

Table of contents

7-	1	SET command
9-	1	System parameters
11-	1	Real device
15-	1	Prompt
16-	1	Syspassword
17-	1	Endstartup
18-	1	Window
20-	1	Printwindow
21-	1	Priority
22-	1	Maxpriority
23-	1	Error
23-	12	Shutdown
24-	1	Log
26-	1	Logoff
27-	1	Subprocess
28-	1	Kmon
30-	1	LD
34-	1	SL

```

1          .TITLE  TSKST1 -- Keyboard SET Command routines
2          .ENABL  LC
3          .DSABL  QBL
4 000000   .CSECT  TSKST1
5 000000   TSKST1:
6          ;
7          ; TSKST1 is the portion of TSKMON that contains the code
8          ; to implement the SET 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 .QTLIN, .QTIM, .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 TSKST1, CMDHD, CMDOFF, KDDCIN, 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 LSW11, $PWKEY, SETCL, SETHST, DVFLAG, DX$NST, TM$MRS
33         .GLOBL SC$SUC, SC$WRN, SC$ERR, SC$SEV, SC$FTL, SC$UNC, SC$NON
34         .GLOBL SETPRC, SETTTY, STVRFY, STNOVR, PO$SYS, RCLREV
35         .GLOBL PA$GRC, PA$UKC, PA$DSC, PA$BLD, PA$ULN, PA$DWD, PA$HQL, PA$LET
36         .GLOBL EM$PTA, EM$PTU, SBPSUF, CHKCLU, EM$CLX, PA$BEL, EM$OPR, PA$NWD
37         .GLOBL SCNOPS, TM$XBK, CL$XLN, CKCLUS, ACRSTR, R5OKMN, SJEMT, RJEMT
38         .GLOBL PEKEMT, PEKADR, PEKSIZ, TM$NNR, CDBUF, CDGET, TM$IN1, TM$IN2
39         .GLOBL SYPSWD, EM$SPL, ACRTXT, JPWDEV, JPWTYP, JPWFLG
40         .GLOBL EM$WCO, EM$WC1, EM$WC2, EM$WC3, EM$WCM, P2$CGR
41         .GLOBL CXTRMN, R$CHN, R$XCHN, CHNSIZ, C. USED, CL$EPN, CL$EPS, CLEOFS
42         .GLOBL EDMALD, EM$STL, EM$IST, CLSFEP, EM$IDR, VPRIDF, SETWRD
43         .GLOBL CLRPRV, OPTLST, PFSO, PFCO, PVNPW, PO$SPV, PRIVSO, PRVOPT
44         .GLOBL TM$PVA, TM$PVC, PRIVAO, EM$CND, EM$CPD, EM$CAP, RSTPRV
45         .GLOBL CHKEQ, CKACQJ, PO$OPR, CKSYPV, P2$TRM, PRIVC2, PO$NAM, EM$NPR
46         .GLOBL INSTBL, INGADR, INGEMT, IIBUF, II$NAM, II$FLG, II$SZ, EM$NAD
47         .GLOBL INSTBN, AF$SCA, AF$NOW, AF$MEM, PO$DBG, PRIVCO, PRVLST
48         .GLOBL ABRTAD, ABRTCD, CINFLG, $VNOTT, II$PRV, II$NPV
49         .GLOBL TM$RD1, TM$RD2, TM$LCL, TM$GBL, SPACE1, RC$OWN, RC$CNT
50         .GLOBL RC$EXC, RC$AGE, RC$AEP, RC$USE, RC$FLG, RC$GBL, RC$NAM
51         .GLOBL RC$LEN, RC$PVT, RCBBAS, RCBEND, RC$SZ, SHRRCB, SHRRCN
52         .GLOBL LP$SPD, LP$PAR, LP$ODD, LP$7BT, EM$ICL, PROSLT, RC$LCC
53         .GLOBL EM$NPD, EM$ILN, EM$CIP, EM$NSF, EM$IUN, EM$CLN
54         .GLOBL EM$ILN, EM$ACL, EM$TSL, EM$CLB, EM$NSL, EM$SLT, EM$SLW
55         .GLOBL SLKDON, SLKDOF, EM$UID, TM$PR1, TM$PR2, TM$LPR, TM$HPR
56         .GLOBL TM$HPE, TM$CNG, TM$CDS, TM$CEN, TRMHD1, TRMHD2, AT$DEV
57         .GLOBL OPRTXT, CLLINE, LCLTXT, REMTXT, TM$AUT, CLFREE, CLUNIT

```

```

58 . GLOBL TM#CLO, TM#CL1, TM#CL2, TM#CL3, TM#CL4, TM#CL5, TM#CL6
59 . GLOBL QHDMS1, QHDMS2, DVSHH1, DVSHH2, DVSHH3, SYASHD, DKASHD
60 . GLOBL TM#NAD, ALCHD1, ALCHD2, TM#NSD, TM#SDN, LNAME
61 . GLOBL CORUSR, LSW, #CTRL0, SERFLG, IOABFL, #CHACT, #STSNG
62 . GLOBL LSTHL, LCLUNT, FSTIOL, LSTIOL, CL$LIX, CW$PRO, CONFG2
63 . GLOBL CL$RQH, CL$WQH, MAXALC, ALCTBL, ALCEND
64 . GLOBL AD$DVU, AD$JOB, AD$SZ, UCIDEF, HANCHN
65 . GLOBL NEDCHR, LOUITR, LINIR, LINRTS, CLOTIR
66 . GLOBL CO$DEF, CL$COL, L$DTYP, SOPALC, SOPDAT, SOPTIM
67 . GLOBL UTRPAD, JSWLOC, ERRLOC, MAXMEM, MAXPRI
68 . GLOBL USRSTK, #KINIT, CFSTK, MXJMEM, DFJMEM, EM$HNI
69 . GLOBL SPUBUF, SXBPNT, MXJADR
70 . GLOBL TMTOTH, TMTOTL, TMUSRH, TMIOWH, LDMNT, EM$CSE
71 . GLOBL TMSWTH, TMIDLH, TMIOH, TMSWPH, LDCLN
72 . GLOBL WILDFL, #NOIN, #NOWTT, $HITTY
73 . GLOBL TECO, EDIT, KED, K52, $1STLG, $DIBOL
74 . GLOBL SH$VAL, SH$NAM, SH$SZ, SH$RTN, SH$FLG
75 . GLOBL SO$NVL, SO$OCT, SO$NO, HANENT, HANSIZ
76 . GLOBL H. CSR, H. VEC, DVSTAT, SFID, ACRSPD, HANPAR, LSTSPL
77 . GLOBL HAZEL, HAZLFL, HAZLNO, $MLOCK, MDT, GETKCH
78 . GLOBL LINBUF, LINNXT, LSTACT, PRGTOP, PRGSIZ, KMNHI
79 . GLOBL KMNTOP, KMNPGS, KMNSTK, KMNSTR, CXTAG, FSTIOL
80 . GLOBL LIMPNT, LINCNT, LACTIV, LRDTIM, CS$RON
81 . GLOBL LOTBUF, LOTNXT, LOTPNT, $VTESE
82 . GLOBL LOTSIZ, LOTSPC, LCOL, $SLKED, ESC, VDBFLG
83 . GLOBL LAFSIZ, LFWLIM, LINCUR, NUMON, ILSW2
84 . GLOBL $DBKMN
85 . GLOBL $CARUP, DOASGN, UKMNAM, $UKMON, LSW9
86 . GLOBL LSUCF, $CCLRN, VLDSYS, EM$NUK, S$QMIO
87 . GLOBL KL3CLR, $PRGLK, LSW5, PVON, S$SPND, $AUTO
88 . GLOBL S$TWFN, S$TTFN, S$OTFN, S$IOFN, S$OTLO
89 . GLOBL LSTD, FSTD, $DETCH, UMSYTP, S$TTSC
90 . GLOBL $DISCN, LPROJ, LPROG, LUNAME, S$RT, S$LOW
91 . GLOBL LCPUI, LCPULO, LCONTM, $CTRLS, $SPLJB, TXTCL
92 . GLOBL STPFLG, TON, USPLCH, SPLCHN, S$HICP
93 . GLOBL S$INWT, S$OTWT, S$TMWT, S$SFWT, S9600
94 . GLOBL S$MSWT, CFBUF, CFEND, CCLSAV, KMNCHN
95 . GLOBL MINTIM, LSECPT, MAXSEC, $EMTTR, VCSHNB
96 . GLOBL OKFILE, OKFEND, $CLTST, UCISPC, MHNSIZ
97 . GLOBL CASTBR, CASCBR, CASTBW, CASCUP, MHNSMS
98 . GLOBL CASTRO, CASTWO, CLTOTL, CO$DTR, CLSFSP
99 . GLOBL CO$CR, CO$FF, CO$FFO, CO$LC, CO$TAB, CO$CTL
100 . GLOBL CO$LFI, CO$LFO, CO$BNI, CO$BNO, CL$OPT
101 . GLOBL CL$LEN, CL$SKP, CL$WID, CL$LIN, PHYMEM
102 . GLOBL LJSW, CTRLTT, NEWJSW, JSTKND, VIMAGE
103 . GLOBL USTART, GENTOP, BOTDEV, BOTUNI, CSHALC
104 . GLOBL $CTRLC, LSW2, $INKMN, CHAIN, UFORM
105 . GLOBL $SQO, $SQO3, LITIME
106 . GLOBL MAXASN, $CFABT, INDSTA, INDERR
107 . GLOBL RUNDEV, LNBLKS, CXTBAS, CXTWDS, UHIMEM
108 . GLOBL $DILUP, CSHDEV, CSHDVN, LNSBLK
109 . GLOBL LSW3, LSW2S, $DUPRN
110 . GLOBL $FORM, $TAB, LSCCA, $CFSOT, LOFSPC, R50COM
111 . GLOBL $PAGE, $SCOPE, $ECHO, $LC, $BBIT, CHKALC
112 . GLOBL UCHAN, $FORMO, $CFALL, $CFDCC, $CFCLL
113 . GLOBL LNPRIM, LNMAP, CW$50H, CONFIG, $SUCF
114 . GLOBL $DOOFF, NUCHN, LRBFIL, CFIND, TALEMT

