

Table of contents

| | | |
|-----|----|---|
| 4- | 1 | Data areas |
| 5- | 1 | SHOW command |
| 6- | 1 | ALL |
| 8- | 1 | MEMORY |
| 9- | 1 | TERMINALS |
| 10- | 1 | CL |
| 11- | 1 | JOBS |
| 12- | 1 | QUEUE |
| 13- | 1 | COMMANDS |
| 14- | 1 | DEVICES |
| 18- | 1 | ASSIGNS |
| 19- | 1 | ALLOCATIONS |
| 20- | 1 | MOUNTS |
| 21- | 1 | MODEM |
| 22- | 1 | DATE |
| 22- | 15 | TIME |
| 22- | 23 | VERSION |
| 22- | 39 | USE |
| 23- | 1 | INSTALL |
| 25- | 1 | REGIONS |
| 27- | 1 | PRIVILEGES |
| 28- | 1 | SL |
| 30- | 1 | RUN-TIMES |
| 31- | 1 | SPOOL |
| 32- | 1 | SUBSET |
| 33- | 1 | VM |
| 34- | 1 | SYSTAT (& WHO) command |
| 34- | 9 | USE command |
| 35- | 1 | PRTUSE -- Print system usage statistics |
| 36- | 1 | MEMORY command |

```

1          .TITLE  TSKSHO -- Keyboard SHOW Command Routines
2          .ENABL  LC
3          .DSABL  GBL
4 000000   .CSECT  TSKSHO
5 000000   TSKSHO:
6          ;
7          ; TSKSHO is the portion of TSKMON that contains the code
8          ; to implement the SHOW command.
9          ;
10         ; Copyright 1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985.
11         ; S&H Computer Systems, Inc.
12         ; Nashville, Tennessee
13         ;
14         ; Macro calls
15         ;
16         .MCALL  .CSISPC, .TTOUTR, .SRESET
17         .MCALL  .READW, .TTYIN, .TTYOUT, .PURGE
18         .MCALL  .CSIGEN, .SAVEST, .REOPEN
19         .MCALL  .GTLIN, .GTIM, .DATE, .SPFUN
20         .MCALL  .PRINT, .CLOSE, .LOOKUP
21         .MCALL  .WRITW, .ENTER, .EXIT
22         .MCALL  .SERR, .HERR, .FPROT, .GVAL, .PVAL
23         ;
24         ; Global definitions
25         ;
26         .GLOBL  TSKSHO, CMDHD, CMDOFF, KDOCIN, SKPSPC, UCLCMD
27         .GLOBL  DORUN, CMDFRM, CMDDSN, STLGCN, DATTIM, PRGALL
28         .GLOBL  DLCEMT, ALCDEV, CMDSHO, CMDSET, CMDWHO, CMDMEM, CMDUSE
29         ;
30         ; Global references
31         ;
32         .GLOBL  AF$DUP, AF$IND, AF$UCL, AF$SET, AF$CCA, TSXVRS, AF$NPW
33         .GLOBL  CKCLUS, SHOKEY, HANIOC, SYPSWD, TM$NSP, TM$SL1, TM$OFF, TM$ON
34         .GLOBL  TM$KED, TM$TTY, TM$SUB, TM$NO, VMXMSG, VMAXMC, VMXMRB
35         .GLOBL  PEKEMT, PEKADR, PEKSIZ, TM$NNR, CDBUF, CDGET, TM$IN1, TM$IN2
36         .GLOBL  CLRPRV, OPTLST, PFSO, PFCO, PVNPW, TSXVER
37         .GLOBL  PO$SYS, PO$SPV, PRIVSO, PRVOPT, KBPARN, PARNNL
38         .GLOBL  TM$PVA, TM$PVC, PRIVAO, EM$CNO, EM$CPO, EM$CAP, RSTPRV
39         .GLOBL  CHKEQ, CKACQJ, PO$OPR, CKSYPV, AF$BYA, TM$PVL, DMYDEV, AF$TPO
40         .GLOBL  INSTBL, INGADR, INGEMT, IIBUF, II$NAM, II$FLG, II$$SZ, EM$NAD
41         .GLOBL  INSTBN, AF$SCA, AF$NOW, AF$MEM, PO$DBG, PRIVCO, PRVLST
42         .GLOBL  ABRTAD, ABRTCD, CINFLG, $VNOTT, VPRILO, VPRIHI, II$PRV, II$NPV
43         .GLOBL  TM$RD1, TM$RD2, TM$LCL, TM$GBL, SPACE1, RC$OWN, RC$CNT, RC$BAS
44         .GLOBL  RC$EXC, RC$AGE, RC$AEP, RC$USE, RC$FLG, RC$GBL, RC$NAM
45         .GLOBL  RC$LEN, RC$PVT, RCBBAS, RCBEND, RC$$SZ, SHRRCB, SHRRCN
46         .GLOBL  LP$SPD, LP$PAR, LP$ODD, LP$7BT, EM$IICL, PROSLT, RC$LCC
47         .GLOBL  EM$NPD, EM$ILN, EM$CIP, EM$NSF, EM$IUN, EM$CLN
48         .GLOBL  EM$IILN, EM$ACL, EM$TSL, EM$CLB, EM$NSL, EM$SLT, EM$SLW
49         .GLOBL  SLKDON, SLKDOF, EM$UIO, TM$PR1, TM$PR2, TM$LPR, TM$HPR
50         .GLOBL  TM$HPE, TM$CNG, TM$CDS, TM$CEN, TRMHD1, TRMHD2
51         .GLOBL  OPRTXT, CLLINE, LCLTXT, REMTXT, TM$AUT, CLFREE, CLUNIT, CLVERS
52         .GLOBL  TM$CLO, TM$CL1, TM$CL2, TM$CL3, TM$CL4, TM$CL5, TM$CL6, TM$CL7
53         .GLOBL  TM$CL8, QHDMS1, QHDMS2, DVSHH1, DVSHH2, DVSHH3, SYASHD, DKASHD
54         .GLOBL  TM$NAD, ALCHD1, ALCHD2, TM$NSD, TM$SDN, LNAME, TM$C13
55         .GLOBL  CORUSR, LSW, $CTRL0, SERFLG, IOABFL, $CHACT, $STSNG
56         .GLOBL  LSTHL, LCLUNT, FSTIOL, LSTIOL, CL$LIX, CW$PRO, CONFQ2
57         .GLOBL  CL$RQH, CL$WQH, MAXALC, ALCTBL, ALCEND

```

```

58 . GLOBL AD$DVU, AD$JOB, AD$$SZ, UCIDEF, HANCHN
59 . GLOBL NEDCHR, LOUITR, LINIR, LINRTS, CLOTIR
60 . GLOBL CO$DEF, CL$COL, LCDTYP, SOPALC, SOPDAT, SOPTIM
61 . GLOBL UTRPAD, JSWLOC, ERRLOC, MAXMEM, MAXPRI
62 . GLOBL USRSTK, $KINIT, CFSTK, MXJMEM, DFJMEM, EM$HNI
63 . GLOBL SPUBUF, SXBPNT, VSWPFL, MXJADR, CLSFCH
64 . GLOBL TMTOTH, TMTOTL, TMUSRH, TMIOWH, LDMNT, EM$CSE
65 . GLOBL TMSWTH, TMIDLH, TMIOH, TMSWPH, LDCLEN
66 . GLOBL WILDFL, $NOIN, $NOWTT, $HITTY
67 . GLOBL TECO, EDIT, KED, K52, $1STLG, $DIBOL
68 . GLOBL SH$VAL, SH$NAM, SH$$SZ, SH$RTN, SH$FLG
69 . GLOBL SO$NVL, SO$OCT, SO$NO, HANENT, HANSIZ
70 . GLOBL H. CSR, H. VEC, DVSTAT, SFID, ACRSPD, HANPAR, LSTSPL
71 . GLOBL HAZEL, HAZLFL, HAZLNO, $MLOCK, MDT, GETKCH
72 . GLOBL LINBUF, LINNXT, LSTACT, PRGTOP, PRGSIZ, KMNHI
73 . GLOBL KMNTOP, KMNPGS, KMNSTK, KMNSTR, CXPAG, FSTIOL
74 . GLOBL LIMPNT, LINCNT, LACTIV, LRDTIM, CS$RON
75 . GLOBL LOTBUF, LOTNXT, LOTPNT, $VTESC
76 . GLOBL LOTSIZ, LOTSPC, LCOL, $SLKED, ESC
77 . GLOBL LAFSIZ, LFWLIM, LINCUR, NUMON, ILSW2
78 . GLOBL VPRIDF, VPRIVR, $DBKMN
79 . GLOBL $CARUP, DOASGN, UKMNAM, $UKMON, LSW9
80 . GLOBL LSUCF, $CCLRN, EM$NUK, S$QMIO, S$$RUN
81 . GLOBL KL3CLR, $PRGLK, LSW5, PVON, S$SPND, $AUTO
82 . GLOBL S$TWFN, S$TTFN, S$OTFN, S$IOFN, S$OTLO
83 . GLOBL LSTD, FSTD, $DETCH, UMSYTP, S$TTSC
84 . GLOBL $DISCN, LPROJ, LPROG, LUNAME, S$RT, S$LOW
85 . GLOBL LCPUI, LCPULO, LCONTM, $CTRLS, $SPLJB, TXTCL
86 . GLOBL STPFLG, TON, USPLCH, SPLCHN, S$HICP, TXTC1
87 . GLOBL S$INWT, S$OTWT, S$TMWT, S$SFWT, S9600
88 . GLOBL S$MSWT, CFBUF, CFEND, CCLSAV, KMNCHN
89 . GLOBL MINTIM, LSECPT, MAXSEC, $EMTTR, VCSHNB
90 . GLOBL OKFILE, OKFEND, $CLTST, UCISPC, MHNSIZ
91 . GLOBL CASTBR, CASCBR, CASTBW, CASCUP, MHNSMS
92 . GLOBL CASTRO, CASTWO, CLTOTL, CO$DTR, CLSFSP
93 . GLOBL CO$CR, CO$FF, CO$FFO, CO$LC, CO$TAB, CO$CTL
94 . GLOBL CO$LFI, CO$LFO, CO$BNI, CO$BNO, CL$OPT
95 . GLOBL CL$LEN, CL$SKP, CL$WID, CL$LIN, PHMEM
96 . GLOBL LJSW, CTRLTT, NEWJSW, JSTKND, VIMAGE
97 . GLOBL USTART, GENTOP, BOTDEV, BOTUNI, CSHALC
98 . GLOBL $CTRLC, LSW2, $INKMN, CHAIN, UFORM
99 . GLOBL $SGQO, $SGQ3, LITIME
100 . GLOBL MAXASN, AT$$SZ, $CFABT, INDSTA, INDERR
101 . GLOBL RUNDEV, LNBLKS, CXTBAS, CXTWDS, UHMEM
102 . GLOBL ASNTBL, $DILUP, CSHDEV, CSHDVN, LNSBLK
103 . GLOBL ASNEND, LSW3, LSW2S, $DUPRN
104 . GLOBL $FORM, $TAB, LSCCA, $CFSOT, LOFSPC, R50COM
105 . GLOBL $PAGE, $SCOPE, $ECHO, $LC, $8BIT, CHKALC
106 . GLOBL UCHAN, $FORMO, $CFALL, $CFDCC, $CFCL
107 . GLOBL LNPRIM, LNMAP, CW$50H, CONFIG, $SUCF
108 . GLOBL $DOOFF, NUCHN, LRBFIL, CFIND, TALEMT
109 . GLOBL C. CSW, C. DEVQ, C. SBLK, NLINES, CO$BBT
110 . GLOBL CD$NAM, CD$DVU, CD$BAS, CD$JOB, CD$$SZ, CD$$UB
111 . GLOBL LTSCMD, LNSPAC, CFNEST, UCLNAM
112 . GLOBL $CFOPN, CFSEND, PBFEND, CFSP, $TTGAG
113 . GLOBL UFPTRP, SDSFCB, SD$DEL, CFLFL4, $UCLCF
114 . GLOBL SDFLAG, SD$FLK, SD$WFM, SDFORM, $UCLRN

```

```

115 . GLOBL SDBUF1, SDBLK, NSPLDV, LD$RON, $UCLCM, $UCLCL
116 . GLOBL LDNAME, LDSIZE, LDFLAG, LDBASE, LDPDEV
117 . GLOBL LSW8, $SQQ1, $SQQ1A, $SQQ1B, $SQQ1C, $SQQ2, $SGIIO, $SGHIO
118 . GLOBL $DEFER, CFCHAN, SCHAIN, LDDEVX, $SGALL
119 . GLOBL CFPNT, CFBLK, $QUIET, DIABFL
120 . GLOBL DIABNO, VT52NO, LA36NO, LA36FL
121 . GLOBL LSW4, KL4CLR, SDSKIP, SDBU, SD$BAK
122 . GLOBL $INCOR, $KED, VQUN1B, VINTIO, VQUN1C
123 . GLOBL SF$BSY, SFFORM, SD$SNG, SFNMBL, NFRESB
124 . GLOBL SD$HLD, SF$HLD, CURPRM, PRMPNT, SF$1ST
125 . GLOBL LSTPRM, PRMBUF, PRMEND, CFSPND
126 . GLOBL SDFHD, SFFLAG, SFQLNK, CFHOLD, LOGDVU, LOGBAS
127 . GLOBL LCCL, $QTSET, $TECO, CD$TOP, LOGCHK
128 . GLOBL $WILD, ERRSEV, UERSEV, PASLIN, LOGBAS, LOGDVU
129 . GLOBL LSTPL, SDCB, SDCBND, VQUANO, VQUAN3
130 . GLOBL VQUAN1, VQUN1A, VQUAN2, VHIPCT, VQUANO, VQUAN3
131 . GLOBL DCTRD, DCCRD, DCTWR, DCCWR
132 . GLOBL AT$LOG, AT$SIZ, AT$DEV, AT$FIL, AT$EXT, AT$$SZ
133 . GLOBL VCORTM, NUMDCD, KMPRMT, MXPRMT, C1DEVX
134 . GLOBL RDB, RDBEND, RT$DEV, RT$NAM, RT$$SZ, CLDEVX, SDDVU
135 . GLOBL SDNAME, SDCBSZ, LSTSL, LSTATE
136 . GLOBL TK1VAL, CINDAT, SYSDAT, SYTIMH, SYTIML
137 . GLOBL BASMAP, LOMAP, HIMAP, JCXPGS
138 . GLOBL SMRSIZ, SRTSIZ, CSHSIZ, TK1SEC
139 . GLOBL TSXLN, TSXSIT, GRT1, TRGRET, LICTXT, SUPCOD, NAMTOP, SUMS, SUCS
140 . GLOBL LPRG1, LPRG2, S$QUSR, S$IOWT
141 . GLOBL S$SPDB, S$SPCB, SFUSER, SFFILE, VT200, VT2007, VT2008
142 . GLOBL LCBIT, LA36, LA120, VT52, VT100, DIABLO, QUME
143 . GLOBL ADM3A, LTRMTP, LA12FL, LA12NO, VT52FL
144 . GLOBL VT10FL, VT10NO, QUMEFL, QUMENO, ADM3FL
145 . GLOBL VT20FL, VT20NO
146 . GLOBL ADM3NO, SYINDX, SYUNIT, NUMDEV, PNAME
147 . GLOBL OF$DEV, OF$UNT, OF$FIL, OF$FLG, SYNAME
148 . GLOBL OF$$SZ, OT$RON, RESDEV, $TAPE
149 . GLOBL KMNBAS, ODTBAS, $CTRLD
150 . GLOBL LSW6, $SNWTT, PF$SYS, PF$IOW, $DEBUG
151 . GLOBL RSR, TSR, LMXNUM, LSTMX, MXDTR, ZCLR, MXCSR
152 . GLOBL $INDDF, $INDRN, IN$ACT, IN$CNT, IN$CMD, INDSAV
153 . GLOBL $PHONE, INVEC, LMXLN, MXVEC, $INIT, $DEAD, $HARD
154 . GLOBL ITRMTP, LMXPRM, LSW7, $INDAB, CFSTS, CF$IND, CF$QUT
155 . GLOBL CFABLV, MONVEC, LBSPRI, MAXPRI, MXJPRI, LPRI
156 . GLOBL LOGCHN, LOGFLG, LOGPTR, LOGBUF, LOGBLK
157 . GLOBL LF$OPN, LF$WRT, UCLBLK, UCLDAT
158 . GLOBL CSHHD, FC$CDX, FC$LNK, FD$NAM, UC$NDC, UC$MDC, CVTUC
159 . GLOBL CMDBUF, PAUMSG, RDCMD, DKSAV, SYSAV, CVTTAB, RUNHD, SEARCH
160 . GLOBL FKILL, ABRTCF, ACRFN, XAREA, FILNAM, NOPRG, FPRINT
161 . GLOBL PUSHCF, TRMSTR, FILNAM, R5ODIR, R5OSY, R5OIND, R5OSAV
162 . GLOBL INDACT, R5ODUP, R5OPIP, R5OKED, R5OK52, R5OKEX, R5OTSX, R5OUCL
163 . GLOBL BLKO, RDERM, R5OVIR, NOSTRT, RUNEMT, OVRCOR
164 . GLOBL BADSAV, LDNAM, NOPRG, NOCIN, SIZVAL, ASKLNM, BADCMD, KCSIBF
165 . GLOBL ASDEX, KCSIMS, ASNOVF, GTRD50, R5OBUF, R5OLD0, MNTDEV, DMTARG
166 . GLOBL DEADEV, CHKMNT, CHKMTX, INFOMT, NOFLAG, MTOPHD, ILLCMD
167 . GLOBL R5OLD, INVLDN, R5ODSK, ACRFIL, BDFNAM, LOGASN, MNTFUL, R5OLD7
168 . GLOBL TBLOVF, SETHD, CSIMS2, CKPRIV, R5ONO, AMBOPT, ACRDEC
169 . GLOBL MAXAVL, PRTDEC, DEVUNT, PNAME, HNBUF
170 . GLOBL ACROCT, HANBSY, CSIMS1, MISSEQ, NOIND, POPCF
171 . GLOBL BADPMT, BADPRI, TOTXT, CRLF, HIPRI, STLGH, LOGCLS, R5OLOG

```

| | | |
|-----|---------|---|
| 172 | . GLOBL | BDLGOP, SPLHLA, NOCCL, LDOPHD, PRTFIX, PRTSPC |
| 173 | . GLOBL | DLTXT, OCTFIX, PRTTTP, NATXT, SPDTX1, NOTXT, YESTXT, NINTXT |
| 174 | . GLOBL | PRTUNM, SYHD1, SYHD2, PRTLN, SPACE2, DETTXT, SPACE3, RNMS, WTMS |
| 175 | . GLOBL | SWPTX, LOCKTX, SPACE5, PRTDC3, KBMSG, DIVIDE, PRTDC2 |
| 176 | . GLOBL | COLOO, CPUAH, CPUAL, PRTTMV, NOFIL, CMDBUF, CALUCL |
| 177 | . GLOBL | NOUDC, DEVHD1, ASNHD1, ASNHD2, SHMTH1, SHMTH2, PRTTMD |
| 178 | . GLOBL | CVDVNM, SPACE6, PRTBUF, PRTFNM, NONEMS, NODAT, NOLDMT |
| 179 | . GLOBL | SUBARO, EDTFIL, RONTXT, NOTAVL, KBTX, MNFLGS, MNBPC |
| 180 | . GLOBL | DELSPC, MNBASE, MNTOP, MONHD, MONAR1, NOPMGN, PMBUSY, MONAR2 |
| 181 | . GLOBL | NSWPMS, MAXMTX, CURMTX, CHKDLM, SPLHD, AMBOPT, INVOPT |
| 182 | . GLOBL | DEVIDL, COAL, ALDEX, COAD, SPACTV, SPWFM, DEVIDL, SPSNG |
| 183 | . GLOBL | COAL, ALDEX, ALDBLK, COAD, SPACTV, SPWFM, DEVIDL |
| 184 | . GLOBL | SPSNG, SPFUL, SPCF, SPFLK, NOFIL, SPGEMT, NOOPTT |
| 185 | . GLOBL | BDLIN, MSGBUF, MSGEND, NOTON, GAGMSG |
| 186 | . GLOBL | LINFRE, DJABMS, DLMSG, INVTIM, DMTALL |
| 187 | . GLOBL | SHTMSG, AUTHFN, SPLACT, DOSTOP, OFFEMT, KILEMT, UPTMMS |
| 188 | . GLOBL | TMTOTH, DIVSOR, TMTOTL, PRTPCT, SUM1, SUM2, SUM3, SUM4 |
| 189 | . GLOBL | SUM5, SUM6, SUM7, OTHRON, SPLPND, STPASK, SRTSMS |
| 190 | . GLOBL | SIZEMT, ASNOVF, INVLDM, CSIMS4, MNTARG, HUPARG, R50TT |
| 191 | . GLOBL | KMNNAM, NOKMON, CCLNAM, OTRMNT, CHKDEV, DMTSUB, CMDCCCL |
| 192 | . GLOBL | SHOHD, SUBTXT, MNTTXT, SRTTXT, TOTMMS, UMSSMS, SSRMAP |
| 193 | . GLOBL | TSXSMS, USRMMS, JCSMS, DZTXT, OCTPRT |
| 194 | . GLOBL | PRTR50, PRTDAT, PRTTOD, PRTTIM, INVDEV, ALFN, R50DK |
| 195 | . GLOBL | DETHD, DETARG, RUNMS, NOFRDL, R50MON, INV DAT, MUL32, COAF |
| 196 | . GLOBL | AR\$PRJ, AR\$PRG, AR\$CON, AR\$CNT, AR\$CPH, AR\$CPL, AR\$UNM |
| 197 | . GLOBL | AR\$DMY, AR\$\$SZ, ARNRPB, \$SLON, \$SLTTY, \$SLLET |
| 198 | . GLOBL | PRTWRN, SLMXLN, \$LOFCF, CSHMSG, \$CARMN, VUSPHN |
| 199 | . GLOBL | AF\$HIE, AF\$NOI, \$NDINT, AF\$PLK, AF\$DBG |
| 200 | . GLOBL | AF\$IOP, \$RNIOP, SHVTX1, SHVTX2, SHVTX3, SHVTX4, SJSPPN |
| 201 | . GLOBL | VONTM, VOFFTM, VTMIN, VTMOU, VTMLOC |
| 202 | . GLOBL | MDMTXT, OFTTXT, ONTTXT, TMITXT, TMOTXT, TMLTXT, PHNTXT |

```
1          ;
2          ; Assembly constants
3          ;
4          000012      LF      =      12      ; LINE FEED
5          000015      CR      =      15      ; CARRIAGE RETURN
6          000040      BLANK   =      40      ; ASCII SPACE
7          000007      BELL    =      07      ; ASCII BELL
8          000011      TAB     =      11      ; HORIZONTAL TAB
9          000014      FF      =      14      ; FORM FEED
10         000054      COMMA   =      54      ; COMMA
11         000400      BLKWDS  =      256.    ; # OF WORDS IN DISK BLOCK
12         132500      WLDNAM  =      132500  ; RAD50 /*/ (WILDCARD)
```

```

1      ; -----
2      ; Macro to cause a fatal error message to be printed.
3      ;
4      .MACRO FERR MSG
5      MOV R5, -(SP)
6      MOV MSG, R5
7      CALL FPRINT
8      MOV (SP)+, R5
9      .ENDM FERR
10     ;
11     ; -----
12     ; Macro to print a fatal error message, clean up
13     ; and then jump to RDCMD.
14     ;
15     .MACRO FABORT MSG
16     MOV MSG, R5
17     JMP FKILL
18     .ENDM FABORT
19     ;
20     ; -----
21     ; Macro to print a warning message
22     ;
23     .MACRO FWARN MSG
24     MOV R5, -(SP)
25     MOV MSG, R5
26     CALL PRTWRN
27     MOV (SP)+, R5
28     .ENDM FWARN
29     ;
30     ; -----
31     ; Macro to start a standard option table.
32     ; Name = 1 to 4 character table name.
33     ; NA = Number of arguments per table entry.
34     ;
35     .MACRO TBLDEF NAME, NA
36     NARGS = NA
37     .CSECT CMDVSH
38     NAME 'HD: .WORD 2*NA
39     .ENDM TBLDEF
40     ;
41     ; -----
42     ; Macro to enter an option text name and a set of parameters
43     ; into the currently open table.
44     ; STRNG = Ascii name
45     ; A, B, C = Set of option parameters to store in table with name.
46     ;
47     .MACRO CMDDEF STRNG, A, B, C
48     .CSECT NAMESH
49     L =
50     .ASCIZ /STRNG/
51     .CSECT CMDVSH
52     .WORD L ; POINTER TO NAME STRING
53     .WORD A
54     .IIF GE, <NARGS-2> .WORD B
55     .IIF GE, <NARGS-3> .WORD C
56     .ENDM CMDDEF
57     ;

