALUE SP,DP,W,C;                                                     01485000
    BEGIN                                                                    01486000
        SI:=S; SI:=SI+SP;                                                    01487000
        DI:=D; DI:=DI+DP;                                                    01488000
        W(DS:=8 CHR); DS:=C CHR;                                             01489000
    END MOVC;                                                                01490000
    BOOLEAN PROCEDURE WITHIN(ARAY,AMAX,ITEM,RESULT,PLUS);                    01491000
        VALUE AMAX,ITEM;                                                     01492000
        ARRAY ARAY[0];                                                       01493000
        INTEGER AMAX,ITEM,RESULT,PLUS;                                       01494000
    BEGIN                                                                    01495000
        BOOLEAN FOUND;                                                       01496000
        LABEL EXIT;                                                          01497000
        IF AMAX≥0 THEN                                                       01498000
        FOR I:=0 STEP 1 UNTIL AMAX DO                                        01499000
        IF FOUND:=                                                           01500000
            (ARAY[I].CF≤ITEM AND ITEM ≤ARAY[I].FF) THEN                      01501000
        BEGIN                                                                01502000
            RESULT:=ARAY[I].[8:10];                                          01503000
            PLUS:=ITEM-ARAY[I].CF;                                           01504000
            GO TO EXIT;                                                      01505000
        END;                                                                 01506000
        EXIT: WITHIN:=FOUND;                                                 01507000
    END WITHIN;                                                              01508000
    ARRAY PROGS[0:3,0:255]; INTEGER PROWS;                                   01509000
    PROCEDURE GETBETTERTOS(TOS,DELTA,INC,BOS,NEWTOS);                        01510000
    VALUE TOS,DELTA,INC,BOS;                                                 01511000
    INTEGER TOS,DELTA,INC,BOS,NEWTOS;                                        01512000
    BEGIN                                                                    01513000
    COMMENT:THIS PROC IS INTENDED TO BE USED WITH CONTROL STATE              01514000
    STACKS WHOSE ACTUAL TOS IS UNKNOWN(ONLY A MAXIMUM VALUE IS               01515000
    KNOWN). WE ARE "DONE" WHEN,BARRING ABNORMAL CONDITIONS, WE RETURN        01516000
    IN "NEWTOS" THE SMALLER OF:                                              01517000
    1) "TOS"   AND                                                           01518000
    2) THE ADDRESS OF THE "TOP VALID CONTROL WORD" +DELTA.                   01519000
    TO OBTAIN THE "TOP VALID CONTROL WORD" WE SCAN DOWN THE STACK STARTING   01520000
    AT "TOS" UNTIL WE FIND A CONTROLWORD(CW). IF THE F-FIELD(FF)             01521000
    OF THIS "ORIGINAL" CW POINTS TO ANOTHER CW AND THE                       01522000
    ADDRESS OF THIS LATTER CW FALLS IN THE RANGE "BOS" THROUGH               01523000
     ("BOS" +"INC") WE ARE "DONE". IF NOT IN THIS RANGE WE                   01524000
    CHECK FOR A CW AT THE ADDRESS GIVEN BY THE FF OF                         01525000
    THIS LATTER CW AND REPEAT THE RANGE TEST UNTIL WE ARE "DONE" OR NO       01526000
    LONGER LINK TO A CW.  IN THIS LAST CASE WE LOOK FOR ANOTHER "ORIGINAL"   01527000
    CW STARTING AT THE NEW VALUE OF "TOS" AND REPEAT THE ABOVE DESCRIBED     01528000
    PROCESS**************************************************************;   01529000
```

```
REAL TOSORIG,X,Y;BOOLEAN WOOPS,CD;                                  01530000
LABEL SQUEEZE,CUTBACK,LAST,XIT;                                     01531000
        TOSORIG:=TOS;                                               01532000
        IF BOS=0 THEN GO LAST ELSE TOS:=TOS+1;                      01533000
SQUEEZE: DO TOS:=TOS-1 UNTIL CONTROLDESC(TOS,X)  OR                 01534000
                             (WOOPS:=TOS LEQ BOS);                  01535000
        IF WOOPS THEN GO LAST;                                      01536000
CUTBACK: IF (Y:=X.FF) GEQ BOS  AND                                  01537000
             Y LEQ (BOS +INC )   AND                                01538000
                CD:=CONTROLDESC(Y,X) THEN                           01539000
        BEGIN                                                       01540000
LAST:             NEWTOS:=MIN(TOSORIG,TOS+DELTA);                   01541000
                  GO XIT;                                           01542000
        END                                                         01543000
        ELSE                                                        01544000
        IF Y GTR BOS+1 AND Y LSS TOS AND CD                         01545000
        THEN GO CUTBACK;                                            01546000
        GO SQUEEZE;                                                 01547000
XIT:    END OF GETBETTERTOS;                                        01548000
 BOOLEAN PROCEDURE PRUNLNK(TOS,BOS,C);                              01549000
  VALUE TOS,BOS;  REAL TOS,BOS,C;                                   01550000
  BEGIN                                                             01551000
   REAL R,F; BOOLEAN OK; R:=TOS;                                    01552000
   WHILE NOT OK AND R GTR BOS DO                                    01553000
   IF OK:=RCWMSCWSEARCH(R,BOS,F) THEN                               01554000
   WHILE NOT(CONTROLDESC(R+1,F)) AND R≠TOS DO                       01555000
   R:=R+1  ELSE R:=R-1;                                             01556000
  C:=R; PRUNLNK:=OK;                                                01557000
  END PRUNLNK;                                                      01558000
   BOOLEAN HOOKORUNHOOKING;                                         01558100
   REAL LOCIRCW;                                                    01558110
   PROCEDURE DUMPSTACK(INX);                                        01559000
      VALUE INX;                                                    01560000
      INTEGER INX;                                                  01561000
   BEGIN                                                            01562000
      LABEL SKIPEEE,ERROWT,DUMPIT;                                  01563000
      LABEL AGAIN;                                                  01563100
      BOOLEAN RECYCLING;                                            01563200
      REAL X,Y;                                                     01564000
      INTEGER TOS,BOS,SEG,ADR,H,L,PRO,I;                            01565000
      INTEGER IY;                                                   01565100
      DEFINE FORKED=BOOLEAN(STAX[INX],[6:1])#;                      01566000
      DEFINE JUSTBELOWTHEFENCE=BOOLEAN(STAX[INX],[5:1])#;           01567000
      DEFINE SPEC=BOOLEAN(STAX[INX],[4:1])#;                        01567100
      DEFINE EYESTACK=BOOLEAN(STAX[INX],[3:1])#;                    01567200
      INTEGER DELTA,INC,W;                                          01568000
      DEFINE MAXSTACKSIZE=512#;%MAX STACK BELOW TOS WE"LL DP        01569000
      DEFINE CONTROL=NOT NORMAL#;                                   01570000
      REAL V,R;                                                     01571000
      BOOLEAN NORMAL,FOUND;                                         01572000
      FORMAT HD(X8,"STACK FROM ",A5," DOWN TO ",A5),                01573000
    INTSTACK(A5," = ",O,X1,O),                                      01574000
  S(X8,"BED ENTRY AT ",A5,X2,"=",2(X1,O),                           01575000
   ", ","MASK AT ",A5,X2,"=",2(X1,O)),                              01576000
     C(X8,3A6,X1,A4,"(",I4,")","  + ",A4,                          01577000
       "(",I4,":",I1,")");                                          01578000
      NEXTITEM;                                                     01579000
      WRITE(P[DBL],HD,                                              01580000
          IF (TOS:=STAX[INX].CF) ≤1 THEN "*****"                    01581000
             ELSE OCTAL(TOS),                                       01582000
```

```
                IF (BOS:=STAX[INX].FF) = 0 THEN "*****"                    01583000
                  ELSE OCTAL(BOS));                                        01584000
            IF BOS LSS 64 THEN BOS:=0;                                     01585000
            IF TOS ≤1 THEN GO ERROWT;                                      01587000
            IF SPEC THEN GO DUMPIT;                                        01587100
            NORMAL:=BOOLEAN(STAX[INX].[7:1]);                              01588000
            IF INX EQL 0 OR INX GTR BEDSTK THEN % % ~ % % % % % % % % % % % 01589000
      %****STACK TO BE DUMPED IS NOT IN THE BED. WE CURRENTLY DEAL WITH     01590000
      %THE FOLLOWING CASES:                                                01591000
      %1. NORMAL STATE(N.S.) STACKS NOT ASLEEP                             01592000
      %2. CONTROL STATE(C.S.) STACKS NOT ASLEEP                            01593000
      %       A) THE INTERRUPT STACK AT @100                               01594000
      %       B) ISTACK (IF NOT ASLEEP)                                    01595000
      %       C) STACKS BELOW THE FENCE (STACKQ,SPACESTACK)                01596000
      %       D) TYPE 12 FORKED STACKS                                     01597000
      %   FOR N.S. STACKS WE PRINT THE WORD AT R-1,CUTBACK TO THE          01598000
      %LAST WORD OF "33"S, PRINT THAT WORD, CUT BACK FURTHER TO ELIMINATE   01599000
      %STACK THAT IS NO LONGER VALID, AND PRINT FROM A MORE RECENT          01600000
      % TOS DOWN TO BOS.                                                   01601000
      %   FOR THE INTERRUPT STACK WE DUMP FROM @177 DOWN TO PRTBASE        01602000
      %GIVING THE IDENTIFIERS DEFINED AT THESE LOCATIONS(IF ANY).           01603000
      %THEN CUT BACK FROM PRTBASE-1 TO THE LAST WORD OF "25"S               01604000
      %AND THEN DUMP FROM THAT WORD DOWN TO @100.                          01605000
      %   FOR ISTACK WE CUT BACK TO THE LAST WORD OF "25"S AND             01606000
      %PRINT FROM THAT WORD DOWN TO THE BASE OF ISTACK .                    01607000
      %   FOR FORKED STACKS AND STACKS BELOW THE FENCE WE ELIMINATE        01608000
      %STACK AT THE TOP, OBTAIN A LOWER  TOS AND DUMP DOWN TO BOS.         01609000
      % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % 01610000
          BEGIN%ALL CASES OF NON-BEDDED STACKS                             01611000
            IF NORMAL OR EYESTACK THEN %                                   01612000
            BEGIN                                                          01613000
                WRITE(P[DBL],ITEM,OCTAL(TOS),                              01614000
                OCTAL(HIHALF(TOS)),OCTAL(LOHALF(TOS)),                     01615000
                "      ");                                                 01616000
                TOS:=TOS-1; % PT AT 33"S OR 25"S                           01617000
            END;                                                           01618000
            IF CONTROL AND BOS=64 THEN %INTERRUPT STACK                    01619000
            FOR TOS:=127 STEP-1 UNTIL ACTUALPRTBASE DO                     01620000
            BEGIN                                                          01621000
                WRITE(LINE[*],INTSTACK,OCTAL(TOS),                         01622000
                OCTAL(HIHALF(TOS)),OCTAL(LOHALF(TOS)));                    01623000
                MOVE(XNAMS[XNAME[TOS].CF],LINE[5],XNAME[TOS].FF);          01624000
                WRITE(P[DBL],8,LINE[*]);                                   01625000
            END;%TOS=ACTUALPRTBASE-1 AT LOOP XIT                           01626000
      %THE FOLLOWING APPLIES TO 1)THE STACK AT @100 AND ISTACK.2) N.S. STACKS 01627000
          SKIPEEE:                                                         01628000
            IF OPERAND(TOS,V) THEN%GET RID OF "25"S OR "33"S               01629000
            IF ((V="EEEE"&"EEEE"[1:25:23]) AND CONTROL)                    01630000
            OR ((V="[[[["&"[[[["[1:25:23]) AND NORMAL)                     01631000
            OR (V=0 AND JUSTBELOWTHEFENCE)                                 01632000
            THEN WHILE OPERAND(TOS-1,R) AND R=V DO TOS:=TOS-1;             01633000
            IF CONTROL THEN IF OPERAND(TOS-2,R) THEN                       01634000
            IF R=V THEN                                                    01635000
            BEGIN                                                          01636000
                FOR I:=TOS,TOS-1 DO                                        01637000
                WRITE(P[DBL],ITEM,OCTAL(I),OCTAL(HIHALF(I)),               01638000
                      OCTAL(LOHALF(I)),"      ");                          01639000
                TOS:=TOS-2;                                                01640000
                GO SKIPEEE;                                                01641000
            END;                                                           01642000
```