```

```

115 . GLOBL C, CSW, C, DEVG, C, SBLK, NLINES, CD$BBT
116 . GLOBL CD$NAM, CD$DVU, CD$BAS, CD$JOB, CD$$SZ, CD$$UB
117 . GLOBL LTSCMD, LNSPAC, CFNEST, UCLNAM
118 . GLOBL $CFOPN, CFSEND, PBFEND, CFSP, $TTGAG
119 . GLOBL UFPTRP, SDSFCB, SD$DEL, CFLFL4, $UCLCF
120 . GLOBL SDFLAG, SD$FLK, SD$WFM, SDFORM, $UCLRN
121 . GLOBL SDBUF1, SDBLK, NSPLDV, LD$RON, $UCLCM, $UCLCL
122 . GLOBL LDNAME, LDSIZE, LDFLAG, LDBASE, LDPDEV
123 . GLOBL LSWB, $SQ01, $SQ01A, $SQ01B, $SQ01C, $SQ02, $SQIID, $SGHIO
124 . GLOBL $DEFER, CFCHAN, SCHAIN, LDDEVX, $SGALL
125 . GLOBL CFPNT, CFBLK, $QUIET, DIABFL
126 . GLOBL DIABNO, VT52NO, LA36NO, LA36FL
127 . GLOBL LSW4, KL4CLR, SDSKIP, SDBU, SD$BAK
128 . GLOBL $INCOR, $KED, VQUN1B, VINTIO, VQUN1C
129 . GLOBL SF$BSY, SFFORM, SD$SNG, SFNMBL, NFRESB
130 . GLOBL SD$HLD, SF$HLD, CURPRM, PRMPNT, SF$1ST
131 . GLOBL LSTPRM, PRMBUF, PRMEND, CFSPND
132 . GLOBL SDFHD, SFFLAG, SF$LNK, CFHOLD, LOGDVU, LOGBAS
133 . GLOBL LCOL, $QTSET, $TECO, CD$TOP, LOGCHK
134 . GLOBL $WILD, ERRSEV, UERSEV, PASLIN, LOGBAS, LOGDVU
135 . GLOBL LSTPL, SDCB, SDCBND, VQUANO, VQUAN3
136 . GLOBL VQUAN1, VQUN1A, VQUAN2, VHIPCT, VQUANO, VQUAN3
137 . GLOBL DCTRD, DCCRD, DCTWR, DCCWR, ASNSRC
138 . GLOBL VCORTM, NUMDCD, VNUMDC, KMPRMT, MXPRMT
139 . GLOBL RDB, RDBEND, RT$NAM, RT$$SZ, CLDEVX, SDDVU
140 . GLOBL SDNAME, SDCBSZ, LSTSL, LSTATE
141 . GLOBL TK1VAL, CINDAT, SYSDAT, SYTIMH, SYTIML
142 . GLOBL BASMAP, LOMAP, HIMAP, JCXPGS
143 . GLOBL SMRSIZ, SRTSIZ, CSHSIZ, TK1SEC
144 . GLOBL TSXLN, TSXSIT, GRT1, TRGRET, LICTXT, SUPCOD, NAMTOP, SUMS, SUCS
145 . GLOBL LPRG1, LPRG2, S$QUSR, S$IOWT, S$SFWT
146 . GLOBL S$SPDB, S$SPCB, SFUSER, SFFILE, VT200, VT2007, VT2008
147 . GLOBL LCBIT, LA36, LA120, VT52, VT100, DIABLD, QUME
148 . GLOBL ADM3A, LTRMTP, LA12FL, LA12NO, VT52FL
149 . GLOBL VT10FL, VT10NO, QUMEFL, QUMEND, ADM3FL
150 . GLOBL VT20FL, VT20NO
151 . GLOBL ADM3NO, SYINDX, SYUNIT, NUMDEV, PNAME
152 . GLOBL OF$DEV, OF$UNT, OF$FIL, OF$FLG, SYNAME
153 . GLOBL OF$$SZ, OT$RON, RESDEV, $TAPE
154 . GLOBL KMNBAS, ODTBAS, $CTRLD
155 . GLOBL LSW6, $SNWTT, PF$SYS, PF$IOW, $DEBUG
156 . GLOBL RSR, TSR, LMXNUM, LSTMX, MXDTR, ZCLR, MXCSR
157 . GLOBL $INDDF, $INDRN, IN$ACT, IN$CNT, IN$CMD, INDSAV
158 . GLOBL $PHONE, INVEC, LMXLN, MXVEC, $INIT, $DEAD, $HARD
159 . GLOBL ITRMTP, LMXPRM, LSW7, $INDAB, CFSTS, CF$IND, CF$QUT
160 . GLOBL CFABLV, MONVEC, LBSPRI, MAXPRI, MXJPRI, LPRI, $SYSPS
161 . GLOBL LOGCHN, LOGFLG, LOGPTR, LOGBUF, LOGBLK, LF$IN, LF$OUT
162 . GLOBL LF$OPN, LF$WRT, UCLBLK, UCLDAT
163 . GLOBL CSHHD, FC$CDX, FC$LNK, FD$NAM, UC$NDC, UC$MDC, CVTUC
164 . GLOBL CMDBUF, PAUMSG, RDCMD, DKSAV, SYSAV, CVTTAB, RUNHD, SEARCH
165 . GLOBL INVOPT, FKILL, ABRTCF, ACRFN, XAREA, FILNAM, NOPRG, FPRINT
166 . GLOBL PUSHCF, TRMSTR, FILNAM, R5ODIR, R5OSY, R5OIND, R5OSAV
167 . GLOBL INDACT, R5ODUP, R5OPIP, R5OKED, R5OK52, R5OKEX, R5OTSX, R5OUCL
168 . GLOBL BLKO, RDERM, R5OVIR, NOSTRT, RUNEMT, OVRCOR
169 . GLOBL BADSAV, LDNAM, NOPRG, NDCIN, SIZVAL, ASKLNM, BADCMD, KCSIBF
170 . GLOBL ASDEX, KCSIMS, ASNOVF, GTRD50, R5OBUF, R5OLD0, MNTDEV, DMTARG
171 . GLOBL DEADEV, CHKMNT, CHKMTX, INFOMT, NOFLAG, MTOPHD, INVOPT, ILLCMD