```

```

58 ; -----
59 ; Macro to end a set of table entries.
60 ;
61 ; .MACRO TBLEND
62 ; .CSECT CMDVSH
63 ; .WORD 0
64 ; .CSECT TSKSHO
65 ; .ENDM TBLEND
66 ; -----
67 ; Define options for SHOW command
68 ;
69 000000 TBLDEF SHO, 1
70 000002 CMDDEF ALL, SOPALL
71 000006 CMDDEF ALLO*CATE, SOPALC
72 000012 CMDDEF ALLO*CATIONS, SOPALC
73 000016 CMDDEF AS*SIGNS, SOPASN
74 000022 CMDDEF CA*CHE, SOPCSH
75 000026 CMDDEF C*ONFIGURATION, SOPCON
76 000032 CMDDEF CL, SOPCL
77 000036 CMDDEF COM*MANDS, SOPCMD
78 000042 CMDDEF COR*TIM, SOPCTM
79 000046 CMDDEF C1, SOPCL
80 000052 CMDDEF D*EVICES, SOPDEV
81 000056 CMDDEF DA*TE, SOPDAT
82 000062 CMDDEF E*RRORS, RDCMD
83 000066 CMDDEF H*IPRCT, SOPHIO
84 000072 CMDDEF INS*TALL, SOPINS
85 000076 CMDDEF INT*IOC, SOPIID
86 000102 CMDDEF J*OBS, SOPJOB
87 000106 CMDDEF KEY*S, SHOKEY
88 000112 CMDDEF LD, SOPSUB
89 000116 CMDDEF M*EMORY, SOPMEM
90 000122 CMDDEF MAXMC, SOPMC
91 000126 CMDDEF MAXMRB, SOPMR
92 000132 CMDDEF MAXMSG, SOPMB
93 000136 CMDDEF MO*UNTS, SOPMNT
94 000142 CMDDEF MOD*EM, SOPMDM ; Synonym with PHONE
95 000146 CMDDEF N*UMDC, SOPNDC
96 000152 CMDDEF PH*ONE, SOPMDM ; Synonym with MODEM
97 000156 CMDDEF PRI*ORITY, SOPPRI
98 000162 CMDDEF PRIL*OW, SOPPLO
99 000166 CMDDEF PRIH*I, SOPPHI
100 000172 CMDDEF PRID*EF, SOPPDF
101 000176 CMDDEF PRIV*ILEGES, SOPPRV
102 000202 CMDDEF PRIVIR, SOPPVR
103 000206 CMDDEF Q*UEUE, SOPQUE
104 000212 CMDDEF QUANO, SOPQO
105 000216 CMDDEF QUAN1, SOPQ1
106 000222 CMDDEF QUAN1A, SOPQ1A
107 000226 CMDDEF QUAN1B, SOPQ1B
108 000232 CMDDEF QUAN1C, SOPQ1C
109 000236 CMDDEF QUAN2, SOPQ2
110 000242 CMDDEF QUAN3, SOPQ3
111 000246 CMDDEF REG*IONS, SOPREG
112 000252 CMDDEF RUN*-TIMES, SOPSRT
113 000256 CMDDEF S*UBSETS, SOPSUB
114 000262 CMDDEF SL*E, SOPSLE

```


| | | |
|------------|--------|----------------------|
| 115 000266 | CMDDEF | SP*OOL, SOPSPL |
| 116 000272 | CMDDEF | SYSP*ASSWORD, SOPSYP |
| 117 000276 | CMDDEF | T*TERMINALS, SOPTRM |
| 118 000302 | CMDDEF | TI*ME, SOPTIM |
| 119 000306 | CMDDEF | TT*Y, SOPTRM |
| 120 000312 | CMDDEF | USE, SOPUSE |
| 121 000316 | CMDDEF | USERS, SOPJOB |
| 122 000322 | CMDDEF | VE*RSION, SOPVER |
| 123 000326 | CMDDEF | VM, SOPVM |
| 124 000332 | TBLEND | |

Data areas

```

1                                     .SBTTL  Data areas
2                                     ;-----
3                                     ; Data areas
4                                     ;
5 000000 075250 014644 000000 HANNAM: .RAD50 /SY ddd   TSX/ ;File spec for device handler
   000006 100020
6 000010 000000          DVEC:  .WORD  0
7 000012 000000          DCSR:  .WORD  0
8 000014 012276          R5OCLO: .RAD50 /CLO/
9 000016 000000          CDBASE: .WORD  0
10 000020 000000          CDDVU:  .WORD  0
11 000022 000000          CDBAS5: .WORD  0
12                                     ;
13                                     ; Table of terminal device type names.
14                                     ; Note, the order of the entries in this table must match the order
15                                     ; of the values of the CDX$xx entries as defined in TSGEN.
16                                     ;
17 000024 015340          CTLNAM: .RAD50 /DL /           ; CDX$DL
18 000026 016420          .RAD50 /DZ /           ; CDX$DZ
19 000030 015100          .RAD50 /DH /           ; CDX$DZ
20 000032 015126          .RAD50 /DHV/         ; CDX$VH
21 000034 062170          .RAD50 /PC /           ; CDX$PI
22 000036 012500          .RAD50 /CP /           ; CDX$PC
23 000040 063200          .RAD50 /PP /           ; CDX$PP
24 000042 066300          .RAD50 /QP /           ; CDX$QP
25 000044 114700          .RAD50 /XX /

```

SHOW command

```

1                                     .SBTTL  SHOW command
2                                     -----
3                                     ; Process the SHOW command
4                                     ;
5 000046 000240                      CMDSHD: NOP
6 000050 004767 000000G              CALL    CVTTAB          ; CONVERT TABS AND FF'S TO SPACES
7 000054 105713                      TSTB    @R3            ; IF NO OPTION WAS SPECIFED WITH COMMAND,
8 000056 001555                      BEQ     SOPASN        ; TREAT COMMAND LIKE "SHOW ASSIGNS"
9 000060 012704 000000'              MOV     #SHOHD,R4    ; POINT TO TABLE OF SHOW OPTIONS
10 000064 004767 000000G             CALL    SEARCH        ; LOOK UP OPTION
11 000070 103401                      BCS    BDSO          ; BR IF INVALID OPTION
12 000072 000134                      JMP     @(R4)+        ; JUMP TO PROCESSING ROUTINE
13                                     ;
14                                     ; Invalid option
15                                     ;
16 000074 005704                      BDSO:   TST    R4      ; Invalid or ambiguous option?
17 000076 001404                      BEQ    1$           ; Br if invalid
18 000100                              FABORT #AMBOPT      ; Ambigious option
19 000110                              1$:   FABORT #INVOPT ; Invalid option

```

ALL

```

1
2
3
4
5 000120 004767 010022      .SBTTL      ALL
6 000124
7 000132 004767 005052      ; SHOW ALL
8 000136
9 000144 004767 006074      ;
10 000150
11 000156 004767 006430      SOPALL: CALL  SHOVER      ; SHOW VERSION
12 000162      .PRINT #CRLF
13 000170 004767 003534      CALL  SHODEV      ; SHOW DEVICES
14 000174 004767 001446      .PRINT #CRLF
15 000200      CALL  SHOASN      ; SHOW ASSIGNS
16 000206 004767 002630      .PRINT #CRLF
17 000212      CALL  SHOALC      ; SHOW ALLOCATIONS
18 000220 004767 001056      .PRINT #CRLF
19 000224      CALL  SHOJOB      ; SHOW JOBS
20 000232 004767 012012      CALL  SHOTRM      ; SHOW TERMINALS
21 000236      .PRINT #CRLF
22 000244      CALL  SHOCL      ; SHOW CL
23 000252 004767 012102      .PRINT #CRLF
24 000256      CALL  SHOMEM      ; SHOW MEMORY
25 000264      .PRINT #CRLF
26 000272 004767 006532      CALL  SHOSPL      ; SHOW SPOOL
27 000276      .PRINT #CRLF
28 000304      .PRINT #SUBTXT
29 000312 004767 011630      CALL  SHOSUB      ; SHOW SUBSETS
30 000316      .PRINT #CRLF
31 000324 004767 010560      .PRINT #MNTTXT
32 000330      CALL  SHOMNT      ; SHOW MOUNTS
33 000336 032767 000000G 000000G .PRINT #CRLF
34 000344 001405      CALL  SHOSRT      ; SHOW RUN-TIMES
35 000346 004767 007656      .PRINT #CRLF
36 000352      CALL  SHOREG      ; SHOW REGIONS
37 000360 004767 011264      .PRINT #CRLF
38 000364      CALL  SHOPRV      ; SHOW PRIVILEGES
39 000372 004767 011372      .PRINT #CRLF
40 000376      CALL  SHOSLE      ; SHOW SL
41 000404 004767 012220      .PRINT #CRLF
42 000410 000426      CALL  SHOWVM      ; SHOW VM (Base, Top and Size)
43      BR      SOPJMP
44      ;
45      ; SHOW ASSIGNS
46 000412 004767 005626      ;
47 000416 000423      SOPASN: CALL  SHOASN      ; SHOW ASSIGNS
48      BR      SOPJMP
49      ;
50      ; SHOW COMMANDS
51 000420      ;
52 000420 004767 004526      UCLCMD:
53 000424 000420      SOPCMD: CALL  SHOCMD      ; SHOW COMMANDS
54      BR      SOPJMP
55      ;
56      ; SHOW CONFIGURATION
57 000426 004767 004556      ;
      SOPCON: CALL  SHODEV      ; SHOW DEVICES

```

ALL

```

58 000432 004767 001210          CALL  SHOTRM          ; SHOW TERMINALS
59 000436 000413          BR    SOPJMP
60                               ;
61                               ; SHOW DATE
62                               ;
63 000440 004767 007430  SOPDAT: CALL  SHODAT          ; SHOW DATE
64 000444 000410          BR    SOPJMP
65                               ;
66                               ; SHOW DEVICES
67                               ;
68 000446 004767 004536  SOPDEV: CALL  SHODEV          ; SHOW DEVICES
69 000452 000405          BR    SOPJMP
70                               ;
71                               ; SHOW JOBS
72                               ;
73 000454 004767 003250  SOPJOB: CALL  SHOJOB          ; SHOW JOBS
74 000460 000402          BR    SOPJMP
75                               ;
76                               ; SHOW MEMORY
77                               ;
78 000462 004767 000614  SOPMEM: CALL  SHOMEM          ; SHOW MEMORY
79 000466 000167 000000G SOPJMP: JMP   RDCMD
80                               ;
81                               ; SHOW ALLOCATIONS
82                               ;
83 000472 004767 006114  SOPALC: CALL  SHOALC          ; SHOW ALLOCATIONS
84 000476 000167 000000G      JMP   RDCMD
85                               ;
86                               ; SHOW CL
87                               ;
88 000502 004767 002334  SOPCL:  CALL  SHOCL           ; SHOW CL
89 000506 000767          BR    SOPJMP
90                               ;
91                               ; SHOW INSTALL
92                               ;
93 000510 004767 007514  SOPINS: CALL  SHOINS          ; SHOW INSTALL
94 000514 000764          BR    SOPJMP
95                               ;
96                               ; SHOW MOUNTS
97                               ;
98 000516 004767 006306  SOPMNT: CALL  SHOMNT          ; SHOW MOUNTS
99 000522 000761          BR    SOPJMP
100                              ;
101                              ; SHOW MODEM (ONTIM, OFFTIM, TIMIN, TIMOUT, TIMLOC)
102                              ;
103 000524 004767 007172  SOPMDM: CALL  SHOMDM          ; SHOW MODEM
104 000530 000756          BR    SOPJMP
105                              ;
106                              ; SHOW PRIVILEGES
107                              ;
108 000532 004767 011112  SOPPRV: CALL  SHOPRV          ; SHOW PRIVILEGES
109 000536 000753          BR    SOPJMP
110                              ;
111                              ; SHOW QUEUE
112                              ;
113 000540 004767 004310  SOPQUE: CALL  SHOQUE          ; SHOW QUEUE
114 000544 000750          BR    SOPJMP

```

ALL

```

115 ;
116 ; SHOW RUN-TIMES
117 ;
118 000546 004767 011374 SOPSRT: CALL SHOSRT ; SHOW RUN-TIMES
119 000552 000745 BR SOPJMP
120 ;
121 ; SHOW SL
122 ;
123 000554 004767 011210 SOPSLE: CALL SHOSLE ; SHOW SL
124 000560 000742 BR SOPJMP
125 ;
126 ; SHOW VM
127 ;
128 000562 004767 012042 SOPVM: CALL SHOWVM ; SHOW VM
129 000566 000737 BR SOPJMP
130 ;
131 ; SHOW REGIONS
132 ;
133 000570 004767 010314 SOPREG: CALL SHOREG ; SHOW REGIONS
134 000574 000734 BR SOPJMP
135 ;
136 ; SHOW SUBSETS
137 ;
138 000576 004767 011556 SOPSUB: CALL SHOSUB ; SHOW SUBSETS
139 000602 000731 BR SOPJMP
140 ;
141 ; SHOW SPOOL
142 ;
143 000604 004767 011440 SOPSPL: CALL SHOSPL ; SHOW SPOOL
144 000610 000726 BR SOPJMP
145 ;
146 ; SHOW SYSPASSWORD
147 ;
148 000612 004767 000432 SOPSYP: CALL SHOSYP ; SHOW SYSPASSWORD
149 000616 000723 BR SOPJMP
150 ;
151 ; SHOW TERMINALS
152 ;
153 000620 004767 001022 SOPTRM: CALL SHOTRM ; SHOW TERMINALS
154 000624 000720 BR SOPJMP
155 ;
156 ; SHOW TIME
157 ;
158 000626 004767 007300 SOPTIM: CALL SHOTIM ; SHOW TIME
159 000632 000715 BR SOPJMP
160 ;
161 ; SHOW USE
162 ;
163 000634 004767 007362 SOPUSE: CALL SHOUSE ; SHOW USE
164 000640 000712 BR SOPJMP
165 ;
166 ; SHOW VERSION
167 ;
168 000642 004767 007300 SOPVER: CALL SHOVER ; SHOW VERSION
169 000646 000707 BR SOPJMP

```

ALL

```

1      ;
2      ; SHOW MAXMSG
3      ;
4 000650 016705 000000G  SOPMB:  MOV    VMXMSG, R5
5 000654 000427          BR      SHOVAL
6      ;
7      ; SHOW MAXMC
8      ;
9 000656 016705 000000G  SOPMC:  MOV    VMAXMC, R5
10 000662 000424         BR      SHOVAL
11     ;
12     ; SHOW MAXMRB
13     ;
14 000664 016705 000000G  SOPMR:  MOV    VMXMRB, R5
15 000670 000421         BR      SHOVAL
16     ;
17     ; SHOW QUANO
18     ;
19 000672 016705 000000G  SOPQ0:  MOV    VQUANO, R5      ;Get value
20 000676 000416         BR      SHOVAL
21     ;
22     ; SHOW QUAN1
23     ;
24 000700 016705 000000G  SOPQ1:  MOV    VQUAN1, R5     ;GET VALUE
25 000704 000413         BR      SHOVAL
26     ;
27     ; SHOW QUAN1A
28     ;
29 000706 016705 000000G  SOPQ1A: MOV    VQUN1A, R5
30 000712 000410         BR      SHOVAL
31     ;
32     ; SHOW QUAN1B
33     ;
34 000714 016705 000000G  SOPQ1B: MOV    VQUN1B, R5
35 000720 000405         BR      SHOVAL
36     ;
37     ; SHOW QUAN1C
38     ;
39 000722 016705 000000G  SOPQ1C: MOV    VQUN1C, R5
40 000726 000402         BR      SHOVAL
41     ;
42     ; SHOW QUAN2
43     ;
44 000730 016705 000000G  SOPQ2:  MOV    VQUAN2, R5
45 000734          SHOVAL: .PRINT #SPACE3      ;Print 3 spaces
46 000742 004767 000000G          CALL  PRTDEC
47 000746          .PRINT #CRLF
48 000754 000167 000000G          JMP    RDCMD
49     ;
50     ; SHOW QUAN3
51     ;
52 000760 016705 000000G  SOPQ3:  MOV    VQUAN3, R5     ;Get QUAN3 value
53 000764 000763          BR      SHOVAL      ;Display it
54     ;
55     ; SHOW CORTIM
56     ;
57 000766 016705 000000G  SOPCTM: MOV    VCORTM, R5

```

ALL

```

58 000772 000760          BR      SHOVAL
59                          ;
60                          ; SHOW INTIOC
61                          ;
62 000774 016705 000000G  SOPIID: MOV      VINTIO,R5
63 001000 000755          BR      SHOVAL
64                          ;
65                          ; SHOW HIPRCT
66                          ;
67 001002 016705 000000G  SOPHID: MOV      VHIPCT,R5
68 001006 000752          BR      SHOVAL
69                          ;
70                          ; SHOW NUMDC
71                          ;
72 001010 016705 000000G  SOPNDC: MOV      NUMDCD,R5
73 001014 000747          BR      SHOVAL
74                          ;
75                          ; SHOW PRILOW
76                          ;
77 001016 116705 000000G  SOPPLO: MOV      VPRILO,R5
78 001022 000744          BR      SHOVAL
79                          ;
80                          ; SHOW PRIHI
81                          ;
82 001024 116705 000000G  SOPPHI: MOV      VPRIHI,R5
83 001030 000741          BR      SHOVAL
84                          ;
85                          ; SHOW PRIDEF
86                          ;
87 001032 116705 000000G  SOPPDF: MOV      VPRIDF,R5
88 001036 000736          BR      SHOVAL
89                          ;
90                          ; SHOW PRIVIR
91                          ;
92 001040 116705 000000G  SOPPVR: MOV      VPRIVR,R5
93 001044 000733          BR      SHOVAL
94                          ;
95                          ; SHOW PRIORITY
96                          ;
97 001046          SOPPRI: .PRINT #TM#PR1          ;"Current priority = "
98 001054 116105 000000G      MOV      LPRI(R1),R5      ;Get current priority
99 001060 004767 000000G      CALL     PRTDEC          ;Print it
100 001064          .PRINT #TM#PR2          ;"; maximum authorized priority = "
101 001072 116705 000000G      MOV      MXJPRI,R5      ;Get max authorized priority
102 001076 004767 000000G      CALL     PRTDEC          ;Print it
103 001102          .PRINT #CRLF          ;End of line
104 001110          .PRINT #TM#LPR          ;"Low priority range = 0 to "
105 001116 116705 000000G      MOV      VPRILO,R5      ;Get top of low priority range
106 001122 004767 000000G      CALL     PRTDEC          ;Print the value
107 001126          .PRINT #CRLF
108 001134          .PRINT #TM#HPR          ;"High priority range = "
109 001142 116705 000000G      MOV      VPRIHI,R5      ;Get base of high priority range
110 001146 004767 000000G      CALL     PRTDEC          ;Print it
111 001152          .PRINT #TM#HPE          ;Finish message
112 001160 000167 000000G      JMP      RDCMD
113                          ;
114                          ; SHOW CACHE

```


ALL

```

115
116 001164 016705 000000G      ;
SOPCSH: MOV      CSHALC,R5      ;Was caching genned into system?
117 001170 001004              BNE      1$                    ;Br if yes
118 001172                      .PRINT  #TM$CNG                    ;Caching not genned into system
119 001200 000421              BR       9$
120 001202 005767 000000G      1$:   TST      VCSHNB          ;Is caching enabled?
121 001206 001004              BNE      2$                    ;Br if yes
122 001210                      .PRINT  #TM$CDS                    ;Caching is disabled
123 001216 000412              BR       9$
124 001220                      2$:   .PRINT  #TM$CEN                    ;Data cache size is ...
125 001226 016705 000000G      MOV      VCSHNB,R5            ;Get current number of blocks for cache
126 001232 004767 000000G      CALL    PRTDEC                ;Print size of cache
127 001236                      .PRINT  #CRLF                    ;Terminate print line
128 001244 000167 000000G      9$:   JMP      RDCMD
129
130
131
132      ; SHOW SYSPASSWORD
133 001250 004767 000000G      SHOSYP: CALL   CKSYPV          ;Make sure user has SYSPRV privilege
134 001254 105767 000000G      TSTB    SYPSWD                ;Is there a system password?
135 001260 001004              BNE      1$                    ;Br if yes
136 001262                      .PRINT  #TM$NSP                    ;No system password
137 001270 000403              BR       9$
138 001272                      1$:   .PRINT  #SYPSWD                ;Print the password
139 001300 000207              9$:   RETURN

```

MEMORY

```

1          .SBTTL          MEMORY
2          ;-----
3          ; SHOW MEMORY
4          ;
5 001302 010546 SHOMEM: MOV      R5, -(SP)
6          ; Total installed memory
7 001304          .PRINT #TOTMMS          ; TOTAL MEMORY
8 001312 016705 000000G MOV      PHYMEM, R5          ; Get total # 64-byte blocks of memory
9 001316 072527 177775 ASH      #-3, R5          ; Convert to # 512-byte pages
10 001322 042705 160000 BIC      #160000, R5      ; Kill possible sign extension
11 001326 004767 012230 CALL     PRTKB          ; DISPLAY THE VALUE
12          ; Size of unmapped system & unmapped handlers
13 001332          .PRINT #UMSSMS          ; SIZE OF UNMAPPED PORTION OF SYSTEM
14 001340 016705 000000G MOV      UMSYTP, R5      ; GET ADDRESS OF TOP OF TSX
15 001344 062705 001777 ADD      #1777, R5      ; BOUND UP TO KB
16 001350 000241 CLC          ; CVT TO KB
17 001352 006005 ROR      R5          ; /2 without propagating sign
18 001354 072527 177767 ASH      #-9, R5          ; /2/512. =/1024.
19 001360 004767 000000G CALL     PRTDEC          ; Display #Kb
20 001364          .PRINT #KBPARN          ; "Kb ("
21 001372 016702 000000G MOV      UMSYTP, R2      ; Get limit again
22 001376 004767 000000G CALL     OCTPRT          ; Display high address
23 001402          .PRINT #PARNNL          ; ")<CR><LF>"
24          ; Size of mapped system regions
25 001410          .PRINT #SSRMAMP          ; SIZE OF SYSTEM MAPPED REGION
26 001416 016705 000000G MOV      SMRSIZ, R5      ; # 64-byte blocks for mapped regions
27 001422 062705 000007 ADD      #7, R5          ; Round up
28 001426 072527 177775 ASH      #-3, R5          ; Convert to 512-byte block units
29 001432 004767 012124 CALL     PRTKB
30 001436 010546 MOV      R5, -(SP)      ; SAVE MAPPED REGION SIZE
31          ; Total TSX-Plus size
32 001440          .PRINT #TSXSMS          ; TSX SIZE
33 001446 016705 000000G MOV      LOMAP, R5      ; PAGE AT TOP OF TSX
34 001452 166705 000000G SUB      BASMAP, R5      ; # PAGES FOR TSX AND HANDLERS
35 001456 005305 DEC      R5
36 001460 062605 ADD      (SP)+, R5      ; ADD MAPPED REGION SIZE
37 001462 004767 012074 CALL     PRTKB
38          ; Mapped handlers
39 001466          .PRINT #MHNSMS          ; Size of mapped handlers
40 001474 016705 000000G MOV      MHNSIZ, R5      ; # 64-byte blocks for mapped handlers
41 001500 062705 000007 ADD      #7, R5          ; Round up
42 001504 072527 177775 ASH      #-3, R5          ; Convert to 512-byte block units
43 001510 004767 012046 CALL     PRTKB
44          ; Shared run-time systems
45 001514          .PRINT #SRTSMS          ; SIZE OF SHARABLE RUN-TIME SYSTEMS
46 001522 016705 000000G MOV      SRTSIZ, R5      ; # 64-byte blocks for shared run-times
47 001526 062705 000007 ADD      #7, R5          ; Round up
48 001532 072527 177775 ASH      #-3, R5          ; Convert to 512-byte block units
49 001536 004767 012020 CALL     PRTKB
50          ; Data cache buffers
51 001542          .PRINT #CSHMSG          ; Size of data cache
52 001550 016705 000000G MOV      CSHSIZ, R5      ; # 64-byte blocks for data cache
53 001554 062705 000007 ADD      #7, R5          ; Round up
54 001560 000241 CLC          ; Clear carry to convert to 512 byte
55 001562 006005 ROR      R5          ; block units without sign extension
56 001564 072527 177776 ASH      #-2, R5          ; in case cache >=2Mb (100000 64bytes)
57 001570 004767 011766 CALL     PRTKB

```

MEMORY

```

58                                     ; User memory space
59 001574                             .PRINT #USRMMS           ;USER MEMORY SPACE
60 001602 016705 000000G              MOV  HIMAP,R5
61 001606 166705 000000G              SUB  LOMAP,R5
62 001612 004767 011744                CALL  PRTKB
63                                     ; Job context area size
64 001616                             .PRINT #JCXSMS           ;SIZE OF JOB CONTEXT AREA
65 001624 016705 000000G              MOV  JCXPGS,R5
66 001630 005205                       INC  R5                ;ROUND UP
67 001632 004767 011724                CALL  PRTKB
68 001636 004767 011620                CALL  DSPMEM           ;SHOW JOB MEMORY LIMITS
69 001642 012605                       MOV  (SP)+,R5
70 001644 000207                       RETURN

```

TERMINALS

```

1          .SBTTL          TERMINALS
2          ;-----
3          ; Display information about terminals.
4          ;
5 001646 010146 SHOTRM: MOV      R1,-(SP)
6 001650 010246      MOV      R2,-(SP)
7 001652 010346      MOV      R3,-(SP)
8 001654 010446      MOV      R4,-(SP)
9 001656 010546      MOV      R5,-(SP)
10         ;
11         ; Print heading lines
12         ;
13 001660      .PRINT  #CRLF          ;Print a blank line
14 001666      .PRINT  #TRMHD1       ;Heading line 1
15 001674      .PRINT  #TRMHD2       ;Heading line 2
16         ;
17         ; Begin loop to print info for each terminal
18         ;
19 001702 012701 000002      MOV      #2,R1          ;Get first terminal index number
20         ;
21         ; If we are on a pro, do not display info about uninstalled lines
22         ;
23 001706 032761 000000G 000000G 1$: BIT      #$DEAD,LSW3(R1) ;Is this line installed?
24 001714 001406      BEQ      33$          ;Br if yes
25 001716 032767 000000G 000000G      BII     #CW$PRO,CONF02 ;Are we running on a pro?
26 001724 001402      BEQ      33$          ;Br if not
27 001726 000167 001044      JMP      30$          ;Go check next line
28         ;
29         ; Display unit number
30         ;
31 001732 010105 33$:      MOV      R1,R5          ;Get terminal index #
32 001734 006205      ASR      R5          ;Convert to unit #
33 001736 012703 000002      MOV      #2,R3          ;Print using 2 columns
34 001742 004767 000000G      CALL     PRTFIX        ;Print unit number
35         ;
36         ; Print "*" if this is the current line
37         ;
38 001746 116700 000000G      MOVVB   CORUSR,R0          ;Get current job index #
39 001752 116000 000000G      MOVVB   LNPRIM(R0),R0 ;Get primary line #
40 001756 012703 000004      MOV      #4,R3          ;Set to print 4 spaces
41 001762 020100      CMP      R1,R0          ;Is this our line?
42 001764 001035      BNE     15$          ;Br if not
43 001766      .TTYOUT #'*          ;Flag our line
44 001776 005303      DEC      R3          ;Print only 3 more spaces
45 002000 004767 000000G 15$: CALL     PRTSPC        ;Print spaces
46         ;
47         ; Display type of terminal
48         ;
49 002004 012703 000000G      MOV      #OPRTXT,R3      ;Assume this is operator's console
50 002010 120167 000000G      CMPB   R1,CTRLTT      ;Is this operator's console?
51 002014 001430      BEQ      2$          ;Br if yes
52 002016 012703 000000G      MOV      #CLLINE,R3      ;Assume this is a CL line
53 002022 020127 000000G      CMP      R1,#LSTPL      ;Is this a CL line?
54 002026 101023      BHI     2$          ;Br if yes
55 002030 012703 000000G      MOV      #LCLTXT,R3      ;Assume this is a local terminal
56 002034 032761 000000G 000000G      BIT      #$PHONE,ILSW2(R1);Is this a dial-up terminal?
57 002042 001415      BEQ      2$          ;Br if not dial-up