```
%LEAVE TOS POINTING AT THE LAST WORD OF "25"S OR "33"S                    01643000
%FINISH OFF N.S. STACKS NEXT                                             01644000
          IF NORMAL THEN                                                 01645000
          BEGIN                                                          01646000
1              WRITE(P[DBL],ITEM,OCTAL(TOS),OCTAL(HIHALF(TOS)),          01647000
2             OCTAL(LOHALF(TOS)),"      ");                            01648000
3                    TOS:=TOS-1;                                        01648500
4                    IF FALSE THEN % OMIT THIS CARD FOR LESS STACK(NS)  01648600
5             IF BOS NEQ 0 THEN                                          01649000
6            IF PRUNLNK(TOS,BOS,W)                                      01650000
7             THEN TOS:=W;                                              01651000
8          END;%OF GETTING NEW TOS                                       01652000
9     %FINALLY FINISH OFF C.S. STACKS                                    01653000
10         IF CONTROL THEN IF FORKED THEN                                01654000
11        GETBETTERTOS(TOS,DELTA:=35,INC:=1,BOS,TOS);                    01655000
12    END% OF NON-BEDDED STUFF                                           01659000
13        ELSE                                                           01660000
14        BEGIN                                                          01661000
15    WRITE(P[DBL],S,OCTAL(V:=(STAX[INX].CF-1)),                         01662000
16            OCTAL(HIHALF(V)),                                          01663000
17                  OCTAL(LOHALF(V)),                                    01664000
18            OCTAL(V-2),                                                01665000
19                  OCTAL(R:=HIHALF(V-2)),                               01666000
20                  OCTAL(H:=LOHALF(V-2)));                              01667000
21      TOS:=TOS+4;                                                      01668000
22            IF OCTAL(R.[24:6])="74" AND CONTROLDESC(V-1,V) THEN        01669000
23            IF WITHIN(MCPROG,MAXMCPROG,V:=V.CF,SEG,ADR) THEN           01670000
24            BEGIN                                                      01671000
25                WRITE(LINE[*],C,"COMPLE","X SLEE","P AT  ",            01672000
26                   OCTAL(SEG),SEG,OCTAL(ADR),ADR,R.[34:2]);            01673000
27                MOVE(NAMS[NAME[SEG].CF], LINE[7],                      01674000
28                          NAME[SEG].FF);                               01675000
29                WRITE(P[DBL],15,LINE[*]);                              01676000
30            END                                                        01677000
31            ELSE WRITE(P);                                             01678000
32        END;                                                           01679000
33        IF BOS=0 THEN BOS:=MAX(0,TOS-(IF NORMAL THEN                   01680000
34                            MAXSTACKSIZE-1 ELSE 128)) ELSE             01680500
35        IF BOS >TOS THEN BOS:=MAX(0,TOS-(IF NORMAL THEN                01681000
36                            MAXSTACKSIZE-1 ELSE  1));                  01681500
37        IF TOS-BOS≥MAXSTACKSIZE THEN BOS:=MAX(0,TOS-MAXSTACKSIZE-1);   01682000
38    DUMPIT:%                                                           01683900
39        IF MYSTACKADR >0 AND MYSTACKSIZE=0 THEN                        01683910
40        IF NOT MYSTACKDUMPED THEN                                      01683920
41        IF MYSTACKADR≤TOS THEN                                         01683930
42        IF MYSTACKADR >BOS THEN                                        01683940
43        MYSTACKDUMPED:=TRUE;                                           01683950
44        FOR I:=TOS STEP -1 UNTIL BOS DO                                01684000
45        BEGIN                                                          01685000
46            WRITE(P[DBL],ITEM,OCTAL(I),                                01686000
47                  OCTAL(H:=HIHALF(I)),                                 01687000
48                  OCTAL(L:=LOHALF(I)),"      ");                       01688000
49            PRO:=-1;                                                   01689000
50            RECYCLING:=FALSE;                                          01689100
51            FOUND:=FALSE;                                              01690000
52            L:=L.CF;                                                   01691000
53            IF  CONTROLDESC(I,X) AND X.CF GTR 0 AND                    01692000
54            X.FF GTR BOS AND X.FF LSS TOS AND I NEQ IY THEN            01692100
55            BEGIN                                                      01693000
56        AGAIN:                                                         01693500
57
```

```
                IF FOUND:=                                                  01694000
                    WITHIN(MCPROG,MAXMCPROG,L.CF,SEG,ADR) AND               01695000
                   TRUE THEN                                                01696000
                BEGIN                                                       01697000
                    WRITE(LINE[*],C,"MCP SE","GMENT ","AT    ",            01698000
                          OCTAL(SEG),SEG,OCTAL(ADR),ADR,H.[34:2]);          01699000
                    MOVE(NAMS[NAME[SEG].CF],LINE[7],                        01700000
                         NAME[SEG].FF);                                     01701000
                END                                                         01702000
                ELSE                                                        01703000
                IF FOUND:=WITHIN(OUTERBLOCK,0,L.CF,SEG,ADR) THEN            01704000
                    WRITE(LINE[*],C,"MCP OU","TER BL","OCK   ",             01705000
                     0,0,OCTAL(L),L,H.[34:2])                               01706000
                ELSE IF NORMAL THEN                                         01707000
            BEGIN                                                           01707200
                IF FOUND:=WITHIN(INTSP,EXTRAINTSPMAX,L,                     01708000
                                 SEG,ADR)  THEN                             01708100
                BEGIN                                                       01709000
                    WRITE(LINE[*],C,"INTRIN","SIC NU","MBER  ",             01710000
                        OCTAL(SEG),SEG,OCTAL(ADR),ADR,H.[34:2]);            01711000
                    MOVE(INAMS[INAME[SEG].CF],LINE[7],                      01712000
                         INAME[SEG].FF);                                    01713000
                END                                                         01714000
                ELSE                                                        01715000
                WHILE (PRO:=PRO+1)≤SPROWS AND NOT FOUND DO                  01717000
                IF FOUND:=WITHIN(PROGS[PRO,*],PROGS[PRO,255],               01718000
                    L,SEG,ADR)          THEN                                01719000
                WRITE(LINE[*],C,"SEGMEN","T NUMB","ER    ",                 01720000
                OCTAL(SEG),SEG,OCTAL(ADR),ADR,H,[34:2]);                    01721000
                END;                                                        01722000
            END ELSE                                                        01722500
                IF NOT(FOUND OR RECYCLING) AND NORMAL  THEN                 01723000
                 IF OPERAND(I,X) AND X.[1:1]=1 AND X.[3:1]=0                01724000
                 THEN % POSSIBLE CODE (OVERLAID)                            01725000
                 IF FOUND:=((Y:=X.FF) GTR BOS AND                           01726000
                            Y LSS I AND                                     01727000
                              (SEG:=X.CF) NEQ 0 AND                         01728000
                              SEG LEQ 1023 AND                              01729000
                                CONTROLDESC(Y,R))                           01730000
                 THEN                                                       01731000
                 BEGIN                                                      01731100
                 IY:=Y;                                                     01731200
                 IF HOOKORUNHOOKING THEN                                    01731500
                 IF TOS GTR 0 AND TOS GTR LOCIRCW THEN                      01731510
                 BEGIN                                                      01731520
                     RECYCLING:=TRUE;                                       01731530
                     GO AGAIN;                                              01731540
                 END;                                                       01731550
                 WRITE(LINE[*],C,"SEGMEN","T NUMB","ER(*) ",                01732000
                     OCTAL(SEG),SEG,OCTAL(ADR:=R.CF),ADR,X.[10:2]);         01733000
                 END;                                                       01733100
                IF FOUND THEN                                               01734000
                WRITE(F[DBL],10,LINE[*]);                                   01735000
          END;                                                              01736000
        ERROWT:                                                             01737000
        END DUMPING A STACK;                                                01738000
          PROCEDURE DUMPSPACESTACK(INX,I);                                  01739000
          VALUE INX;  REAL INX,I;                                           01740000
          BEGIN                                                             01741000
          FORMAT SSHD(X6,"THIS STACK IS A SPACESTACK.  IT LINKS ",          01742000
```

```
                "DIRECTLY TO THE NEXT STACK PRINTED.");                    01743000
    NEXTITEM;                                                              01744000
    WRITE (P[0BL],SSHD);                                                   01745000
    DUMPSTACK(INX);                                                        01746000
    STAX[I:=0]:=SSTACK;                                                    01747000
    END DUMP SPACESTACK;                                                   01748000
  INTEGER PROCEDURE SPREAD(AT,F,C);                                        01749000
    VALUE AT;                                                              01750000
    INTEGER AT,F,C;                                                        01751000
  BEGIN                                                                    01752000
    SPREAD:=CHRS(M[AT.ROW,AT.COL],0,3);                                    01753000
    C:=(F:=CHRS(M[AT.ROW,AT.COL],3,5)).CF;                                 01754000
    F:=F.FF;                                                               01755000
  END SPREAD;                                                              01756000
  PROCEDURE DUMPROGRAMS;                                                   01757000
  BEGIN                                                                    01758000
    INTEGER MIX,TYP,PRTH,PRTF,H,F,C,FPB,SD,AIT,S;                          01759000
    INTEGER L; BOOLEAN SPEC;                                               01760000
    INTEGER A;                                                             01761000
    REAL JAR00;                                                            01762000
 REAL STARS,JAR0; BOOLEAN SWOK;                                            01763000
    FORMAT HD(A5," = ",A6,2(X1,A5),X2,A1,X1,*A6),                          01764000
           H1(X8,"PROGRAM:",X8,"/",X7,", MIX = ",I2);                      01765000
    INTEGER PCOL,S,Z,RU,CL,RP;                                             01766000
    REAL SEGS,SGM;                                                         01767000
    REAL SEG,ADR;                                                          01768000
    ARRAY LSTU[0:,];                                                       01769000
    FORMAT SEGH(X8,"PRESENT PROGRAM SEGMENTS & INTRINSICS"//               01770000
          "SEGMENT SIZE  AT     THRU  TYP    INT #  "//),                 01771000
      SEGMENT(X3,2(A4,X1),X1,A5,X2,A5,X2,A2,X4,A3),                        01772000
      INTSEG(X8,A4,X1,X1,A5,X2,A5,X2,A2,X4,A3),                            01772100
      PD(A5," = ",A6,X1,2(A5,X1),", R+",A4,", +",A4);                      01773000
    STARS:="****"; STARS.[6:18]:=STARS.[24:18];                            01774000
  IF PDATADESC(JAR,JAR0) THEN                                              01775000
    JAR0:=JAR0.CF;                                                         01776000
IF (PRTOK:=PDATADESC(PRT,PRT0)) THEN                                       01777000
PRTH:=SPREAD(PRT,PRTF,PRT0);                                               01778000
 IF PRT0>0 AND (SQCK:=PDATADESC(SW,TYP) AND TYP:=TYP.CF>0) THEN            01779000
    FOR MIX:=1 STEP 1 UNTIL MIXMAX DO                                      01780000
      IF ONEMIX=0 OR ONEMIX=MIX THEN                                       01780500
   IF (R:=TYP +MIX) GTR 0 AND OPERAND(R,R) AND BOOLEAN(R.[1:1]) AND        01781000
   R≠0 AND (R:=R.[18:6])≠63 AND                                            01782000
   (R=0 OR R=1 OR R=4 OR R=5 OR R=16 OR R=32) THEN                         01783000
   IF (HOOKORUNHOOKING:=(R=0 OR R=1 OR R=5)) OR TRUE THEN                  01783100
    IF PDATADESC(PRT0+MIX,R) THEN                                          01784000
      BEGIN                                                                01785000
      NEXTPAGE;                                                            01786000
        WRITE(LINE[*],H1,MIX);                                             01787000
IF JAR0≠0 AND DESCRIPTOR(JAR0+MIX,JAR00,F) AND F.[39:3]=5                  01788000
          AND JAR00.CF>0                          THEN                     01789000
        BEGIN                                                              01790000
          MOVC(M[(A:=JAR00).ROW,A.COL],1,                                  01791000
            LINE[2],1,0,7);                                                01792000
          MOVC(M[(A:=A+1).ROW,A.COL],1,                                    01793000
            LINE[3],1,0,7);                                                01794000
        END                                                                01795000
        ELSE                                                               01796000
        BEGIN                                                              01797000
          MOVC(STARS,1,LINE[2],1,C,7);                                     01798000
          MOVC(STARS,1,LINE[3],1,0,7);                                     01799000
```