```

```

172 . GLOBL R5OLD, INVLDN, R5ODSK, ACRFIL, BDFNAM, LOGASN, MNTFUL, R5OLD7
173 . GLOBL TBLOVF, SETHD, CSIMS2, CKPRIV, R5OND, AMBOPT, ACRDEC
174 . GLOBL MAXAVL, PRTDEC, DEVUNT, PNAME, HNBUF, CKTERM
175 . GLOBL ACROCT, HANBSY, CSIMS1, MISSEQ, NOIND, POPCF
176 . GLOBL BADPMT, BADPRI, TOTXT, CRLF, HIPRI, STLGH, LOGCLS, R5OLOG
177 . GLOBL BDLGOP, SPLHLA, NOCCL, LDOPHD, PRTPFX, PRTSPC
178 . GLOBL DLTXT, OCTFIX, PRTPTP, NATXT, SPDTX1, NOTXT, YESTXT, NINTXT
179 . GLOBL PRTUNM, SYHD1, SYHD2, PRTLN, SPACE2, DETTXT, SPACE3, RNMS
180 . GLOBL SWPTX, LOCKTX, SPACE5, PRTDC3, KBMSG, DIVIDE, PRTDC2
181 . GLOBL COLOO, CPUAH, CPUAL, PRTPMV, NOFIL, CMDBUF, CALUCL
182 . GLOBL NOUDC, DEVHD1, ASNHD1, ASNHD2, SHMTH1, SHMTH2, PRTPMD
183 . GLOBL CVDVNM, SPACE6, PRTBUF, PRTPFM, NONEMS, NODAT, NOLDMT
184 . GLOBL SUBARD, EDTFIL, RONTXT, NOTAVL, KBTX, MNFLGS, MNBPC
185 . GLOBL DELSPC, MNBASE, MNTOP, MONHD, MONAR1, NOPMGN, PMBUSY, MONAR2
186 . GLOBL NSWPMS, MAXMTX, CURMTX, CHKDLN, SPLHD, AMBOPT, INVOPT
187 . GLOBL DEVIDL, COAL, ALDEX, COAD, SPACTV, SPWFM, DEVIDL, SPSNG
188 . GLOBL COAL, ALDEX, ALDBLK, COAD, SPACTV, SPWFM, DEVIDL
189 . GLOBL SPSNG, SPFUL, SPCF, SPFLK, NOFIL, SPGEMT, NOOPTT
190 . GLOBL BDLIN, MSGBUF, MSGEND, NOTON, GAGMSG
191 . GLOBL LINFRE, DJABMS, DLMSG, INVTIM, DMTALL
192 . GLOBL SHTMSG, AUTHFN, SPLACT, DOSTOP, OFFEMT, KILEMT, UPTMMS
193 . GLOBL TMTOTH, DIVSOR, TMTOTL, PRTPCT, SUM1, SUM2, SUM3, SUM4
194 . GLOBL SUM5, SUM6, SUM7, OTHRON, SPLPND, STPASK, SRTSMS, CHKTDD
195 . GLOBL SIZEMT, ASNOVF, INVLDN, CSIMS4, MNTARG, HUPARG, R5OTT
196 . GLOBL KMNNAM, NOKMON, CCLNAM, OTRMNT, CHKDEV, DMTSUB, CMDCCCL
197 . GLOBL SHOHD, SUBTXT, MNTTXT, SRTTXT, TOTMMS, UMSSMS, SSRMAP
198 . GLOBL TSXSMS, USRMMS, JCXSMS, DZTXT, OCTPRT
199 . GLOBL PRTR50, PRTPAT, PRTPOD, PRTPIM, INVDEV, ALFN, R5ODK
200 . GLOBL DETHD, DETARG, RUNMS, NOFRDL, R5OMON, INV DAT, MUL32, COAF
201 . GLOBL AR$PRJ, AR$PRG, AR$CON, AR$CNT, AR$CPH, AR$CPL, AR$UNM
202 . GLOBL AR$DMY, AR$SZ, ARNRPB, $SLON, $SLTTY, $SLLET
203 . GLOBL PRTWRN, SLMXLN, VSLEDT, $LOFCF, CSHMSG
204 . GLOBL AF$HIE, AF$NOI, $NOINT, AF$PLK, AF$DBG
205 . GLOBL AF$IOP, $RNIOP, VOFFTM, VTMOUT

```

```
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 CMDVS1
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 NAMES1
49     L =
50     .ASCIZ /STRNG/
51     .CSECT CMDVS1
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 CMDVSI  
63 .WORD 0  
64 .CSECT TSKST1  
65 .ENDM TBLEND
```

```

1          ; -----
2          ; Define options for the SET command.
3          ;
4          ; Table of "device" names for the SET command and a
5          ; pointer to a set of sub-options for the device.
6          ;
7          ;
8 000000   TBLDEF  SET, 3
9 000002   CMDDEF  CA*CHE, STCHNB, 0, 0
10 000012  CMDDEF  CC*L, SETSUB, CCLHD, 0
11 000022  CMDDEF  CO*RTIM, SETSVL, VCORTM, 10000.
12 000032  CMDDEF  CTRLD, SETSUB, CTDHD, 0
13 000042  CMDDEF  ED*IT, SETSUB, EDITHD, 0
14 000052  CMDDEF  EM*T, SETSUB, EMTHD, 0
15 000062  CMDDEF  ENDS*TARTUP, SETESU, 0, 0
16 000072  CMDDEF  ER*ROR, SETSUB, ERRHD, 0
17 000102  CMDDEF  EX*IT, RDCMD, 0, 0
18 000112  CMDDEF  HIP*RCT, SETSVL, VHIPCT, 10000.
19 000122  CMDDEF  HO*ST, SETHST, 0, 0
20 000132  CMDDEF  IND, SETSUB, INDHD, 0
21 000142  CMDDEF  INT*IOC, SETSVL, VINTIO, 10000.
22 000152  CMDDEF  IO*ABORT, SETPRV, IOABHD, 0
23 000162  CMDDEF  KMO*N, SETSUB, KMONHD, 0
24 000172  CMDDEF  LAN*GUAGE, SETSUB, LANGHD, 0
25 000202  CMDDEF  LOG, SETLOG, 0, 0
26 000212  CMDDEF  LOGO*FF, SETLOF, 0, 0
27 000222  CMDDEF  MAX*PRIORITY, SETMPR, 0, 0
28 000232  CMDDEF  NUM*DC, SETNDC, NUMDCD, 0
29 000242  CMDDEF  OFF*TIM, SETSVL, VOFFTM, 10000.
30 000252  CMDDEF  PROC*ESS, SETPRC, 0, 0
31 000262  CMDDEF  PRGM*PT, SETPRT, 0, 0
32 000272  CMDDEF  PRI*ORITY, SETPRI, 0, 0
33 000302  CMDDEF  PRINTS*CREEN, SETPRS, 0, 0
34 000312  CMDDEF  PRINTW*INDOW, SETPRS, 0, 0
35 000322  CMDDEF  QUANO, SETSVL, VQUANO, 10000.
36 000332  CMDDEF  QUAN1, SETSVL, VQUAN1, 10000.
37 000342  CMDDEF  QUAN1A, SETSVL, VQUN1A, 10000.
38 000352  CMDDEF  QUAN1B, SETSVL, VQUN1B, 10000.
39 000362  CMDDEF  QUAN1C, SETSVL, VQUN1C, 10000.
40 000372  CMDDEF  QUAN2, SETSVL, VQUAN2, 10000.
41 000402  CMDDEF  QUAN3, SETSVL, VQUAN3, 10000.
42 000412  CMDDEF  REC*ALL, SETREC, 0, 0
43 000422  CMDDEF  SHUT*DOWN, SETSHT, 0, 0
44 000432  CMDDEF  NOSHUT*DOWN, SETNSH, 0, 0
45 000442  CMDDEF  SIG*NAL, SETSUB, SIGHD, 0
46 000452  CMDDEF  SL, SETSL, 0, 0
47 000462  CMDDEF  SUB*PROCESS, SETSBP, 0, 0
48 000472  CMDDEF  SYSP*ASSWORD, SETSYP, 0, 0
49 000502  CMDDEF  SYSP*WD, SETSYP, 0, 0
50 000512  CMDDEF  TE*RMINAL, SETTTY, 0, 0
51 000522  CMDDEF  TT*Y, SETTTY, 0, 0
52 000532  CMDDEF  TIM*OUT, SETSVL, VTMOUT, 10000.
53 000542  CMDDEF  UC*L, SETSUB, UCLHD, 0
54 000552  CMDDEF  US*R, RDCMD, 0, 0
55 000562  CMDDEF  VER*IFY, STVRFY, 0, 0
56 000572  CMDDEF  NOVER*IFY, STNOVR, 0, 0
57 000602  CMDDEF  WI*LDCARDS, SETSUB, WILDHD, 0