```

TERMINALS

```

58 002044 032761 000000G 000000G      BIT    #$INIT,LSW(R1) ;Is line in use now?
59 002052 001407                      BEQ    17$             ;Print "PHONE" if inactive dial-up line
60 002054 105767 000000G                TSTB   VUSPHN         ;Are phone lines always phone?
61 002060 001004                      BNE    17$             ;If so, print "PHONE"
62 002062 032761 000000G 000000G      BIT    #$CARMN,LSW5(R1) ;Is line being used as dial-up?
63 002070 001402                      BEQ    2$              ;Print "LOCAL" if active but no carrier
64 002072 012703 000000G                17$:  MOV    #REMTXT,R3    ;This is a remote terminal
65 002076                      2$:   .PRINT R3        ;Print type
66                                     ;
67                                     ;   Display vector address and DL/DZ type
68                                     ;
69 002102 016100 000000G                MOV    LCDTYP(R1),R0  ;Get device type code for this line
70 002106 016000 000024'                MOV    CTLNAM(R0),R0  ;Get RAD50 name of controlling device
71 002112 004767 000000G                CALL   PRTR50         ;Print the device name
72 002116 016105 000000G                MOV    LMXLN(R1),R5   ;Get line # within mux
73 002122 016104 000000G                MOV    LMXNUM(R1),R4  ;Get mux index number
74 002126 001007                      BNE    3$              ;Br if this is a mux line
75                                     ;   This is not a mux line
76 002130 012703 000004                MOV    #4,R3          ;Print 4 spaces
77 002134 004767 000000G                CALL   PRTSPC         ;
78 002140 016105 000000G                MOV    INVEC(R1),R5   ;Get input vector address
79 002144 000420                      BR     4$              ;
80                                     ;   This is a mux line
81 002146                      3$:   .TTYOUT #'-         ;Put in hyphen
82 002156 016105 000000G                MOV    LMXLN(R1),R5   ;Get line # within the mux
83 002162 012703 000002                MOV    #2,R3          ;Print in 2 column field
84 002166 004767 000000G                CALL   PRTFIX         ;Print line number
85 002172 012703 000001                MOV    #1,R3          ;Print 1 space
86 002176 004767 000000G                CALL   PRTSPC         ;
87 002202 016405 000000G                MOV    MXVEC(R4),R5   ;Get DZ-11 interrupt vector address
88 002206 012703 000003                4$:  MOV    #3,R3          ;Print 3 digits
89 002212 004767 000000G                CALL   OCTFIX         ;Print vector address
90 002216 012703 000002                MOV    #2,R3          ;Print 2 spaces
91 002222 004767 000000G                CALL   PRTSPC         ;
92                                     ;
93                                     ;   Print address of Control Status Register (CSR)
94                                     ;
95 002226 016104 000000G                MOV    LMXNUM(R1),R4  ;Get mux index number
96 002232 001003                      BNE    10$            ;Br if DZ-11
97                                     ;   This is a DL-11 line
98 002234 016102 000000G                MOV    RSR(R1),R2    ;Get address of receiver status register
99 002240 000402                      BR     11$            ;
100                                    ;   This is a DZ-11 line
101 002242 016402 000000G                10$: MOV    MXCSR(R4),R2 ;Get address of CSR
102 002246 004767 000000G                11$: CALL   OCTPRT      ;Print the address
103 002252 012703 000002                MOV    #2,R3          ;Print 2 spaces
104 002256 004767 000000G                CALL   PRTSPC         ;
105                                     ;
106                                     ;   Print information about terminal type
107                                     ;
108 002262 004767 000000G                CALL   PRTTTP         ;Print terminal type
109 002266 012703 000002                MOV    #2,R3          ;Print 2 spaces
110 002272 004767 000000G                CALL   PRTSPC         ;
111                                     ;
112                                     ;   Print information about speed
113                                     ;
114 002276 032761 000000G 000000G      BIT    #$AUTO,ILSW2(R1);Is autobaud specified for this line?

```

TERMINALS

```

115 002304 001413          BEQ      24$          ;Br if not
116 002306 005761 000000G   TST      LCLUNT(R1)   ;Is this line being used as a CL unit?
117 002312 002010          BGE      24$          ;Br if yes -- No autobaud for CL
118 002314 032761 000000G 000000G   BIT      ##DILUP,LSW(R1) ;Is a job active on this line?
119 002322 001004          BNE      24$          ;Br if yes -- speed is known if job on
120 002324          .PRINT  #TM$AUT      ;Print "auto"
121 002332 000424          BR       7$
122 002334 016103 000000G   24$:    MOV      LMXPRM(R1),R3 ;Get line parameters for mux line
123 002340 000303          SWAB     R3           ;Right justify speed code
124 002342 042703 177760          BIC      #177760,R3   ;Clear all but speed code
125 002346 001004          BNE      5$
126 002350          .PRINT  #NATXT      ;Print "N/A"
127 002356 000412          BR       7$
128 002360 070327 000005   5$:     MUL      #5,R3       ;Values are stored 5 characters each
129 002364 062703 000000G   ADD      #SPDXT1,R3   ;Point to text string
130 002370 012704 000005   MOV      #5,R4        ;Print 5 characters
131 002374 112300          6$:     MOVVB   (R3)+,R0      ;Get next char of speed
132 002376          .TTYOUT ;Print it
133 002402 077404          SOB     R4,6$        ;Loop till all printed
134 002404          7$:     .TTYOUT #40         ;Print a space
135          ;
136          ; Print information about character length and parity
137          ;
138 002414 116103 000001G   MOVVB   LMXPRM+1(R1),R3 ;Get line parameter flags
139 002420 032703 000000G   BIT      #LP$7BT,R3   ;7 bit characters wanted?
140 002424 001005          BNE      27$
141 002426          .TTYOUT #'8         ;Say length is 8 bits
142 002436 000404          BR      28$
143 002440          27$:    .TTYOUT #'7         ;Say length is 7 bits
144 002450 032703 000000G   28$:    BIT      #LP$PAR,R3 ;Parity wanted?
145 002454 001005          BNE      29$
146 002456          .TTYOUT #'N        ;Say no parity
147 002466 000414          BR      31$
148 002470 032703 000000G   29$:    BIT      #LP$ODD,R3 ;Odd parity wanted?
149 002474 001005          BNE      32$
150 002476          .TTYOUT #'E        ;Say even parity
151 002506 000404          BR      31$
152 002510          32$:    .TTYOUT #'O        ;Say odd parity
153 002520 012703 000004   31$:    MOV      #4,R3     ;Print 4 spaces
154 002524 004767 000000G   CALL    PRTSPC
155          ;
156          ; Indicate if line is active now
157          ;
158 002530 012703 000000G   MOV      #NOTXT,R3    ;Assume line is not active
159 002534 032761 000000G 000000G   BIT      ##DEAD,LSW3(R1) ;Is line installed?
160 002542 001035          BNE      12$
161 002544 016105 000000G   MOV      LCLUNT(R1),R5 ;Is a CL unit assigned to this line?
162 002550 002012          BGE      16$
163 002552 020127 000000G   CMP      R1,#LSTPL    ;Is this a real or CL line?
164 002556 101027          BHI      12$
165 002560 032761 000000G 000000G   BIT      ##INIT,LSW(R1) ;Is line active now?
166 002566 001423          BEQ      12$
167 002570 012703 000000G   MOV      #YESTXT,R3   ;Point to "Yes" text
168 002574 000420          BR      12$
169 002576 006205          16$:    ASR      R5         ;Convert CL unit index to unit number
170 002600 020527 000007   CMP      R5,#7.      ;Units 0-7 go with CL device
171 002604 101406          BLOS    34$

```

TERMINALS

```

172 002606 162705 000010          SUB      #8,R5          ;Remove C1 unit bias
173 002612                    .PRINT  #TXTC1        ;Print "C1"
174 002620 000403          BR       35$
175 002622                    34$: .PRINT  #TXTCL        ;Print "CL"
176 002630 004767 000000G      35$: CALL  PRTDEC        ;Print CL unit number
177 002634 000402          BR       25$
178 002636                    12$: .PRINT  R3          ;Print Yes/No
179 002642 012703 000003      25$: MOV   #3,R3        ;Print 3 spaces
180 002646 004767 000000G      CALL  PRTSPC
181                                ;
182                                ; Print name of user who is using this line
183                                ;
184 002652 032761 000000G 000000G BIT     #$DEAD,LSW3(R1) ;Is this line installed?
185 002660 001404          BEQ     26$          ;Br if line is installed
186 002662                    .PRINT  #NINTXT        ;Print Not installed
187 002670 000437          BR       14$
188 002672 016105 000000G      26$: MOV   LNAME(R1),R5 ;Is there a descriptive name for this line?
189 002676 001405          BEQ     13$          ;Br if not
190 002700 105715          TSTB   (R5)          ;Is the line blank?
191 002702 001403          BEQ     13$          ;Branch if no line name
192 002704                    .PRINT  R5          ;Print asciz name string
193 002710 000432          BR       30$
194 002712 016105 000000G      13$: MOV   LCLUNT(R1),R5 ;Is this line used as a CL unit?
195 002716 002016          BGE     21$          ;Br if yes
196 002720 020127 000000G      CMP    R1,#LSTPL        ;Is this a real or CL line?
197 002724 101007          BHI     23$          ;Br if CL line
198 002726 032761 000000G 000000G BIT     #$KINIT,LSW(R1) ;Is job logged on?
199 002734 001415          BEQ     14$          ;Br if not
200 002736 004767 000000G      CALL  PRTUNM        ;Print user's name
201 002742 000412          BR       14$
202 002744                    23$: .PRINT  #CLFREE        ;This is a free CL line
203 002752 000406          BR       14$
204 002754                    21$: .PRINT  #CLUNIT        ;Print "CL line "
205 002762 006205          ASR    R5          ;Convert to unit number
206 002764 004767 000000G      CALL  PRTDEC        ;Print decimal value
207 002770                    14$:
208                                ;
209                                ; Terminate print line and loop if more lines to print
210                                ;
211 002770                    .PRINT  #CRLF        ;Terminate print line
212 002776 062701 000002      30$: ADD   #2,R1          ;Advance line index number
213 003002 020127 000000G      CMP    R1,#LSTHL        ;Are there more lines?
214 003006 101007          BHI     20$          ;Br if not
215 003010 020127 000002G      CMP    R1,#LSTPL+2      ;Have we stepped past last real line?
216 003014 001002          BNE     22$          ;Br if not
217 003016 012701 000000G      MOV   #FSTIOL,R1       ;Skip up to 1st CL line
218 003022 000167 176660      22$: JMP   1$          ;Go display info for next line
219                                ;
220                                ; Finished
221                                ;
222 003026 012605      20$: MOV   (SP)+,R5
223 003030 012604          MOV   (SP)+,R4
224 003032 012603          MOV   (SP)+,R3
225 003034 012602          MOV   (SP)+,R2
226 003036 012601          MOV   (SP)+,R1
227 003040 000207          RETURN

```

CL

```

1          .SBTTL          CL
2          ;-----
3          ; Display information about CL units.
4          ;
5 003042 010146          SHOCL:  MOV      R1,-(SP)
6 003044 010346          MOV      R3,-(SP)
7 003046 010446          MOV      R4,-(SP)
8 003050 010546          MOV      R5,-(SP)
9          ;
10         ; See if there are any CL units
11         ;
12 003052 005727 000000G          TST      #CLTOTL          ;Are there any CL units?
13 003056 001005          BNE      11$          ;Br if yes
14 003060          .PRINT  #TM$CLO          ;There are no CL units
15 003066 000167 000466          JMP      9$
16         ;
17         ; There are some CL units.
18         ; Print version number.
19         ;
20 003072          11$:      .PRINT  #TM$CL7          ;"CL emulating XL version "
21 003100 116705 000000G          MOVVB   CLVERS,R5          ;Set number to be printed
22 003104 004767 000000G          CALL   PRTDEC          ;Print the decimal value
23 003110          .PRINT  #TM$CL8          ;" "
24         ;
25         ; Print heading lines.
26         ;
27 003116          1$:      .PRINT  #TM$CL1          ;Heading line 1
28 003124          .PRINT  #TM$CL2          ;Heading line 2
29         ;
30         ; Begin loop to print info about each unit.
31         ;
32 003132 005001          CLR      R1          ;Init CL unit index number
33         ;
34         ; Print CL unit name
35         ;
36 003134 010105          2$:      MOV      R1,R5          ;Get CL unit index
37 003136 006205          ASR      R5          ;Convert to unit number
38 003140 020527 000007          CMP      R5,#7.          ;Unit # in range 0-7 get name CL
39 003144 101406          BLOS    19$
40 003146          .PRINT  #TM$C13          ;Print " C1"
41 003154 162705 000010          SUB      #8.,R5          ;Remove C1 unit number bias
42 003160 000403          BR      20$
43 003162          19$:      .PRINT  #TM$CL3          ;Print " CL"
44 003170 004767 000000G          20$:      CALL   PRTDEC          ;Print unit #
45 003174 012703 000002          MOV      #2,R3          ;Print 2 spaces
46 003200 004767 000000G          CALL   PRTSPC
47         ;
48         ; Print the number of the associated line
49         ;
50 003204 016105 000000G          MOV      CL#LIX(R1),R5  ;Get associated line number
51 003210 001412          BEQ     3$          ;Br if there is no associated line
52 003212 006205          ASR      R5          ;Get line #
53 003214 012703 000003          MOV      #3.,R3          ;Get print field width
54 003220 004767 000000G          CALL   PRTFIX          ;Print the line number
55 003224 012703 000003          MOV      #3.,R3          ;Print 3 spaces
56 003230 004767 000000G          CALL   PRTSPC
57 003234 000407          BR      4$

```


CL

```

58 003236          3$:      .PRINT  #TM$CL4          ;Print "none"
59 003244  012703  000002      MOV      #2,R3          ;Print 2 spaces
60 003250  004767  000000G     CALL     PRTSPC
61
62                ; Print number of any job that is using this CL unit
63
64 003254  010100          4$:      MOV      R1,R0          ;Get CL index number
65 003256  004767  000000G     CALL     CKCLUS        ;See if any job is using CL unit
66 003262  010005          MOV      R0,R5          ;Get # of job that is using the device
67 003264  001411          BEQ      18$           ;Br if device is free
68 003266  012703  000003      MOV      #3,R3          ;Print in 3 character field
69 003272  004767  000000G     CALL     PRTFIX        ;Print the job number
70 003276  012703  000002      MOV      #2,R3          ;Print 2 spaces
71 003302  004767  000000G     CALL     PRTSPC
72 003306  000407          BR       17$
73 003310          18$:      .PRINT  #TM$CL4          ;Print "none"
74 003316  012703  000001      MOV      #1,R3          ;Print 1 space
75 003322  004767  000000G     CALL     PRTSPC
76
77                ; See if this CL unit is spooled
78
79 003326  016705  000000G     17$:     MOV      CLDEVX,R5     ;Get CL device index number
80 003332  010100          MOV      R1,R0          ;Get CL unit # index
81 003334  006200          ASR      R0             ;Convert to unit #
82 003336  020027  000007      CMP      R0,#7         ;CL or C1 type unit?
83 003342  101404          BLOS    21$           ;Br if CL
84 003344  162700  000010      SUB      #8,R0         ;Remove C1 unit bias
85 003350  016705  000000G     MOV      C1DEVX,R5     ;Get C1 device index number
86 003354  000300          21$:     SWAB     R0             ;Put unit number in high order byte
87 003356  050005          BIS      R0,R5         ;Combine with device index #
88 003360  012700  000000G     MOV      #SDCB,R0      ;Point to first SDCB
89 003364  020027  000000G     5$:      CMP      R0,#SDCBND    ;Have we checked all spooled devices?
90 003370  103011          BHIS    7$            ;Br if yes -- This CL unit is not spooled
91 003372  020560  000000G     CMP      R5,SDDVU(R0)  ;Is this SDCB for this CL unit?
92 003376  001403          BEQ     6$            ;Br if yes -- CL unit is spooled
93 003400  062700  000000G     ADD     #SDCBSZ,R0     ;Point to next SDCB
94 003404  000767          BR      5$
95 003406          6$:      .PRINT  #TM$CL5          ;Print "(spooled)"
96
97                ; Print which options are selected for this line
98
99 003414  005761  000000G     7$:      TST     CL$LIX(R1)     ;Is this CL unit assigned to a line?
100 003420  001445          BEQ     10$           ;Br if not -- Don't display options
101 003422  016105  000000G     MOV     CL$OPT(R1),R5   ;Get option flags for this CL unit
102 003426  012703  003572'     MOV     #CLOPTB,R3     ;Point to option name table
103 003432  005004          CLR     R4            ;Say no option names printed yet
104 003434  032305          15$:    BIT     (R3)+,R5     ;Is this option selected?
105 003436  001420          BEQ     16$           ;Br if not
106 003440  005704          TST     R4            ;Is this the 1st option?
107 003442  001004          BNE    12$           ;Br if not
108 003444          .PRINT #TM$CL6        ;Print " ["
109 003452  000404          BR     13$
110 003454          12$:    . TTYOUT #54         ;Print comma
111 003464  010304          13$:    MOV     R3,R4         ;Point to option name string
112 003466  112400          14$:    MOVB   (R4)+,R0     ;Get next char from option name string
113 003470  001403          BEQ     16$           ;Br if hit end of name string
114 003472          . TTYOUT          ;Print next char of option name

```

CL

```

115 003476 000773          BR      14$          ;Loop to print rest of option name
116 003500 105723      16$:  TSTB   (R3)+        ;Search for end of option name
117 003502 001376          BNE   16$          ;Loop till end found
118 003504 005203          INC   R3           ;Point to next word
119 003506 042703 000001  BIC   #1,R3
120 003512 020327 003730'  CMP   R3,#CLOPND  ;Checked all options?
121 003516 103746          BLD   15$          ;Loop if not
122 003520 005704          TST   R4           ;Were any options printed?
123 003522 001404          BEQ   10$          ;Br if not
124 003524          .TTYOUT #'J          ;Close option list
125          ;
126          ; Terminate the print line
127          ;
128 003534      10$:  .PRINT #CRLF          ;End of print line
129          ;
130          ; See if there are more CL units
131          ;
132 003542 062701 000002  ADD   #2,R1        ;Advance CL unit index
133 003546 020127 000000C  CMP   R1,#2*CLTOTL ;Have we done all CL units?
134 003552 103002          BHIS  9$           ;Br if yes
135 003554 000167 177354  JMP   2$           ;Go show info for next unit
136          ;
137          ; Finished
138          ;
139 003560 012605      9$:  MOV   (SP)+,R5
140 003562 012604          MOV   (SP)+,R4
141 003564 012603          MOV   (SP)+,R3
142 003566 012601          MOV   (SP)+,R1
143 003570 000207          RETURN
144          ;
145          ; Table of CL option flags and option names
146          ;
147          .MACRO  CLOP   FLAG,NAME
148          .WORD  FLAG
149          .ASCIZ  /'NAME'/
150          .EVEN
151          .ENDM   CLOP
152          ;
153 003572      CLOPTB:
154 003572          CLOP   CO$FF,FORM
155 003602          CLOP   CO$TAB,TAB
156 003610          CLOP   CO$LC,LC
157 003616          CLOP   CO$BBT,EIGHTBIT
158 003632          CLOP   CO$LFO,LFOUT
159 003642          CLOP   CO$LFI,LFIN
160 003652          CLOP   CO$FFO,FORMO
161 003662          CLOP   CO$BND,BINDOUT
162 003674          CLOP   CO$BNI,BININ
163 003704          CLOP   CO$CR,CR
164 003712          CLOP   CO$CTL,CTRL
165 003722          CLOP   CO$DTR,DTR
166 003730      CLOPND:

```

JOBS

```

1          . SBTTL          JOBS
2          ;-----
3          ; Show information about all jobs.
4          ;
5          ; Define /PPN option for SHOW JOBS command
6          ;
7 003730          TBLDEF    SJD,1
8 000336          CMDDEF    P*PN,0
9 000342          TBLEND
10         ;
11 003730 010146  SHOJOB:  MOV    R1,-(SP)
12 003732 010246          MOV    R2,-(SP)
13 003734 010346          MOV    R3,-(SP)
14 003736 010446          MOV    R4,-(SP)
15 003740 010546          MOV    R5,-(SP)
16         ;
17         ; See if we should display PPN instead of User names
18         ;
19 003742 105713          TSTB    (R3)          ; ANY COMMAND OPTIONS?
20 003744 001420          BEQ     40$          ; BRANCH IF NOT
21 003746 121327 000057  CMPB    (R3),#'/          ; VALID SWITCH LEADIN?
22 003752 001015          BNE     40$          ; IGNORE IF NOT
23 003754 005203          INC     R3          ; SKIP /
24 003756 012704 000334'  MOV     #SJOHD,R4          ; POINT TO SHOW JOBS OPTION TABLE
25 003762 004767 000000G  CALL    SEARCH          ; VALID OPTION?
26 003766 103004          BCC    41$          ; REMEMBER /PPN IF SO
27 003770          FABORT   #INVOPT
28 004000 112767 000001 000000G 41$:  MOVVB  #1,SJSPPN          ; /PPN ONLY VALID OPTION AT THIS TIME
29         ;
30         ; Print system usage statistics
31         ;
32 004006 004767 007122 40$:  CALL    PRTUSE          ; PRINT SYSTEM USAGE STATISTICS
33         ;
34         ; Print heading lines
35         ;
36 004012          .PRINT  #SYHD1          ; PRINT FIRST LINE OF JOB HEADING
37 004020          .PRINT  #SYHD2          ; AND UNDERLINE IT
38         ;
39         ; Begin loop to display information about each active line
40         ;
41 004026 012701 000002          MOV     #2,R1          ; GET 1ST LINE INDEX #
42         ;
43         ; See if line is active
44         ;
45 004032 032761 000000G 000000G 1$:  BIT     ##KINIT,LSW(R1) ; HAS LINE BEEN INITIALIZED?
46 004040 001002          BNE     20$          ; BR IF YES
47 004042 000167 000624          JMP     30$          ; TRY NEXT LINE
48         ;
49         ; Print line number
50         ;
51 004046 010105 20$:  MOV     R1,R5          ; GET LINE INDEX #
52 004050 004767 000000G          CALL    PRTLN          ; PRINT THE LINE #
53         ;
54         ; Print '*' if this is our line
55         ;
56 004054 112700 000040          MOVVB  #' ,R0          ; ASSUME THAT IT'S NOT OUR LINE
57 004060 120167 000000G          CMPB    R1,CORUSR          ; IS THIS OUR LINE?