```
END;                                                                    01800000
WRITE(P[DBL],15,LINE[*]);                                               01801000
NEXTITEM;                                                               01802000
WRITE(P,HD,                                                             01803000
    OCTAL(PRT),                                                         01804000
    OCTAL(PRTH),OCTAL(PRTF),OCTAL(PRTO)," ",                            01805000
    2,"PRT[*,","*]    ",                                                01806000
    OCTAL(PRTO+MIX),                                                    01807000
    OCTAL(SPREAD(PRTO+MIX,F,R)),OCTAL(F),OCTAL(R),                      01808000
    IF F=0 THEN " " ELSE "*",                                           01809000
    3,"PRT[MI","X,*], ","CF=R  ",                                       01810000
    OCTAL(R+3),                                                         01811000
    OCTAL(H:=SPREAD(R+3,F,FPB)),OCTAL(F),OCTAL(FPB),                    01812000
    IF OCTAL(H.[30:6])="30" THEN " " ELSE "*",                          01813000
    2,"R+3, F","PB    ",                                                01814000
    OCTAL(R+4),                                                         01815000
    OCTAL(H:=SPREAD(R+4,F,SD)),OCTAL(F),OCTAL(SD),                      01816000
    IF OCTAL(H.[30:6])="30" THEN " " ELSE "*",                          01817000
    4,"R+4, S","EGMENT","," DICTI","ONARY ",                            01818000
    OCTAL(R+6),                                                         01819000
    OCTAL(H:=SPREAD(R+6,F,AIT)),OCTAL(F),OCTAL(AIT),                    01820000
    IF BOOLEAN(H.[30:1]) THEN " " ELSE "*",                             01821000
    2,"R+6, A","IT    ",                                                01822000
    OCTAL(R+7),                                                         01823000
    OCTAL(H:=SPREAD(R+7,F,C)),OCTAL(F),OCTAL(C),                        01824000
    IF H.[30:2]=3 AND F<6 AND C=0 THEN " " ELSE "*",                    01825000
    7,"R+7, L","AST MS","CW FOR","," WHICH","," MSFF ",                 01826000
      "WAS FA","LSE   ",                                                01827000
    OCTAL(R+8),                                                         01828000
    OCTAL(H:=SPREAD(R+8,F,C)),OCTAL(F),OCTAL(C),                        01829000
    " ",                                                                01830000
    2,"R+10, ","INCW  ",                                                01831000
    OCTAL(R+9),                                                         01832000
    OCTAL(H:=SPREAD(R+9,F,C)),OCTAL(F),OCTAL(C),                        01833000
  IF H.[32:16]=0 AND H.[30:1]=0 AND F=0 THEN " " ELSE "*",              01834000
    9,"R+11, ","LITERA","L FOR ","LAST C","OMMUNI",                     01835000
      "CATE OR","," PROGR","AM REL","EASE  ",                           01836000
    OCTAL(R+10),                                                        01837000
    OCTAL(H:=SPREAD(R+10,S,C)),OCTAL(S),OCTAL(C),                       01838000
 IF OCTAL(H.[30:6])="50" AND S GTR 0 AND R=C AND R=1 GTR(S:=S-1)        01839000
  THEN " " ELSE IF (S:=0)=0 THEN "*" ELSE "*",                          01840000
    5,"R+12, ","FF = B","OTTOM ","OF THE","," STACK");                  01841000
  IF CONTRCLDESC(R+8,H) AND (LOCIRCW:=H.CF) GTR 1024                    01841100
  AND H.FF GTR 1024 THEN ELSE LOCIRCW:=-1;                              01841110
  NEXTITEM;                                                             01842000
  IF INTCDE[MIX,0] LSS 0 THEN                                           01842100
 BEGIN                                                                  01842120
  WRITE(P,SEGH);                                                        01842140
  WHILE (A:=INTCDE[MIX,RP:=RP+1]) NEQ 0 DO                              01842160
  IF A LSS 0 THEN                                                       01842180
  BEGIN                                                                 01842200
    WRITE(LINE[*],INTSEG,OCTAL(A.FF-A.CF+1),                            01842220
      OCTAL(A.CF),OCTAL(A.FF),OCTAL(13),                                01842240
      OCTAL(L:=A.[8:10]));                                              01842260
    IF L LEQ INTMAX THEN                                                01842280
    MOVE(INAMS[INAME[L].CF],LINE[5],INAME[L].FF);                       01842300
    WRITE(P,8,LINE[*]);                                                 01842320
  END;                                                                  01842340
END;                                                                    01842360
  PROGS[PROGWS:=0,255]:=PCOL:=-1;                                       01843000
```

```
      IF PDATADESC(R+4,SD) THEN                                          01844000
            IF OPERAND(SD:=SD.CF,SEGS) AND SEGS.[1:37]=0 THEN            01845000
            BEGIN                                                        01846000
 1              IF INTCDE[MIX,0] LSS 0 THEN ELSE                         01846100
 2              WRITE(P,SEGH);                                           01847000
 3              FOR SEG:=1 STEP 1 UNTIL SEGS DO                          01848000
 4              IF OPERAND(SD+SEG,SGM) AND                               01849000
 5                  (ADR:=SGM.FF)>1023 THEN                              01850000
 6              BEGIN                                                    01851000
 7                  SIZ:=                                                01852000
 8            IF (OPERAND(ADR-1,SIZ) AND                                 01853000
 9            SIZ.CF=SEG)          OR                                    01855000
10            (OPERAND(ADR-1,SIZ) AND                                    01856000
11            OPERAND(ADR-2,H)  AND                                      01857000
12      ((H.[3:6]=7 AND H.[9:6]=0) OR (H.[3:6]=1 AND H.[9:6] NEQ 0)) AND 01858000
13            SIZ.[8:10]=SGM.CF )  THEN                                  01859000
14                  SIZ.FF ELSE 0;                                       01860000
15             IF BOOLEAN(SGM.[2:1]) THEN                                01861000
16            BEGIN                                                      01862000
17                  L:=SGM.CF;                                           01863000
18                  SPEC:=FALSE;                                         01864000
19                  FOR I:=0 STEP 1 UNTIL INTSPMAX DO                    01865000
20                  IF INTSP[I].[8:10]=L  THEN                           01866000
21                  BEGIN                                                01867000
22                      IF L=17 THEN SIZ:=4;                             01867100
23                      SPEC:=TRUE;                                      01868000
24                      I:=INTSPMAX+2;                                   01869000
25                  END;                                                 01870000
26                  WRITE(LINE[*],SEGMENT,OCTAL(SEG),                    01871000
27                      IF SIZ=0 THEN STARS ELSE OCTAL(SIZ),             01871100
28                      OCTAL(ADR),IF SIZ=0 THEN STARS ELSE              01872000
29                      OCTAL(ADR+SIZ-1),IF SPEC THEN 7 ELSE 1,          01873000
30                      OCTAL(L));                                       01873100
31                  IF L LEQ INTMAX THEN                                 01874000
32                  MOVE(INAMS[INAME[L].CF],LINE[5],INAME[L].FF);        01875000
33                  WRITE(P,8,LINE[*]);                                  01876000
34        END                                                            01877000
35              ELSE                                                     01878000
36              BEGIN                                                    01879000
37                  IF (PCOL:=PCOL+1)=255 THEN                           01880000
38                  BEGIN                                                01881000
39                      PROGS[PROWS,255]:=254;                           01882000
40                      PROWS:=PROWS+1;                                  01883000
41                      PCOL:=0;                                         01884000
42                  END;                                                 01885000
43                  PROGS[PROWS,PCOL]:=                                  01886000
44                      ADR&                                             01887000
45                      (IF SIZ=0 THEN 0 ELSE ADR+SIZ-1)[18:33:15]&      01888000
46                      SEG[8:38:10];                                    01889000
47              END;                                                     01890000
48        END;                                                           01891000
49        IF (PROGS[PROWS,255]:=PCOL)≥0 THEN                             01892000
50        BEGIN                                                          01893000
51            FOR RO:=0 STEP 1 UNTIL PROWS DO                            01894000
52            SEQUENCE(PROGS[RO,*],PROGS[RO,255]);                       01895000
53            SEGS:=PROWS+1;                                             01896000
54            FOR I:=0 STEP 1 UNTIL PROWS DO                             01897000
55            BEGIN                                                      01898000
56                SEGS:=SEGS+PROGS[I,255];                               01899000
57                LSTD[I]:=0;                                            01900000
```

```
                    END;                                                    01901000
                    FOR SEG:=1 STEP 1 UNTIL SEGS DO                         01902000
                    BEGIN                                                   01903000
1                       RO:=0;                                              01904000
2                       IF PROWS>0 THEN                                     01905000
3                       FOR I:=1 STEP 1 UNTIL PROWS DO                      01906000
4                       IF LSTD[RO]=255 OR                                  01907000
5                           (LSTD[I]<255 AND                                01908000
6                           PROGS[I,LSTD[I]].CF<PROGS[RO,LSTD[RO]].CF)      01909000
7                                                           THEN            01910000
8                       RO:=I;                                              01911000
9                       WRITE(P,SEGMENT,                                    01912000
10                              OCTAL((SGM:=PROGS[RO,LSTD[RO]]).[8:10]),    01913000
11                              IF SGM.FF=0 THEN STARS ELSE                 01914000
12                                  OCTAL(SGM.FF-SGM.CF+1),                 01915000
13                                  OCTAL(SGM.CF), IF SGM.FF=0 THEN         01916000
14                                  STARS ELSE OCTAL(SGM.FF),1);            01916100
15                      LSTD[RO]:=LSTD[RO]+1;                               01917000
16                  END;                                                    01918000
17              END;                                                        01919000
18          END;                                                            01920000
19      IF (H:=MIXSTK[MIX])=0 THEN BEGIN                                    01921000
20        STAX[0]:=(R=1) & S  CTF  & 1[7:47:1]  END                         01922000
21    ELSE BEGIN MIXSTK[MIX]:=0;                                            01923000
22        IF H=SSTACKINX THEN DUMPSPACESTACK(H,H);                          01924000
23    IF STAX[H].FF=0 THEN STAX[H].FF:=5; END;                              01925000
24     DUMPSTACK(H);                                                        01926000
25     HOOKORUNHOOKING:=FALSE;                                              01926100
26          END;                                                            01927000
27     IF PRTOK AND SOOK THEN                                               01928000
28     FOR MIX:=1 STEP 1 UNTIL MIXMAX DO                                    01929000
29     IF H:=MIXSTK[MIX]≠0 THEN STAX[H].[7:1]:=0;                           01930000
30       END DUMPING NORMAL STATE PROGRAM INFO;                             01931000
31       PROCEDURE DUMPCONTROLSTACKS;                                       01932000
32       BEGIN                                                              01933000
33          FORMAT H(X8,"CONTROL STATE STACKS");                            01934000
34          REAL V,R,A,I;                                                   01935000
35          LABEL LOW,NONE,DUMPIT,BOT;                                      01935100
36          INTEGER INX,K;  ARRAY MCPST[0:4];                               01936000
37          NEXTPAGE;                                                       01937000
38          WRITE(P[DBL],H);                                                01938000
39          STAX[0]:=127&64  CTF ;                                          01939000
40          DUMPSTACK(0);                                                   01940000
41          NEXTPAGE;                                                       01941000
42    NONE: IF OPERAND(WORKERSTACK,R) AND R>PRTO AND VFENCE>R THEN          01952000
43          FOR I:=0 STEP 1 UNTIL 2 DO BEGIN                                01952100
44          MCPST[I]:=(R+127) & R CTF & 1[5:47:1];                          01952123
45          IF I=0 THEN MCPST[0].[3:1]:=1;                                  01952140
46          R:=R+128 END;                                                   01952150
47          I:=I-1;                                                         01952160
48    % THE ABOVE LOCATES WORKERSTACK AND TWO STACKQ STACKS EACH             01952200
49    % PRESUMED TO BE 128 WORDS LONG AND CONTIGUOUS.                       01953000
50          IF PDATADESC(ISTACK,R) AND (V:=R.[8:10])>0 AND (R:=R.CF)>0 THEN 01955000
51          MCPST[I:=I+1]:=(R+V-1)&R CTF & 1[3:47:1];                       01956000
52          % ISTACK                                                        01957000
53          % NOW DUMP OUT THE BEDDED STACKS                                01958000
54          FOR INX:=1 STEP 1 UNTIL BEDSTK DO                               01959000
55          IF NOT BOOLEAN((R:=STAX[INX]).[7:1]) THEN                       01960000
56          BEGIN R:=R.CF;                                                  01961000
57              IF (K:=INX)  = SSTACKINX THEN                               01962000
```