```

58 000612
59 000622
60 000632

CMDDEF WIN*DOWS, SETWIN, 0, 0
CMDDEF WINP*RT, SETPRS, 0, 0
TBLEND

```

1      ;
2      ; Define options for SET EDIT command
3      ;
4 000000      TBLDEF  EDIT, 3
5 000636      CMDDEF  ED*IT, SEEDIT, 0
6 000646      CMDDEF  KED, SEKED, 0
7 000656      CMDDEF  K52, SEK52, 0
8 000666      CMDDEF  T*ECO, SETECO, 0
9 000676      TBLEND
10     ;
11     ; Define options for SET ERROR command
12     ;
13 000000      TBLDEF  ERR, 2
14 000702      CMDDEF  W*ARNING, SETSEV, SC$WRN
15 000710      CMDDEF  E*RROR, SETSEV, SC$ERR
16 000716      CMDDEF  S*EVERE, SETSEV, SC$SEV
17 000724      CMDDEF  F*ATAL, SETSEV, SC$FTL
18 000732      CMDDEF  U*NCONDITIONAL, SETSEV, SC$UNC
19 000740      CMDDEF  N*ONE, SETSEV, SC$NON
20 000746      TBLEND
21     ;
22     ; Define options for SET WILDCARDS command
23     ;
24 000000      TBLDEF  WILD, 3
25 000752      CMDDEF  IM*PLICIT, SETBIT, LSW5, $WILD
26 000762      CMDDEF  EX*PLICIT, CLRBIT, LSW5, $WILD
27 000772      TBLEND
28     ;
29     ; Define options for the SET KMON command
30     ;
31 000000      TBLDEF  KMON, 3
32 000776      CMDDEF  IND, SETIND, LSW5, $INDDF
33 001006      CMDDEF  NOI*ND, RSTIND, LSW5, $INDDF
34 001016      CMDDEF  SY*STEM, CLRBIT, LSW7, $UKMON
35 001026      CMDDEF  UCI, SETUKM, 0, 0
36 001036      CMDDEF  USE*R, SETUKM, 0, 0
37 001046      CMDDEF  NEW, SETKNW, 0, 0
38 001056      CMDDEF  DEBUG, SETBIT, LSW9, $DBKMN
39 001066      CMDDEF  NODEBUG, CLRBIT, LSW9, $DBKMN
40 001076      TBLEND
41     ;
42     ; Define options for the SET CCL command
43     ;
44 000000      TBLDEF  CCL, 3
45 001102      CMDDEF  T*EST, SETBIT, LSW5, $CLTST
46 001112      CMDDEF  NOT*EST, CLRBIT, LSW5, $CLTST
47 001122      CMDDEF  NEW, SETCNW, 0, 0
48 001132      TBLEND
49     ;
50     ; Define options for the SET UCL command
51     ;
52 000000      TBLDEF  UCL, 3
53 001136      CMDDEF  F*IRST, STUCLF, LSW7, $UCLCF
54 001146      CMDDEF  M*IDDLE, STUCLM, LSW7, $UCLCM
55 001156      CMDDEF  L*AST, STUCLL, LSW7, $UCLCL
56 001166      CMDDEF  N*ONE, STUCLN, LSW7, 0
57 001176      TBLEND

```

```

58      ;
59      ; Define options for the SET IND command
60      ;
61      TBLDEF  IND, 3
62      CMDDEF  AB*ORT, SFTBIT, LSW7, $INDAB
63      CMDDEF  NOA*BORT, CLRBIT, LSW7, $INDAB
64      TBLEND
65      ;
66      ; Define options for the SET LD command
67      ;
68      TBLDEF  LDOP, 1
69      CMDDEF  CL*EAN, SLDCLN
70      CMDDEF  EMP*TY, SLDEMP
71      CMDDEF  FR*EE, SLDFRE
72      CMDDEF  WR*ITE, SLDWRT
73      CMDDEF  NOWR*ITE, SLDNWR
74      TBLEND
75      ;
76      ; Define options for the SET SL command
77      ;
78      TBLDEF  SLE, 1
79      CMDDEF  AS*K, SLONOP
80      CMDDEF  KED, SLOKED
81      CMDDEF  K52, SLOKED
82      CMDDEF  KEX, SLOKED
83      CMDDEF  NOKED, SLORT
84      CMDDEF  LEA*RN, SLOUNI
85      CMDDEF  NOLEA*RN, SLONOP
86      CMDDEF  LET, SLOLET
87      CMDDEF  NOLET, SLONLT
88      CMDDEF  OF*F, SLOOFF
89      CMDDEF  ON, SLOON
90      CMDDEF  KM*ON, SLOON
91      CMDDEF  RT*11, SLORT
92      CMDDEF  SUB*STITUTE, SLOLET
93      CMDDEF  NOSUB*STITUTE, SLONLT
94      CMDDEF  SYS*GEN, SLONOP
95      CMDDEF  TT*YIN, SLOTTY
96      CMDDEF  NOTT*YIN, SLONTT
97      CMDDEF  WID*TH, SLOWID
98      CMDDEF  REC*ALL, SLONOP
99      CMDDEF  NOREC*ALL, SLONOP
100     TBLEND
101     ;
102     ; Define options for the SET RECALL
103     ;
104     TBLDEF  RCL, 1
105     CMDDEF  N*ORMAL, 0
106     CMDDEF  R*EVERSE, 1
107     TBLEND
108     ;
109     ; Define options for the SET WINDOW command
110     ;
111     TBLDEF  WIN, 2
112     CMDDEF  COL*UMNS, WINCOL, 0
113     CMDDEF  DA*RK, WINBIT, WFNORM
114     CMDDEF  DEL*ETE, WINBIT, WFDEL

```

```

115 001444          CMDDEF  LI*GHT, WINBIT, WFREV
116 001452          CMDDEF  NAR*ROW, WINBIT, WFBO
117 001460          CMDDEF  NOR*MAL, WINBIT, WFNORM
118 001466          CMDDEF  OFF, WINBIT, WFDEL
119 001474          CMDDEF  ON, WINBIT, O
120 001502          CMDDEF  REV*ERSE, WINBIT, WFREV
121 001510          CMDDEF  SCR*OLL, WINSR, O
122 001516          CMDDEF  NOSCR*OLL, WINNOS
123 001524          CMDDEF  WI*DE, WINBIT, WF132
124 001532          TBLEND
125                ;
126                ; Define options for the SET PRINTWINDOW command
127                ;
128 000000          TBLDEF  PRS, 1
129 001536          CMDDEF  BEL*L, PRSBEL
130 001542          CMDDEF  NOBEL*L, PRSNBL
131 001546          CMDDEF  DEV*ICE, PRSDEV
132 001552          CMDDEF  DR*AFT, PRSDRF
133 001556          CMDDEF  KEY*PRINT, PRSKEY
134 001562          CMDDEF  NOKEY*PRINT, PRSNKY
135 001566          CMDDEF  LET*TERQUALITY, PRSLET
136 001572          CMDDEF  TY*PE, PRSTYP
137 001576          CMDDEF  WID*TH, PRSWID
138 001602          CMDDEF  NOWID*TH, PRSNWD
139 001606          TBLEND
140                ;
141                ; Printer types for SET PRINTWINDOW/TYPE=type
142                ;
143 000000          TBLDEF  PWT, 2
144 001612          CMDDEF  AS*CI1, 0, 0
145 001620          CMDDEF  FOR*EIGN, 0, 0
146 001626          CMDDEF  PRINT*RONIX, 0, 0
147 001634          CMDDEF  SIM*PLE, 0, 0
148 001642          CMDDEF  ST*ANDARD, 0, 0
149 001650          CMDDEF  LA36, 0, 0
150 001656          CMDDEF  LA120, 0, 0
151 001664          CMDDEF  LA100, 1, PA$GRC!PA$UKC!PA$ULN!PA$HQL!PA$DWD
152 001672          CMDDEF  LA12, 1, PA$GRC!PA$UKC!PA$BLD!PA$ULN!PA$HQL!PA$DWD
153 001700          CMDDEF  LA50, 1, PA$GRC!PA$UKC!PA$DSC!PA$BLD!PA$ULN!PA$HQL!PA$DWD
154 001706          CMDDEF  LA210, 1, PA$GRC!PA$UKC!PA$DSC!PA$BLD!PA$ULN!PA$HQL!PA$DWD
155 001714          CMDDEF  LQPO2, 1, PA$BLD!PA$ULN
156 001722          CMDDEF  LQPO3, 1, PA$BLD!PA$ULN
157 001730          CMDDEF  LN03, 1, PA$GRC!PA$BLD!PA$ULN!PA$DWD
158 001736          TBLEND
159                ;
160                ; Define options for the SET LANGUAGE command
161                ;
162 000000          TBLDEF  LANG, 3
163 001742          CMDDEF  DI*BOL, SETBIT, LSW6, $DIBOL
164 001752          CMDDEF  DBL, CLRBIT, LSW6, $DIBOL
165 001762          TBLEND
166                ;
167                ; Define options for the SET EMT command
168                ;
169 000000          TBLDEF  EMT, 3
170 001766          CMDDEF  TR*ACE, SETENT, LSW6, $EMTTR
171 001776          CMDDEF  NOTR*ACE, CLRBIT, LSW6, $EMTTR