```

JOBS

```

58 004064 001002          BNE      2$          ;BR IF NOT
59 004066 112700 000052   MOVVB   #'*,R0       ;PRINT * IF YES
60 004072          2$:   .TTYOUT          ;
61 004076          .PRINT  #SPACE2      ;PRINT 2 SPACES
62          ;
63          ; Print information about owner jobs
64          ;
65 004104 020127 000000G   CMP     R1,#LSTPL    ;IS THIS A PRIMARY LINE?
66 004110 101407          BLOS   3$           ;BR IF PRIMARY LINE
67 004112 020127 000000G   CMP     R1,#LSTD   ;DETACHED JOB?
68 004116 101004          BHI    3$           ;BR IF NOT
69 004120          .PRINT  #DETTXT     ;PRINT "Detached"
70 004126 000443          BR     18$         ;
71 004130 116105 000000G   3$:   MOVVB  LNPRIM(R1),R5 ;GET OWNER LINE NUMBER
72 004134 004767 000000G   CALL   PRTLN       ;PRINT OWNER LINE NUMBER
73 004140          .TTYOUT #'('       ;ENCLOSE VIRTUAL LINE # IN ( )
74 004150 020127 000000G   CMP     R1,#LSTPL    ;IS THIS A PRIMARY LINE?
75 004154 101412          BLOS   19$         ;BR IF YES
76 004156 016504 000000G   MOV     LSECT(R5),R4 ;POINT TO VIRTUAL LINE # TABLE FOR OWNER JOB
77 004162 012700 000000G   MOV     #MAXSEC,R0  ;GET HIGHEST VIRTUAL LINE #
78 004166 012705 000001   MOV     #1,R5       ;START WITH VIRTUAL LINE #1
79 004172 120124          21$:  CMPB   R1,(R4)+     ;SEARCH FOR LINE WITHIN TABLE
80 004174 001403          BEQ    22$         ;BR IF FOUND RIGHT ONE
81 004176 005205          INC    R5           ;INC VIRTUAL LINE #
82 004200 077004          SOB   R0,21$      ;LOOP TILL FOUND
83 004202 005005          19$:  CLR    R5           ;SAY THIS IS LINE 0
84 004204 004767 000000G   22$:  CALL   PRTDEC     ;PRINT VIRTUAL LINE #
85 004210          .TTYOUT #' )      ;CLOSE VIRTUAL LINE #
86 004220 020527 000011   CMP     R5,#9       ;1 OR 2 DIGIT VIRTUAL LINE #?
87 004224 101004          BHI    18$         ;BR IF 2 DIGIT
88 004226          .TTYOUT #40      ;PRINT A SPACE
89 004236          18$:  .PRINT  #SPACE2     ;PRINT 2 SPACES
90          ;
91          ; Print job priority
92          ;
93 004244 116105 000000G   MOVVB  LPRI(R1),R5  ;GET JOB'S PRIORITY VALUE
94 004250 012703 000003   MOV     #3,R3       ;PRINT IN 3 SPACES
95 004254 004767 000000G   CALL   PRTFIX      ;
96 004260          .PRINT  #SPACE2     ;PRINT 2 SPACES
97          ;
98          ; Print current line state
99          ;
100 004266 005003          CLR    R3           ;
101 004270 016102 000000G   MOV     LSTATE(R1),R2 ;GET USER'S CURRENT EXECUTION STATE
102 004274 020263 004730'  4$:   CMP     R2,STBIT(R3) ;LOOK UP THE STATE
103 004300 001416          BEQ    6$           ;BR WHEN FOUND
104 004302 062703 000002   ADD     #2,R3       ;TRY NEXT FLAG
105 004306 020327 000050   CMP     R3,#NUMST   ;DONE ALL?
106 004312 103770          BLO   4$           ;BR IF NOT
107 004314 012700 000000G   MOV     #RNMS,R0    ;ASSUME 'RN' STATE
108 004320 020227 000000G   CMP     R2,#S##RUN  ;IS JOB EXECUTABLE?
109 004324 101402          BLOS   25$         ;USE 'RN' IF SO
110 004326 012700 000000G   MOV     #WTMS,R0    ;GENERIC WAIT IF STATE NOT RN OR IN TABLE
111 004332          25$:  .PRINT          ;PRINT 'RN' OR 'WT' AS STATE
112 004334 000406          BR     5$         ;
113 004336 016367 005000' 000504 6$:   MOV     STNAM(R3),STPBUF ;MOVE STATE NAME TO PRINT BUFFER
114 004344          .PRINT  #STPBUF   ;PRINT STATE NAME

```

JOB5

```

115 ;
116 ; Print "SWP" if job is outswapped
117 ;
118 004352 032761 000000G 000000G 5$: BIT ##INCOR,LSW(R1) ; IS JOB IN MEMORY NOW?
119 004360 001004 BNE 12$ ; BR IF YES
120 004362 .PRINT #SWPTX ; PRINT "SWP"
121 004370 000413 BR 13$
122 ;
123 ; Print "Lock" if job is locked in memory
124 ;
125 004372 032761 000000G 000000G 12$: BIT ##MLOCK,LSW6(R1); IS JOB LOCKED IN MEMORY?
126 004400 001404 BEQ 14$ ; BR IF NOT
127 004402 .PRINT #LOCKTX ; PRINT "LOCK"
128 004410 000403 BR 13$
129 004412 14$: .PRINT #SPACE5 ; PRINT SPACES
130 ;
131 ; Print current memory size
132 ;
133 004420 13$: .TTYOUT #40 ; PRINT A SPACE
134 004430 .TTYOUT #40 ; PRINT ANOTHER SPACE
135 004440 016105 000000G MOV LNBLKS(R1),R5 ; GET # 256-WORD BLOCKS ASSIGNED TO JOB
136 004444 066105 000000G ADD LNSBLK(R1),R5 ; ADD # BLOCKS USED FOR PLAS REGIONS
137 004450 005205 INC R5 ; ROUND UP
138 004452 006205 ASR R5 ; CVT TO # K BYTES
139 004454 004767 000000G CALL PRTDC3 ; PRINT # K-BYTES OF MEMORY USED
140 004460 .PRINT #KBMSG ; PRINT "Kb"
141 004466 .PRINT #SPACE2 ; PUT IN 2 SPACES
142 ;
143 ; Print job connect time
144 ;
145 004474 016705 000000G MOV MINTIM,R5 ; GET CURRENT MINUTE TIMER VALUE
146 004500 166105 000000G SUB LCONTM(R1),R5 ; CALCULATE CONNECT TIME FOR LINE
147 004504 005205 INC R5 ; CHARGE A MINIMUM OF 1 MINUTE
148 004506 005004 CLR R4 ; CLEAR HIGH-ORDER FOR DIVIDE
149 004510 012703 000074 MOV #60.,R3 ; SET TO DIVIDE BY 60.
150 004514 004767 000000G CALL DIVIDE ; DIVIDE BY 60 TO GET HOURS AND MINUTES
151 004520 010046 MOV R0,-(SP) ; SAVE NUMBER OF MINUTES
152 004522 020527 000144 CMP R5,#100. ; DO WE NEED MORE THAN 2 DIGITS FOR HOURS?
153 004526 103405 BLD 23$ ; BR IF NOT
154 004530 012703 000003 MOV #3,R3 ; PRINT 3 DIGITS
155 004534 004767 000000G CALL PRTFIX ; PRINT # HOURS
156 004540 000406 BR 24$
157 004542 23$: .TTYOUT #40 ; PRINT A SPACE
158 004552 004767 000000G CALL PRTDC2 ; PRINT # HOURS WITH 2 DIGITS
159 004556 012605 24$: MOV (SP)+,R5 ; GET # MINUTES CONNECTED
160 004560 .TTYOUT #'; ; PRINT COLON AFTER HOURS
161 004570 004767 000000G CALL PRTDC2 ; PRINT # MINUTES CONNECTED
162 004574 .PRINT #COL00 ; PRINT ':00' SECONDS
163 ;
164 ; Print cpu time
165 ;
166 004602 .PRINT #SPACE2 ; PRINT 2 SPACES
167 004610 016104 000000G MOV LCPUHI(R1),R4 ; GET HIGH ORDER CPU TIME (CLOCK TICKS)
168 004614 016105 000000G MOV LCPULO(R1),R5 ; GET LOW ORDER CPU TIME (CLOCK TICKS)
169 004620 004767 000000G CALL PRTTMV ; PRINT TIME VALUE
170 ;
171 ; Print name of running program

```

JOBS

```

172
173 004624          .PRINT #SPACE2
174 004632 016100 000000G  MOV   LPRG1(R1),R0 ;GET 1ST 3 CHARS OF NAME (RAD50)
175 004636 004767 000000G  CALL  PRTR50       ;PRINT THEM
176 004642 016100 000000G  MOV   LPRG2(R1),R0 ;PRINT 2ND 3 CHARS
177 004646 004767 000000G  CALL  PRTR50
178
179 ; Print user name or PPN
180
181 004652          .PRINT #SPACE3 ;PRINT 3 SPACES
182 004660 004767 000000G  CALL  PRTUNM      ;PRINT USER NAME
183 004664          .PRINT #CRLF
184
185 ; Do next line
186
187 004672 062701 000002   30$:  ADD   #2,R1 ;GET NEXT LINE INDEX #
188 004676 020127 000000G  CMP   R1,#LSTSL  ;CHECKED ALL?
189 004702 101002          BHI   99$        ;BR IF YES
190 004704 000167 177122   JMP   1$         ;DO NEXT LINE
191
192 ; Finished
193
194 004710 105067 000000G  99$:  CLRB  SJSPPN ;TURN OFF /PPN SWITCH
195 004714 012605          MOV   (SP)+,R5
196 004716 012604          MOV   (SP)+,R4
197 004720 012603          MOV   (SP)+,R3
198 004722 012602          MOV   (SP)+,R2
199 004724 012601          MOV   (SP)+,R1
200 004726 000207          RETURN
201
202
203 -----
204 ; JOB STATE CODES
205
206 004730 000000G  STBIT: .WORD  S$RT ;REAL-TIME
207 004732 000000G  .WORD  S$TTSC ;TT INPUT DONE AND SINGLE CHAR ACTIVATION
208 004734 000000G  .WORD  S$TTFN ;TT INPUT DONE
209 004736 000000G  .WORD  S$OTFN ;TT OUTPUT BUFFER EMPTY
210 004740 000000G  .WORD  S$HICP ;INTERACTIVE JOB COMPUTATION
211 004742 000000G  .WORD  S$TWFN ;TIMED WAIT FINISH
212 004744 000000G  .WORD  S$OTLO ;OUTPUT BUFFER ALMOST EMPTY
213 004746 000000G  .WORD  S$I OFN ;I/O WAIT COMPLETION
214 004750 000000G  .WORD  S$LOW  ;LOW PRIORITY COMPUTATION
215 004752 000000G  .WORD  S$INWT ;INPUT WAIT
216 004754 000000G  .WORD  S$OTWT ;OUTPUT WAIT
217 004756 000000G  .WORD  S$TMWT ;TIMED WAIT
218 004760 000000G  .WORD  S$SPND ;JOB HAS DONE A . SPND
219 004762 000000G  .WORD  S$MSWT ;MESSAGE WAIT
220 004764 000000G  .WORD  S$QUSR ;WAITING FOR USR
221 004766 000000G  .WORD  S$IOWT ;WAITING FOR I/O TO FINISH
222 004770 000000G  .WORD  S$SFWT ;WAITING FOR SHARED FILE RECORD
223 004772 000000G  .WORD  S$QMIO ;WAITING FOR MAPPED I/O BUFFER
224 004774 000000G  .WORD  S$SPDB ;WAITING FOR SPOOLED FILE SPACE
225 004776 000000G  .WORD  S$SPCB ;WAITING FOR SPOOLED FILE
226          000050  NUMST = <.-STBIT>
227
228 ; PARALLEL VECTOR OF STATE NAMES

```

JOBS

| | | | | | |
|-----|--------|-----|-----|-------------|------------------------|
| 229 | | | | | |
| 230 | 005000 | 122 | 124 | STNAM: | . ASCII /RT/ ; S\$RT |
| 231 | 005002 | 111 | 116 | | . ASCII /IN/ ; S\$TTSC |
| 232 | 005004 | 111 | 116 | | . ASCII /IN/ ; S\$TTFN |
| 233 | 005006 | 111 | 116 | | . ASCII /IN/ ; S\$OTFN |
| 234 | 005010 | 111 | 116 | | . ASCII /IN/ ; S\$HICP |
| 235 | 005012 | 110 | 111 | | . ASCII /HI/ ; S\$TWFN |
| 236 | 005014 | 110 | 111 | | . ASCII /HI/ ; S\$OTLO |
| 237 | 005016 | 110 | 111 | | . ASCII /HI/ ; S\$IOFN |
| 238 | 005020 | 114 | 117 | | . ASCII /LO/ ; S\$LOW |
| 239 | 005022 | 124 | 111 | | . ASCII /TI/ ; S\$INWT |
| 240 | 005024 | 124 | 117 | | . ASCII /TO/ ; S\$OTWT |
| 241 | 005026 | 123 | 114 | | . ASCII /SL/ ; S\$TMWT |
| 242 | 005030 | 123 | 114 | | . ASCII /SL/ ; S\$SPND |
| 243 | 005032 | 115 | 123 | | . ASCII /MS/ ; S\$MSWT |
| 244 | 005034 | 125 | 123 | | . ASCII /US/ ; S\$QUSR |
| 245 | 005036 | 111 | 117 | | . ASCII /IO/ ; S\$IOWT |
| 246 | 005040 | 123 | 106 | | . ASCII /SF/ ; S\$SFWT |
| 247 | 005042 | 115 | 111 | | . ASCII /MI/ ; S\$QMIO |
| 248 | 005044 | 123 | 120 | | . ASCII /SP/ ; S\$SPDB |
| 249 | 005046 | 123 | 120 | | . ASCII /SP/ ; S\$SPCB |
| 250 | | | | | |
| 251 | 005050 | 130 | 130 | 200 STPBUF: | . ASCII /XX/<200> |
| 252 | | | | | . EVEN |

QUEUE

| 1 | . SBTTL | QUEUE |
|----|---------|----------------|
| 2 | ----- | |
| 3 | | |
| 4 | | SHOW QUEUE |
| 5 | | |
| 6 | 005054 | 010146 |
| 7 | 005056 | 012701 000000G |
| 8 | 005062 | 020127 000000G |
| 9 | 005066 | 103024 |
| 10 | 005070 | 005761 000000G |
| 11 | 005074 | 001003 |
| 12 | 005076 | 062701 000000G |
| 13 | 005102 | 000767 |
| 14 | 005104 | |
| 15 | 005112 | |
| 16 | 005120 | 004767 000000G |
| 17 | 005124 | 062701 000000G |
| 18 | 005130 | 020127 000000G |
| 19 | 005134 | 103771 |
| 20 | 005136 | 000403 |
| 21 | 005140 | |
| 22 | 005146 | 012601 |
| 23 | 005150 | 000207 |

| | | | |
|---------|---------|-------------|---|
| SHOQUE: | MOV | R1, -(SP) | |
| | MOV | #SDCB, R1 | ; POINT TO FIRST SDCB |
| 4#: | CMP | R1, #SDCBND | ; CHECKED ALL SDCB'S? |
| | BHIS | 3# | ; BR IF YES -- THERE ARE NO QUEUED FILES |
| | TST | SDFHD(R1) | ; ANY FILES QUEUED FOR THIS DEVICE? |
| | BNE | 1# | ; BR IF YES |
| | ADD | #SDCBSZ, R1 | ; POINT TO NEXT SDCB |
| | BR | 4# | ; GO CHECK IT |
| 1#: | . PRINT | #QHDMS1 | ; PRINT QUEUE HEAD MESSAGE |
| | . PRINT | #QHDMS2 | ; Underline the heading |
| 5#: | CALL | LSTSPL | ; LIST INFO ABOUT FILES QUEUED FOR THIS DEV |
| | ADD | #SDCBSZ, R1 | ; POINT TO NEXT SDCB |
| | CMP | R1, #SDCBND | ; DONE ALL SDCB'S? |
| | BLO | 5# | ; BR IF MORE TO DO |
| | BR | 2# | ; FINISHED |
| 3#: | . PRINT | #NOFIL | ; THERE ARE NO PENDING SPOOL FILES |
| 2#: | MOV | (SP)+, R1 | |
| | RETURN | | |

COMMANDS

```

1          .SBTTL      .      COMMANDS
2          ;-----
3          ; SHOW COMMANDS Calls TSXUCL to display specific or all command definitions.
4          ;
5          ; Inputs:
6          ;     R3      Points to next keyword in command buffer or end of command
7          ;     R4      Points to next word in SHOW command table
8          ;
9          ; Outputs:
10         ;     R3      Points to end of input command in CMDBUF
11         ;     R4      Points to end of string to be passed to TSXUCL in CMDBUF
12         ;
13 005152  SHOCMD:
14         ;
15         ; See if there are any user defined commands
16         ;
17 005152 005767 000000G      TST      UCLBLK      ;Any user defined commands?
18 005156 001410              BEQ      1$          ;Br if not
19         ;
20         ; There may be user defined commands.
21         ; Pass any specific command to be displayed to TSXUCL.
22         ; Enter TSXUCL to show the commands.
23         ;
24 005160 012704 000000G      MOV      #CMDBUF,R4      ;Point to beginning of command buffer
25         ; Probably holds "SHOW ..." at this point
26 005164 112724 000077      MOVB     #'?(R4)+      ;Tell TSXUCL we want to show commands
27 005170 112324 3$:        MOVB     (R3)+,(R4)+      ;Pass next character from command buffer
28 005172 001376              BNE      3$          ;Until remainder of command line passed
29 005174 000167 000000G      JMP      CALUCL      ;Call TSXUCL
30         ;
31         ; There are no user defined commands
32         ;
33 005200 1$:        .PRINT  #NOUDC      ;No user defined commands
34 005206 000207      RETURN

```

DEVICES

```

1          .SBTTL      DEVICES
2          ;-----
3          ; SHOW DEVICES
4          ;
5 005210   010246     SHODEV: MOV      R2,-(SP)
6          ;
7          ; Print heading lines
8          ;
9 005212          .PRINT #DVSHH1
10 005220         .PRINT #DVSHH2
11 005226         .PRINT #DVSHH3
12          ;
13          ; List status of each device that is installed in the system
14          ;
15 005234   005002          CLR      R2          ;Start with the first device
16 005236   004767   000016 1$:      CALL    DEVDSP       ;Display status about this device
17 005242   062702   000002          ADD     #2,R2        ;Get next device index
18 005246   020267   000000G        CMP     R2,NUMDEV     ;Finished all devices?
19 005252   101771          BLOS    1$          ;Loop if not
20          ;
21          ; Finished
22          ;
23 005254   012602          MOV     (SP)+,R2
24 005256   000207          RETURN

```

DEVICES

```

1          ; -----
2          ;   Display a line of status information for a device.
3          ;
4          ;   Inputs:
5          ;   R2 = Device index number.
6          ;
7 005260   010346
8 005262   010546
9          ;
10         ;   Print the device name
11         ;
12 005264   012703   000002           MOV     #2,R3           ;Print 2 spaces
13 005270   004767   000000G        CALL    PRTSPC
14 005274   016200   000000G        MOV     PNAME(R2),R0    ;Get RAD50 device name
15 005300   004767   000000G        CALL    PRTR50         ;Print it
16         ;
17         ;   Print the current I/O count
18         ;
19 005304   012703   000003           MOV     #3,R3           ;Print spaces
20 005310   004767   000000G        CALL    PRTSPC
21 005314   016205   000000G        MOV     HANIOC(R2),R5   ;Get # pending I/O operations for dev
22 005320   012703   000005           MOV     #5,R3           ;Print in 5 column field
23 005324   004767   000000G        CALL    PRTFIX        ;Print it
24         ;
25         ;   Print the device status word
26         ;
27 005330   012703   000003           MOV     #3,R3           ;Print spaces
28 005334   004767   000000G        CALL    PRTSPC
29 005340   016205   000000G        MOV     DVSTAT(R2),R5  ;Get status word value
30 005344   012703   000006           MOV     #6,R3           ;Print 6 digits
31 005350   004767   000000G        CALL    OCTFIX        ;Print octal value
32         ;
33         ;   Print the handler virtual base address
34         ;
35 005354   012703   000003           MOV     #3,R3           ;Print spaces
36 005360   004767   000000G        CALL    PRTSPC
37 005364   016205   000000G        MOV     HANENT(R2),R5  ;Get base address of handler
38 005370   020527   000100           CMP     R5,#100         ;Is this a real or pseudo handler?
39 005374   103440           BLO    9#              ;Branch of pseudo handler
40 005376   162705   000006           SUB     #6,R5           ;Get actual handler base address
41 005402   012703   000006           MOV     #6,R3           ;Print 6 digits
42 005406   004767   000000G        CALL    OCTFIX        ;Print octal value
43         ;
44         ;   Print the handler physical base address
45         ;
46 005412   012703   000003           MOV     #3,R3           ;Print spaces
47 005416   004767   000000G        CALL    PRTSPC
48 005422   016205   000000G        MOV     HANPAR(R2),R5  ;Get physical base address of handler
49 005426   012703   000006           MOV     #6,R3           ;Print 6 digits
50 005432   004767   000000G        CALL    OCTFIX        ;Print octal value
51         ;
52         ;   Print the handler size
53         ;
54 005436   012703   000002           MOV     #2,R3           ;Print 2 spaces
55 005442   004767   000000G        CALL    PRTSPC
56 005446   016205   000000G        MOV     HANSIZ(R2),R5  ;Get the handler size
57 005452   012703   000005           MOV     #5,R3           ;Print 5 digits

```

DEVICES

```

58 005456 004767 0000000 CALL PRTFIX ;Print decimal value
59 005462 012703 000002 MOV #2,R3 ;Print spaces
60 005466 004767 0000000 CALL PRTSPC
61 ;
62 ; Print CSR and Vector addresses for this device
63 ;
64 005472 004767 000014 CALL DSPCSR
65 ;
66 ; Terminate this print line
67 ;
68 005476 9$: .PRINT #CRLF ;End the print line
69 ;
70 ; Finished
71 ;
72 005504 012605 MOV (SP)+,R5
73 005506 012603 MOV (SP)+,R3
74 005510 000207 RETURN

```

```

1
2 ; -----
3 ; Extract CSR and Vector addresses from a device handler and display them.
4 ;
5 ; Inputs:
6 ; R2 = Device index number.
7 005512 010146 DSPCSR: MOV R1, -(SP)
8 005514 010246 MOV R2, -(SP)
9 005516 010346 MOV R3, -(SP)
10 005520 010446 MOV R4, -(SP)
11 005522 010546 MOV R5, -(SP)
12 ;
13 ; Build file spec for device handler
14 ;
15 005524 016767 000000G 172246 MOV SYNAME, HANNAM ;Set physical device name for SY device
16 005532 016267 000000G 172242 MOV PNAME(R2), HANNAM+2 ;Set RAD50 device name
17 ;
18 ; Try to open the device handler file
19 ;
20 005540 .SERR ;Don't abort on lookup errors
21 005546 .LOOKUP #XAREA, #1, #HANNAM ;Try to open handler file
22 005566 103547 BCS 9$ ;Br if unable to open handler
23 ;
24 ; Read block 0 of handler and save information about CSR address
25 ;
26 005570 .READW #XAREA, #1, #BLKO, #256., #0 ;Read block 0 of handler
27 005626 103527 BCS 9$
28 005630 016767 000001C 172154 MOV BLKO+H. CSR, DCSR ;Save CSR info
29 ;
30 ; Read blocks 1 and 2 of handler
31 ;
32 005636 .READW #XAREA, #1, #BLKO, #512., #1 ;Read blocks 1 and 2 of handler
33 005676 016767 000001C 172104 MOV BLKO-1000+H. VEC, DVEC ;Save vector info
34 ;
35 ; If both the CSR and vector are zero, don't display anything
36 ;
37 005704 016700 172102 MOV DCSR, R0 ;Get CSR value
38 005710 056700 172074 BIS DVEC, R0 ;Add Vector value
39 005714 001474 BEQ 9$ ;Br if both are zero
40 ;
41 ; Determine if this is a handler for a PRO device with floating
42 ; vector and CSR addresses.
43 ;
44 005716 005001 CLR R1 ;Assume this is not a PRO device
45 005720 012704 000000C MOV #BLKO-1000+H. VEC, R4 ;Point to handler vector cell
46 005724 012400 MOV (R4)+, R0 ;Get address of vector
47 005726 002013 BGE 3$ ;Br if not pointer to vector list
48 005730 006300 ASL R0 ;Get byte offset to vector list
49 005732 060004 ADD R0, R4 ;Point to start of vector list
50 005734 005714 TST (R4) ;Is this a PRO device with floating addresses?
51 005736 002007 BGE 3$ ;Br if not
52 005740 005724 TST (R4)+ ;Point to word with PRO device ID
53 005742 012401 MOV (R4)+, R1 ;Get PRO device ID
54 005744 010100 MOV R1, R0 ;Get PRO device ID
55 005746 004767 000226 CALL PIDCSR ;Get CSR address for PRO device
56 005752 010005 MOV R0, R5 ;Get to R5 for OCTFIX
57 005754 000402 BR 4$ ;Go display it

```

DEVICES

```

58
59 ; Display the CSR value
60 ;
61 005756 016705 172030 3$: MOV DCSR,R5 ;Get CSR value
62 005762 012703 000002 4$: MOV #2,R3 ;Print 2 spaces
63 005766 004767 000000G CALL PRTSPC
64 005772 012703 000006 MOV #6,R3 ;Print 6 digits
65 005776 004767 000000G CALL OCTFIX ;Print octal value
66
67 ; Display vector addresses
68 ;
69 006002 012703 000002 MOV #2,R3 ;Print 2 spaces
70 006006 004767 000000G CALL PRTSPC
71 006012 016705 171772 MOV DVEC,R5 ;Single vector or vector list?
72 006016 001433 BEQ 9$ ;Br if no vector
73 006020 002405 BLT 1$ ;Br if multiple vector
74 006022 012703 000003 MOV #3,R3 ;Print 3 digits
75 006026 004767 000000G CALL OCTFIX ;Print octal value
76 006032 000425 BR 9$
77
78 ; We have multiple vectors (and possibly floating PRO vectors)
79 ;
80 006034 005701 1$: TST R1 ;Do we have floating vectors for PRO device?
81 006036 001404 BEQ 5$ ;Br if not
82 006040 010100 MOV R1,R0 ;Get PRO device ID
83 006042 004767 000154 CALL PIDVEC ;Get base vector based on device ID
84 006046 010001 MOV R0,R1 ;Save base vector location for device
85 006050 011405 5$: MOV (R4),R5 ;Get address of next vector
86 006052 060105 ADD R1,R5 ;Add base vector address
87 006054 012703 000003 MOV #3,R3 ;Print 3 digits
88 006060 004767 000000G CALL OCTFIX ;Print octal value
89 006064 062704 000006 ADD #6,R4 ;Point to next vector entry
90 006070 005714 TST (R4) ;Are there more vectors?
91 006072 003405 BLE 9$ ;Br if not
92 006074 012703 000001 MOV #1,R3 ;Print 1 space
93 006100 004767 000000G CALL PRTSPC
94 006104 000761 BR 5$ ;Print next vector
95
96 ; Finished
97 ;
98 006106 9$: .CLOSE #1 ;Close handler file
99 006114 .HERR ;Reset error aborts
100 006122 012605 MOV (SP)+,R5
101 006124 012604 MOV (SP)+,R4
102 006126 012603 MOV (SP)+,R3
103 006130 012602 MOV (SP)+,R2
104 006132 012601 MOV (SP)+,R1
105 006134 000207 RETURN

```

DEVICES

```

1
2 ; -----
3 ; Convert a PRO device ID number into the option slot # for the device.
4 ;
5 ; Inputs:
6 ;   RO = PRO device ID.
7 ;
8 ; Outputs:
9 ;   C-flag set ==> Could not find device ID.
10 ;   RO = Option slot number.
11 ;
12 ;-----
13 ;
14 ;
15 ;
16 ;
17 ;
18 ;
19 ;
20 ;
21 ;
22 ;
23 ;
24 ;
25 ;
26 ;-----
27 ; Determine the address of the CSR for a PRO device given the device ID.
28 ;
29 ; Inputs:
30 ;   RO = PRO device ID
31 ;
32 ; Outputs:
33 ;   C-flag set ==> Do not recognize the device.
34 ;   RO = CSR address
35 ;
36 ;-----
37 ;
38 ;
39 ;
40 ;
41 ;
42 ;
43 ;-----
44 ; Determine the base vector for a PRO device given the device ID.
45 ;
46 ; Inputs:
47 ;   RO = PRO device ID
48 ;
49 ; Outputs:
50 ;   C-flag set ==> Did not recognize device.
51 ;   RO = Base vector location for device.
52 ;
53 ;-----
54 ;
55 ;
56 ;
57 ;