```
                BEGIN                                                                   01963000
                    DUMPSPACESTACK(K,K); R:=STAX[0].CF+2;                               01964000
                END;                                                                    01965000
                IF R>PRTO AND VFENCE>R THEN                                             01966000
                BEGIN                                                                   01967000
                    FOR A:=0 STEP 1 UNTIL I DO                                          01968000
                    IF (V:=MCPST[A]).CF>R AND R>(V:=V.FF) THEN                          01969000
                    BEGIN STAX[K].FF:=V;                                                01970000
                        MCPST[A]:=0; GO TO DUMPIT;                                      01971000
                    END;                                                                01972000
                    V:=PRTO; GO TO BOT;                                                 01973000
                END;                                                                    01974000
                IF STAX[K].FF = (V:=0) THEN                                             01975000
BOT:            STAX[K].FF:=IF RCWMSCWSLARCH(R-2,V,R) THEN (R-1) ELSE 0;                01976000
DUMPIT:         DUMPSTACK(K);                                                           01977000
            END;                                                                        01978000
            % MOVE UNBEDDED MCP STACKS TO MCP LIST                                     01979000
            IF SSTACKINX=0 THEN %SPACESTACK NOT DUMPED                                  01980000
            IF OPERAND(SPACESTACK,V) AND V.[1:32]=0  THEN                               01981000
            IF (V:=V.CF) GTR PRTO AND V LSS VFENCE THEN                                 01982000
            MCPST[I:=I+1]:=(V+ 99) & V CTF & 1[5:47:1];                                 01983000
            FOR A:=0 STEP 1 UNTIL I DO                                                  01984000
            IF MCPST[A]≠0 THEN STAX[MAXSTK:=MAXSTK+1]:=MCPST[A];                        01985000
            FOR INX:=(BEGSTK+1) STEP 1 UNTIL MAXSTK DO                                  01986000
            BEGIN                                                                       01987000
                DUMPSTACK(INX);                                                         01988000
            END;                                                                        01989000
            IF NOT MYSTACKDUMPED THEN                                                   01989500
            IF MYSTACKADR >0 THEN                                                       01989510
            BEGIN % SPECIAL DUMP OF USER SELECTED "STACK" AREA                          01989520
                STAX[0]:=MYSTACKADR &                                                   01989530
                        MAX(0,MYSTACKADR+1-                                             01989540
                        (IF MYSTACKSIZE≠0 THEN MYSTACKSIZE ELSE 200))CTF &              01989550
                            1[4:47:1];                                                  01989560
                DUMPSTACK(0);                                                           01989570
                MYSTACKDUMPED:=TRUE;                                                    01989575
            END;                                                                        01989580
        END DUMPING CONTROL STATE STACKS;                                               01990000
        REAL VFORK,FORKLNK,PREVLNK,FORKPRO,FQPARAM;                                     01991000
    PROCEDURE MEMORYMAP;                                                                01992000
    BEGIN                                                                               01993000
        INTEGER JARO,UVO,SQO,CTO;                                                       01994000
        INTEGER MIX,I,A,STAR,CHUNKS;                                                    01995000
        REAL PRTX,SQX,S,CTX,JARX,UVX;                                                   01996000
        BOOLEAN JAROK,UVOK,SQOK,CTOK,RUNNING,ABOVE,INCORE,SQTOG;                        01997000
        FORMAT    BADPRT(X1,I2,X38,A1/),                                                01998000
                  SKIP(/),                                                              01999000
                  BISKIP(//),                                                           02000000
                  TRISKIP(///),                                                         02001000
                  MUSE(X1,I2,X9,A1,X20,V2,X6,A1,X4 ,24(X1,A1,X1)),                      02002000
                  MAP(X41,"CORE ASSIGNMENT MAP"//),                                     02003000
                  LEGER(//X8,"LEGER:"/                                                  02003100
                X14,"X = JOB IS BELOW THE FENCE"/,                                      02004000
                X14,"R = JOB RUNNING WITH THIS CHUNK"/,                                 02005000
                X14,"P = CHUNK POSSESSED BY A JOB BUT JOB NOT RUNNING"/,                02006000
                X14,"A = CHUNK ASSIGNED TO A JOB BUT NOT POSSESSED"/,                   02007000
                X14,"* = AN ARRAY HAS BEEN CLOBBERED(PRT,JAR,UV,SQ,CT)"/,               02008000
                X14,"I = # OF JOBS READY TO RUN USING THIS CHUNK",                      02009000
                    "(CTIN].[36:6])"/,                                                  02009100
                X14,"J = # OF JOBS ASSIGNED TO THIS CHUNK",                             02010000
```

```
            "(CT[N],[42:6])"//),                                          02010100
          TOTRDY(X20,"I",X24,24(X1,I2)),                                  02011000
          TOTASG(X20,"J",X24,24(X1,I2)),                                  02012000
          HDR("MIX",X5,"JOB NAME",X8,"USER",X3,"STATUS",X2,               02013000
            "BELOW",X2,24(A2,X1));                                        02014000
IF MOMOK(PRT,PRTO) AND VFENCE NEQ 0 AND VFENCE.[38:10]=0  THEN            02015000
BEGIN                                                                     02016000
     NEXTITEM;                                                            02017000
     I:=A:=-1;                                                            02018000
     CTOK:=MOMOK(CT,CTO);                                                 02019000
     WRITE(P,MAP);                                                        02020000
     SGLTOG:=TRUE;                                                        02020100
     DISPLAY(FENCE,FALSE);                                                02020500
     DISPLAY(PRT,FALSE);                                                  02020510
     DISPLAY(JAR,FALSE);                                                  02020520
     DISPLAY(UV,FALSE);                                                   02020530
     DISPLAY(CT,FALSE);                                                   02020540
     SGLTOG:=FALSE;                                                       02020650
     WRITE(P,LEGER);                                                      02020700
     WRITE(P,TOTRDY,FOR MIX:=VFENCE STEP 1024 UNTIL 31744 DO              02021000
          IF CTOK AND OPERAND(CTO+(I:=I+1),CTX) THEN                      02022000
          CTX.[36:6] ELSE 100);                                           02023000
     WRITE(P,TOTASG,FOR MIX:=VFENCE STEP 1024 UNTIL 31744 DO              02024000
          IF CTOK AND OPERAND(CTO+(A:=A+1),CTX) THEN                      02025000
          CTX.[42:6] ELSE 100); WRITE(P,SKIP);                            02026000
     WRITE(P,HDR,FOR MIX:=VFENCE STEP 1024 UNTIL 31744 DO                 02027000
          IF (S:=OCTAL(MIX).[18:12]).[42:6] NEQ 0 THEN                    02028000
            (S&" "[36:42:6]) ELSE S.[36:12]);                             02029000
     WRITE(P,BISKIP);                                                     02030000
     JAROK:=MOMOK(JAR,JARO);                                              02031000
     UVOK:=MOMOK(UV,UVO);                                                 02032000
     SQOK:=MOMOK(SQ,SQO);                                                 02033000
     STAR:="*";                                                           02034000
     CHUNKS:=IF OPERAND(CHUNKMAX,A) AND A#0 AND                           02035000
            A.[1:42]=0 THEN MIN(23,A) ELSE 23;                            02035100
     FOR MIX:=1 STEP 1 UNTIL MIXMAX DO                                    02036000
     IF OPERAND(PRTO+MIX,A) AND A=0 THEN ELSE                             02037000
     IF NOT MOMOK(PRTO+MIX,PRTX) THEN                                     02038000
     WRITE(P,BADPRT,MIX,STAR ) ELSE                                       02039000
     BEGIN                                                                02040000
          RUNNING:=ABOVE:=FALSE;                                          02041000
          SQTOG:=SQOK AND OPERAND(SQO+MIX,SQX) AND SQX NEQ 0;             02042000
          RUNNING:=(S:=SQX.[18:6])=16 AND SQTOG;                          02044000
          INCORE:=NOT (ABOVE:=(VFENCE LSS PRIX  )) OR                     02045000
               (ABOVE AND SQTOG AND                                       02046000
               (S=0 OR S=1 OR S=4 OR S=5 OR S=16 OR S=32));               02046500
          WRITE(LINE[*],MUSE,MIX,                                         02047000
               IF INCORE THEN "/" ELSE " ",                               02048000
               IF SQTOG THEN "I" ELSE "A",                                02049000
               IF SQTOG THEN S ELSE " *",                                 02050000
               IF ABOVE THEN " " ELSE "x",                                02051000
               FOR I:=0 STEP 1 UNTIL CHUNKS DO                            02052000
               IF ABOVE THEN                                              02053000
               IF SQTOG THEN                                              02054000
               IF I LEQ SQX.[30:6] THEN                                   02055000
               IF I GEQ SQX.[36:6] THEN                                   02056000
               IF ABOVE AND CTOK AND                                      02057000
                    OPERAND(CTO+I,CTX) THEN                               02058000
               IF CTX.[30:6]=MIX THEN                                     02059000
               IF RUNNING THEN "R" & RUNNING                              02060000
```

```
                                ELSE "P" & POSSESSED                                02061000
                                ELSE "A" & ASSIGNED                                 02062000
                                ELSE "*" & BAD CT[*] EL.                            02063000
                                ELSE " " &   NOT                                    02064000
                                ELSE " " & ASSIGNED                                 02065000
                                ELSE "*" & BAD SG[*] EL.                            02066000
                                ELSE " " & BELOW FENCE                              02067000
                                     );&                                            02068000
                    IF JAROK AND MOMOK(JARO+MIX,JARX) AND INCORE THEN               02069000
                    BEGIN                                                           02070000
                         MOVC(M[(A:=JARX),ROW,A.COL],1,                             02071000
                              LINE[0],5,0,7);                                       02072000
                         MOVC(M[(A:=A+1),ROW,A.COL],1,                              02073000
                              LINE[1],5,0,7);                                       02074000
                    END                                                             02075000
                    ELSE IF INCORE THEN MOVC(STAR,7,LINE[1],4,0,1);                 02076000
                    IF UVOK AND MOMOK(UVO+MIX,UVX) THEN                             02077000
                    IF INCORE THEN                                                  02078000
                         MOVC(M[(A:=UVX),ROW,A.COL],33,                             02079000
                              LINE[3],0,0,7) ELSE                                   02080000
                    ELSE MOVC(STAR,7,LINE[3],2,0,1);                                02081000
                    WRITE(P,15,LINE[*]);                                            02082000
               END LOOP;                                                            02084000
          END;                                                                      02085000
     END MEMORYMAP;                                                                 02086000
     STREAM PROCEDURE MOVD(S,D,W);  VALUE W;                                        02086500
          BEGIN LABEL EXIT;                                                         02086510
               SI:=S; DI:=D;                                                        02086524
               W(8(IF SC="+" THEN JUMP OUT 2 TO EXIT ; DS:=CHR));                   02086526
               SI:=SI-1; DI:=DI-1;                                                  02086530
               EXIT:DS:=CHR;                                                        02086540
     END MOVD;                                                                      02086550
     PROCEDURE DUMPMCPINFO;                                                         02087000
     BEGIN                                                                          02088000
          REAL R,A,N,L,TA,TS,MA,MS,LA,LS,RA,RS,PA,PS;                               02089000
          DEFINE MAXMESSAGES=100#;                                                  02090000
          INTEGER TYP,S,C;                                                          02091000
          BOOLEAN TINUOK;                                                           02092000
          BOOLEAN COMSAVE;                                                          02093000
          FORMAT SE(A3," = ",2(O,X1),X24,"(",O,X1,O,")"),                           02102000
     TFXI(X8,"BIT ",I2," = ",I1,", ",X28),                                          02103000
               TF(X8,"BIT ",I2," = ",I1,", ",*A6/);                                 02104000
     ARRAY TB[0:146];                                                               02105000
          REAL UA,US,IA,IS,FA,FS;                                                   02106000
          FORMAT LUN(A3),                                                           02107000
            LQUEHDR("EL",X2,"IOQUE-INDEX", X2,"DISK-ADDRESS"),                      02107100
            BADLQUENTRY(A2,"  # BAD ENTRY IS  ",2(O,X1)),                          02107110
            BADLQUE("## THE LQUE IS INCORRECT"),                                    02107120
            FLQUE(A2,X6,A3,X8,A1,A6),                                               02107130
            NULLQUE("### NO ENTRIES IN THE LQUE"),                                  02107140
             FT(A1,X1,A2,X1,A1,X1,A3,X1,A2,X1,A1,X1,A2,X1,A3),                      02108000
             RT(A3,X1,A2,X1,A4,X1,A6,X1,A3),                                        02109000
             PT(A1,X1,A1,X1,A5,X1,A5,X1,A6),                                        02110000
             IOATH(X8,"FIELDS OF WORDS IN THE I/O ASSIGNMENT TABLES:"//             02111000
             X8,"TINU"/                                                             02112000
             X12,"  [0:3]"/                                                         02113000
             X12,"1 [3:5]   HARDWARE UNIT NUMBER"/                                  02114000
             X12,"  [8:3]"/                                                         02115000
             X12,"2 [11:7]  POWER OF 2"/                                            02116000
             X12,"  [18:6]"/                                                        02117000
```