```

```

172 002006          TBLEND
173                ;
174                ; Define options for the SET IOABORT command
175                ;
176 000000          TBLDEF  IOAB,3
177 002012          CMDDEF  AB*ORT,STRWRD,IOABFL,1
178 002022          CMDDEF  NOAB*ORT,STRWRD,IOABFL,0
179 002032          TBLEND
180                ;
181                ; Define options for the SET SIGNAL command
182                ;
183 000000          TBLDEF  SIG,3
184 002036          CMDDEF  H*IPRCT,SETBIT,LSWB,$SGHID
185 002046          CMDDEF  NOH*IPRCT,CLRBIT,LSWB,$SGHID
186 002056          CMDDEF  I*NTIOC,SETBIT,LSWB,$SGIID
187 002066          CMDDEF  NOI*NTIOC,CLRBIT,LSWB,$SGIID
188 002076          CMDDEF  QUANO,SETBIT,LSWB,$SGQ0
189 002106          CMDDEF  NOQUANO,CLRBIT,LSWB,$SGQ0
190 002116          CMDDEF  QUAN1,SETBIT,LSWB,$SGQ1
191 002126          CMDDEF  NOQUAN1,CLRBIT,LSWB,$SGQ1
192 002136          CMDDEF  QUAN1A,SETBIT,LSWB,$SGQ1A
193 002146          CMDDEF  NOQUAN1A,CLRBIT,LSWB,$SGQ1A
194 002156          CMDDEF  QUAN1B,SETBIT,LSWB,$SGQ1B
195 002166          CMDDEF  NOQUAN1B,CLRBIT,LSWB,$SGQ1B
196 002176          CMDDEF  QUAN1C,SETBIT,LSWB,$SGQ1C
197 002206          CMDDEF  NOQUAN1C,CLRBIT,LSWB,$SGQ1C
198 002216          CMDDEF  QUAN2,SETBIT,LSWB,$SGQ2
199 002226          CMDDEF  NOQUAN2,CLRBIT,LSWB,$SGQ2
200 002236          CMDDEF  QUAN3,SETBIT,LSWB,$SGQ3
201 002246          CMDDEF  NOQUAN3,CLRBIT,LSWB,$SGQ3
202 002256          CMDDEF  OFF,CLRBIT,LSWB,$SCALL
203 002266          TBLEND
204                ;
205                ; Define options for the SET LOG command
206                ;
207 000000          TBLDEF  STLO,1
208 002272          CMDDEF  A*LL,STLOAL
209 002276          CMDDEF  CLE*AN,STLOCN
210 002302          CMDDEF  CLO*SED,STLOCL
211 002306          CMDDEF  FI*LE,STLOFL
212 002312          CMDDEF  I*NPOT,STLOIN
213 002316          CMDDEF  O*UTPUT,STLOOT
214 002322          CMDDEF  WR*ITE,STLOWR
215 002326          CMDDEF  NOWR*ITE,STLOWW
216 002332          TBLEND
217                ;
218                ; Define options for the SET LOGOFF command
219                ;
220 000000          TBLDEF  STLO,1
221 002336          CMDDEF  FI*LE,STLOFL
222 002342          TBLEND
223                ;
224                ; Define options for the SET SUBPROCESS command
225                ;
226 000000          TBLDEF  SBP,1
227 002346          CMDDEF  FI*LE,SBPFIL
228 002352          TBLEND

```

```
229      ;  
230      ; Define options for the SET CTRLD command  
231      ;  
232 000000      TBLDEF   CTD, 1  
233 002356      CMDDEF   D*EBUG, STCDUN  
234 002362      CMDDEF   NOD*EBUG, STCDF  
235 002366      TBLEND
```

```

1 ; -----
2 ; Data areas
3 ;
4 000000 HANSAV: .BLKW 5 ;Space for .SAVESTATUS
5 000012 012276 R5OCLO: .RAD50 /CLO/
6 000014 013666 R5OC10: .RAD50 /C10/
7 000016 000000 WINFLG: .WORD 0
8 ;
9 ; Flags stored in WINFLG
10 ;
11 000001 WFDEL = 1 ;Delete the window
12 000002 WPREV = 2 ;Put terminal in reverse video
13 000004 WFNORM = 4 ;Put terminal in normal video mode
14 000010 WF132 = 10 ;Put terminal in 132 column mode
15 000020 WF80 = 20 ;Put terminal in 80 column mode
16 ;
17 ; Terminal control sequences
18 ;
19 000020 033 133 077 TCREV: .ASCII <33>/[?5h/<200> ;Select reverse video display
20 000023 065 150 200
21 000026 033 133 077 TCNORM: .ASCII <33>/[?5l/<200> ;Select normal video display
22 000031 065 154 200
23 000034 033 133 077 TC132: .ASCII <33>/[?3h/<200> ;Select 132 column mode
24 000037 065 150 200
25 000042 033 133 077 TC80: .ASCII <33>/[?3l/<200> ;Select 80 column mode
26 000045 065 154 200
27 ; .EVEN
28 ;
29 ; Emt to set line speed
30 ;
31 000050 000 154 LSPEMT: .BYTE 0,154
32 000052 000000 .WORD 0 ;Line number
33 000054 000000 .WORD 0 ;Speed code
34 ;
35 ; Emt to reinitialize data caching
36 ;
37 000056 000 126 DCHNEW: .BYTE 0,126
38 000060 000006 .WORD 6
39 ;
40 ; Emt to set job priority
41 ;
42 000062 000 150 PRIEMT: .BYTE 0,150
43 000064 000000 .WORD 0
44 ;
45 ; Emt to create window # 1
46 ;
47 000066 000 161 WINMAK: .BYTE 0,161
48 000070 001 001 .BYTE 1,1
49 000072 120 020 .BYTE 30,16
50 000074 000000 000000 .WORD 0,0
51 ;
52 ; Emt to map to window # 1
53 ;
54 000100 001 161 WINMAP: .BYTE 1,161
55 000102 001 000 .BYTE 1,0
56 ;
57 ; Emt to delete window # 1

```

54				;
55	000104	007	161	WINDEL: .BYTE 2,161
56	000106	001	000	.BYTE 1,0
57				;
58				; Emt to dismount all LD's
59				;
60	000110	005	135	DMTALD: .BYTE 5,135
61	000112	000000		.WORD 0