```

| | | | |
|----|--------|--------|---------|
| 11 | 006136 | 010146 | |
| 12 | 006140 | 012701 | 000000G |
| 13 | 006144 | 021127 | 177777 |
| 14 | 006150 | 001410 | |
| 15 | 006152 | 020021 | |
| 16 | 006154 | 001373 | |
| 17 | 006156 | 162701 | 000002G |
| 18 | 006162 | 006201 | |
| 19 | 006164 | 010100 | |
| 20 | 006166 | 000241 | |
| 21 | 006170 | 000401 | |
| 22 | 006172 | 000261 | |
| 23 | 006174 | 012601 | |
| 24 | 006176 | 000207 | |
| 36 | 006200 | 004767 | 177732 |
| 37 | 006204 | 103405 | |
| 38 | 006206 | 072027 | 000007 |
| 39 | 006212 | 062700 | 174000 |
| 40 | 006216 | 000241 | |
| 41 | 006220 | 000207 | |
| 53 | 006222 | 004767 | 177710 |
| 54 | 006226 | 103405 | |
| 55 | 006230 | 072027 | 000003 |
| 56 | 006234 | 062700 | 000300 |
| 57 | 006240 | 000241 | |

```

PIDSLT: MOV      R1, -(SP)
        MOV      #PROSLT, R1      ; Point to entry for slot 0
1$:     CMP      (R1), #-1        ; Checked all entries?
        BEQ      2$              ; Br if yes
        CMP      RO, (R1)+        ; Search for correct entry
        BNE      1$              ; Loop if this is not it
        SUB      #PROSLT+2, R1    ; Get byte index for slot
        ASR      R1              ; Get word index
        MOV      R1, RO          ; Return in RO
        CLC                       ; Signal success on return
        BR      9$
2$:     SEC                       ; Signal failure on return
9$:     MOV      (SP)+, R1
        RETURN

PIDCSR: CALL     PIDSLT          ; Determine slot where device is installed
        BCS     9$              ; Br if don't recognize device
        ASH     #7., RO         ; CSR addresses are 200 apart per slot
        ADD     #174000, RO     ; CSR for slot 0 is here
        CLC                       ; Signal success on return
9$:     RETURN

PIDVEC: CALL     PIDSLT          ; Determine what slot has controller
        BCS     9$              ; Br if don't recognize device
        ASH     #3., RO         ; Vectors are 8 bytes apart per slot
        ADD     #300, RO        ; Vector for slot 0 is at 300
        CLC                       ; Signal success on return

```

58 006242 000207

9#: RETURN

ASSIGNS

```

1          . SBTTL          ASSIGNS
2          ; -----
3          ;   Display assignments
4          ;
5 006244 010246          SHOASN: MOV      R2, -(SP)
6 006246          .PRINT  #ASNHD1          ;PRINT HEADING
7          ;
8          ;   If there is no assignment for SY, print default assignment
9          ;
10 006254 012702 000000G          MOV      #ASNTBL,R2          ;Point to assign table
11 006260 026762 000000G 000000G 5$:  CMP      R5OSY,AT$LOG(R2);Is this entry for SY?
12 006266 001423          BEQ      6$          ;Br if yes
13 006270 062702 000000G          ADD      #AT#$SZ,R2          ;Point to next entry
14 006274 020227 000000G          CMP      R2,#ASNEND          ;Checked all assign entries?
15 006300 103767          BLD      5$          ;Loop if not
16 006302          .PRINT  #SYASHD          ;Print "SY --> "
17 006310 016700 000000G          MOV      SYNAME,R0          ;Get default device
18 006314 004767 000000G          CALL    PRTR50
19 006320          .TTYOUT #':
20 006330          .PRINT  #CRLF
21          ;
22          ;   If there is no assignment for DK, print default assignment
23          ;
24 006336 012702 000000G          6$:  MOV      #ASNTBL,R2          ;Point to assign table
25 006342 026762 000000G 000000G 7$:  CMP      R5ODK,AT$LOG(R2);Is this entry for DK?
26 006350 001423          BEQ      8$          ;Br if yes
27 006352 062702 000000G          ADD      #AT#$SZ,R2          ;Point to next entry
28 006356 020227 000000G          CMP      R2,#ASNEND          ;Checked all assign entries?
29 006362 103767          BLD      7$          ;Loop if not
30 006364          .PRINT  #DKASHD          ;Print "DK --> "
31 006372 016700 000000G          MOV      SYNAME,R0          ;Get default device
32 006376 004767 000000G          CALL    PRTR50          ;Print it
33 006402          .TTYOUT #':
34 006412          .PRINT  #CRLF
35          ;
36          ;   Now print user assign table entries
37          ;
38 006420 012702 000000G          8$:  MOV      #ASNTBL,R2          ;POINT TO ASSIGN TABLE
39 006424 005762 000000G          2$:  TST      AT$LOG(R2)          ;IS THIS ASSIGN TABLE ENTRY IN USE?
40 006430 001461          BEQ      3$          ;BR IF NOT
41 006432 016200 000000G          MOV      AT$LOG(R2),R0          ;GET LOGICAL DEVICE NAME
42 006436 004767 000000G          CALL    PRTR50          ;DISPLAY LOGICAL DEVICE NAME
43 006442          .PRINT  #ASNHD2          ;PRINT ARROW
44 006450 016200 000000G          MOV      AT$DEV(R2),R0          ;GET PHYSICAL DEVICE NAME
45 006454 004767 000000G          CALL    PRTR50          ;PRINT IT
46 006460          .TTYOUT #':          ;PUT IN COLON
47 006470 016200 000000G          MOV      AT$FIL(R2),R0          ;GET 1ST 3 CHARS OF FILE NAME
48 006474 001434          BEQ      4$          ;BR IF NO NAME
49 006476 004767 000000G          CALL    PRTR50          ;PRINT 1ST PART OF NAME
50 006502 016200 000002G          MOV      AT$FIL+2(R2),R0          ;GET 2ND 3 CHARS OF FILE NAME
51 006506 004767 000000G          CALL    PRTR50          ;PRINT THE NAME
52 006512          .TTYOUT #':          ;PUT IN A PERIOD
53 006522 016200 000000G          MOV      AT$EXT(R2),R0          ;GET EXTENSION
54 006526 001417          BEQ      4$          ;BR IF NONE SPECIFIED
55 006530 004767 000000G          CALL    PRTR50          ;DISPLAY THE EXTENSION
56 006534 016205 000000G          MOV      AT$SIZ(R2),R5          ;GET FILE SIZE
57 006540 001412          BEQ      4$          ;BR IF NONE SPECIFIED

```

ASSIGNS

| | | | | | | |
|----|--------|--------|---------|----------|---------------|---------------------------------|
| 58 | 006542 | | | .TTYOUT | #'[| ; START SIZE SPEC |
| 59 | 006552 | 004767 | 000000G | CALL | PRTDEC | ; DISPLAY THE FILE SIZE |
| 60 | 006556 | | | .TTYOUT | #'] | ; TERMINATE THE SIZE SPEC |
| 61 | 006566 | | | .PRINT | #CRLF | ; TERMINATE THE LINE |
| 62 | 006574 | 062702 | 000000G | 4\$: ADD | #AT\$\$SZ, R2 | ; POINT TO NEXT ASSIGN ENTRY |
| 63 | 006600 | 020227 | 000000G | 3\$: CMP | R2, #ASNEND | ; HAVE WE REACHED END OF TABLE? |
| 64 | 006604 | 103707 | | BLO | 2\$ | ; BR IF NOT |
| 65 | 006606 | 012602 | | MOV | (SP)+, R2 | |
| 66 | 006610 | 000207 | | RETURN | | |

ALLOCATIONS

```

1          .SBTTL      ALLOCATIONS
2          ;-----
3          ; Show allocations
4          ;
5 006612   010146     SHOALC: MOV      R1,-(SP)
6 006614   010246     MOV      R2,-(SP)
7 006616   010346     MOV      R3,-(SP)
8 006620   010546     MOV      R5,-(SP)
9          ;
10         ; Make a fast scan to see if there are any allocated devices
11         ;
12 006622   012702   000000G     MOV      #ALCTBL,R2      ;Point to start of allocation table
13 006626   105762   000000G     1$: TSTB   AD$JOB(R2)     ;Is this entry used?
14 006632   001011           BNE     2$              ;Br if yes
15 006634   062702   000000G     ADD     #AD$$SZ,R2     ;Point to next entry
16 006640   020227   000000G     CMP     R2,#ALCEND     ;Checked all entries?
17 006644   103770           BLO     1$              ;Loop if not
18         ;
19         ; There are no allocated devices
20         ;
21 006646           .PRINT  #TM$NAD      ;No allocated devices
22 006654   000460     BR      9$              ;Finished
23         ;
24         ; There are some allocated devices.
25         ; Print heading lines.
26         ;
27 006656     2$: .PRINT  #CRLF        ;Print a blank line
28 006664           .PRINT  #ALCHD1     ;Print heading line 1
29 006672           .PRINT  #ALCHD2     ;Print heading line 2
30         ;
31         ; Begin loop to print information about each allocated device
32         ;
33 006700   012702   000000G     MOV     #ALCTBL,R2     ;Point to start of allocation table
34 006704   116205   000000G     3$: MOVB  AD$JOB(R2),R5 ;Is this entry used?
35 006710   001435           BEQ     4$              ;Br if not
36         ;
37         ; Print the device name
38         ;
39 006712           .TTYOUT #40        ;Put a space in front of the device name
40 006722   016200   000000G     MOV     AD$DVU(R2),R0  ;Get device and unit number
41 006726   004767   000000G     CALL   CVDVNM         ;Convert dev and unit # to device name
42 006732   004767   000000G     CALL   PRTR50         ;Print the device name
43 006736   012703   0000004     MOV     #4.,R3        ;Print 4 spaces
44 006742   004767   000000G     CALL   PRTSPC
45         ;
46         ; Print the number of the job to which the device is allocated
47         ;
48 006746   012703   0000002     MOV     #2.,R3        ;Print job number in 2 col field
49 006752   010501           MOV     R5,R1         ;Save job index number in R1
50 006754   006205           ASR     R5            ;Convert job index number to job number
51 006756   004767   000000G     CALL   PRTFIX        ;Print the job number
52 006762   012703   0000003     MOV     #3.,R3        ;Space over 3 columns
53 006766   004767   000000G     CALL   PRTSPC
54         ;
55         ; Print the user's name
56         ;
57 006772   004767   000000G     CALL   PRTUNM        ;Print the user's name

```

ALLOCATIONS

```

58 ;
59 ; Terminate this print line
60 ;
61 006776 .PRINT #CRLF ;End of line
62 ;
63 ; See if there are more allocated devices to display
64 ;
65 007004 062702 000000G 4$: ADD #AD$$SZ,R2 ;Point to next allocation entry
66 007010 020227 000000G CMP R2,#ALCEND ;Finished all entries?
67 007014 103733 BLD 3$ ;Br if more to do
68 ;
69 ; Finished
70 ;
71 007016 012605 9$: MOV (SP)+,R5
72 007020 012603 MOV (SP)+,R3
73 007022 012602 MOV (SP)+,R2
74 007024 012601 MOV (SP)+,R1
75 007026 000207 RETURN

```

MOUNTS

```

1
2
3
4
5 007030 010146
6 007032 010246
7 007034 010346
8 007036 010446
9 007040 010546
10
11
12
13 007042 016704 000000G
14 007046 012705 000000G
15 007052 010400
16 007054 004767 000000G
17 007060 005767 000001C
18 007064 001404
19 007066 020527 001774G
20 007072 103001
21 007074 010425
22 007076 062704 000000G
23 007102 020467 000000G
24 007106 103761
25 007110 005015
26
27
28
29
30 007112 012705 000000G
31 007116 012503
32 007120 001437
33 007122 010504
34 007124 010300
35 007126 004767 000000G
36 007132 016702 000001C
37 007136 000302
38 007140 016767 000001C 170650
39 007146 012401
40 007150 001762
41 007152 010100
42 007154 004767 000000G
43 007160 016700 000001C
44 007164 000300
45 007166 020002
46 007170 101366
47 007172 103404
48 007174 026767 000001C 170614
49 007202 103361
50 007204 010364 177776
51 007210 010165 177776
52 007214 010103
53 007216 000742
54
55
56
57 007220 005767 000000G

```

. SBTTL MOUNTS

```

; Show mounts
;
SHOMNT: MOV        R1, -(SP)
         MOV        R2, -(SP)
         MOV        R3, -(SP)
         MOV        R4, -(SP)
         MOV        R5, -(SP)
;
; Set up a table of pointers to active mount entries in BLKO area.
;
         MOV        CSHDEV, R4            ; Point to start of mount table
         MOV        #BLKO, R5            ; Point to area where we will build table
20$:     MOV        R4, R0                ; Get address of next mount entry
         CALL       CDGET                ; Get mount entry into CDBUF
         TST        CDBUF+CD$DVU        ; Is this mount entry in use?
         BEQ        21$                 ; Br if not
         CMP        R5, #BLKO+1020.     ; Is table full?
         BHS        21$                 ; Br if yes
21$:     MOV        R4, (R5)+            ; Save pointer to active mount entry
         ADD        #CD$SZ, R4          ; Point to next entry in mount table
         CMP        R4, CSHDEV          ; Have we checked all mount entries?
         BLO        20$                 ; Loop if not
         CLR        (R5)                ; Put null pointer at end of table
;
; Now sort the list of mount table pointers using two keys:
; key 1 -- Device # & unit #; key 2 -- base block number of logical disks
;
24$:     MOV        #BLKO, R5            ; Point to first entry in table
         MOV        (R5)+, R3            ; Get pointer to a mount entry
         BEQ        26$                 ; Br if finished sorting table
23$:     MOV        R5, R4                ; Get pointer to next entry in table
         MOV        R3, R0               ; Get address of mount entry
         CALL       CDGET                ; Copy entry into CDBUF
         MOV        CDBUF+CD$DVU, R2    ; Get device # and unit # of mounted device
         SWAB       R2                  ; Put dev # in high-order, unit # in low-order
25$:     MOV        CDBUF+CD$BAS, CDBASE ; Save base block number
         MOV        (R4)+, R1            ; Get pointer to another mount entry
         BEQ        24$                 ; Br if checked all beyond reference entry
         MOV        R1, R0               ; Get address of mount entry
         CALL       CDGET                ; Copy entry into CDBUF
         MOV        CDBUF+CD$DVU, R0    ; Get device # and unit # of this entry
         SWAB       R0                  ; Rearrange dev # and unit # order
         CMP        R0, R2               ; Look for smallest dev # and unit #
         BHI        25$                 ; Br if this device is higher
         BLO        22$                 ; Br if this device is smaller than ref one
22$:     CMP        CDBUF+CD$BAS, CDBASE ; Same device, now check base block #'s
         BHS        25$                 ; Br if not smaller
         MOV        R3, -2(R4)          ; Swap table entries
         MOV        R1, -2(R5)
         MOV        R1, R3               ; Set new reference pointer
         BR         23$                 ; Continue sort
;
; If there are any mounted devices, print table heading
;
26$:     TST        BLKO                 ; Are there any mounted devices?

```

Mounts

```

58 007224 001406          BEQ      30$          ;Br if not
59 007226          .PRINT  #SHMTH1      ;Print heading line
60 007234          .PRINT  #SHMTH2      ;Underline it
61
62          ; Now print information about each device.
63
64 007242 012702 000000G 30$:   MOV      #BLKO,R2    ;Get pointer to table of sorted pointers
65 007246 012201          1$:   MOV      (R2)+,R1    ;Get next mount table pointer
66 007250 001002          BNE     27$          ;Br if got another entry
67 007252 000167 000414          JMP     2$           ;Finished all devices
68 007256 010100          27$:  MOV      R1,R0        ;Get address of mount entry
69 007260 004767 000000G          CALL   CDGET         ;Move entry into CDBUF
70 007264 016700 000001C          MOV    CDBUF+CD$DVU,R0 ;GET UNIT # / DEVICE # OF MOUNTED DEVICE
71 007270 001576          BEQ    8$           ;BR IF NULL -- THIS ENTRY IS FREE
72
73          ; Found a mounted device.
74          ; Print device name and unit number
75
76 007272 004767 000000G          CALL   CVDVNM       ; CONVERT DEVICE & UNIT #'S TO DEVICE NAME
77 007276 004767 000000G          CALL   PRTR50      ; PRINT THE DEVICE NAME
78 007302          .TTYOUT #';        ; PRINT ":"
79
80          ; If this is a logical disk, print the LD file name
81
82 007312 005767 000001C          TST    CDBUF+CD$BAS  ; IS THIS A LOGICAL DISK?
83 007316 001005          BNE    29$          ; BR IF YES
84 007320          .PRINT  #SPACE6    ; SPACE OVER 6 COLUMNS
85 007326 000167 000234          JMP    4$           ;
86 007332 010146          29$:  MOV      R1,-(SP)    ; SAVE ORIGINAL MOUNT TABLE INDEX
87 007334 005046          CLR    -(SP)        ; PUT NULL ON STACK TO SIGNAL END
88 007336 010146          12$:  MOV      R1,-(SP)    ; SAVE POINTER TO INNER-MOST LOGICAL DISK
89 007340 016703 000001C          MOV    CDBUF+CD$BAS,R3 ; GET BASE BLOCK NUMBER OF THIS LD
90 007344 005005          CLR    R5           ; SAY NO ENCLOSING LD FOUND YET
91 007346 016767 000001C 170444          MOV    CDBUF+CD$DVU,CDDVU ; Save device and unit info
92 007354 016704 000000G          MOV    CSHDEV,R4    ; SEARCH THROUGH MOUNTED DEVICE TABLE
93 007360 010400          13$:  MOV      R4,R0        ; Get address of mount entry
94 007362 004767 000000G          CALL   CDGET         ; Move into CDBUF
95 007366 026767 170426 000001C          CMP    CDDVU,CDBUF+CD$DVU; LOOKING FOR ONE WITH SAME PHYS DEVICE
96 007374 001023          BNE    14$          ; BR IF NOT THIS ONE
97 007376 005767 000001C          TST    CDBUF+CD$BAS  ; IS THIS ALSO A LOGICAL DISK?
98 007402 001420          BEQ    14$          ; BR IF NOT
99 007404 020367 000001C          CMP    R3,CDBUF+CD$BAS ; SEE IF THIS LD ENCLOSES INNER LD
100 007410 101415          BLOS   14$          ; BR IF NOT
101 007412 020367 000001C          CMP    R3,CDBUF+CD$TOP ; CHECK UPPER RANGE
102 007416 103012          BHIS   14$          ; BR IF NOT ENCLOSING
103 007420 005705          TST    R5           ; ANY OTHER ENCLOSING LD'S FOUND SO FAR?
104 007422 001404          BEQ    15$          ; BR IF NOT
105 007424 026767 000001C 170370          CMP    CDBUF+CD$BAS,CDBAS5; SAVE ONE WITH HIGHEST BASE
106 007432 101404          BLOS   14$          ;
107 007434 010405          15$:  MOV      R4,R5        ; THIS IS AN ENCLOSING LOGICAL DISK
108 007436 016767 000001C 170356          MOV    CDBUF+CD$BAS,CDBAS5
109 007444 062704 000000G          14$:  ADD     #CD$$SZ,R4    ; POINT TO NEXT MOUNT TABLE ENTRY
110 007450 020467 000000G          CMP    R4,CSHDVN    ; CHECKED ALL?
111 007454 103741          BLO    13$          ; LOOP IF NOT
112 007456 010501          MOV    R5,R1        ; DID WE FIND AN ENCLOSING LOGICAL DISK?
113 007460 001404          BEQ    16$          ; BR IF NOT
114 007462 010500          MOV    R5,R0        ; WE FOUND AN OUTER LD

```

MOUNTS

```

115 007464 004767 000000G          CALL    CDGET          ;GET OUTER LD DATA INTO CDBUF
116 007470 000722                   BR      12$           ;KEEP SCANNING OUTWARD
117 007472 012605                   16$:    MOV    (SP)+,R5 ;GET POINTER TO OUTER-MOST LOGICAL DISK
118 007474 012703 000000G          MOV    #PRTBUF,R3    ;GET POINTER TO TEXT BUFFER FOR PRTFNM
119 007500 010500                   17$:    MOV    R5,R0      ;Get address of mount entry
120 007502 004767 000000G          CALL    CDGET          ;Get mount entry into CDBUF
121 007506 012700 000000C          MOV    #CDBUF+CD$NAM,R0;GET POINTER TO FILE NAME
122 007512 004767 000000G          CALL    PRTFNM        ;FORMAT THE FILE NAME
123 007516 112723 000072          MOV    #'',(R3)+    ;PUT COLON AFTER FILE NAME
124 007522 012605                   MOV    (SP)+,R5     ;IS THERE ANOTHER NAME?
125 007524 001365                   BNE    17$          ;BR IF YES
126 007526 005303                   DEC    R3            ;POINT BACK TO LAST COLON
127 007530 020327 000006G          18$:    CMP    R3,#PRTBUF+6. ;TAB UP TO COLUMN 6
128 007534 103003                   BHS    19$          ;
129 007536 112723 000040          MOV    #40,(R3)+    ;FILL WITH TRAILING SPACES
130 007542 000772                   BR      18$          ;
131 007544 112713 000200          19$:    MOV    #200,(R3)    ;PUT IN END OF STRING MARKER
132 007550                   .PRINT #PRTBUF      ;PRINT THE LOGICAL DISK SPECIFICATION
133 007556 012601                   MOV    (SP)+,R1     ;GET BACK MOUNT TABLE INDEX
134 007560 010100                   MOV    R1,R0        ;Get address of mount entry
135 007562 004767 000000G          CALL    CDGET          ;Move entry into CDBUF
136                                     ;
137                                     ; Print numbers of jobs that have this device mounted
138                                     ;
139 007566 012705 000001          4$:    MOV    #1,R5      ;START WITH JOB # 1
140 007572 012703 000001          MOV    #1,R3        ;GET MOUNT FLAG FOR JOB # 1
141 007576 012704 000000C          MOV    #CDBUF+CD$JOB,R4;POINT TO TABLE WITH JOB BIT FLAGS
142 007602 130314                   7$:    BITB   R3,(R4)    ;IS DEVICE MOUNTED BY THIS JOB?
143 007604 001415                   BEQ    5$            ;BR IF NOT
144 007606                   .TTYOUT #40         ;PRINT A SPACE
145 007616 020527 000012          CMP    R5,#10.     ;IS THIS A 2 DIGIT JOB NUMBER?
146 007622 103004                   BHS    10$          ;BR IF YES
147 007624                   .TTYOUT #40         ;PRINT EXTRA SPACE FOR COLUMN ALIGNMENT
148 007634 004767 000000G          10$:   CALL    PRTDEC       ;PRINT JOB NUMBER
149 007640 106303                   5$:    ASLB   R3         ;SHIFT OVER MOUNT FLAG
150 007642 103002                   BCC    6$            ;BR IF DID NOT SHIFT OUT OF BYTE
151 007644 005204                   INC    R4            ;POINT TO NEXT BYTE
152 007646 006103                   ROL    R3            ;RESET FLAG BIT
153 007650 005205                   6$:    INC    R5        ;ADVANCE JOB NUMBER
154 007652 020527 000000G          CMP    R5,#NLINES  ;CHECKED ALL JOBS?
155 007656 101751                   BLOS   7$            ;LOOP IF MORE TO CHECK
156                                     ;
157                                     ; Finished with this mount entry
158                                     ;
159 007660                   .PRINT #CRLF        ;END LINE
160 007666 000167 177354          8$:    JMP    1$         ;Go back and print info for next device
161                                     ;
162                                     ; Finished checking all entries in mount table
163                                     ;
164 007672 005767 000000G          2$:    TST    BLKO      ;WERE THERE ANY MOUNTED DEVICES?
165 007676 001003                   BNE    3$            ;BR IF YES
166 007700                   .PRINT #NONEMS     ;PRINT "(NONE)"
167                                     ;
168                                     ; Finished
169                                     ;
170 007706 012605                   3$:    MOV    (SP)+,R5
171 007710 012604                   MOV    (SP)+,R4

```

MGUNTS

| | | | | |
|-----|--------|--------|--------|-----------|
| 172 | 007712 | 012603 | MOV | (SP)+, R3 |
| 173 | 007714 | 012602 | MOV | (SP)+, R2 |
| 174 | 007716 | 012601 | MOV | (SP)+, R1 |
| 175 | 007720 | 000207 | RETURN | |

MODEM

| | | | | | |
|----|--------|--------|------------------------------------|------------|----------------------|
| 1 | | | . SBTTL | MODEM | |
| 2 | | | ;----- | | |
| 3 | | | ; Display modem control parameters | | |
| 4 | | | ; | | |
| 5 | 007722 | | SHOMDM: .PRINT | #PHNTXT | ; Start with PHONE |
| 6 | 007730 | 012700 | MOV | #'0, R0 | ; Assume phone = 0 |
| 7 | 007734 | 105767 | TSTB | VUSPHN | ; Is phone set? |
| 8 | 007740 | 001401 | BEG | 1\$ | ; Br if not |
| 9 | 007742 | 005200 | INC | R0 | ; PHONE=1 |
| 10 | 007744 | | 1\$: .TTYOUT | R0 | |
| 11 | 007750 | | .PRINT | #MDMTXT | ; Print modem header |
| 12 | 007756 | | .PRINT | #ONTTXX | ; ONTIM |
| 13 | 007764 | 016705 | MOV | VONTM, R5 | |
| 14 | 007770 | 004767 | CALL | PRTDEC | |
| 15 | 007774 | | .PRINT | #OFTTXX | ; OFFTIM |
| 16 | 010002 | 016705 | MOV | VOFFTM, R5 | |
| 17 | 010006 | 004767 | CALL | PRTDEC | |
| 18 | 010012 | | .PRINT | #TMITXX | ; TIMIN |
| 19 | 010020 | 016705 | MOV | VTMIN, R5 | |
| 20 | 010024 | 004767 | CALL | PRTDEC | |
| 21 | 010030 | | .PRINT | #TMLTXX | ; TIMLOC |
| 22 | 010036 | 016705 | MOV | VTMLQC, R5 | |
| 23 | 010042 | 004767 | CALL | PRTDEC | |
| 24 | 010046 | | .PRINT | #TMOTXX | ; TIMOUT |
| 25 | 010054 | 016705 | MOV | VTMOUT, R5 | |
| 26 | 010060 | 004767 | CALL | PRTDEC | |
| 27 | 010064 | | .PRINT | #CRLF | |
| 28 | 010072 | 000207 | RETURN | | |