```
        X12,"3 [24:1]  IN=0, OUT=1"/                                    02118000
        X12,"  [25:5]"/                                                 02119000
        X12,"4 [30:18] UNIT MNEMUNIC"//                                 02120000
        X8,"RDCTABLE"/                                                  02121000
        X12,"  [0:8]"/                                                  02122000
        X12,"1 [8:6]   MIX INDEX IF ASSIGNED"/                          02123000
        X12,"2 [14:10] REEL NUMBER"/                                    02124000
        X12,"3 [24:17] CREATION DATE"/                                  02125000
        X12,"4 [41:7]  CYCLE"//                                         02126000
        X8,"PRNTABLE"/                                                  02127000
        X12,"  [0:1]"/                                                  02128000
        X12,"1 [1:1]   IF WRITE RING PRESENT"/                          02129000
        X12,"  [2:13]"/                                                 02130000
        X12,"2 [15:15] ADDRESS OF TOP I/O DESCRIPTOR"/                  02131000
        X12,"3 [30:18] PHYSICAL REEL NUMBER"//                          02132000
        "LUN",X6,                                                       02133000
        "TINU",X24,                                                     02134000
        "MULTITABLE",X4,                                                02135000
        "LABELTABLE",X4,                                                02136000
        "RDCTABLE",X24,                                                 02137000
        "PRNTABLE"/                                                     02138000
        X9,                                                             02139000
        X2,"1",X4,"2",X8,"3",X4,"4",X8,                                 02140000
        X14,X14,                                                        02141000
        X4,"1",X2,"2",X4,"3",X6,"4",X8,                                 02142000
        X2,"1",X7,"2",X5,"3",X5);                                       02143000
    BOOLEAN PROCEDURE VERIFY(WHAT,A,S);                                 02144000
        VALUE WHAT;                                                     02145000
        INTEGER WHAT;                                                   02146000
        REAL A,S;                                                       02147000
    BEGIN                                                               02148000
        DISPLAY(WHAT,FALSE);                                            02149000
        VERIFY:=                                                        02150000
    PDATADESC(WHAT,A) AND                                               02151000
            (S:=A.[8:10])>0 AND                                         02152000
            (A:=A.CF)>0 AND                                             02153000
            (A+S-1)<MAXCOR;                                             02154000
      END VERIFY;                                                       02155000
      FORMAT IOQSH("FIELDS OF WORDS IN THE I/O-QUEUE TABLES"//          02156000
        X8,"UNIT"/                                                      02157000
        X12,"  [0:1]"/                                                  02158000
        X12,"1 [1:4]   UNIT TYPE"/                                      02159000
        X12,"2 [5:8]   ERROR FIELD"/                                    02160000
        X12,"3 [13:1]  UNIT NOT READY"/                                 02161000
        X12,"4 [14:1]  ERROR FLAG"/                                     02162000
        X12,"5 [15:1]  WAITING FOR AN I/O CHANNEL"/                     02163000
        X12,"6 [16:2]  I/O IN PROCESS"/                                 02164000
        X12,"7 [18:15] INDEX OF FIRST I/O REQUEST"/                     02165000
        X12,"8 [33:15] INDEX OF LAST I/O REQUEST"//                     02166000
        X8,"LOCATQUE"/                                                  02167000
        X12,"  [0:3]   = 5, DESCRIPTOR BITS"/                          02168000
        X12,"1 [3:5]   MIX INDEX"/                                      02169000
        X12,"  [8:4]"/                                                  02170000
        X12,"2 [12:6]  LOGICAL UNIT NUMBER"/                            02171000
        X12,"3 [18:15] INDEX OF NEXT I/O REQUEST"/                      02172000
        X12,"4 [33:15] ADDRESS OF I/O DESCRIPTOR"//                     02173000
    "LUN/  TINU  ",                                                     02174000
        "UNIT",X31,                                                     02175000
        "IOQUE",X19,                                                    02176000
        "LOCATQUE",X21,                                                 02177000
```

```
                "FINALQUE",X11/                                                  02178000
                "INDEX",X7,                                                      02179000
                X2,"1",X2,"2",X3,"3",X1,"4",X1,"5",                              02180000
                    X1,"6",X1,"7",X5,"8",X11,X24,                                02181000
                X2,"1",X5,"2",X2,"3",X5,"4"//),                                  02182000
        IFO(A5,X2,A3),                                                           02183000
                UFO(A1,X1,A2,X1,A3,4(X1,A1),2(X1,A5)),                           02184000
                WFO(0,X1,0),                                                     02185000
                LFO(A1,3(X1,A2),2(X1,A5));                                       02186000
    LABEL FQBAD, FQEND,Q;                                                        02187000
     PROCEDURE WRITEFORKQUE;                                                     02188000
       BEGIN                                                                     02189000
     FORMAT FQITEM(X48,2(X1,O),X6);                                              02190000
           PREVLNK:=FORKLNK; FQPARAM:=FORKLNK+2;                                 02191000
           WRITE(LINE[*],FQITEM,OCTAL(HIHALF(FQPARAM)),                          02192000
                OCTAL(LOHALF(FQPARAM)));                                         02193000
           MOVE(NAMS[NAME[FORKPRO].CF],LINE[2],                                  02194000
                NAME[FORKPRO].FF);                                               02195000
      WRITE (P[DBL],9,LINE[*]);                                                  02196000
       FORKLNK:=VFORK.CF;                                                        02197000
         END WRITE FORKQUE;                                                      02198000
    FORMAT FQEMPTY(X8," THE FORK QUEUE IS EMPTY"),                               02199000
           FQHD(X8,"THE FORK QUEUE CONTENTS ARE :"//                             02200000
                X16," (INDEPENDENT RUNNERS WAITING ",                            02201000
                "TO BE STARTED BY THE MCP )"//                                   02202000
                X16,"PROCEDURE NAME",X18,                                        02203000
                "PARAMETER TO BE PASSED TO PROCEDURE"//);                        02204000
    FORMAT SQBAD(X24,"THE SQ IS NOT CORRECT."),                                  02205000
           SQOUT(X8,A2,X2,V2,X7,2(A1,X1),2(A2,X1),                               02206000
                3(A1,X1),5(A2,X1),X2,2A6),                                       02206100
             SQHD(X8,"FIELDS OF THE WORDS IN THE SQ:"//,X8,"SQ"/                 02207000
           ,X12,"  [0:1]"/                                                       02208000
           ,X12,"1 [1:1]  INTERLOCK BIT"/                                        02208100
           ,X12,"  [2:6]"/                                                       02208200
           ,X12,"2 [8:4]  NLS"/                                                  02208300
           ,X12,"  [12:1] "/                                                     02208400
           ,X12,"3 [13:2] FLAGS FOR EXPANDING A JOBS AREA"/                      02209000
           ,X12,"4 [15:3] NUMBER OF SWAPS FOR JOB"/                              02210000
           ,X12,"5 [18:6] SWAP STATUS OF JOB"/                                   02211000
           ,X12,"6 [24:6] (NUMBER OF CHUNKS POSSESSED BY JOB)-1"/                02212000
           ,X12,"7 [30:6] CHUNK NUMBER OF LAST CHUNK FOR JOB"/                   02213000
           ,X12,"8 [36:6] CHUNK NUMBER OF FIRST CHUNK FOR JOB"/                  02214000
           ,X12,"9 [42:6] MIX INDEX OF NEXT JOB IN THE SWAP OR ",                02215000
                "READY QUE"//                                                    02216000
           ,X8,"EL MIX",X9,"1",X4,"2",X4,"3",X1,"4",X1,"5"                       02216100
           ,X2,"6",X2,"7",X2,"8",X2,"9",X4,"STATUS"/);                           02217000
      FORMAT FQBD("*******"," THE FORKQUE IS BROKEN.  A BACKWARDS ",             02218000
           "SEARCH WILL BE MADE AND EVERYTHING THAT CAN BE FOUND",               02219000
           " WILL BE PRINTED.");                                                 02220000
    REAL LOGZERO;                                                                02221000
    DEFINE MIX      =LOGZERO.[2:5]#,                                             02222000
           REM      =LOGZERO.[7:1]#,                                             02223000
           LL       =LOGZERO.[8:8]#,                                             02224000
           TYPE     =LOGZERO.[16:7]#,                                            02225000
           SPOED    =LOGZERO.[23:1]#,                                            02226000
           XCLOCK   =LOGZERO.[25:23]#;                                           02227000
    DEFINE LOGTYPEMAX=17#;                                                       02228000
    DEFINE LOGBUF=TB#;                                                           02229000
    DEFINE ATIMES=ITIMES#;%TRICK SAVES ONE ARRAY TO CALL GETIMES                 02230000
    INTEGER ARRAY ITIMES[0:3];                                                   02231000
```

```
ALPHA ARRAY YESNO[0:1];                                                  02232000
BOOLEAN DESCOK,EOFOK;                                                    02233000
FORMAT LOGHDR( "MIX",X1,"LL",X1,"TYPE",X1,"REMOTE",X1,"SPOED",          02234000
               X5,"TIME",X7,"LOG ENTRY" ),                               02235000
       FLOG(2(X1,I2),X3,I2,2(X2,A3,X2),A2,3("!",A2),X4,X80),             02236000
       BADZERO(X40,5("*"),"WORD ZERO HAS THE FLAGBIT ON..."),            02237000
       BADDESC(5("*"),"LOGARRAY DESCRIPTOR HAS BEEN CLOBBERED..."),      02238000
       BADEOF(5("*"),"LOGARRAY[30] IS INCORRECT...");                    02239000
 NEXTPAGE;                                                               02240000
 COMSAVE:=COM;                                                           02240100
 COM:=BOOLEAN(64); % OCTAL ONLY WHEN CALL PRINT                          02240200
 DISPLAY(TOGLE,FALSE);                                                   02241000
 IF OPERAND(TOGLE,R) THEN                                                02242000
 BEGIN                                                                   02243000
  FOR I:=1,2,3,5 STEP 1 UNTIL 11,14 STEP 1 UNTIL 17 DO                   02244000
  IF BITON(R,I) THEN WRITE(P,TF,I,1,-1);                                 02245000
  FOR I:=27 STEP 1 UNTIL 31,35,43 DO                                     02246000
  IF BITON(R,I) THEN WRITE(P,TF,I,1,-1);                                 02247000
     FILL TB[*] WITH                                                     02248000
"          ","          ","          ",%00(NEVER USED)                   02249000
"          ","          ","          ",%01                               02250000
"          ","          ","          ",%02                               02251000
"          ","          ","          ",%03                               02252000
"PBUSY     ","          ","          ",%04                               02253000
"          ","          ","          ",%05                               02254000
"          ","          ","          ",%06                               02255000
"          ","          ","          ",%07                               02256000
"          ","          ","          ",%08                               02257000
"          ","          ","          ",%09                               02258000
"          ","          ","          ",%10                               02259000
"          ","          ","          ",%11                               02260000
"DIRECTOR","YTOG       ","          ",%12                               02261000
"SEPTICTA","NKING      ","          ",%13                               02262000
"          ","          ","          ",%14                               02263000
"          ","          ","          ",%15                               02264000
"          ","          ","          ",%16                               02265000
"          ","          ","          ",%17                               02266000
"NU MEM *","  COUNT     ","          ",%18                               02267000
"       *","  =====     ","          ",%19                               02268000
"       *","  TOGLE     ","          ",%20                               02269000
"CDFREE    ","          ","          ",%21                               02270000
"FINDINGA","DDRESS     ","          ",%22                               02271000
"SCRATCHD","IRECTORY","READY       ",%23                                02272000
"MCPFREE   ","          ","          ",%24                               02273000
"AREASNEE","DED        ","          ",%25                               02274000
"AREARDY   ","          ","          ",%26                               02275000
"          ","          ","          ",%27                               02276000
"          ","          ","          ",%28                               02277000
"          ","          ","          ",%29                               02278000
"          ","          ","          ",%30                               02279000
"          ","          ","          ",%31                               02280000
"CANDEINP","UTREADY    ","          ",%32                                02281000
"WORKING   ","          ","          ",%33                               02282000
"INTFREE   ","          ","          ",%34                               02283000
"          ","          ","          ",%35                               02284000
"NUBACKTA","LK         ","          ",%36                               02285000
"KEYBOARD","READY      ","          ",%37                               02286000
"NEEDSELE","CT         ","          ",%38%ONLY MARK XI                  02287000
"SYSDISKT","OG         ","          ",%39                               02288000
"NSECONDR","EADY       ","          ",%40                               02289000
```