SET command

```

1          .SBTTL  SET command
2          ;-----
3          ; Process the "SET" command.
4          ;
5 000114  004767  0000000  CMDSET: CALL    CVTTAB      ; CONVERT TAB AND FF CHARS TO SPACES
6 000120  004767  0000000          CALL    SKPSPC      ; Skip up to start of device name
7 000124  010305          MOV     R3,R5        ; SAVE COMMAND STRING POINTER
8 000126  105067  0000000          CLRB   NOFLAG      ; ASSUME "NO" OPTION NOT SEEN YET
9          ;
10         ; If device name ends with a colon (e.g., "TT:") replace the colon
11         ; with a space.
12         ;
13 000132  112300          7$:  MOV     (R3)+,R0    ; Get next char of device name
14 000134  120027  000040          CMPB   RO,#40       ; Reached end of name?
15 000140  101410          BLOS   B$           ; Br if yes
16 000142  120027  000057          CMPB   RO,#'/'      ; Start of qualifiers?
17 000146  001405          BEQ    B$           ; Br if yes
18 000150  120027  000072          CMPB   RO,#':'      ; Is this character colon?
19 000154  001365          BNE    7$          ; Br if not
20 000156  112743  000040          MOV     #40,-(R3)   ; Replace colon with space
21 000162  010503          B$:  MOV     R5,R3    ; Get back pointer to start of device name
22         ;
23         ; Look up device name and see if we recognize it as a pseudo-device
24         ; such as TT, WILDCARD, etc.
25         ;
26 000164  012704  000000'  MOV     #SETHD,R4    ; GET TABLE OF PSEUDO-DEVICE NAMES
27 000170  004767  0000000  CALL    SEARCH       ; SEARCH FOR DEVICE NAME
28 000174  103401          BCS    1$           ; BR IF NOT A PSEUDO-DEVICE NAME
29         ;
30         ; This is a pseudo-device name.
31         ; Jump off to appropriate processing routine.
32         ;
33 000176  000134          JMP     @(R4)+       ; JUMP OFF TO PROCESSING ROUTINE
34         ;
35         ; Device name is not recognized as a pseudo device.
36         ; Apply any logical device assignment to the device name.
37         ;
38 000200  010503          1$:  MOV     R5,R3    ; POINT BACK TO DEVICE NAME
39 000202  005067  0000020  CLR     R50BUF+2     ; CLEAR 2ND WORD OF RAD50 NAME
40 000206  004767  0000000  CALL    @TRD50      ; ACCRUE RAD50 VALUE
41 000212  016700  0000000  MOV     R50BUF,R0   ; GET DEVICE NAME
42 000216  005767  0000020  TST    R50BUF+2     ; MAKE SURE NAME ONLY 3 CHARS LONG
43 000222  001404          BEQ    2$           ; BR IF OK
44 000224          FABORT #CSIMS2    ; INVALID DEVICE FOR SET
45 000234  004767  0000000  2$:  CALL    ASNSRC     ; Search for device name in assign table
46 000240  103404          BCS    6$           ; Br if name was not assigned
47 000242  016200  0000000  MOV     AT$DEV(R2),R0 ; Get the physical device name
48 000246  010067  0000000  MOV     R0,R50BUF   ; Set new device name for SET
49         ;
50         ; See if this is a SET LDn command
51         ;
52 000252  020067  0000000  6$:  CMP     RO,R50LD    ; SEE IF IT IS LD
53 000256  103410          BLD    3$           ; BR IF NOT
54 000260  020067  0000000  CMP     RO,R50LD7   ; CHECK HIGH RANGE
55 000264  101005          BHI    3$           ;
56 000266  105767  0000000  TSTB   VLDSYS      ; IS LD SUPPORT GENNED IN?
57 000272  001415          BEQ    5$           ; BR IF NOT

```

SET command

```

58 000274 000167 005412          JMP      SETLD          ;GO PROCESS SET LD COMMAND
59                               ;
60                               ; See if this device is allocated to another user
61                               ;
62 000300 004767 0000000        3$:     CALL      CHKALC          ;Check for allocation conflict
63                               ;
64                               ; See if this is a SET CLn command
65                               ;
66 000304 005727 0000000        TST      #CLTOTL          ;Are there any CL lines?
67 000310 001406                BEQ      5$              ;Br if not
68 000312 004767 0000000        CALL     CHKCLU          ;See if this is a CL or C1 unit
69 000316 103403                BCS     5$              ;Br if not CL or C1 unit
70 000320 010002                MOV     R0,R2           ;Pass CL unit number in R2
71 000322 000167 0000000        JMP     SETCL           ;Process SET CL command
72 000326 000167 000446        5$:     JMP     SETDEV

```

SET command

```

1          ; -----
2          ; Process SET commands that have sub-options
3          ;
4          ; Inputs:
5          ; R3 = Pointer to start of qualifier in command line.
6          ; R4 = Points to start of command table built with TBLDEF...TBLEND macros.
7          ;
8          ; Outputs:
9          ; R3 = Pointer beyond end of qualifier.
10         ; R4 = Pointer to argument value word in command table.
11         ;
12         ; See if set option is preceded by "NO".
13         ;
14 000332 004767 0000000 SETPRV: CALL CKSYPV          ;Must have SYSPRV privilege
15 000336 010246 SETSUB: MOV R2,-(SP)
16 000340 010546      MOV R5,-(SP)
17 000342 122327 000072      CMPB (R3)+,#':          ;WAS DEVICE NAME FOLLOWED BY COLON?
18 000346 001401      BEQ 3$          ;BR IF YES
19 000350 005303      DEC R3          ;RESET POINTER IF NOT
20 000352 004767 0000000 3$: CALL SKPSPC          ;SKIP OVER SPACES
21 000356 111300      MOVB (R3),R0          ;Get next character from command
22 000360 120027 000075      CMPB R0,#'='          ;Is there an equal sign?
23 000364 001403      BEQ 5$          ;Br if yes -- skip over equal sign
24 000366 120027 000072      CMPB R0,#':          ;Colon separator?
25 000372 001001      BNE 6$          ;Br if not
26 000374 005203      5$: INC R3          ;Skip over separator
27 000376 010302      6$: MOV R3,R2          ;SAVE COMMAND STRING POINTER
28 000400 004767 0000000      CALL @TRD50          ;ACCRUE NEXT WORD IN RAD50 FORM
29 000404 026767 0000000 0000000      CMP R50BUF,R50ND          ;IS WORD "NO"?
30 000412 001005      BNE 1$          ;BR IF NOT
31         ;
32         ; Option is preceded by NO. Concatenate NO with following qualifier.
33         ;
34 000414 010305      MOV R3,R5          ;GET POINTER INTO COMMAND PAST "NO"
35 000416 004767 0000000      CALL SKPSPC          ;SKIP OVER SPACES FOLLOWING "NO"
36 000422 112325      4$: MOVB (R3)+,(R5)+          ;CONCATENATE OPTION WORD WITH "NO"
37 000424 001376      BNE 4$          ;MOVE ALL OF COMMAND
38 000426 010203      1$: MOV R2,R3          ;GET BACK POINTER TO OPTION WORD
39         ;
40         ; Look up the option word.
41         ;
42 000430 011404      MOV @R4,R4          ;GET POINTER TO TABLE OF OPTIONS FOR COMMAND
43 000432 004767 0000000      CALL SEARCH          ;LOOK FOR THE CORRECT OPTION WORD
44 000436 103403      BCS BDSO          ;BR IF INVALID OPTION
45         ;
46         ; Enter Option processing routine.
47         ;
48 000440 012605      MOV (SP)+,R5
49 000442 012602      MOV (SP)+,R2
50 000444 000134      JMP @(R4)+          ;PERFORM PROCESSING FOR OPTION
51         ;
52         ; Invalid option.
53         ;
54 000446 005704      BDSO: TST R4          ;INVALID OR AMBIGUOUS OPTION?
55 000450 001407      BEQ INVSOP          ;BR IF INVALID
56 000452      FABORT #AMBOPT          ;AMBIGUOUS OPTION
57 000462      INVLDO: .PURGE #HANCHN          ;MAKE SURE THE CHANNEL IS CLOSED

```

5B 000470

INVSOP: FABORT #INVOPT ; INVALID OPTION

System parameters

```

1          .SBTTL      System parameters
2          ;-----
3          ; Set a system parameter value.
4          ;
5          ; Inputs:
6          ;   R4 Points to cell with address of parameter cell to be modified,
7          ;   following word contains maximum legal value for parameter.
8          ;
9 000500 004767 0000000 SETSVL: CALL   CKSYPV           ;Must have SYSPRV privilege
10         ;
11         ; Accrue numeric value
12         ;
13 000504 004767 0000000         CALL   ACRDEC           ;ACCRUE A DECIMAL VALUE
14 000510 016400 0000002         MOV    2(R4),R0        ;GET MAX LEGAL VALUE FOR PARAMETER
15         ;
16         ; Make sure the value is valid.
17         ; R1 = Accrued value, R0 = Maximum legal value.
18         ;
19 000514 005701 SETSVV: TST    R1             ;CHECK FOR NEGATIVE VALUE SPECIFIED
20 000516 002402         BLT    3$             ;BR IF INVALID VALUE SPECIFIED
21 000520 020100         CMP    R1,R0          ;IS SPECIFIED VALUE OK?
22 000522 101413         BLOS   1$             ;BR IF OK
23 000524 010001 3$:      MOV    R0,R1          ;GET MAX LEGAL VALUE
24 000526 010005         MOV    R0,R5
25 000530         .PRINT #MAXAVL           ;VALUE TOO LARGE
26 000536 004767 0000000         CALL   PRTDEC           ;PRINT MAX LEGAL VALUE
27 000542         .PRINT #CRLF           ;END LINE
28 000550 000401         BR     4$             ;FINISHED
29         ;
30         ; Store value into parameter cell
31         ;
32 000552 010134 1$:      MOV    R1,@(R4)+      ;STORE NEW VALUE INTO PARAMETER CELL
33 000554 000167 0000000 4$:      JMP    RDCMD          ;FINISHED
34         ;
35         ; SET NUMDC value
36         ;
37 000560 004767 0000000 SETNDC: CALL   CKSYPV           ;Must have SYSPRV privilege
38         ;
39         ; Accrue numeric value
40         ;
41 000564 004767 0000000         CALL   ACRDEC           ;ACCRUE A DECIMAL VALUE
42 000570 016700 0000000         MOV    VNUMDC,R0        ;GET MAX LEGAL VALUE FOR PARAMETER
43 000574 000747         BR     SETSVV          ;Check and store value