DATE

```

1          . SBTTL          DATE
2          ;-----
3          ; Display current date.
4          ;
5 010074   SHODAT: . DATE          ; GET CURRENT DATE
6 010102   TST      R0            ; IS SYSTEM DATE KNOWN?
7 010104   BEQ      1$           ; BR IF NO DATE ENTERED
8 010106   CALL    PRDAT         ; DISPLAY DATE
9 010112   . PRINT #CRLF
10 010120   BR      9$
11          ; Date is unknown.
12 010122   1$: . PRINT #NODAT    ; NO DATE
13 010130   9$: RETURN
14
15          . SBTTL          TIME
16          ;-----
17          ; Display current time of day.
18          ;
19 010132   SHOTIM: CALL    PRTTOD ; DISPLAY CURRENT TIME
20 010136   . PRINT #CRLF
21 010144   RETURN
22
23          . SBTTL          VERSION
24          ;-----
25          ; Display the TSX-Plus system version number.
26          ;
27 010146   SHOVER: . PRINT #TSXVER ; "TSX-Plus Version="
28 010154   MOV     #TSXVRS,R3    ; Get system version number
29 010160   CLR     R2            ; Clear high-order for divide
30 010162   DIV    #100.,R2      ; Divide R2-R3 by 100
31 010166   MOV    R2,R5         ; Get major version number
32 010170   CALL   PRTDEC        ; Print major version number
33 010174   . TTYOUT #'.         ; Print decimal point
34 010204   MOV    R3,R5         ; Get fractional version number
35 010206   CALL   PRTDC2        ; Print fractional version number
36 010212   . PRINT #CRLF        ; Terminate print line
37 010220   RETURN
38
39          . SBTTL          USE
40          ;-----
41          ; Display computer usage information for current job.
42          ;
43 010222   SHOUSE: CALL    PRTTIM ; PRINT CONNECT AND CPU TIME FOR JOB
44 010226   RETURN

```

INSTALL

```

1          .SBTTL      INSTALL
2          ;-----
3          ; Show information about installed programs
4          ;
5 010230   010546
6 010232   004767   000000G  SHOINS:  MOV      R5, -(SP)
7          ;                               CALL      CKSYPV          ;Require SYSPRV privilege to do this
8          ; Print title lines
9          ;
10 010236          .PRINT  #TM#IN1
11 010244          .PRINT  #TM#IN2
12          ;
13          ; Begin loop to print information about each installed program
14          ;
15 010252   016705   000000G          MOV      INSTBL, R5          ;Point to 1st install table entry
16 010256   010567   000000G  1$:   MOV      R5, INGADR          ;Set address of entry to get
17 010262   012700   000000G          MOV      #INGEMT, R0          ;Get EMT arg block
18 010266   104375          EMT      375          ;Get the entry to IIBUF
19 010270   005767   000001C          TST      IIBUF+II$NAM          ;Is this entry in use?
20 010274   001402          BEQ      2$          ;Br if not
21 010276   004767   000016          CALL     INSPRT          ;Print info about this program
22 010302   062705   000000G  2$:   ADD      #II$#SZ, R5          ;Point to next install table entry
23 010306   020567   000000G          CMP      R5, INSTBN          ;Done all entries?
24 010312   103761          BLO      1$          ;Loop if not
25          ;
26          ; Finished
27          ;
28 010314   012605          MOV      (SP)+, R5
29 010316   000207          RETURN

```

INSTALL

```

1          ; -----
2          ; Print information about the program whose install entry is in IIBUF.
3          ;
4          ; Inputs:
5          ; IIBUF contains install entry for program.
6          ;
7 010320 010146 INSPRT: MOV      R1,-(SP)
8 010322 010246      MOV      R2,-(SP)
9 010324 010346      MOV      R3,-(SP)
10 010326 010446      MOV      R4,-(SP)
11         ;
12         ; Print the file spec for the program.
13         ;
14 010330 012704 000000C      MOV      #IIBUF+II$NAM,R4;Point to file spec to be converted
15 010334 012703 000000G      MOV      #BLKO,R3      ;Point to result area
16 010340 004767 000000G      CALL     EDTFIL      ;Convert file spec to asciz form
17 010344 112723 000040      1$:  MOVVB  #' ,(R3)+      ;Store space following file spec
18 010350 020327 000021G      CMP      R3,#BLKO+17.  ;Filled out to flag area?
19 010354 103773              BLD      1$           ;Loop if not
20 010356 112713 000200      MOVVB  #200,(R3)      ;Store null at end of name
21 010362              .PRINT  #BLKO      ;Print the name
22         ;
23         ; Print names of attribute flags
24         ;
25 010370 012704 177756      MOV      #-18.,R4      ;No attributes printed yet
26 010374 012703 010620'      MOV      #INSANT,R3    ;Point to attribute name table
27 010400 032367 000001C      2$:  BIT      (R3)+,II$FLG+IIBUF ;Is this attribute set?
28 010404 001422              BEQ      4$           ;Br if not
29 010406 005704              TST      R4           ;Is this the first attribute for program?
30 010410 002406              BLT      8$           ;Br if yes
31 010412              .TTYOUT #'/'      ;Print slash
32 010422 005204              INC      R4           ;Count column used by slash
33 010424 000402              BR       3$           ;
34 010426 012704 000022      8$:  MOV      #18.,R4      ;Initialize column number
35 010432 011301      3$:  MOV      (R3),R1      ;Get address of name string
36 010434              .PRINT  R1      ;Print attribute name
37 010440 122127 000200      10$:  CMPB   (R1)+,#200     ;End of keyword?
38 010444 001402              BEQ      4$           ;Br if yes
39 010446 005204              INC      R4           ;Count columns
40 010450 000773              BR       10$          ;
41 010452 005723      4$:  TST      (R3)+      ;Skip name pointer
42 010454 005713              TST      (R3)        ;Any more attributes to check?
43 010456 001350              BNE     2$           ;Loop if yes
44         ;
45         ; See if program has any associated privileges
46         ;
47 010460 012702 000000C      MOV      #II$PRV+IIBUF,R2;Point to privilege flag info
48 010464 012703 000000C      MOV      #II$NPV+IIBUF,R3;Point to negative privilege flags
49 010470 012700 000000G      MOV      #PVNPW,R0     ;Get # privilege words
50 010474 005722      12$:  TST      (R2)+      ;Any privilege flags set?
51 010476 001004              BNE     13$          ;Br if yes
52 010500 005723              TST      (R3)+      ;Any negative privilege flags set?
53 010502 001002              BNE     13$          ;Br if yes
54 010504 077005              SOB     R0,12$     ;Check all privilege words
55 010506 000434              BR       11$        ;No privileges specified for program
56         ;
57         ; Print information about special privileges associated with program

```

INSTALL

```

58
59 010510          13$:      .PRINT  #TM$PVL          ;Print "/PRIV=(
60 010516 005704   TST      R4              ;Any attributes printed?
61 010520 002401   BLT      14$              ;Br if not
62 010522 005404   NEG      R4              ;Make R4 negative to suppress leading comma
63 010524 162704 000007 14$:      SUB      #7.,R4          ;Increase column count
64 010530 012702 000000C 15$:      MOV      #II$PRV+IIBUF,R2;Point to words with privilege flags
65 010534 012703 000001  MOV      #+1,R3          ;Select positive flags this time
66 010540 012700 000022  MOV      #18.,R0         ;Wrap-around to column 18
67 010544 004767 000000G CALL     PRVLST          ;Print selected privileges
68 010550 012702 000000C MOV      #II$NPV+IIBUF,R2;Point to words with deselected priv flags
69 010554 012703 177777  MOV      #-1,R3         ;Select negative flags
70 010560 012700 000022  MOV      #18.,R0         ;Set wrap-around column
71 010564 004767 000000G CALL     PRVLST          ;Print deselected privileges
72 010570          .TTYOUT #')          ;Terminate privilege list
73
74                ; Terminate the print line
75
76 010600          11$:      .PRINT  #CRLF          ;End line
77
78                ; Finished
79
80 010606 012604   MOV      (SP)+,R4
81 010610 012603   MOV      (SP)+,R3
82 010612 012602   MOV      (SP)+,R2
83 010614 012601   MOV      (SP)+,R1
84 010616 000207   RETURN
85
86                ; Table of install program attributes and names
87
88 010620 000000G 010722'  INSANT: .WORD  AF$SCA,1$
89 010624 000000G 010735'  .WORD  AF$NOW,2$
90 010630 000000G 010744'  .WORD  AF$HIE,3$
91 010634 000000G 010751'  .WORD  AF$NOI,4$
92 010640 000000G 010770'  .WORD  AF$IOP,5$
93 010644 000000G 010777'  .WORD  AF$MEM,6$
94 010650 000000G 011007'  .WORD  AF$PLK,7$
95 010654 000000G 011014'  .WORD  AF$DBG,8$
96 010660 000000G 011022'  .WORD  AF$BYA,9$
97 010664 000000G 011031'  .WORD  AF$TPD,10$
98 010670 000000G 011045'  .WORD  AF$DUP,11$
99 010674 000000G 011051'  .WORD  AF$IND,12$
100 010700 000000G 011055'  .WORD  AF$UCL,13$
101 010704 000000G 011064'  .WORD  AF$SET,14$
102 010710 000000G 011072'  .WORD  AF$CCA,15$
103 010714 000000G 011077'  .WORD  AF$NPW,16$
104 010720 000000  .WORD  0
105
106 010722          123      111      116 1$:      .ASCII  /SINGLECHAR/<200>
      010725          107      114      105
      010730          103      110      101
      010733          122      200
107 010735          116      117      127 2$:      .ASCII  /NOWAIT/<200>
      010740          101      111      124
      010743          200
108 010744          110      111      107 3$:      .ASCII  /HIGH/<200>
      010747          110      200

```

INSTALL

| | | | | | | | |
|-----|--------|-----|-----|-----|------|---------|-----------------------|
| 109 | 010751 | 116 | 117 | 116 | 4#: | . ASCII | /NONINTERACTIVE/<200> |
| | 010754 | 111 | 116 | 124 | | | |
| | 010757 | 105 | 122 | 101 | | | |
| | 010762 | 103 | 124 | 111 | | | |
| | 010765 | 126 | 105 | 200 | | | |
| 110 | 010770 | 111 | 117 | 120 | 5#: | . ASCII | /IOPAGE/<200> |
| | 010773 | 101 | 107 | 105 | | | |
| | 010776 | 200 | | | | | |
| 111 | 010777 | 115 | 105 | 115 | 6#: | . ASCII | /MEMLOCK/<200> |
| | 011002 | 114 | 117 | 103 | | | |
| | 011005 | 113 | 200 | | | | |
| 112 | 011007 | 114 | 117 | 103 | 7#: | . ASCII | /LOCK/<200> |
| | 011012 | 113 | 200 | | | | |
| 113 | 011014 | 104 | 105 | 102 | 8#: | . ASCII | /DEBUG/<200> |
| | 011017 | 125 | 107 | 200 | | | |
| 114 | 011022 | 102 | 131 | 120 | 9#: | . ASCII | /BYPASN/<200> |
| | 011025 | 101 | 123 | 116 | | | |
| | 011030 | 200 | | | | | |
| 115 | 011031 | 124 | 122 | 101 | 10#: | . ASCII | /TRANSPARENT/<200> |
| | 011034 | 116 | 123 | 120 | | | |
| | 011037 | 101 | 122 | 105 | | | |
| | 011042 | 116 | 124 | 200 | | | |
| 116 | 011045 | 104 | 125 | 120 | 11#: | . ASCII | /DUP/<200> |
| | 011050 | 200 | | | | | |
| 117 | 011051 | 111 | 116 | 104 | 12#: | . ASCII | /IND/<200> |
| | 011054 | 200 | | | | | |
| 118 | 011055 | 124 | 123 | 130 | 13#: | . ASCII | /TSXUCL/<200> |
| | 011060 | 125 | 103 | 114 | | | |
| | 011063 | 200 | | | | | |
| 119 | 011064 | 123 | 105 | 124 | 14#: | . ASCII | /SETUP/<200> |
| | 011067 | 125 | 120 | 200 | | | |
| 120 | 011072 | 123 | 103 | 103 | 15#: | . ASCII | /SCCA/<200> |
| | 011075 | 101 | 200 | | | | |
| 121 | 011077 | 116 | 117 | 127 | 16#: | . ASCII | /NOWINDOW/<200> |
| | 011102 | 111 | 116 | 104 | | | |
| | 011105 | 117 | 127 | 200 | | | |
| 122 | | | | | | . EVEN | |

REGIONS

```

1
2
3
4
5 011110 010246
6 011112 010446
7 011114 005002
8
9
10
11 011116 012704 000000G
12 011122 004767 000106
13 011126 062704 000000G
14 011132 020427 000000G
15 011136 103771
16
17
18
19 011140 016704 000000G
20 011144 020467 000000G
21 011150 103021
22 011152 010467 000000G
23 011156 012767 000000G 000000G
24 011164 012700 000000G
25 011170 104375
26 011172 012704 000000G
27 011176 004767 000032
28 011202 016704 000000G
29 011206 062704 000000G
30 011212 000754
31
32
33
34 011214 005702
35 011216 001003
36 011220
37
38
39
40 011226 012604
41 011230 012602
42 011232 000207

```

```

          .SBTTL      REGIONS
          -----
          ; Show named regions
          ;
SHOREG:  MOV      R2,-(SP)
          MOV      R4,-(SP)
          CLR      R2          ;Clear count of # regions shown
          ;
          ; First show information about all private regions
          ;
          MOV      #RCBBAS,R4   ;Point to first private Region Control Blk
1$:      CALL     RGNDSR        ;Display info about this region
          ADD      #RC$$SZ,R4   ;Point to next RCB
          CMP      R4,#RCBEND   ;Checked all RCB's?
          BLD     1$           ;Br if not
          ;
          ; Now show information about all shared regions
          ;
          MOV      SHRRCB,R4     ;Point to 1st shared RCB
2$:      CMP      R4,SHRRCN     ;Checked all shared RCB's?
          BHIS    3$           ;Br if yes
          MOV      R4,PEKADR     ;Set address of RCB
          MOV      #RC$$SZ,PEKSIZ ;Get # bytes to fetch
          MOV      #PEKEMT,R0    ;Point to EMT arg block
          EMT     375           ;Move RCB to BLKO buffer
          MOV      #BLKO,R4      ;Point to RCB in our buffer
          CALL     RGNDSR        ;Display info about the region
          MOV      PEKADR,R4     ;Recover real RCB address
          ADD      #RC$$SZ,R4    ;Point to next RCB
          BR      2$
          ;
          ; If there were no named regions, print a message.
          ;
3$:      TST      R2            ;Were there any named regions?
          BNE     9$           ;Br if yes
          .PRINT  #TM$NNR      ;No named regions
          ;
          ; Finished
          ;
9$:      MOV      (SP)+,R4
          MOV      (SP)+,R2
          RETURN

```

REGIONS

```

1          ; -----
2          ; Display a line of information about a named region.
3          ;
4          ; Inputs:
5          ;   R4 = Pointer to Region Control Block.
6          ;   R2 = Count of number of active regions so far.
7          ;
8          ; Outputs:
9          ;   R2 = Incremented if this is an active region.
10         ;
11 011234  RGNDSP:
12         ;
13         ; See if this RCB is for an active, named region
14         ;
15 011234 032764 000000G 000000G          BIT    #RC$USE,RC$FLG(R4) ;Is this an active RCB?
16 011242 001410          BEQ    1$          ;Br if not
17 011244 032764 000000G 000000G          BIT    #RC$LCG,RC$FLG(R4) ;Is this a local copy of a global RCB?
18 011252 001004          BNE    1$          ;Br if yes -- We will list global RCB
19 011254 032764 000000G 000000G          BIT    #RC$GBL,RC$FLG(R4) ;Is this a named region?
20 011262 001001          BNE    2$          ;Br if yes
21 011264 000207          1$:    RETURN
22         ;
23         ; This is an RCB for an active named region.
24         ;
25 011266 010146          2$:    MOV    R1,-(SP)
26 011270 010346          MOV    R3,-(SP)
27 011272 010546          MOV    R5,-(SP)
28         ;
29         ; See if this is the 1st region
30         ;
31 011274 005702          TST    R2          ;Is this the first region?
32 011276 001006          BNE    3$          ;Br if not
33 011300          .PRINT #TM$RD1          ;Display title line 1
34 011306          .PRINT #TM$RD2          ;Display title line 2
35 011314 005202          3$:    INC    R2          ;Count another region displayed
36         ;
37         ; Display name of the region
38         ;
39 011316 016400 000000G          MOV    RC$NAM(R4),R0 ;Get 1st 3 chars of name
40 011322 004767 000000G          CALL   PRTR50          ;Print them
41 011326 016400 000002G          MOV    RC$NAM+2(R4),R0 ;Get 2nd 3 chars of name
42 011332 004767 000000G          CALL   PRTR50          ;Print them
43         ;
44         ; Display size of region
45         ;
46 011336 016401 000000G          MOV    RC$LEN(R4),R1 ;Get # 64-byte blocks allocated for region
47 011342 005000          CLR    R0          ;Clear high-order for divide
48 011344 071027 000020          DIV   #16.,R0          ;Conver to # Kb
49 011350 010005          MOV    R0,R5          ;Get # whole K
50 011352 012703 000006          MOV    #6.,R3          ;Print in 6 digit field
51 011356 004767 000000G          CALL   PRTFIX          ;Print # whole Kb
52 011362 116105 011630'          MOVB  FRAC64(R1),R5 ;Convert to decimal Kb fraction
53 011366          .TTYOUT #'.'          ;Print decimal point
54 011376 004767 000000G          CALL   PRTDEC          ;Print decimal digit
55         ;
56         ; Display type of region
57         ;

```


REGIONS

```

58 011402                .PRINT  #SPACE2          ;Print 2 spaces
59 011410 032764 000000G 000000G  BIT    #RC$PVT,RC$FLG(R4) ;Is this a private or shared region?
60 011416 001407                BEQ    4$                ;Br if shared
61 011420                .PRINT  #TM$LCL          ;Say region is local to job
62 011426                .PRINT  #SPACE1          ;Print extra space
63 011434 000403                BR     5$                ;
64 011436                4$:   .PRINT  #TM$GBL          ;Say region is global
65                        ;
66                        ; Print number of job that created region
67                        ;
68 011444 116405 000000G 5$:   MOV    RC$OWN(R4),R5 ;Get # of job that created region
69 011450 006205                ASR    R5                ;Convert to #
70 011452 012703 000005                MOV    #5.,R3          ;Print 5 digit field
71 011456 004767 000000G  CALL   PRTFIX          ;Print job #
72                        ;
73                        ; Print attachment (use) count
74                        ;
75 011462 116405 000000G  MOV    RC$CNT(R4),R5 ;Get attachment count
76 011466 012703 000005                MOV    #5.,R3          ;Print in 5 digit field
77 011472 004767 000000G  CALL   PRTFIX          ;Print use count
78                        ;
79                        ; Show if shared
80                        ;
81 011476 012703 000004                MOV    #4.,R3          ;Print 4 spaces
82 011502 004767 000000G  CALL   PRTSPC
83 011506 032764 000000G 000000G  BIT    #RC$EXC,RC$FLG(R4) ;Is this region sharable?
84 011514 001404                BEQ    6$                ;Br if yes
85 011516                .PRINT  #NOTXT          ;Print 'No'
86 011524 000403                BR     7$                ;
87 011526                6$:   .PRINT  #YESTXT          ;Print 'Yes'
88                        ;
89                        ; Show if AGE is set
90                        ;
91 011534                7$:   .PRINT  #SPACE3          ;Print 3 spaces
92 011542 032764 000000C 000000G  BIT    #RC$AGE!RC$AEP,RC$FLG(R4) ;Is AGE enabled?
93 011550 001004                BNE    8$                ;Br if yes
94 011552                .PRINT  #NOTXT          ;Print 'No'
95 011560 000403                BR     9$                ;
96 011562                8$:   .PRINT  #YESTXT          ;Print 'Yes'
97                        ;
98                        ; Show region base 64-byte block #
99                        ;
100 011570                9$:   .PRINT  #SPACE2          ;Print 2 spaces
101 011576 010246                MOV    R2,-(SP)
102 011600 016402 000000G  MOV    RC$BAS(R4),R2 ;Get base block #
103 011604 004767 000000G  CALL   OCTPRT          ;Display it
104 011610 012602                MOV    (SP)+,R2
105                        ;
106                        ; Terminate print line
107                        ;
108 011612                .PRINT  #CRLF
109                        ;
110                        ; Finished
111                        ;
112 011620 012605                MOV    (SP)+,R5
113 011622 012603                MOV    (SP)+,R3
114 011624 012601                MOV    (SP)+,R1

```

REGIONS

115 011626 000207

RETURN

116

117

118

; Values to convert # fractional 64-byte blocks to tenths of Kb

119 011630 000 001 001

FRAC64: .BYTE 0.,1.,1.,2.,2.,3.,4.,4.,5.,6.,6.,7.,7.,8.,9.,9.

011633 002 002 003

011636 004 004 005

011641 006 006 007

011644 007 010 011

011647 011

120

.EVEN

PRIVILEGES

```

1          .SBTTL          PRIVILEGES
2          ;-----
3          ; Show privileges
4          ;
5 011650 010246  SHOPRV: MOV      R2,-(SP)
6 011652 010346      MOV      R3,-(SP)
7 011654 010446      MOV      R4,-(SP)
8          ;
9          ; List authorized privileges
10         ;
11 011656          .PRINT  #TM$PVA          ; Authorized privileges
12 011664 012702 000000G  MOV      #PRIVA0,R2          ; Point to priv flag vector
13 011670 012703 000001  MOV      #+1,R3          ; Show positive privileges only
14 011674 012704 177750  MOV      #-24.,R4         ; Start at column 24
15 011700 012700 000030  MOV      #24.,R0         ; Wrap around to col 24
16 011704 004767 000000G  CALL    PRVLST          ; List privileges
17 011710          .PRINT  #CRLF          ; Terminate last line
18 011716          .PRINT  #CRLF          ; Put in a blank line
19         ;
20         ; List current privileges
21         ;
22 011724          .PRINT  #TM$PVC          ; Current privileges
23 011732 012702 000000G  MOV      #PRIVC0,R2         ; Current privilege flags
24 011736 012704 177750  MOV      #-24.,R4         ; Start at column 24
25 011742 012700 000030  MOV      #24.,R0         ; Wrap around to col 24
26 011746 004767 000000G  CALL    PRVLST          ; List privileges
27 011752          .PRINT  #CRLF
28         ;
29         ; Finished
30         ;
31 011760 012604      MOV      (SP)+,R4
32 011762 012603      MOV      (SP)+,R3
33 011764 012602      MOV      (SP)+,R2
34 011766 000207      RETURN

```

SL

```

1          .SBTTL      SL
2          ;-----
3          ; Display single line editor status.
4          ;
5 011770   010246     SHOSLE: MOV      R2,-(SP)
6 011772   010346     MOV      R3,-(SP)
7 011774   010446     MOV      R4,-(SP)
8          ;
9          ; Print message heading
10         ;
11 011776           .PRINT  #TM$SL1      ;"SL status: "
12         ;
13         ; Print ON or OFF
14         ;
15 012004   032761   000000G 000000G     BIT      ##SLON,LSW7(R1) ;Is SL turned on?
16 012012   001003     BNE      1$          ;Br if yes
17 012014   012702   000000G     MOV      #TM$OFF,R2      ;Point to OFF message
18 012020   000402     BR       2$          ;
19 012022   012702   000000G     1$:    MOV      #TM$ON,R2      ;Point to ON message
20 012026           2$:    .PRINT  R2          ;Print text
21         ;
22         ; Check for KED mode
23         ;
24 012032   012704   000000G     MOV      ##$LKED,R4      ;Get Ked flag
25 012036   012703   000000G     MOV      #TM$KED,R3
26 012042   004767   000046     CALL     SLCKFL
27         ;
28         ; Check for TTY mode
29         ;
30 012046   012704   000000G     MOV      ##$LTTY,R4      ;Get flag
31 012052   012703   000000G     MOV      #TM$TTY,R3      ;Point to text string
32 012056   004767   000032     CALL     SLCKFL
33         ;
34         ; Check for SUBSTITUTE mode
35         ;
36 012062   012704   000000G     MOV      ##$LLET,R4      ;Get flag
37 012066   012703   000000G     MOV      #TM$SUB,R3      ;Point to text string
38 012072   004767   000016     CALL     SLCKFL
39         ;
40         ; Terminate the print line
41         ;
42 012076           .PRINT  #CRLF      ;Terminate print line
43         ;
44         ; Finished
45         ;
46 012104   012604     MOV      (SP)+,R4
47 012106   012603     MOV      (SP)+,R3
48 012110   012602     MOV      (SP)+,R2
49 012112   000207     RETURN

```

SL

```

1          ; -----
2          ; Check of a SL option flag is set in LSW7, and if it is print that the
3          ; option is on; otherwise, print that the option is off.
4          ;
5          ; Inputs:
6          ; R1 = Job index number
7          ; R3 = Pointer to text string for option name
8          ; R4 = Flag bit to be tested in LSW7
9          ;
10         012114 SLCKFL:
11         ;
12         ; Print a leading comma
13         ;
14         012114 .TTYOUT #54          ;Print comma
15         ;
16         ; If option flag is not set, print "NO"
17         ;
18         012124 030461 000000G      BIT    R4,LSW7(R1)    ;Is the option flag set?
19         012130 001003              BNE    1$          ;Br if yes
20         012132                      .PRINT  #TM$NO      ;Print NO
21         ;
22         ; Print the name of the option
23         ;
24         012140 1$: .PRINT  R3          ;Print the option name
25         ;
26         ; Finished
27         ;
28         012144 000207              RETURN

```

```

1          .SBTTL      .      RUN-TIMES
2          ;-----
3          ; Show run-times
4          ;
5 012146  010146      SHOSRT: MOV      R1, -(SP)
6 012150  012701  000000G  MOV      #RDB, R1      ; POINT TO FIRST RUN-TIME DESCRIPTOR BLOCK
7 012154  005005      CLR      R5      ; Count run-times in R5
8 012156  020127  000000G  1$:  CMP      R1, #RDBEND    ; ANY RUN-TIMES?
9 012162  103023      BHIS     2$      ; BR IF FINISHED
10 012164  026127  000000G 000000G  CMP      RT$DEV(R1), #DMYDEV ; Is this dummy entry for patching?
11 012172  001414      BEQ      4$      ; Br if yes
12 012174  016100  000000G  MOV      RT$NAM(R1), R0    ; GET 1ST 3 CHARS OF RUN-TIME NAME
13 012200  005205      INC      R5      ; Count another run-time
14 012202  004767  000000G  CALL     PRTR50           ; PRINT THEM
15 012206  016100  000002G  MOV      RT$NAM+2(R1), R0 ; GET 2ND 3 CHARS OF NAME
16 012212  004767  000000G  CALL     PRTR50           ; PRINT THEM
17 012216      .PRINT  #CRLF      ; END LINE
18 012224  062701  000000G  4$:  ADD      #RT$$SZ, R1    ; POINT TO NEXT DESCRIPTOR BLOCK
19 012230  000752      BR      1$
20 012232  005705      2$:  TST      R5      ; WERE THERE ANY RUN-TIMES?
21 012234  001003      BNE     3$      ; BR IF YES
22 012236      .PRINT  #NONEMS    ; PRINT NONE
23 012244  012601      3$:  MOV      (SP)+, R1
24 012246  000207      RETURN
  