```
        "HOLDFREE","          ","          ",%41                                  02290000
        "USERDISK","READY     ","          ",%42%USERSPACEREADY(MARK X)           02291000
        "          ","          ","          ",%43                                  02292000
        "STACKUSE","          ","          ",%44                                  02293000
        "SHEETFRE","E         ","          ",%45                                  02294000
        "STATUSBI","T         ","          ",%46                                  02295000
        "HP2TOG  ","          ","          ",%47                                  02296000
        FOR I:=4,12,13,18,19,20,21,22,23,24,25,26,32,33,34,36,37,38,39,          02297000
            40,41,42,44,45,46,47 DO                                              02297100
        BEGIN                                                                    02298000
          WRITE(LINE[*],TFX1,I,REAL(BITON(R,I)));                                02299000
        MOVC(TB[3*I],0,LINE[2],4,3,0);                                           02300000
          WRITE(P[DBL],6,LINE[*]);                                               02301000
        END;                                                                     02302000
        DISPLAY(NOPROCESSTOG,FALSE);                                             02302200
        NEXTITEM;                                                                02303000
        DUMPARRAY(TAR);                                                          02303100
        DISPLAY(TOGLE,FALSE); % FOR COMPARISION                                  02303200
        NEXTITEM;                                                                02303300
        NEXTPAGE;                                                                02303400
        DISPLAY(OPTION,FALSE);                                                   02304000
        IF OPERAND(OPTION,R) THEN                                                02305000
        BEGIN                                                                    02306000
          IF BITON(R,1) THEN                                                     02308000
          WRITE(P,TF,1,1,-1);                                                    02309000
          FILL TB[*] WITH                                                        02310000
            "USEDRA","          ",                                               02311000
            "USEDRB","          ",                                               02312000
            "BOJMES","S         ",                                               02313000
            "EOJMES","S         ",                                               02314000
            "OPNMES","S         ",                                               02315000
            "TERMGO","          ",                                               02316000
            "GIVEDA","IE        ",                                               02317000
            "GIVETI","ME        ",                                               02318000
            "NOT US","ED        ",%39                                            02319000
            "AUTOPR","INT       ",                                               02320000
            "NOT US","ED        ",%37                                            02321000
            "NOT US","ED        ",%36                                            02322000
            "COPNME","SS        ",                                               02323000
            "CLOSEM","ESS       ",                                               02324000
            "ERRORM","SG        ",                                               02325000
            "RETMSG","          ",                                               02326000
            "LIBMSG","          ",                                               02327000
            "SCHEDM","SG        ",                                               02328000
            "SECMSG","          ",                                               02329000
            "DSKTOG","          ",                                               02330000
            "RELTOG","          ",                                               02331000
            "PBDREL","          ",                                               02332000
            "CHECK ","          ",                                               02333000
            "DISKMS","G         ",                                               02334000
            "DKLOG ","          ",                                               02335000
            "LIBERR","          ",                                               02336000
            "USEPBD","          ",                                               02337000
            "SVPBT ","          ",                                               02338000
            "RSTOG ","          ",                                               02339000
            "AUTOUN","LD        ",                                               02340000
            "RNALL ","          ",                                               02340100
            "CODECL","AY        ",                                               02340200
            "NOT US","ED        ",%15                                            02340300
            "DATACL","AY        ",                                               02340400
```

```
                "HALTSE","T      ",                                    02340500
                "REMOTE","       ",                                    02340600
                "CANDYM","ESS    ",                                    02340700
                "BATCHT","OG     ",                                    02341000
                "BACKGR","OUND   ",                                    02342000
                "STOPTE","ST     ",                                    02342010
                "PUNCHL","CK     ",                                    02342020
                "CDONLY","       ",                                    02342030
                "PKTONL","Y      ",                                    02342040
                "SEPARA","TE     ",                                    02342050
                "AUTOCE","       ",                                    02342060
                0;%                                                    02343000
            R.[9:1]:=REAL(NOT BOOLEAN(R.[9:1]));% NOT BACKGROUND       02343100
            WRITE(P,TF,2,R.[2:1],2,"MODBIO","S      ");                02344000
        FOR I:=3 STEP 1 UNTIL 47 DO                                    02345000
            WRITE(P,TF,1,REAL(BITON(R,I)),2,                           02346000
                TB[A:=2*(47-I)],TB[A+1]);                              02347000
        END;                                                           02348000
    END;%OF TOGLE STUFF                                                02349000
     NEXTITEM;                                                         02350000
     DISPLAY(MESSAGEHOLDER,FALSE);                                     02351000
      I:=0;                                                            02352000
     IF OPERAND(MESSAGEHOLDER,R) AND (R:=R.CF)#0 THEN                  02353000
     DO                                                                02354000
     BEGIN                                                             02355000
         WRITE(LINE[*],ITEM,OCTAL(R),                                  02356000
             OCTAL(HIHALF(R)),OCTAL(LOHALF(R)),"      ");               02357000
         MOVD(M[(R+1).ROW,(R+1).COL],LINE[4],MIN(9,MAXCOR-R));         02358000
         WRITE(P,15,LINE[*]);                                          02359000
     END                                                               02360000
     UNTIL                                                             02361000
         (I:=I+1) GEQ MAXMESSAGES  OR                                  02362000
         NOT OPERAND(R,R) OR                                           02363000
         (R:=R.FF)=0;                                                  02364000
     NEXTITEM;                                                         02365000
IF DESCOK:=VERIFY(LOGARRAY,R,A) AND                                    02366000
     EOFOK:=(OPERAND(R+30,A) AND A=REAL(NOT FALSE)) THEN               02367000
     BEGIN %DUMP LOGARRAY                                              02368000
         WRITE(P[DBL],LOGHDR);                                         02369000
         YESNO[0]:="NO ";YESNO[1]:="YES";                              02370000
         FOR I:=0,1,2 DO                                               02371000
         IF NOT OPERAND(A:=R+10*I,LOGZERO) THEN                        02372000
             WRITE(P,BADZERO) ELSE                                     02373000
         BEGIN                                                         02374000
             MOVE(M[(A+1).ROW,(A+1).COL],LOGBUF,9);                    02375000
             GETIMES(XCLOCK,ITIMES,ATIMES);%NOTE:ATIMES=ITIMES         02376000
            WRITE(LINE[*],FLOG,MIX,LL,TYPE,YESNO[REAL(BOOLEAN(REM))],  02377000
                 YESNO[REAL(BOOLEAN(SPOED))],                          02378000
                  FOR C:=0 STEP 1 UNTIL 3 DO ATIMES[C]);               02379000
             IF TYPE NEQ 8 AND TYPE NEQ 9 AND                          02380000
                TYPE LEQ LOGTYPEMAX THEN                               02381000
             MOVD(LOGBUF[0],LINE[5],9)  ELSE                           02382000
             MOVE(LOGBUF[0],LINE[5],IF TYPE=8 THEN 3                   02383000
                 ELSE IF TYPE=9 THEN 5 ELSE 9);                        02384000
             WRITE(P,15,LINE[*]);                                      02385000
         END;                                                          02386000
     END;                                                              02387000
IF NOT DESCOK THEN WRITE(P[DBL],BADDESC);                              02388000
IF NOT EOFOK THEN WRITE(P[DBL],BADEOF);                                02389000
NEXTITEM;% END LOGARRAY DUMP                                           02390000
```

```
COMMENT ... THE FOLLOWING SECTION PRINTS THE CONTENTS                    02391000
          OF THE FORK QUEUE SINCE THE FORKQUE HAS                        02392000
          REPLACED THE SLATE. THE QUE CONTAINS A 2                       02393000
          WORD ENTRY FOR EACH PROCEDURE. THESE ENTRIES                   02394000
          ARE DISJOINT AND ARE LINKED BY THE FIRST                       02395000
                          WORD IN EACH ENTRY. THE HEAD AND TAIL OF       02396000
          QUE ARE KEPT IN FORKQUE.  THIS QUE IS                          02397000
          ANALYZED LIKE THE BED BUT ONLY A FORWARD                       02398000
          SEARCH IS DONE. ........ FGB ........;                         02399000
HAH ... THERE IS A BACKWARD SEARCH ... WPM.....;                         02400000
IF ( PDATADESC(FORKQUE,VFORK) AND                                        02401000
    (VFORK.[9:9]=511) AND                                                02402000
    (FORKLNK:=VFORK.CF)≠FORKQUE ) THEN                                   02403000
    BEGIN                                                                02404000
     WRITE(P,FQHD);                                                      02405000
     PREVLNK:=FORKQUE;                                                   02406000
     WHILE FORKLNK≠FORKQUE DO                                            02407000
   IF (OPERAND(FORKLNK,VFORK) AND                                        02408000
        VFORK.FF=PREVLNK AND                                             02409000
        OPERAND(FORKLNK+1,FORKPRO) AND                                   02410000
                          (FORKPRO:=FORKPRO.CF) GEQ 129 AND              02411000
       FORKPRO LEQ PRTMAX) THEN                                          02412000
 WRITEFORKQUE                                                            02413000
       ELSE GO TO FQBAD;                                                 02414000
    END ELSE WRITE(P,FQEMPTY); GO TO FQEND;                              02415000
FQBAD:   PREVLNK:=FORKQUE;                                               02416000
IF ( PDATADESC(FORKQUE,VFORK) AND                                        02417000
    (VFORK.[9:9]=511) AND                                                02418000
    (FORKLNK:=VFORK.FF)≠FORKQUE ) THEN                                   02419000
     WHILE FORKLNK≠FORKQUE DO                                            02420000
   IF (OPERAND(FORKLNK,VFORK) AND                                        02421000
        VFORK.CF=PREVLNK AND                                             02422000
        OPERAND(FORKLNK+1,FORKPRO) AND                                   02423000
                          (FORKPRO:=FORKPRO.CF) GEQ 129 AND              02424000
       FORKPRO LEQ PRTMAX) THEN BEGIN                                    02425000
  PREVLNK:=FORKLNK;  FORKLNK:=VFORK.FF;  END ELSE GO TO Q;                02426000
   Q: WRITE(P,FQBD);  FORKLNK:=PREVLNK;                                  02427000
     WHILE FORKLNK≠FORKQUE  DO                                           02428000
      WRITEFORKQUE;                                                      02429000
   FQEND:                                                                02430000
BEGIN LABEL MAKESEG; MAKESEG:%                                           02430100
       NEXTITEM; NEXTPAGE;                                               02431000
        SGLTOG:=TRUE;                                                    02431100
IF TINUOK:=VERIFY(TINU,TA,TS)    AND                                     02432000
        VERIFY(MULTITABLE,MA,MS) AND                                     02433000
        VERIFY(LABELTABLE,LA,LS) AND                                     02434000
        VERIFY(RDCTABLE,RA,RS) AND                                       02435000
        VERIFY(PRNTABLE,PA,PS) THEN                                      02436000
       BEGIN                                                             02437000
        S:=MAX(TS,MS,LS,RS,PS)-1;                                        02438000
        WRITE(P,ICATH);                                                  02439000
        FOR I:=0 STEP 1 UNTIL S DO                                       02440000
        BEGIN                                                            02441000
        WRITE(TB[*],LUN,OCTAL(I));                                       02442000
        IF I<TS THEN                                                     02443000
        BEGIN                                                            02444000
          WRITE(LINE[*],FT,                                              02445000
               (A:=HIHALF(TA+I)).[24:3],                                 02446000
               OCTAL(A.[27:5]),                                          02447000
               A.[32:3],                                                 02448000
```