```

System parameters

```

1          ;
2          ; SET CACHE = value
3          ;
4 000576 004767 0000000 STCHNB: CALL  CKSYPV          ;Must have SYSPRV privilege
5          ;
6          ; Accrue numeric value
7          ;
8 000602 004767 0000000          CALL  ACRDEC          ;ACCRUE A DECIMAL VALUE
9 000606 016700 0000000          MOV   CSHALC,RO        ;GET MAX LEGAL VALUE FOR PARAMETER
10 000612 005701          TST   R1              ;CHECK FOR NEGATIVE VALUE SPECIFIED
11 000614 002402          BLT   3$              ;BR IF INVALID VALUE SPECIFIED
12 000616 020100          CMP   R1,RO          ;IS SPECIFIED VALUE OK?
13 000620 101412          BLOS  1$              ;BR IF OK
14 000622 010001          3$:  MOV   RO,R1          ;GET MAX LEGAL VALUE
15 000624 010005          MOV   RO,R5
16 000626          .PRINT #MAXAVL          ;VALUE TOO LARGE
17 000634 004767 0000000          CALL  PRTDEC          ;PRINT MAX LEGAL VALUE
18 000640          .PRINT #CRLF          ;END LINE
19          ;
20          ; Store value into parameter cell
21          ;
22 000646 010167 0000000          1$:  MOV   R1,VCSHNB          ;STORE NEW VALUE INTO PARAMETER CELL
23          ;
24          ; Now clean out the data cache
25          ;
26 000652 012700 000056'          MOV   #DCHNEW,RO        ;EMT to clean out the cache
27 000656 104370          EMT   375
28 000660 000167 0000000          JMP   RDCMD          ;FINISHED
29          ;
30          ; SET CTRLD [NO]DEBUG
31          ;
32 000664 105767 0000000          STCDON: TSTB  VDBFLG          ;Was debugger genned into the system?
33 000670 001004          BNE   1$              ;Br if yes
34 000672          FABORT #EM#NPD          ;No program debugger
35 000702 032767 0000000 0000000 1$:  BIT   #PO#DBG,PRIVCO        ;Can we use debugger?
36 000710 001004          BNE   2$              ;Br if yes
37 000712          FABORT #EM#NAD          ;Not allowed to use debugger
38 000722 052761 0000000 0000000 2$:  BIS   ##CTRLD,LSW9(R1);Enable ctrl-D debugger entry
39 000730 000167 0000000          JMP   RDCMD
40 000734 042761 0000000 0000000 STCDOF: BIC   ##CTRLD,LSW9(R1);Disable ctrl-D debugger entry
41 000742 000167 0000000          JMP   RDCMD
42          ;
43          ; SET EMT TRACE
44          ;
45 000746 032767 0000000 0000000 SETEMT: BIT   #PO#DBG,PRIVCO        ;Are we allowed to use debugger?
46 000754 001004          BNE   1$              ;Br if yes
47 000756          FABORT #EM#NAD          ;Not allowed to use debugger
48 000766 052761 0000000 0000000 1$:  BIS   ##EMTTR,LSW6(R1);Enable EMT trace
49 000774 000167 0000000          JMP   RDCMD

```

Real device

```

1          .SBTTL      Real device
2          ;-----
3          ; The device name is not a pseudo-device.
4          ; See if it is the name of a real device.
5          ;
6 001000 016701 0000000 SETDEV: MOV      R50BUF,R1      ;GET THE DEVICE NAME AND UNIT
7 001004 005000          CLR      RO              ;CLEAR HIGH ORDER REGISTER
8 001006 071027 000050          DIV     #50,RO      ;STRIP THE LAST RAD 50 CHARACTER
9 001012 010146          MOV      R1,-(SP)        ;SAVE THE UNIT NUMBER
10 001014 070027 000050          MUL     #50,RO      ;GET THE DEVICE NAME SHIFTED
11 001020 010167 0000000          MOV     R1,R50BUF      ;AND STORE BACK INTO THE RAD 50 BUFFER
12 001024 012601          MOV     (SP)+,R1      ;GET THE UNIT NUMBER
13 001026 001407          BEQ     2$           ;BR IF NO UNIT NUMBER WAS SPECIFIED
14 001030 162701 000036          SUB     #36,R1      ;NORMALIZE TO ZERO
15 001034 002450          BLT     SETIVD      ;ERROR IF LESS THAN ZERO
16 001036 020127 000007          CMP     R1,#7      ;CHECK UNIT RANGE
17 001042 003045          BGT     SETIVD      ;ERROR IF GREATER THAN SEVEN
18 001044 000407          BR      3$           ;CONTINUE
19 001046 012701 100000          2$:  MOV     #100000,R1 ;SET NO UNIT WAS SPECIFIED FLAG
20 001052 010167 0000000          3$:  MOV     R1,DEVUNT    ;SAVE THE UNIT SPECIFIED
21 001056 122327 000072          CMPB   (R3)+,#':'    ;WAS DEVICE NAME FOLLOWED BY COLON?
22 001062 001401          BEQ     4$           ;BR IF YES
23 001064 005303          DEC     R3          ;BACKUP POINTER
24          ;
25          ; Lookup the handler file and read in the first 2 blocks.
26          ;
27 001066 016767 0000000 0000020 4$:  MOV     R50BUF,HNBUFF+2 ;SET UP HANDLER NAME
28 001074          .LOOKUP #XAREA,#HANCHN,#HNBUFF;TRY TO LOOKUP HANDLER FILE
29 001114 103420          BCS     SETIVD      ;BR IF CAN'T FIND HANDLER
30 001116          .READW #XAREA,#HANCHN,#BLKO,#512.,#0 ;READ IN 1ST 2 BLOCKS
31 001154 103007          BCC     SETCKP      ;Branch if read ok
32          ;
33          ; Invalid device name.
34          ;
35 001156          SETIVD: .PURGE #HANCHN      ;MAKE SURE CHANNEL IS CLOSED
36 001164          FABORT #CSIMS2      ;INVALID DEVICE NAME
37          ;
38          ; We are performing a set to a device handler.
39          ; User must be privileged to do this.
40          ;
41 001174          SETCKP: .SAVEST #XAREA,#HANCHN,#HANSAV ;Save channel status in HANSAV
42 001214 032767 0000000 0000000          BIT     #PO#OPR,PRIVCO ;Is user privileged?
43 001222 001007          BNE     NXTOPT      ;Br if yes
44 001224          .PURGE #HANCHN      ;Close handler channel
45 001232          FABORT #EM#OPR      ;Not-privileged error message