```

SPOOL

```

1          .SBTTL          SPOOL
2          ;-----
3          ; SHOW SPOOL
4          ; List the names of the spooled devices.
5          ;
6 012250   010246        SHOSPL: MOV      R2,-(SP)
7          ;
8          ; See if there are any spooled devices
9          ;
10 012252   105767   000000G      TSTB     NSPLDV      ;Are there any spooled devices?
11 012256   001405          BEQ      3$          ;Br if not
12 012260   012702   000000G      MOV      #SDCB,R2      ;Point to first spooled device control block
13 012264   020227   000000G      CMP      R2,#SDCBND    ;Are there any spooled devices?
14 012270   103404          BLO     1$          ;Br if there are spooled devices
15 012272          3$: .PRINT  #TM$NSD      ;There are no spooled devices
16 012300   000425          BR      9$
17          ;
18          ; There are spooled devices, print their names
19          ;
20 012302          1$: .PRINT  #TM$SDN      ;Print heading
21 012310   016200   000000G      2$: MOV      SDNAME(R2),R0 ;Get RAD50 name of spooled device
22 012314   020027   000000G      CMP      R0,#DMYDEV    ;Uninstalled device?
23 012320   001405          BEQ      4$          ;Br if yes
24 012322   004767   000000G      CALL     PRTR50        ;Print the name
25 012326          .PRINT  #SPACE2        ;Print 2 spaces
26 012334   062702   000000G      4$: ADD      #SDCBSZ,R2 ;Point to next SDCB
27 012340   020227   000000G      CMP      R2,#SDCBND    ;Are there more?
28 012344   103761          BLO     2$          ;Br if yes
29 012346          .PRINT  #CRLF         ;Terminate the line
30          ;
31          ; Finished
32          ;
33 012354   012602          9$: MOV      (SP)+,R2
34 012356   000207          RETURN

```

SUBSET

```

1          .SBTTL          SUBSET
2          ;-----
3          ; SHOW SUBSET
4          ; Display information about logical disks.
5          ;
6 012360 010146 SHOSUB: MOV      R1,-(SP)
7 012362 010246      MOV      R2,-(SP)
8 012364 010346      MOV      R3,-(SP)
9 012366 010446      MOV      R4,-(SP)
10 012370 010546     MOV      R5,-(SP)
11         ;
12         ; First do a SET LD CLEAN to update logical disk information
13         ;
14 012372 004767 000000G CALL    LDCLEN          ; DO SET LD CLEAN
15         ;
16         ; Make a fast scan and see if any logical disks are mounted
17         ;
18 012376 012703 000010      MOV      #8,R3          ; GET # LOGICAL DISK ENTRIES
19 012402 012704 000000G     MOV      #LDNAME,R4       ; POINT TO FILE NAME TABLE
20 012406 005714 5$:      TST      (R4)          ; IS THIS DISK ASSIGNED TO A FILE?
21 012410 001007      BNE      6$          ; BR IF YES
22 012412 062704 000010     ADD      #8,R4          ; POINT TO NEXT ENTRY IN TABLE
23 012416 077305      SOB      R3,5$          ; LOOP IF MORE TO CHECK
24 012420      .PRINT  #NOLDMT        ; NO LOGICAL DISKS MOUNTED
25 012426 000472      BR       9$
26         ;
27         ; Now begin to display logical disk information
28         ;
29 012430 016702 000000G 6$:      MOV      R5OLD0,R2       ; GET "LD0" NAME
30 012434 005003      CLR      R3          ; INIT DISK TABLE INDEX
31 012436 012704 000000G     MOV      #LDNAME,R4       ; POINT TO FILE NAME TABLE
32         ;
33         ; See if next logical disk unit is assigned
34         ;
35 012442 005714 2$:      TST      (R4)          ; IS DISK ASSIGNED TO A FILE?
36 012444 001453      BEQ      1$          ; BR IF NOT
37         ;
38         ; Print logical disk name
39         ;
40 012446 010200      MOV      R2,R0          ; GET DISK NAME
41 012450 004767 000000G     CALL    PRTR50          ; PRINT THE DISK NAME
42 012454      .PRINT  #SUBARO        ; " --> "
43         ;
44         ; Print the file name
45         ;
46 012462 010346      MOV      R3,-(SP)        ; SAVE R3
47 012464 012703 000000G     MOV      #BLKO,R3        ; EDIT FILE NAME INTO BLKO
48 012470 004767 000000G     CALL    EDTFIL          ; EDIT FILE NAME
49 012474 112723 000133     MOVVB   #133,(R3)+       ; "[" START OF FILE SIZE
50 012500 112713 000200     MOVVB   #200,(R3)       ; TERMINATE NAME STRING
51 012504      .PRINT  #BLKO        ; PRINT NAME
52 012512 012603      MOV      (SP)+,R3
53 012514 016305 000000G     MOV      LDSize(R3),R5   ; GET FILE SIZE
54 012520 004767 000000G     CALL    PRTDEC          ; PRINT IT
55 012524      .TTYOUT #135        ; "]"
56         ;
57         ; Print "read only" if that is the case

```


SUBSET

```

58
59 012534 032763 000000G 000000G ; BIT #LD#RON,LDFLAG(R3) ; IS IS MOUNTED READ-ONLY
60 012542 001403 ; BEQ 3$ ; BR IF NOT
61 012544 ; .PRINT #RONTXT ; PRINT "(read only)"
62 ;
63 ; Print "not available" if that is the case
64 ;
65 012552 005763 000000G 3$: TST LDPDEV(R3) ; IS FILE CURRENTLY ACTIVE?
66 012556 001003 ; BNE 4$ ; BR IF YES
67 012560 ; .PRINT #NOTAVL ; NOT AVAILABLE
68 012566 ; 4$: .PRINT #CRLF ; PRINT THE LINE
69 ;
70 ; Advance to next logical disk
71 ;
72 012574 005202 1$: INC R2 ; ADVANCE LOGICAL DISK NAME
73 012576 062704 000010 ; ADD #8.,R4 ; ADVANCE NAME POINTER
74 012602 062703 000002 ; ADD #2,R3 ; ADVANCE TABLE POINTER
75 012606 020327 000016 ; CMP R3,#14. ; DONE ALL?
76 012612 101713 ; BLOS 2$ ; BR IF NOT
77 ;
78 ; Finished
79 ;
80 012614 012605 9$: MOV (SP)+,R5
81 012616 012604 ; MOV (SP)+,R4
82 012620 012603 ; MOV (SP)+,R3
83 012622 012602 ; MOV (SP)+,R2
84 012624 012601 ; MOV (SP)+,R1
85 012626 000207 ; RETURN

```

VM

```

1          . SBTTL          VM
2          ; -----
3          ; SHOW CURRENT VM BASE, TOP AND DEVICE SIZE
4          ;
5 012630 012767 105646 000000G SHOWVM: MOV    #^RVMO,FILNAM ;Set device name VM:
6 012636 005067 000002G CLR    FILNAM+2      ;No file name
7 012642          .SERR          ;Trap .LOOKUP errors
8 012650          .LOOKUP #XAREA,#1,#FILNAM ;Get channel to VMO:
9 012670 103004 BCC    1$          ;Branch if we got VM
10 012672          .PRINT #SHVTX4      ;"VM not installed"
11 012700 000476 BR     9$          ;Exit if no VM
12          ;
13          ; Get current base and top
14          ;
15 012702 1$:          .SPFUN #XAREA,#1,#372,#BLKO,#0,#0 ;Request VM base and top
16 012744 103451 BCS    8$          ;Ignore command on error
17 012746          .PRINT #SHVTX1      ;" VM Base="
18 012754 016705 000000G MOV    BLKO,R5      ;Retrieve base
19 012760 012703 000006 MOV    #6,R3       ;Six digit display
20 012764 004767 000000G CALL   OCTFIX      ;Display it
21 012770          .PRINT #SHVTX2      ;" Top="
22 012776 016705 000002G MOV    BLKO+2,R5    ;Retrieve top
23 013002 004767 000000G CALL   OCTFIX      ;Display it
24          ;
25          ; Get current device size in blocks from handler
26          ;
27 013006          .SPFUN #XAREA,#1,#373,#BLKO,#0,#0 ;Request VM device size
28 013050 103407 BCS    8$          ;Skip size if bad
29 013052          .PRINT #SHVTX3      ;" Size="
30 013060 016705 000000G MOV    BLKO,R5      ;Retrieve size
31 013064 004767 000000G CALL   PRTDEC      ;Display decimal size in blocks
32          ;
33 013070 8$:          .CLOSE #1          ;Only close if successful open
34 013076 9$:          .PRINT #CRLF      ;Format display
35 013104          .HERR          ;Give back error trapping
36 013112 000207 RETURN

```

```
1          .SBTTL SYSTAT (& WHO) command
2          ;-----
3          ; THE WHO COMMAND PRINTS OUT A LIST OF ALL LINE NUMBERS
4          ; WHICH ARE LOGGED ON.
5          ;
6 013114 004767 170610 CMDWHO: CALL SHOJOB ; SHOW JOB INFORMATION
7 013120 000167 000000G JMP RDCMD
8
9          .SBTTL USE command
10         ;-----
11         ; Process the USE command.
12         ;
13 013124 004767 000000G CMDUSE: CALL PRTTIM ; PRINT CONNECT AND CPU TIME FOR JOB
14 013130 000167 000000G JMP RDCMD
```

PRTUSE -- Print system usage statistics

```

1          .SBTTL  PRTUSE -- Print system usage statistics
2          ;-----
3          ; PRTUSE is called to print the system usage statistics.
4          ;
5 013134 010146 PRTUSE: MOV     R1,-(SP)
6 013136 010246      MOV     R2,-(SP)
7 013140 010346      MOV     R3,-(SP)
8 013142 010446      MOV     R4,-(SP)
9          ; Print system up-time.
10 013144      .PRINT  #UPTMMS      ; "UPTIME:"
11 013152 016704 000000G  MOV     TMTOTH,R4      ; GET TOTAL UP-TIME (OR TIME SINCE LAST RESET)
12 013156 016705 000000G  MOV     TMTOTL,R5
13 013162 000241      CLC          ; DIVIDE TIME VALUE BY 2 TO GET 1/10 SEC UNITS
14 013164 006004      ROR     R4          ; SHIFT HIGH-ORDER PART
15 013166 006005      ROR     R5          ; AND LOW-ORDER PART
16 013170 004767 000000G  CALL    PRTTMD         ; PRINT TIME VALUE
17 013174      .PRINT  #CRLF      ; TERMINATE LINE
18          ; Print system usage statistics
19 013202 016767 000000G 000000G  MOV     TMTOTH,DIVSOR  ; SET TOTAL UP-TIME AS DIVISOR FOR PERCENTAGE
20 013210 016767 000000G 000002G  MOV     TMTOTL,DIVSOR+2
21 013216 012702 013254'  MOV     #SUMVEC,R2    ; POINT TO DRIVER VECTOR TABLE
22 013222 012200      1$.  MOV     (R2)+,R0    ; GET ADDRESS OF ASCIZ STRING TO PRINT
23 013224 001406      BEQ    2$          ; BR IF END OF LIST HIT
24 013226      .PRINT          ; PRINT TEXT MESSAGE
25 013230 012201      MOV     (R2)+,R1    ; GET ADDRESS OF TIME CELL TO PRINT
26 013232 001773      BEQ    1$          ; BR IF NONE WITH THIS TEXT
27 013234 004767 000000G  CALL    PRTPCT        ; CONVERT TO PERCENTAGE AND PRINT VALUE
28 013240 000770      BR     1$
29          ;
30          ; Finished
31          ;
32 013242 012604      2$:  MOV     (SP)+,R4
33 013244 012603      MOV     (SP)+,R3
34 013246 012602      MOV     (SP)+,R2
35 013250 012601      MOV     (SP)+,R1
36 013252 000207      RETURN
37          ;
38          ; Driver vector for system usage printout.
39          ; First entry of each pair is the address of an asciz string to print.
40          ; Second entry is address of 32-bit time value to be printed as percentage.
41          ;
42 013254 000000G 000000G  SUMVEC: .WORD  SUM1, TMUSRH    ; RUN-TIME
43 013260 000000G 000000G      .WORD  SUM2, TMIDWH    ; I/O WAIT
44 013264 000000G 000000G      .WORD  SUM3, TMSWTH    ; SWAP WAIT
45 013270 000000G 000000G      .WORD  SUM4, TMIDLH    ; IDLE TIME
46 013274 000000G 000000      .WORD  SUM5, 0        ; END OF LINE
47 013300 000000G 000000G      .WORD  SUM6, TMIOH    ; USER I/O TIME
48 013304 000000G 000000G      .WORD  SUM7, TMSWPH    ; SWAP TIME
49 013310 000000G 000000      .WORD  SUM5, 0        ; END OF LINE
50 013314 000000G 000000      .WORD  SUM5, 0
51 013320 000000      .WORD  0              ; END OF LIST

```

```

1          .SBTTL  MEMORY command
2          ;-----
3          ; The MEMORY command is used to set or display the maximum memory limit
4          ; for the current job.
5          ;
6 013322  004767  000000G  CMDMEM: CALL  CVTTAB      ; CONVERT TAB AND FF CHARS TO SPACES
7 013326  111300                MOVB   (R3),R0      ; WAS A MEMORY LIMIT SPECIFIED WITH COMMAND?
8 013330  001004                BNE   SETMEM       ; BR IF YES
9          ;
10         ; Display current memory values
11         ;
12 013332  004767  000124      CALL  DSPMEM       ; DISPLAY JOB MEMORY LIMITS
13 013336  000167  000000G      JMP   RDCMD       ; GO GET NEXT COMMAND
14         ;
15         ; Set a new memory limit for the job.
16         ;
17 013342  105767  000000G  SETMEM: TSTB   VSWPFL      ; IS SWAPPING ALLOWED?
18 013346  001005                BNE   4$          ; BR IF YES
19 013350                .PRINT #NSWPMS      ; CAN'T CHANGE MEMORY SIZE OF NON-SWAP SYSTEM
20 013356  000167  000000G      JMP   RDCMD
21 013362  004767  000000G  4$: CALL  ACRDEC       ; ACCRUE THE VALUE
22 013366  120027  000113      CMPB  RO,#'K       ; DID HE SPECIFY K-SOMETHING?
23 013372  001001                BNE   1$          ; BR IF NOT
24 013374  005203                INC   R3          ; SKIP "K"
25 013376  121327  000127      1$: CMPB  (R3),#'W     ; WAS IT "KW"?
26 013402  001001                BNE   2$          ; BR IF NOT
27 013404  006301                ASL   R1          ; DOUBLE MEMORY VALUE
28         ; Compare request with max limit.
29 013406  020167  000000G  2$: CMP   R1, MXJMEM     ; IS REQUEST LARGET THAN MAX ALLOWED?
30 013412  101413                BLOS  3$          ; BR IF NOT
31 013414  016701  000000G      MOV   MXJMEM, R1    ; SET TO MAX ALLOWED
32 013420  010105                MOV   R1, R5
33 013422                .PRINT #MAXMTX     ; DISPLAY MAX ALLOWED
34 013430  004767  000000G      CALL  PRTDEC
35 013434                .PRINT #KBTX
36 013442  072127  000012      3$: ASH   #10., R1    ; CONVERT # KB TO ADDRESS
37 013446  001002                BNE   5$          ; BR IF DIDN'T OVERFLOW 64KB
38 013450  012701  177774      MOV   #177774, R1   ; SET TO 64KB
39 013454  010167  000000G  5$: MOV   R1, MAXMEM    ; SET AS MAX ADDRESS FOR JOB
40 013460                .EXIT           ; EXIT TO ACTUALLY DO THE MEMORY SIZE CHANGE
41         ;-----
42         ;
43         ; Display information about job memory limits.
44         ;
45 013462  010546      DSPMEM: MOV   R5, -(SP)
46 013464                .PRINT #CURMTX      ; CURRENT MEMORY =
47 013472  016705  000000G      MOV   MAXMEM, R5    ; GET CURRENT HIGH-MEMORY LIMIT FOR JOB
48 013476  020527  177770      CMP   R5, #177770   ; 64KB?
49 013502  103403                BLO   1$          ; BR IF NOT
50 013504  012705  000100      MOV   #64., R5      ; DO THIS TO AVOID OVERFLOW IN CONVERSION
51 013510  000403                BR    2$
52 013512  000305      1$: SWAB  R5          ; CONVERT TO # KB
53 013514  072527  177776      ASH   #-2, R5
54 013520  004767  000000G  2$: CALL  PRTDEC       ; DISPLAY THE VALUE
55 013524                .PRINT #KBTX       ; PRINT "KB"
56 013532                .PRINT #MAXMTX     ; MAX MEMORY =
57 013540  016705  000000G      MOV   MXJMEM, R5    ; MAX SIZE ALLOWED

```

MEMORY command

58 013544 004767 0000000
59 013550
60 013556 012605
61 013560 000207

CALL PRTDEC ; DISPLAY THE VALUE
.PRINT #KBTX ; "KB"
MOV (SP)+, R5
RETURN

MEMORY command

```

1
2
3
4
5 013562 010546
6 013564 006205
7 013566 004767 000000G
8 013572
9 013600 012605
10 013602 000207
11      000001

```

```

-----
; PRTKB is called to convert a value from # of 256-word memory pages
; to # k-bytes and print the value followed by "Kb<cr><lf>"
;
PRTKB:  MOV     R5, -(SP)
        ASR     R5                ; CONVERT # PAGES TO # KB
        CALL    PRTDEC           ; PRINT THE VALUE
        .PRINT  #KBTX           ; PRINT "KB<CR><LF>"
        MOV     (SP)+, R5
        RETURN
        .END

```

Errors detected: 0

*** Assembler statistics

```

Work file reads: 0
Work file writes: 0
Size of work file: 12332 Words ( 49 Pages)
Size of core pool: 18176 Words ( 71 Pages)
Operating system: RT-11

```

```

Elapsed time: 00:01:15.26
,LP:TSKSHD=DK:TSKSHD/C/N:SYM

```

| | | | | |
|---------|-------|--------|-------|-------|
| #1STLG | 1-67 | | | |
| #BBIT | 1-105 | | | |
| #AUTO | 1-81 | 9-114 | | |
| #CARMN | 1-198 | 9-62 | | |
| #CARUP | 1-79 | | | |
| #CCLRN | 1-80 | | | |
| #CFABT | 1-100 | | | |
| #CFALL | 1-106 | | | |
| #CFCCCL | 1-106 | | | |
| #CFDCC | 1-106 | | | |
| #CFOPN | 1-112 | | | |
| #CFSOT | 1-104 | | | |
| #CHACT | 1-55 | | | |
| #CLTST | 1-90 | | | |
| #CTRLC | 1-98 | | | |
| #CTRLD | 1-149 | | | |
| #CTRLO | 1-55 | | | |
| #CTRLS | 1-85 | | | |
| #DBKMN | 1-78 | | | |
| #DEAD | 1-153 | 9-23 | 9-159 | 9-184 |
| #DEBUG | 1-150 | | | |
| #DEFER | 1-118 | | | |
| #DETCH | 1-83 | | | |
| #DIBOL | 1-67 | | | |
| #DILUP | 1-102 | 9-118 | | |
| #DISCN | 1-84 | | | |
| #DOOFF | 1-108 | | | |
| #DUPRN | 1-103 | | | |
| #ECHO | 1-105 | | | |
| #EMTTR | 1-89 | | | |
| #FORM | 1-104 | | | |
| #FORMO | 1-106 | | | |
| #HARD | 1-153 | | | |
| #HITTY | 1-66 | | | |
| #INCOR | 1-122 | 11-118 | | |
| #INDAB | 1-154 | | | |
| #INDDF | 1-152 | | | |
| #INDRN | 1-152 | | | |
| #INIT | 1-153 | 9-58 | 9-165 | |
| #INKMN | 1-98 | | | |
| #KED | 1-122 | | | |
| #KINIT | 1-62 | 9-198 | 11-45 | |
| #LC | 1-105 | | | |
| #LOFCF | 1-198 | | | |
| #MLOCK | 1-71 | 11-125 | | |
| #NOIN | 1-66 | | | |
| #NOINT | 1-199 | | | |
| #NOWTT | 1-66 | | | |
| #PAGE | 1-105 | | | |
| #PHONE | 1-153 | 9-56 | | |
| #PRGLK | 1-81 | | | |
| #QTSET | 1-127 | | | |
| #QUIET | 1-119 | | | |
| #RNIDP | 1-200 | | | |
| #SCOPE | 1-105 | | | |
| #SGALL | 1-118 | | | |

| | | | | | |
|----------|-------|--------|-------|-------|-------|
| ACRFIL | 1-167 | | | | |
| ACRFN | 1-160 | | | | |
| ACROCT | 1-170 | | | | |
| ACRSPD | 1-70 | | | | |
| AD\$\$SZ | 1-58 | 19-15 | | 19-65 | |
| AD\$DVU | 1-58 | 19-40 | | | |
| AD\$JOB | 1-58 | 19-13 | | 19-34 | |
| ADM3A | 1-143 | | | | |
| ADM3FL | 1-144 | | | | |
| ADM3NO | 1-146 | | | | |
| AF\$BYA | 1-39 | 24-96 | | | |
| AF\$CCA | 1-32 | 24-102 | | | |
| AF\$DBG | 1-199 | 24-95 | | | |
| AF\$DUP | 1-32 | 24-98 | | | |
| AF\$HIE | 1-199 | 24-90 | | | |
| AF\$IND | 1-32 | 24-99 | | | |
| AF\$IOP | 1-200 | 24-92 | | | |
| AF\$MEM | 1-41 | 24-93 | | | |
| AF\$NOI | 1-199 | 24-91 | | | |
| AF\$NOW | 1-41 | 24-89 | | | |
| AF\$NPW | 1-32 | 24-103 | | | |
| AF\$PLK | 1-199 | 24-94 | | | |
| AF\$SCA | 1-41 | 24-88 | | | |
| AF\$SET | 1-32 | 24-101 | | | |
| AF\$TPO | 1-39 | 24-97 | | | |
| AF\$UCL | 1-32 | 24-100 | | | |
| ALCDEV | 1-28 | | | | |
| ALCEND | 1-57 | 19-16 | | 19-66 | |
| ALCHD1 | 1-54 | 19-28 | | | |
| ALCHD2 | 1-54 | 19-29 | | | |
| ALCTBL | 1-57 | 19-12 | | 19-33 | |
| ALDBLK | 1-183 | | | | |
| ALDEX | 1-182 | 1-183 | | | |
| ALFN | 1-194 | | | | |
| AMBOPT | 1-168 | 1-181 | | 5-18 | |
| AR\$\$SZ | 1-197 | | | | |
| AR\$CNT | 1-196 | | | | |
| AR\$CON | 1-196 | | | | |
| AR\$CPH | 1-196 | | | | |
| AR\$CPL | 1-196 | | | | |
| AR\$DMY | 1-197 | | | | |
| AR\$PRG | 1-196 | | | | |
| AR\$PRJ | 1-196 | | | | |
| AR\$UNM | 1-196 | | | | |
| ARNRPB | 1-197 | | | | |
| ASDEX | 1-165 | | | | |
| ASKLNM | 1-164 | | | | |
| ASNEND | 1-103 | 18-14 | 18-28 | 18-63 | |
| ASNHD1 | 1-177 | 18-6 | | | |
| ASNHD2 | 1-177 | 18-43 | | | |
| ASNOVF | 1-165 | 1-190 | | | |
| ASNTBL | 1-102 | 18-10 | 18-24 | 18-38 | |
| AT\$\$SZ | 1-100 | 1-132 | 18-13 | 18-27 | 18-62 |
| AT\$DEV | 1-132 | 18-44 | | | |
| AT\$EXT | 1-132 | 18-53 | | | |
| AT\$FIL | 1-132 | 18-47 | 18-50 | | |

| | | | |
|---------|--------|---------|--------|
| CFPNT | 1-119 | | |
| CFSEND | 1-112 | | |
| CFSP | 1-112 | | |
| CFSPND | 1-125 | | |
| CFSTK | 1-62 | | |
| CFSTS | 1-154 | | |
| CHAIN | 1-98 | | |
| CHKALC | 1-105 | | |
| CHKDEV | 1-191 | | |
| CHKDLM | 1-181 | | |
| CHKEQ | 1-39 | | |
| CHKMNT | 1-166 | | |
| CHKMTX | 1-166 | | |
| CINDAT | 1-136 | | |
| CINFLG | 1-42 | | |
| CKACQJ | 1-39 | | |
| CKCLUS | 1-33 | 10-65 | |
| CKPRIV | 1-168 | | |
| CKSYPV | 1-39 | 7-133 | 23-6 |
| CL\$COL | 1-60 | | |
| CL\$LEN | 1-95 | | |
| CL\$LIN | 1-95 | | |
| CL\$LIX | 1-56 | 10-50 | 10-99 |
| CL\$OPT | 1-94 | 10-101 | |
| CL\$RQH | 1-57 | | |
| CL\$SKP | 1-95 | | |
| CL\$WID | 1-95 | | |
| CL\$WQH | 1-57 | | |
| CLDEVX | 1-134 | 10-79 | |
| CLFREE | 1-51 | 9-202 | |
| CLLINE | 1-51 | 9-52 | |
| CLOPND | 10-120 | 10-166# | |
| CLOPTB | 10-102 | 10-153# | |
| CLOTIR | 1-59 | | |
| CLRPRV | 1-36 | | |
| CLSFCH | 1-63 | | |
| CLSFSP | 1-92 | | |
| CLTOTL | 1-92 | 10-12 | 10-133 |
| CLUNIT | 1-51 | 9-204 | |
| CLVERS | 1-51 | 10-21 | |
| CMDBUF | 1-159 | 1-176 | 13-24 |
| CMDCCL | 1-191 | | |
| CMDDSN | 1-27 | | |
| CMDFRM | 1-27 | | |
| CMDHD | 1-26 | | |
| CMDMEM | 1-28 | 36-6# | |
| CMDOFF | 1-26 | | |
| CMDSET | 1-28 | | |
| CMDSHO | 1-28 | 5-5# | |
| CMDUSE | 1-28 | 34-13# | |
| CMDWHO | 1-28 | 34-6# | |
| CO\$BBT | 1-109 | 10-157 | |
| CO\$BNI | 1-94 | 10-162 | |
| CO\$BNO | 1-94 | 10-161 | |
| CO\$CR | 1-93 | 10-163 | |
| CO\$CTL | 1-93 | 10-164 | |