```
                OCTAL(A.[35:7]),                                        02449000
                OCTAL(A.[42:6]),                                        02450000
                (A:=LOHALF(IA+I)).[24:1],                               02451000
                OCTAL(A.[25:5]),                                        02452000
                A.[30:18]);                                             02453000
            MOVC(LINE[0],0,TB[1],1,2,6);                                02454000
        END;                                                            02455000
        IF I<MS THEN                                                    02456000
            MOVC(M[(MA+I).ROW,(MA+I).COL],0,TB[4],5,1,0);               02457000
        IF I<LS THEN                                                    02458000
            MOVC(M[(LA+I).ROW,(LA+I).COL],0,TB[6],3,1,0);               02459000
            IF I<RS THEN                                                02460000
            BEGIN                                                       02461000
                WRITE(LINE[*],RT,                                       02462000
                OCTAL((A:=HIHALF(RA+I)).[24:8]),                        02463000
                OCTAL(A.[32:6]),                                        02464000
                OCTAL(A.[38:10]),                                       02465000
                OCTAL((A:=LOHALF(RA+I)).[24:17]),                       02466000
                OCTAL(A.[41:7]));                                       02467000
                MOVC(LINE[0],0,TB[8],1,2,6);                            02468000
            END;                                                        02469000
            IF I<PS THEN                                                02470000
            BEGIN                                                       02471000
                WRITE(LINE[*],PT,                                       02472000
                (A:=HIHALF(PA+I)).[24:1],                               02473000
                A.[25:1],                                               02474000
                OCTAL(A.[26:13]),                                       02475000
                OCTAL((A:=LOHALF(PA+I)&A[15:39:9]).[15:15]),            02476000
                OCTAL(A.[30:18]));                                      02477000
                MOVC(LINE[0],0,TB[11],5,2,6);                           02478000
            END;                                                        02479000
            WRITE(P,15,TB[*]);                                          02480000
        END;                                                            02481000
    END;                                                                02482000
    NEXTPAGE;                                                           02483000
    DISPLAY(IOQUESLOTS,FALSE);                                          02483500
    DISPLAY(IOQUEAVAIL,FALSE);                                          02484000
    IF  VERIFY(UNIT,UA,US) AND                                          02485000
        VERIFY(IOQUE,IA,IS) AND                                         02486000
        VERIFY(LOCATQUE,LA,LS) AND                                      02487000
        VERIFY(FINALQUE,FA,FS) THEN                                     02488000
    BEGIN                                                               02489000
        S:=MAX(US,IS,LS,FS)-1;                                          02490000
        WRITE(P,IOQSH);                                                 02491000
        FOR I:=0 STEP 1 UNTIL S DO                                      02492000
        BEGIN                                                           02493000
     WRITE(TB[*],IFD,OCTAL(I),IF TINUOK THEN                            02494000
          LOHALF(TA+I).[30:18] ELSE "***");                             02495000
            IF I<US THEN                                                02496000
            BEGIN                                                       02497000
                WRITE(LINE[*],UFD,                                      02498000
                (A:=HIHALF(UA+I)).[24:1],                               02499000
                OCTAL(A.[25:4]),                                        02500000
                OCTAL(A.[29:8]),                                        02501000
                A.[37:1],                                               02502000
                A.[38:1],                                               02503000
                A.[39:1],                                               02504000
                A.[40:2],                                               02505000
                OCTAL((A:=LOHALF(UA+I)&A[18:42:6]).[18:15]),            02506000
                OCTAL(A.[33:15]));                                      02507000
```

```
                    MOVC(LINE[0],0,TB[1],4,3,4);                                02508000
                END;                                                            02509000
                IF I<IS THEN                                                    02510000
                BEGIN                                                           02511000
                    WRITE(LINE[*],WFO,                                          02512000
                    OCTAL(HIHALF(IA+I)),                                        02513000
                    OCTAL(LOHALF(IA+1)));                                       02514000
                    MOVC(LINE[0],0,TB[5],7,2,1);                                02515000
                END;                                                            02516000
                IF I<LS THEN                                                    02517000
                BEGIN                                                           02518000
                    WRITE(LINE[*],LFO,                                          02519000
                    (A:=HIHALF(LA+1)).[24:3],                                   02520000
                    OCTAL(A.[27:5]),                                            02521000
                    OCTAL(A.[32:4]),                                            02522000
                    OCTAL(A.[36:6]),                                            02523000
                    OCTAL((A:=LOHALF(LA+I)&A[18:42:6]).[18:15]),                02524000
                    OCTAL(A.[33:15]));                                          02525000
                    MOVC(LINE[0],0,TB[8],7,2,6),                                02526000
                END;                                                            02527000
                IF I<FS THEN                                                    02528000
                BEGIN                                                           02529000
                    WRITE(LINE[*],WFO,                                          02530000
                    OCTAL(HIHALF(FA+I)),                                        02531000
                    OCTAL(LOHALF(FA+1)));                                       02532000
                    MOVC(LINE[0],0,TB[12],4,2,1);                               02533000
                END;                                                            02534000
                WRITE(P,15,TB[*]);                                              02535000
            END;                                                                02536000
        NEXTITEM;                                                               02536100
        PRINTARRAY(CHANNEL);                                                    02536110
        NEXTITEM;                                                               02536120
        DISPLAY(DISKCUNT,FALSE);                                                02536130
        IF DFX THEN BEGIN                                                       02536140
        DISPLAY(EUW,FALSE);                                                     02536150
        PRINTARRAY(EUQ);                                                        02536160
        END;                                                                    02536170
        IF SHAREDISK THEN                                                       02536400
        BEGIN                                                                   02536410
            NEXTITEM;                                                           02536420
            DISPLAY(LQAVAIL,FALSE);                                             02536430
            IF VERIFY(LQUE,UA,US) AND                                           02536440
            OPERAND(LQAVAIL,FA) AND FA GEQ 0 AND FA LEQ US THEN                 02536450
            IF FA=0 THEN WRITE(P,NULLQUE) ELSE                                  02536460
            BEGIN                                                               02536470
                WRITE(P[DBL],LQUEHDR);                                          02536480
                S:=MIN(20,FA-1);                                                02536490
                FOR C:=0 STEP 1 UNTIL S DO                                      02536500
                IF NOT OPERAND(UA+C,L) OR (LA:=L.[1:7]) GEQ 64 THEN             02536510
                WRITE(P,BADLQUENTRY,OCTAL(C),OCTAL(HIHALF(L)),                  02536520
                      OCTAL(LOHALF(L))) ELSE                                    02536530
                WRITE(P,FLQUE,OCTAL(C),OCTAL(LA),L.[8:4],L.[12:36])             02536540
            END ELSE WRITE(P,BADLQUE);                                          02536550
        END LQUE;                                                               02536560
        END;                                                                    02537000
         SGLTOG:=FALSE;                                                         02537050
         COM:=COMSAVE;                                                          02537075
END BLOCK;                                                                      02537100
  SGLTOG:=TRUE;                                                                 02537900
  IF SHAREDISK THEN NEXTPAGE; DISPLAY(SQ,FALSE);                                02538000
```

```
DISPLAY(READYEND,FALSE);                                                02538100
DISPLAY(FORCEND,FALSE);                                                 02538200
DISPLAY(RDYRPTEND,FALSE);                                               02538300
DISPLAY(SWAPEND,FALSE);                                                 02538400
SGLTOG:=FALSE;                                                          02538500
IF PDATADESC(SQ,A) AND (A:=A.CF)>0 THEN BEGIN                           02539000
 WRITE (P,SQHD);                                                        02540000
  FILL TB[*] WITH                                                       02541000
    "TIMEND","      ",                                                  02542000
    "WAITSW","AP    ",                                                  02543000
    "BOJSTA","TE    ",                                                  02544000
    "SATISF","Y     ",                                                  02545000
    "EOJSTA","TE    ",                                                  02546000
    "FORCES","WAP   ",                                                  02547000
    "TRANSI","T     ",                                                  02548000
    "WAITST","ATE   ",                                                  02549000
    "READYS","TATE  ",                                                  02550000
    "RDYPRT","      ",                                                  02551000
    "READYB","      ",                                                  02552000
    "RUNNIN","G     ",                                                  02553000
    "SELECT","ING   ",                                                  02554000
    "STABLE","      ",                                                  02555000
    "*BAD S","TATUS*";                                                  02556000
 FOR S:=1 STEP 1 UNTIL MIXMAX DO                                        02557000
 FOR S←0 STEP 1 UNTIL MIXMAX,31 DO                                      02557500
 IF OPERAND(A+S,R) AND                                                  02558000
     (R≠0 OR (R=0 AND(S=0 OR S=31))) THEN BEGIN                         02558100
  N:=L:=R.[18:6];                                                       02559000
IF N>6 THEN IF (N:=N-1)>10 THEN IF N=15 THEN N:=11                      02560000
ELSE IF N=31 THEN N:=12 ELSE IF N=55 THEN N:=13 ELSE N:=14;             02561000
 WRITE(P,SQOUT,OCTAL(S),IF EOFOK←(S=0 OR S=31) THEN "A" ELSE "I",       02562000
       IF EOFOK THEN "  " ELSE S,0,OCTAL(R.[1:1]),                     02562100
       OCTAL(R.[2:6]),OCTAL(R.[8:4]),OCTAL(R.[12:1]),OCTAL(R.[13:2]),   02562200
       OCTAL(R.[15:3]),OCTAL(L),OCTAL(R.[24:6]),                        02563000
       OCTAL(R.[30:6]),OCTAL(R.[36:6]),OCTAL(R.[42:6]),                 02564000
       IF EOFOK←(S≠31) THEN TB[N←N+N] ELSE "      ",                    02565000
       IF EOFOK THEN TB[N+1] ELSE "      ");                           02565100
END; END ELSE                                                           02566000
 WRITE (P[DBL],SQBAD);                                                  02567000
 MEMORYMAP;                                                             02568000
 END DUMPING MCP INFO;                                                  02569000
 PROCEDURE DUMPAUXMEM;                                                  02569100
 BEGIN                                                                  02569110
      FORMAT AUX("DR",A1," MEMORY DUMP"),                               02569120
             AUXMESS("# PRINTING CONTENTS OF DR",A1,"...");             02569130
      LABEL TRYANOTHER,EXIT;                                            02569140
      BOOLEAN COMSAVE;                                                  02569150
      RELOAD:=TRUE;                                                     02569160
      TRYANOTHER:%                                                      02569170
      IF NOMO THEN GO EXIT;                                             02569180
      NEXTPAGE;                                                         02569190
      LOAD;                                                             02569200
      IF AUXTYPE NEQ 0 THEN % AUXMEM PRESENT                            02569210
      BEGIN                                                             02569220
           NEXTITEM ;                                                   02569230
           WRITE(P,AUX,16+AUXTYPE DIV 4);                               02569240
           WRITE(SPO,AUXMESS,16+AUXTYPE DIV 4);                         02569250
           NEXTITEM;                                                    02569260
           COMSAVE:=COM;                                                02569270
           COM:=BOOLEAN(128); % ALPHA/OCTAL                             02569280
```

```
            PRINT(0,32768);                                                    02569290
            COM:=COMSAVE;                                                      02569300
            GO TRYANOTHER;                                                     02569310
        END;                                                                   02569320
        EXIT:RELOAD:=FALSE;                                                    02569330
    END DUMPAUXMEM;                                                            02569340
    PROCEDURE FIXFID;                                                          02569350
    BEGIN                                                                      02569360
        REAL A;                                                                02569370
        FORMAT FNEWFILE("#NEXT FILE TO BE ANALYZED IS MEMORY/",A1,A6);         02569380
        STREAM PROCEDURE MINUS1(N,A); VALUE N;                                 02569390
        BEGIN SI:=LOC N; SI:=SI+5; DI:=A; DI:=DI+5; DS:=3 SUB END;             02569400
        MINUS1(1,TDFID);                                                       02569410
        FILL MDUMP WITH *,TDFID;                                               02569420
        WRITE(SPO,FNEWFILE,TDFID,[6:6],TDFID);                                 02569430
    END;                                                                       02569440
FORMAT COLHDR("B5500 COLLATING SEQUENCE"//X8,"01234567"/),                     02569500
        SOFLOW("### POSSIBLE STACK OVERFLOW IN THE INTERRUPT STACK"),          02569505
        COLINE(X5,I1,X2,X8);                                                   02569510
ARRAY COLLATE[0:7];                                                            02569520
REAL HIGOOD,LOGOOD;                                                            02569525
LABEL LINKSGOOD,LINKSBROKEN,MAXLNKBAD,MSTARTBAD,                               02570000
      ALLBAD;                                                                  02571000
PROCEDURE INITIALIZE;                                                          02571100
    BEGIN                                                                      02571110
    INTEGER K,J;                                                               02571115
    LABEL EXIT;                                                                02571120
    COMMENT:ZERO ALL VARIABLES & ARRAYS IN THE EVENT MORE THAN ONE             02571125
    DUMP FILE IS PROCESSED IN A GIVEN RUN;                                     02571130
    IF NOT RELOAD THEN GO EXIT;                                                02571135
    COMMENT:ZERO  REALS & INTEGERS;                                            02571140
    I:=                                                                        02571145
    R:=                                                                        02571150
    MAXSTK:=                                                                   02571155
    BEDSTK:=                                                                   02571160
    VJOBNUM:=                                                                  02571170
    VBED:=                                                                     02571175
    LINKTYPE:=                                                                 02571180
    MSTART:=                                                                   02571185
    MEND:=                                                                     02571190
    VFENCE:=                                                                   02571195
    0;                                                                         02571197
    PRTO:=                                                                     02571200
    SSTACK:=                                                                   02571205
    SSTACKINX:=                                                                02571210
    MAXMCPROG:=                                                                02571215
    ESP:=                                                                      02571220
    INTSPMAX:=                                                                 02571230
    EXTRAINTSPMAX:=                                                            02571235
    PROWS:=                                                                    02571240
    VFORK:=                                                                    02571245
    0;                                                                         02571246
    FORKLNK:=                                                                  02571247
    PREVLNK:=                                                                  02571250
    FORKPRO:=                                                                  02571255
    FQPARAM:=                                                                  02571260
    HIGOOD:=                                                                   02571265
    LOGOOD:=                                                                   02571270
    TABLESLOC:=                                                                02571275
    PRTCODE:=                                                                  02571280
```