| | | | | |
|---------|-------|--------|--------|-------|
| DEVDSP | 14-16 | 15-7# | | |
| DEVHD1 | 1-177 | | | |
| DEVIDL | 1-182 | 1-182 | 1-183 | |
| DEVUNT | 1-169 | | | |
| DFJMEM | 1-62 | | | |
| DIABFL | 1-119 | | | |
| DIABLO | 1-142 | | | |
| DIABNO | 1-120 | | | |
| DIVIDE | 1-175 | 11-150 | | |
| DIVSOR | 1-188 | 35-19* | 35-20* | |
| DJABMS | 1-186 | | | |
| DKASHD | 1-53 | 18-30 | | |
| DKSAV | 1-159 | | | |
| DLCEMT | 1-28 | | | |
| DLMSG | 1-186 | | | |
| DLTXT | 1-173 | | | |
| DMTALL | 1-186 | | | |
| DMTARG | 1-165 | | | |
| DMTSUB | 1-191 | | | |
| DMYDEV | 1-39 | 30-10 | 31-22 | |
| DDASGN | 1-79 | | | |
| DORUN | 1-27 | | | |
| DOSTOP | 1-187 | | | |
| DSPCSR | 15-64 | 16-7# | | |
| DSPMEM | 8-68 | 36-12 | 36-45# | |
| DVEC | 4-6# | 16-33* | 16-38 | 16-71 |
| DVSHH1 | 1-53 | 14-9 | | |
| DVSHH2 | 1-53 | 14-10 | | |
| DVSHH3 | 1-53 | 14-11 | | |
| DVSTAT | 1-70 | 15-29 | | |
| DZTXT | 1-193 | | | |
| EDIT | 1-67 | | | |
| EDTFIL | 1-179 | 24-16 | 32-48 | |
| EM\$ACL | 1-48 | | | |
| EM\$CAP | 1-38 | | | |
| EM\$CIP | 1-47 | | | |
| EM\$CLB | 1-48 | | | |
| EM\$CLN | 1-47 | | | |
| EM\$CNO | 1-38 | | | |
| EM\$CPO | 1-38 | | | |
| EM\$CSE | 1-64 | | | |
| EM\$HNI | 1-62 | | | |
| EM\$ICL | 1-46 | | | |
| EM\$ILN | 1-47 | 1-48 | | |
| EM\$IUN | 1-47 | | | |
| EM\$NAD | 1-40 | | | |
| EM\$NPD | 1-47 | | | |
| EM\$NSF | 1-47 | | | |
| EM\$NSL | 1-48 | | | |
| EM\$NUK | 1-80 | | | |
| EM\$SLT | 1-48 | | | |
| EM\$SLW | 1-48 | | | |
| EM\$TSL | 1-48 | | | |
| EM\$UIO | 1-49 | | | |
| ERRLOC | 1-61 | | | |
| ERRSEV | 1-128 | | | |

| | | | |
|--------|-------|--------|-------|
| LCPUHI | 1-85 | 11-167 | |
| LCPULO | 1-85 | 11-168 | |
| LD#RON | 1-115 | 32-59 | |
| LDBASE | 1-116 | | |
| LDCLEN | 1-65 | 32-14 | |
| LDDEVX | 1-118 | | |
| LDFLAG | 1-116 | 32-59 | |
| LDMNT | 1-64 | | |
| LDNAM | 1-164 | | |
| LDNAME | 1-116 | 32-19 | 32-31 |
| LDOPHD | 1-172 | | |
| LDPDEV | 1-116 | 32-65 | |
| LDSIZE | 1-116 | 32-53 | |
| LF | 2-4# | | |
| LF#OPN | 1-157 | | |
| LF#WRT | 1-157 | | |
| LFWLIM | 1-77 | | |
| LICTXT | 1-139 | | |
| LINBUF | 1-72 | | |
| LINCNT | 1-74 | | |
| LINCUR | 1-77 | | |
| LINFRE | 1-186 | | |
| LINIR | 1-59 | | |
| LINNXT | 1-72 | | |
| LINPNT | 1-74 | | |
| LINRTS | 1-59 | | |
| LITIME | 1-99 | | |
| LJSW | 1-96 | | |
| LMXLN | 1-153 | 9-72 | 9-82 |
| LMXNUM | 1-151 | 9-73 | 9-95 |
| LMXPRM | 1-154 | 9-122 | 9-138 |
| LNAME | 1-54 | 9-188 | |
| LNBLKS | 1-101 | 11-135 | |
| LNMAP | 1-107 | | |
| LNPRIM | 1-107 | 9-39 | 11-71 |
| LNSBLK | 1-102 | 11-136 | |
| LNSPAC | 1-111 | | |
| LOCKTX | 1-175 | 11-127 | |
| LOFSPC | 1-104 | | |
| LOGASN | 1-167 | | |
| LOGBAS | 1-126 | 1-128 | |
| LOGBLK | 1-156 | | |
| LOGBUF | 1-156 | | |
| LOGCHK | 1-127 | | |
| LOGCHN | 1-156 | | |
| LOGCLS | 1-171 | | |
| LOGDVU | 1-126 | 1-128 | |
| LOGFLG | 1-156 | | |
| LOGPTR | 1-156 | | |
| LOMAP | 1-137 | 8-33 | 8-61 |
| LOTBUF | 1-75 | | |
| LOTNXT | 1-75 | | |
| LOTPNT | 1-75 | | |
| LOTSIZ | 1-76 | | |
| LOTSPC | 1-76 | | |
| LOUTIR | 1-59 | | |

| | | | | | | | |
|---------|-------|--------|-------|-------|-------|-------|--------|
| LP\$7BT | 1-46 | 9-139 | | | | | |
| LP\$ODD | 1-46 | 9-148 | | | | | |
| LP\$PAR | 1-46 | 9-144 | | | | | |
| LP\$SPD | 1-46 | | | | | | |
| LPRG1 | 1-140 | 11-174 | | | | | |
| LPRG2 | 1-140 | 11-176 | | | | | |
| LPRI | 1-155 | 7-98 | 11-93 | | | | |
| LPROG | 1-84 | | | | | | |
| LPROJ | 1-84 | | | | | | |
| LRBFIL | 1-108 | | | | | | |
| LRDTIM | 1-74 | | | | | | |
| LSCCA | 1-104 | | | | | | |
| LSECT | 1-89 | 11-76 | | | | | |
| LSTACT | 1-72 | | | | | | |
| LSTATE | 1-135 | 11-101 | | | | | |
| LSTD | 1-83 | 11-67 | | | | | |
| LSTHL | 1-56 | 9-213 | | | | | |
| LSTIOL | 1-56 | | | | | | |
| LSTMX | 1-151 | | | | | | |
| LSTPL | 1-129 | 9-53 | 9-163 | 9-196 | 9-215 | 11-65 | 11-74 |
| LSTPRM | 1-125 | | | | | | |
| LSTSL | 1-135 | 11-188 | | | | | |
| LSTSPL | 1-70 | 12-16 | | | | | |
| LSUCF | 1-80 | | | | | | |
| LSW | 1-55 | 9-58 | 9-118 | 9-165 | 9-198 | 11-45 | 11-118 |
| LSW2 | 1-98 | | | | | | |
| LSW2S | 1-103 | | | | | | |
| LSW3 | 1-103 | 9-23 | 9-159 | 9-184 | | | |
| LSW4 | 1-121 | | | | | | |
| LSW5 | 1-81 | 9-62 | | | | | |
| LSW6 | 1-150 | 11-125 | | | | | |
| LSW7 | 1-154 | 28-15 | 29-18 | | | | |
| LSW8 | 1-117 | | | | | | |
| LSW9 | 1-79 | | | | | | |
| LTRMTP | 1-143 | | | | | | |
| LTSCMD | 1-111 | | | | | | |
| LUNAME | 1-84 | | | | | | |
| MAXALC | 1-57 | | | | | | |
| MAXASN | 1-100 | | | | | | |
| MAXAVL | 1-169 | | | | | | |
| MAXMEM | 1-61 | 36-39* | 36-47 | | | | |
| MAXMTX | 1-181 | 36-33 | 36-56 | | | | |
| MAXPRI | 1-61 | 1-155 | | | | | |
| MAXSEC | 1-89 | 11-77 | | | | | |
| MDMTXT | 1-202 | 21-11 | | | | | |
| MDT | 1-71 | | | | | | |
| MHNSIZ | 1-90 | 8-40 | | | | | |
| MHNSMS | 1-91 | 8-39 | | | | | |
| MINTIM | 1-89 | 11-145 | | | | | |
| MISSEQ | 1-170 | | | | | | |
| MNBASE | 1-180 | | | | | | |
| MNBPC | 1-179 | | | | | | |
| MNFLGS | 1-179 | | | | | | |
| MNTARG | 1-190 | | | | | | |
| MNTDEV | 1-165 | | | | | | |
| MNTFUL | 1-167 | | | | | | |

| | | | | | | | | | | |
|----------|--------|---------|--------|--------|-------|-------|-------|-------|-------|-------|
| NUMDEV | 1-146 | 14-18 | | | | | | | | |
| NUMON | 1-77 | | | | | | | | | |
| NUMST | 11-105 | 11-226# | | | | | | | | |
| OCTFIX | 1-173 | 9-89 | 15-31 | 15-42 | 15-50 | 16-65 | 16-75 | 16-88 | 33-20 | 33-23 |
| OCTPRT | 1-193 | 8-22 | 9-102 | 26-103 | | | | | | |
| ODTBAS | 1-149 | | | | | | | | | |
| OF\$\$SZ | 1-148 | | | | | | | | | |
| OF\$DEV | 1-147 | | | | | | | | | |
| OF\$FIL | 1-147 | | | | | | | | | |
| OF\$FLG | 1-147 | | | | | | | | | |
| OF\$UNT | 1-147 | | | | | | | | | |
| OFFEMT | 1-187 | | | | | | | | | |
| OFTTXT | 1-202 | 21-15 | | | | | | | | |
| OKFEND | 1-90 | | | | | | | | | |
| OKFILE | 1-90 | | | | | | | | | |
| ONTTXT | 1-202 | 21-12 | | | | | | | | |
| OPRTXT | 1-51 | 9-49 | | | | | | | | |
| OPTLST | 1-36 | | | | | | | | | |
| OT\$RON | 1-148 | | | | | | | | | |
| OTHRON | 1-189 | | | | | | | | | |
| OTRMNT | 1-191 | | | | | | | | | |
| OVRCOR | 1-163 | | | | | | | | | |
| PO\$DBG | 1-41 | | | | | | | | | |
| PO\$OPR | 1-39 | | | | | | | | | |
| PO\$SPV | 1-37 | | | | | | | | | |
| PO\$SYS | 1-37 | 6-33 | | | | | | | | |
| PARNNL | 1-37 | 8-23 | | | | | | | | |
| PASLIN | 1-128 | | | | | | | | | |
| PAUMSG | 1-159 | | | | | | | | | |
| PBFEND | 1-112 | | | | | | | | | |
| PEKADR | 1-35 | 25-22* | 25-28 | | | | | | | |
| PEKEMT | 1-35 | 25-24 | | | | | | | | |
| PEKSIZ | 1-35 | 25-23* | | | | | | | | |
| PF\$IOW | 1-150 | | | | | | | | | |
| PF\$SYS | 1-150 | | | | | | | | | |
| PFCO | 1-36 | | | | | | | | | |
| PFSO | 1-36 | | | | | | | | | |
| PHNTXT | 1-202 | 21-5 | | | | | | | | |
| PHMEM | 1-95 | 8-8 | | | | | | | | |
| PIDCSR | 16-55 | 17-36# | | | | | | | | |
| PIDSLT | 17-11# | 17-36 | 17-53 | | | | | | | |
| PIDVEC | 16-83 | 17-53# | | | | | | | | |
| PMBUSY | 1-180 | | | | | | | | | |
| PNAME | 1-146 | 1-169 | 15-14 | 16-16 | | | | | | |
| POPCF | 1-170 | | | | | | | | | |
| PRGALL | 1-27 | | | | | | | | | |
| PRGSIZ | 1-72 | | | | | | | | | |
| PRGTOP | 1-72 | | | | | | | | | |
| PRIVAO | 1-38 | 27-12 | | | | | | | | |
| PRIVCO | 1-41 | 6-33 | 27-23 | | | | | | | |
| PRIVSO | 1-37 | | | | | | | | | |
| PRMBUF | 1-125 | | | | | | | | | |
| PRMEND | 1-125 | | | | | | | | | |
| PRMPNT | 1-124 | | | | | | | | | |
| PROSLT | 1-46 | 17-12 | 17-17 | | | | | | | |
| PRTBUF | 1-178 | 20-118 | 20-127 | 20-132 | | | | | | |

| | | | | | | |
|----------|-------|--------|--------|-------|-------|-------|
| S\$TTSC | 1-83 | 11-207 | | | | |
| S\$TWFN | 1-82 | 11-211 | | | | |
| S9600 | 1-87 | | | | | |
| SCHAIN | 1-118 | | | | | |
| SD\$BAK | 1-121 | | | | | |
| SD\$DEL | 1-113 | | | | | |
| SD\$FLK | 1-114 | | | | | |
| SD\$HLD | 1-124 | | | | | |
| SD\$SNG | 1-123 | | | | | |
| SD\$WFM | 1-114 | | | | | |
| SDBLK | 1-115 | | | | | |
| SDBU | 1-121 | | | | | |
| SDBUF1 | 1-115 | | | | | |
| SDCB | 1-129 | 10-88 | 12-7 | 31-12 | | |
| SDCBND | 1-129 | 10-89 | 12-8 | 12-18 | 31-13 | 31-27 |
| SDCBSZ | 1-135 | 10-93 | 12-12 | 12-17 | 31-26 | |
| SDDVU | 1-134 | 10-91 | | | | |
| SDFHD | 1-126 | 12-10 | | | | |
| SDFLAG | 1-114 | | | | | |
| SDFORM | 1-114 | | | | | |
| SDNAME | 1-135 | 31-21 | | | | |
| SDSFCB | 1-113 | | | | | |
| SDSKIP | 1-121 | | | | | |
| SEARCH | 1-159 | 5-10 | 11-25 | | | |
| SERFLG | 1-55 | | | | | |
| SETHD | 1-168 | | | | | |
| SETMEM | 36-8 | 36-17# | | | | |
| SF\$1ST | 1-124 | | | | | |
| SF\$BSY | 1-123 | | | | | |
| SF\$HLD | 1-124 | | | | | |
| SFFILE | 1-141 | | | | | |
| SFFLAG | 1-126 | | | | | |
| SFFORM | 1-123 | | | | | |
| SFID | 1-70 | | | | | |
| SFNMBL | 1-123 | | | | | |
| SFQLNK | 1-126 | | | | | |
| SFUSER | 1-141 | | | | | |
| SH\$\$SZ | 1-68 | | | | | |
| SH\$FLG | 1-68 | | | | | |
| SH\$NAM | 1-68 | | | | | |
| SH\$RTN | 1-68 | | | | | |
| SH\$VAL | 1-68 | | | | | |
| SHMTH1 | 1-177 | 20-59 | | | | |
| SHMTH2 | 1-177 | 20-60 | | | | |
| SHOALC | 6-11 | 6-83 | 19-5# | | | |
| SHOASN | 6-9 | 6-46 | 18-5# | | | |
| SHOCL | 6-16 | 6-88 | 10-5# | | | |
| SHOCMD | 6-52 | 13-13# | | | | |
| SHODAT | 6-63 | 22-5# | | | | |
| SHODEV | 6-7 | 6-57 | 6-68 | 14-5# | | |
| SHOHD | 1-192 | 3-69# | 5-9 | | | |
| SHOINS | 6-35 | 6-93 | 23-5# | | | |
| SHOJOB | 6-13 | 6-73 | 11-11# | 34-6 | | |
| SHOKEY | 1-33 | 3-87 | | | | |
| SHOMDM | 6-103 | 21-5# | | | | |
| SHOMEM | 6-18 | 6-78 | 8-5# | | | |

| | | | | |
|--------|---------|--------|---------|-------|
| STPASK | 1-189 | | | |
| STPBUF | 11-113* | 11-114 | 11-251# | |
| STPFLG | 1-86 | | | |
| SUBARD | 1-179 | 32-42 | | |
| SUBTXT | 1-192 | 6-22 | | |
| SUCS | 1-139 | | | |
| SUM1 | 1-188 | 35-42 | | |
| SUM2 | 1-188 | 35-43 | | |
| SUM3 | 1-188 | 35-44 | | |
| SUM4 | 1-188 | 35-45 | | |
| SUM5 | 1-189 | 35-46 | 35-49 | 35-50 |
| SUM6 | 1-189 | 35-47 | | |
| SUM7 | 1-189 | 35-48 | | |
| SUMS | 1-139 | | | |
| SUMVEC | 35-21 | 35-42# | | |
| SUPCOD | 1-139 | | | |
| SWPTX | 1-175 | 11-120 | | |
| SXBPNT | 1-63 | | | |
| SYASHD | 1-53 | 18-16 | | |
| SYHD1 | 1-174 | 11-36 | | |
| SYHD2 | 1-174 | 11-37 | | |
| SYINDX | 1-146 | | | |
| SYNAME | 1-147 | 16-15 | 18-17 | 18-31 |
| SYPSWD | 1-33 | 7-134 | 7-138 | |
| SYSAV | 1-159 | | | |
| SYSDAT | 1-136 | | | |
| SYTIMH | 1-136 | | | |
| SYTIML | 1-136 | | | |
| SYUNIT | 1-146 | | | |
| TAB | 2-8# | | | |
| TALEMT | 1-108 | | | |
| TBLOVF | 1-168 | | | |
| TECO | 1-67 | | | |
| TK1SEC | 1-138 | | | |
| TK1VAL | 1-136 | | | |
| TM#AUT | 1-51 | 9-120 | | |
| TM#C13 | 1-54 | 10-40 | | |
| TM#CDS | 1-50 | 7-122 | | |
| TM#CEN | 1-50 | 7-124 | | |
| TM#CLO | 1-52 | 10-14 | | |
| TM#CL1 | 1-52 | 10-27 | | |
| TM#CL2 | 1-52 | 10-28 | | |
| TM#CL3 | 1-52 | 10-43 | | |
| TM#CL4 | 1-52 | 10-58 | 10-73 | |
| TM#CL5 | 1-52 | 10-95 | | |
| TM#CL6 | 1-52 | 10-108 | | |
| TM#CL7 | 1-52 | 10-20 | | |
| TM#CL8 | 1-53 | 10-23 | | |
| TM#CNG | 1-50 | 7-118 | | |
| TM#GBL | 1-43 | 26-64 | | |
| TM#HPE | 1-50 | 7-111 | | |
| TM#HPR | 1-49 | 7-108 | | |
| TM#IN1 | 1-35 | 23-10 | | |
| TM#IN2 | 1-35 | 23-11 | | |
| TM#KED | 1-34 | 28-25 | | |
| TM#LCL | 1-43 | 26-61 | | |

| | | | | |
|---------|-------|-------|-------|-------|
| TM\$LPR | 1-49 | 7-104 | | |
| TM\$NAD | 1-54 | 19-21 | | |
| TM\$NNR | 1-35 | 25-36 | | |
| TM\$NO | 1-34 | 29-20 | | |
| TM\$NSD | 1-54 | 31-15 | | |
| TM\$NSP | 1-33 | 7-136 | | |
| TM\$OFF | 1-33 | 28-17 | | |
| TM\$ON | 1-33 | 28-19 | | |
| TM\$PR1 | 1-49 | 7-97 | | |
| TM\$PR2 | 1-49 | 7-100 | | |
| TM\$PVA | 1-38 | 27-11 | | |
| TM\$PVC | 1-38 | 27-22 | | |
| TM\$PVL | 1-39 | 24-59 | | |
| TM\$RD1 | 1-43 | 26-33 | | |
| TM\$RD2 | 1-43 | 26-34 | | |
| TM\$SDN | 1-54 | 31-20 | | |
| TM\$SL1 | 1-33 | 28-11 | | |
| TM\$SUB | 1-34 | 28-37 | | |
| TM\$TTY | 1-34 | 28-31 | | |
| TMIDLH | 1-65 | 35-45 | | |
| TMIOH | 1-65 | 35-47 | | |
| TMIOWH | 1-64 | 35-43 | | |
| TMITXT | 1-202 | 21-18 | | |
| TMLTXI | 1-202 | 21-21 | | |
| TMOTXT | 1-202 | 21-24 | | |
| TMSWPH | 1-65 | 35-48 | | |
| TMSWTH | 1-65 | 35-44 | | |
| TMTOTH | 1-64 | 1-188 | 35-11 | 35-19 |
| TMTOTL | 1-64 | 1-188 | 35-12 | 35-20 |
| TMUSRH | 1-64 | 35-42 | | |
| TOTMMS | 1-192 | 8-7 | | |
| TOTON | 1-86 | | | |
| TOTXT | 1-171 | | | |
| TRGRET | 1-139 | | | |
| TRMHD1 | 1-50 | 9-14 | | |
| TRMHD2 | 1-50 | 9-15 | | |
| TRMSTR | 1-161 | | | |
| TSKSHO | 1-5# | 1-26 | | |
| TSR | 1-151 | | | |
| TSXLN | 1-139 | | | |
| TSXSIT | 1-139 | | | |
| TSXSMS | 1-193 | 8-32 | | |
| TSXVER | 1-36 | 22-27 | | |
| TSXVRS | 1-32 | 22-28 | | |
| TXTC1 | 1-86 | 9-173 | | |
| TXTCL | 1-85 | 9-175 | | |
| UC\$MDC | 1-158 | | | |
| UC\$NDC | 1-158 | | | |
| UCHAN | 1-106 | | | |
| UCIDEF | 1-58 | | | |
| UCISPC | 1-90 | | | |
| UCLBLK | 1-157 | 13-17 | | |
| UCLCMD | 1-26 | 6-51# | | |
| UCLDAT | 1-157 | | | |
| UCLNAM | 1-111 | | | |
| UERSEV | 1-128 | | | |

| | | | | | | | |
|---------|-------|--------|-------|-------|------|-------|-------|
| UFORM | 1-98 | | | | | | |
| UFPTRP | 1-113 | | | | | | |
| UHIMEM | 1-101 | | | | | | |
| UKMNAM | 1-79 | | | | | | |
| UMSSMS | 1-192 | 8-13 | | | | | |
| UMSYTP | 1-83 | 8-14 | 8-21 | | | | |
| UPTMMS | 1-187 | 35-10 | | | | | |
| USPLCH | 1-86 | | | | | | |
| USRMMMS | 1-193 | 8-59 | | | | | |
| USRSTK | 1-62 | | | | | | |
| USTART | 1-97 | | | | | | |
| UTRPAD | 1-61 | | | | | | |
| VCORTM | 1-133 | 7-57 | | | | | |
| VCSHNB | 1-89 | 7-120 | 7-125 | | | | |
| VHIPCT | 1-130 | 7-67 | | | | | |
| VIMAGE | 1-96 | | | | | | |
| VINTIO | 1-122 | 7-62 | | | | | |
| VMAXMC | 1-34 | 7-9 | | | | | |
| VMXMRB | 1-34 | 7-14 | | | | | |
| VMXMSG | 1-34 | 7-4 | | | | | |
| VOFFTM | 1-201 | 21-16 | | | | | |
| VONTM | 1-201 | 21-13 | | | | | |
| VPRIDF | 1-78 | 7-87 | | | | | |
| VPRIHI | 1-42 | 7-82 | 7-109 | | | | |
| VPRILO | 1-42 | 7-77 | 7-105 | | | | |
| VPRIVR | 1-78 | 7-92 | | | | | |
| VQUANO | 1-129 | 1-130 | 7-19 | | | | |
| VQUAN1 | 1-130 | 7-24 | | | | | |
| VQUAN2 | 1-130 | 7-44 | | | | | |
| VQUAN3 | 1-129 | 1-130 | 7-52 | | | | |
| VQUN1A | 1-130 | 7-29 | | | | | |
| VQUN1B | 1-122 | 7-34 | | | | | |
| VQUN1C | 1-122 | 7-39 | | | | | |
| VSWPFL | 1-63 | 36-17 | | | | | |
| VT100 | 1-142 | | | | | | |
| VT10FL | 1-144 | | | | | | |
| VT10ND | 1-144 | | | | | | |
| VT200 | 1-141 | | | | | | |
| VT2007 | 1-141 | | | | | | |
| VT2008 | 1-141 | | | | | | |
| VT20FL | 1-145 | | | | | | |
| VT20ND | 1-145 | | | | | | |
| VT52 | 1-142 | | | | | | |
| VT52FL | 1-143 | | | | | | |
| VT52ND | 1-120 | | | | | | |
| VTMIN | 1-201 | 21-19 | | | | | |
| VTMLDC | 1-201 | 21-22 | | | | | |
| VTMDUT | 1-201 | 21-25 | | | | | |
| VUSPHN | 1-198 | 9-60 | 21-7 | | | | |
| WILDFL | 1-66 | | | | | | |
| WLDNAM | 2-12# | | | | | | |
| WTMS | 1-174 | 11-110 | | | | | |
| XAREA | 1-160 | 16-21 | 16-26 | 16-32 | 33-8 | 33-15 | 33-27 |
| YESTXT | 1-173 | 9-167 | 26-87 | 26-96 | | | |
| ZCLR | 1-151 | | | | | | |