```
      MINLNK:=                                                                02571285
      MINBAD:=                                                                02571290
      0;                                                                      02571292
      MAXBAD:=                                                                02571295
      MAXLNK:=                                                                02571300
      0;%                                                                     02571400
      COMMENT:MAKE BOOLEANS FALSE;                                            02571500
      MYSTACKDUMPED:=                                                         02571505
      PRTOK:=                                                                 02571510
      SGLTOG:=                                                                02571515
      BADCOMMENT:=                                                            02571520
      COMNT:=                                                                 02571525
      LNKSOK:=                                                                02571530
      FALSE;                                                                  02571532
      AVALNKOK:=                                                              02571535
      SOMOKF:=                                                                02571540
      SOMOKB:=                                                                02571545
      NOTPRINTCALL:=                                                          02571550
      FALSE;                                                                  02571552
      AVALNK:=                                                                02571555
      NEEDCHECKAVAILNKS:=                                                     02571560
      FALSE; %                                                                02571600
      COMMENT:ZERO ARRAYS;                                                    02571700
      FOR K:=0 STEP 1 UNTIL MIXMAX DO                                         02571705
      BEGIN                                                                   02571710
           MIXSTK[K]:=                                                        02571715
           NEXTINT[K]:=                                                       02571720
           0;                                                                 02571740
      END;                                                                    02571750
      FOR K:=0 STEP 1 UNTIL 2*INAMESIZE DO                                    02571755
      INTSP[K]:=0;                                                            02571760
      FOR K:=0 STEP 1 UNTIL MIXMAX DO                                         02571800
      FOR J:=0 STEP 1 UNTIL INAMESIZE-1 DO                                    02571805
      INTCDE[K,J]:=0;                                                         02571830
      EXIT:%                                                                  02571870
      END INITIALIZE;                                                         02571890
    DEFINE DUMPD=IF REAL(NEEDCHECKAVAILNKS) LSS 0 THEN 16 ELSE 36#;           02572500
  RESTART:%%                                                                  02572600
      INITIALIZE;                                                             02572700
      LOAD;                                                                   02573000
      DUMPMCPSPRT;                                                            02574000
      IF NOT COMMON THEN                                                      02574100
      IF DUMPCESSPOOLONLY THEN                                                02574105
      BEGIN                                                                   02574110
           DUMPCESSPOOL;                                                      02574120
           GO EOPROG;                                                         02574130
      END;                                                                    02574140
      IF NOT COMMON THEN                                                      02575000
    IF CHECKMEMORYLINKS THEN GETSTACKSFROMTHEBED                              02576000
    ELSE COM:=TRUE;                                                           02577000
     COMMENT THIS NEXT SECTION PRINTS OUT CORE AND ANY                        02580000
           REMARK WHICH MIGHT APPEAR ON THE MDUMP TAPE.                       02581000
           THE INFO NEEDED IS OBTAINED FROM MEMLOC                            02582000
           AND LINKLOC WHICH ARE SET UP BY THE MEM                            02583000
           LINK CHECK PROCEDURE. ..... FGB .....;                             02584000
      IF NOT COMMON THEN                                                      02584050
      FOR I:=112 STEP -1 UNTIL 108 DO                                         02584200
      IF OPERAND(I,R) AND R="EEEE"&"EELE"[1:25:23] THEN I:=-1;                02584210
      IF I LSS  0 THEN ELSE IF NOT COMMON THEN WRITE(P,SOFLOW);               02584220
    IF COMNT THEN                                                             02585000
```

```
      BEGIN                                                                  02585100
        WRITE(P[DBL],STARS);                                                 02586000
        IF NOT COMMON THEN                                                   02586100
        IF PDATADESC(PUNTER,R) AND R.[8:10] GTR 0 AND (R:=R.CF) GTR 512      02586110
        AND R LSS VFENCE THEN                                                02586120
        IF CHRS(M[R.ROW,R.COL],0,1)="=" THEN                                 02586130
        BEGIN                                                                02586140
             WRITE(LINE[*],FBLANKS);                                         02586145
             MOVD(M[R.ROW,R.COL],LINE[0],14);                                02586150
             WRITE(P[DBL],15,LINE[*]);                                       02586160
        END;                                                                 02586170
        IF BADCOMMENT THEN WRITE(P[DBL],COMNTPAR) ELSE                       02587000
        BEGIN                                                                02588000
         WRITE(P,10,COMMT[*]);                                               02589000
        MOVE(COMMT[10],COMMT[0],10);                                         02590000
        WRITE(P,10,COMMT[*]);                                                02591000
        END;                                                                 02592000
        WRITE(P[PAGE],STARS);                                                02593000
      END;                                                                   02594000
      IF OPERAND(16,R) AND R GEQ 12 AND R LEQ 15 THEN % NOT DP DUMP          02595000
      BEGIN                                                                  02596000
         NEEDCHECKAVAILNKS.[1:1]:=TRUE;                                      02596100
         PRINT(0,R);                                                         02596200
         IF I LSS 0                                                          02596210
         THEN IF OPERAND(96+R,1) AND I NEQ "EEEE"&"EEEE"[1:25:23]            02596220
         THEN WRITE(P[DBL],ITEM,OCTAL(R),OCTAL(HIHALF(I:=96+R)),             02596230
                   OCTAL(LOHALF(I)));                                        02596240
         PRINT(R+1,16);                                                      02596250
      END                                                                    02596300
      ELSE PRINT(0,36); % DP DUMP                                            02596400
    IF NOT COMMON THEN                                                       02597000
    BEGIN                                                                    02597500
      FOR I:=0 STEP 1 WHILE MEMLOC[I]≠-7 DO                                  02598000
      IF I=0 OR ONEMIX=0 OR(ONEMIX≠0 AND ONEMIX=MEMLOC[I].[13:5]) THEN        02598100
       BEGIN                                                                 02599000
       MSTART:=MEMLOC[I].CF; MAXLNK:=MEMLOC[I].FF;                           02600000
       HIGOOD:=LINKLOC[I].CF; LOGOOD:=LINKLOC[I].FF;                         02601000
       IF DUMPALLAVAIL THEN PRINT(MEND+1,MSTART);                            02601500
      MEND:=MAXLNK+2; AVALNKOK:=BOOLEAN(MEMLOC[I].[2:1]);                    02602000
       IF NEEDCHECKAVAILNKS.[1:1] THEN NEEDCHECKAVAILNKS.[47:1]:=            02602500
                                       NOT AVALNKOK;                         02602600
       IF I=0 THEN IF MEMLOC[0].[3:1]=0 THEN % MINLNK VALID                  02602700
       BEGIN                                                                 02602720
            MSTART:=MINLNK;                                                  02602740
            IF NOMCPCODE AND NOT DONTDUMPRT THEN ELSE                        02602760
            PRINT(DUMPD,                                                     02602780
                  IF NOMCPCODE AND DONTDUMPRT THEN (PRTMAX+1)                02602800
                  ELSE MSTART);                                              02602820
       END                                                                   02602840
       ELSE                                                                  02602860
       MSTART:=DUMPD;                                                        02602880
       CASE MEMLOC[I].[3:3] OF                                               02603000
        BEGIN                                                                02604000
       LINKSGOOD:          % 0                                               02606000
         BEGIN                                                               02607000
             DUMPMEMORYANDNOTESTACKS(MSTART,MAXLNK);                         02608000
             PRINT(MAXLNK,MEND+1);                                           02609000
         END;                                                                02610000
       LINKSBROKEN:        % 1                                               02611000
         BEGIN                                                               02612000
```

```
            DUMPMEMORYANDNOTESTACKS(MSTART,HIGOOD);                          02613000
            IF HIGOOD NEQ LOGOOD THEN PRINT(HIGOOD,LOGOOD);                  02614000
            DUMPMEMORYANDNOTESTACKS(LOGOOD,MAXLNK);                          02615000
            PRINT(MAXLNK,MEND+1);                                            02616000
        END;                                                                 02618000
    MAXLNKBAD:          % 2                                                  02619000
        BEGIN                                                                02620000
            DUMPMEMORYANDNOTESTACKS(MSTART,HIGOOD);                          02621000
            PRINT(HIGOOD,MEND+1);                                            02622000
        END;                                                                 02623000
        ;               % 3                                                  02623100
        ;               % 4                                                  02624000
    MSTARTBAD:          % 5                                                  02625000
        BEGIN                                                                02626000
            PRINT(MSTART,LOGOOD);                                            02627000
            DUMPMEMORYANDNOTESTACKS(LOGOOD,MAXLNK);                          02628000
            PRINT(MAXLNK,MEND+1);                                            02629000
        END;                                                                 02630000
    ALLBAD:             % 6                                                  02631000
        PRINT(MSTART,MEND+1);                                                02632000
        ;               % 7                                                  02633000
      END CASE STATEMENT;                                                    02634000
    END MEMLOC LOOP;                                                         02635000
    IF DUMPALLAVAIL THEN PRINT(MEND+1,MAXCOR+1);                             02635500
  END ELSE                                                                   02640000
    PRINT(DUMPD,MAXCOR+1);                                                   02642000
    IF NODUMP THEN                                                           02642100
    BEGIN                                                                    02642110
        NODUMPTOG:=TRUE;                                                     02642120
        COM:=COM AND NOT BOOLEAN(384);                                       02642130
    END;                                                                     02642140
    IF NOT COMMON THEN                                                       02643000
    BEGIN                                                                    02644000
        GETSORTANDLISTMCPROG;                                                02645000
        DUMPMCPINFO;                                                         02646000
        GETSORTANDLISTINTRINSICS;                                            02647000
        DUMPROGRAMS;                                                         02648000
        DUMPCONTROLSTACKS;                                                   02649000
        DUMPSELECTEDARRAYS;                                                  02650000
        DUMPAUXMEM;                                                          02650100
    END;                                                                     02651000
EOPROG:%                                                                     02651100
    FILL COLLATE[*] WITH                                                     02651500
    OCT00010203040506O7,                                                     02651510
    OCT10111213141516l7,                                                     02651520
    OCT20212223242526l7,                                                     02651530
    OCT30313233343536l7,                                                     02651535
    OCT40414243444546l7,                                                     02651540
    OCT50515253545556l7,                                                     02651550
    OCT60616263646566l7,                                                     02651560
    OCT70717273747576l7;                                                     02651570
    WRITE(P[PAGE]);                                                          02651580
    WRITE(P,COLHDR);                                                         02651590
    FOR I:=0 STEP 1 UNTIL 7 DO                                               02651600
    BEGIN                                                                    02651610
        WRITE(LINE[*],COLINE,I);                                             02651620
        MOVE(COLLATE[I],LINE[1],1);                                          02651630
        WRITE(P,2,LINE[*]);                                                  02651640
    END;                                                                     02651650
    IF MULTI AND IOFIO.[30:18] GTR 1 THEN                                    02651700
```

```
      BEGIN                                                                    02051710
            IF NOT REPEATING THEN                                              02051720
            BEGIN                                                              02051730
                  REPEATING:=TRUE;                                             02051740
                  SPACE(MDUMP,- 1)[XX:XX];                                     02051750
                  XX:CLOSE(MDUMP,*);                                           02051760
            END;                                                               02051770
            FIXFID;                                                            02051780
            RELOAD:=TRUE;                                                      02051790
            PRINTCOMMONVALUES;                                                 02051795
            WRITE(P[DBL]); WRITE(P[DBL]);                                      02051800
            WRITE(P,FINI);                                                     02051810
            NEXTPAGE;                                                          02051820
            IF NODUMPTOG THEN BEGIN                                            02051822
            NODUMPTOG:=FALSE;                                                  02051824
COM:=COM OR BOOLEAN(384) END;                                                  02051826
            MYSTACKADR:=-1;                                                    02051826
MYSTACKADR:=-1;                                                                02051828
            GO RESTART;                                                        02051830
      END;                                                                     02051840
   END;% % % % % % % OF INNER BLOCK % % % % % % % % % % % % % % % % % % % %    02052000
PRINTCOMMONVALUES;                                                             02052100
WRITE(P[DBL]); WRITE(P[DBL]);                                                  02053000
WRITE(P,FINI);                                                                 02054000
END.                                                                           02056000
END;END.          LAST CARD ON OCRDING TAPE                                    99999999
                                  04019000 I  0007:2       0000000000000000]x2A41B]
```

LABEL 000000000PRINTER0017510OCC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/TSDUMP↵↵

OBJECT /READ