```

Real device

```

1
2 ; Accrue the set option word.
3 ;
4 001242 105067 0000000 NXTOPT: CLRB NOFLAG ; RESET THE "NO" OPTION FLAG
5 001246 004767 0000000 CALL SKPSPC ; SKIP OVER ANY SPACES
6 ; Check for "NO" preceding the option word.
7 001252 010302 MOV R3,R2 ; SAVE COMMAND STRING POINTER
8 001254 004767 0000000 CALL GETKCH ; GET NEXT COMMAND CHARACTER
9 001260 120027 000116 CMPB R0,#'N ; IS IT "N"?
10 001264 001012 BNE 7$ ; BR IF NOT
11 001266 004767 0000000 CALL GETKCH ; CHECK NEXT CHARACTER
12 001272 120027 000117 CMPB R0,#'O ; IS IT "O"?
13 001276 001005 BNE 7$ ; BR IF NOT
14 001300 105267 0000000 INCB NOFLAG ; REMEMBER "NO" SPECIFIED
15 001304 004767 0000000 CALL SKPSPC ; SKIP OVER SPACES FOLLOWING "NO"
16 001310 000401 BR 9$
17 001312 010203 7$: MOV R2,R3 ; RESET POINTER TO START OF OPTION WORD
18 ; Accrue the option word.
19 001314 004767 0000000 9$: CALL GTRD50 ; ACCRUE AS A RAD50 VALUE
20 ;
21 ; Lookup the option word in the handler header.
22 ;
23 001320 012704 0004000 SETOPT: MOV #BLK0+400,R4 ; POINT TO START OF HANDLER TABLE OF SET OPTIONS
24 001324 005764 0000000 10$: TST SH$VAL(R4) ; HAVE WE REACHED THE END OF THE TABLE?
25 001330 001002 BNE 11$ ; CONTINUE CHECKING
26 001332 000167 177124 JMP INVLDO ; BR IF YES -- CAN'T FIND OPTION WORD
27 001336 026764 0000000 0000000 11$: CMP R50BUF,SH$NAM(R4); COMPARE NAME WITH THIS ENTRY
28 001344 001004 BNE 2$ ; BR IF MISMATCH
29 001346 026764 0000020 0000020 CMP R50BUF+2,SH$NAM+2(R4)
30 001354 001403 BEQ 9$ ; BR IF FOUND ENTRY FOR THIS OPTION
31 001356 062704 0000000 2$: ADD #SH#$SZ,R4 ; POINT TO NEXT OPTION ENTRY
32 001362 000760 BR 10$ ; CONTINUE SEARCH
33 ;
34 ; Found entry for this option.
35 ;
36 001364 116402 0000000 9$: MOVB SH$RTN(R4),R2 ; GET OFFSET TO ROUTINE TO CALL
37 001370 042702 177400 BIC #^C377,R2 ; CLEAR SIGN EXTENSION
38 001374 006302 ASL R2 ; CONVERT FROM WORD TO BYTE OFFSET
39 001376 062702 0004000 ADD #BLK0+400,R2 ; ADD BASE ADDRESS
40 ; See if "NO" option is allowed.
41 001402 105767 0000000 TSTB NOFLAG ; WAS "NO" SPECIFIED WITH OPTION?
42 001406 001410 BEQ 3$ ; BR IF NOT
43 001410 132764 0000000 0000000 BITB #SO$NO,SH$FLG(R4); IS NO ALLOWED WITH THIS OPTION?
44 001416 001002 BNE 14$ ; BR IF YES
45 001420 000167 000464 JMP SETIVS ; INVALID OPTION
46 001424 062702 000004 14$: ADD #4,R2 ; GET ADDRESS OF SET ROUTINE FOR NO-OPTION
47 ; See if option takes a numeric parameter.
48 001430 132764 0000000 0000000 3$: BITB #SO$NVL,SH$FLG(R4); DOES OPTION REQUIRE A NUMERIC PARAMETER?
49 001436 001430 BEQ 4$ ; BR IF NOT
50 001440 004767 0000000 CALL SKPSPC ; SKIP OVER ANY SPACES
51 001444 122327 000075 CMPB (R3)+,#'=' ; IS OPTION WORD FOLLOWED BY EQUAL SIGN?
52 001450 001401 BEQ 7$ ; BR IF YES
53 001452 005303 DEC R3 ; BACKUP POINTER
54 001454 004767 0000000 7$: CALL SKPSPC ; SKIP OVER ANY SPACES
55 001460 010305 MOV R3,R5 ; SAVE POINTER TO START OF NUMBER
56 001462 132764 0000000 0000000 BITB #SO$OCT,SH$FLG(R4); IS NUMERIC VALUE OCTAL?
57 001470 001003 BNE 6$ ; BR IF YES

```

Real device

```

58 001472 004767 0000000          CALL   ACRDEC          ; ACCRUE A DECIMAL VALUE
59 001476 000402                   BR      5$
60 001500 004767 0000000          6$:   CALL   ACROCT          ; ACCRUE AN OCTAL VALUE
61 001504 010100                   5$:   MOV    R1,R0          ; GET VALUE IN R0 FOR SET ROUTINE
62 001506 020305                   CMP    R3,R5          ; DID WE ACTUALLY GET A NUMBER?
63 001510 001002                   BNE   15$            ; BR IF YES
64 001512 000167 000372          JMP    SETIVS         ; BR IF NOT
65 001516 122327 000056          15$:  CMPB   (R3)+,#'    ; WAS DECIMAL POINT SPECIFIED?
66 001522 001401                   BEQ   4$              ; BR IF YES
67 001524 005303                   DEC   R3              ; BACKUP POINTER
68                                     ;
69                                     ;   Open channel 17 (which is normally used for KMON overlays) to
70                                     ;   the handler file so that some handlers can use this channel to
71                                     ;   read in and updated additional blocks in the handler.
72                                     ;
73 001526 010046          4$:   MOV    R0,-(SP)          ; Save numeric parameter value
74 001530                   .PURGE #17            ; Purge channel 17 (KMON overlay channel)
75 001536                   .REOPEN #XAREA,#17,#HANSAV ; Open channel 17 to handler file
76 001556 012600          MOV    (SP)+,R0        ; Restore numeric parameter value
77                                     ;
78                                     ;   Call routine in handler that actually does the set.
79                                     ;   On entry the following registers are set up:
80                                     ;   R0 = Optional numeric parameter value.
81                                     ;   R3 = Fixed option value found in option word (SH$VAL)
82                                     ;
83 001560 010346          MOV    R3,-(SP)          ; SAVE COMMAND STRING POINTER
84 001562 016403 0000000          MOV    SH$VAL(R4),R3   ; GET VALUE FROM OPTION ENTRY
85 001566 016701 0000000          MOV    DEVUNT,R1      ; GET THE DEVICE UNIT SPECIFIED
86 001572 000241          CLC                    ; CLEAR C-BIT BEFORE ENTERING HANDLER
87 001574 004712          CALL   @R2            ; CALL SET PROCESSING ROUTINE IN HANDLER
88 001576 012603          MOV    (SP)+,R3      ; RECOVER COMMAND STRING POINTER
89 001600 012702 0000000          MOV    #0,R2          ; SAY NO ERROR (DON'T AFFECT C-FLAG)
90 001604 103001          BCC   13$            ; BR IF NO ERROR OCCURRED
91 001606 005202          INC   R2              ; SET ERROR FLAG IN R2
92                                     ;
93                                     ;   Restore status of HANCHN and channel 17
94                                     ;
95 001610          13$:  .PURGE #17            ; Purge channel 17
96 001616                   .REOPEN #XAREA,#17,#KMNCHN ; Reopen channel 17 to KMON
97 001636                   .REOPEN #XAREA,#HANCHN,#HANSAV ; Reopen HANCHN to handler
98 001656 005702          TST   R2              ; Any error detected in handler?
99 001660 001113          BNE   SETIVS         ; Br if yes
100                                     ;
101                                     ;   See if there are more options.
102                                     ;
103 001662 004767 0000000          CALL   SKPSPC         ; SKIP SPACES
104 001666 112300          MOVB   (R3)+,R0        ; GET NEXT CHAR
105 001670 001406          BEQ   SETUP          ; BR IF REACHED END
106 001672 120027 000054          CMPB   R0,#'          ; IS COMMA THE DELIMITER?
107 001676 001401          BEQ   12$            ; BR IF YES
108 001700 005303          DEC   R3              ; BACKUP STRING POINTER
109 001702 000167 177334          12$:  JMP    NXTOPT        ; GO PROCESS THE NEXT OPTION

```

Real device

```

1      ;
2      ; Finished with all options.
3      ; Write updated handler back to its disk file.
4      ;
5 001706 SETUP: .WRITW #XAREA,#HANCHN,#BLKO,#512.,#0
6 001744 .CLOSE #HANCHN
7      ;
8      ; Now try to update the running handler in memory.
9      ;
10 001752 016702 0000000 MOV NUMDEV,R2 ;GET INDEX TO LAST DEVICE IN DEVICE TABLES
11 001756 016700 0000020 MOV HNBUF+2,R0 ;Get device name
12 001762 020062 0000000 1$: CMP RO,PNAME(R2) ;LOOKUP DEVICE NAME
13 001766 001412 BEQ 3$ ;BR IF FOUND
14 001770 162702 0000002 SUB #2,R2 ;MORE ENTRIES TO CHECK?
15 001774 003372 BGT 1$ ;BR IF YES
16 001776 FWARN #EM$HNI ;Warn that handler is not installed
17 002012 000434 BR 2$
18 ;
19 ; We have located the entry for the handler.
20 ; See if we are allowed to change the in-memory handler image.
21 ;
22 002014 032762 0000000 0000000 3$: BIT #DX$NST,DVFLAG(R2);Are we allowed to reload running handler?
23 002022 001407 BEQ 4$ ;Br if yes
24 002024 FWARN #TM$MRS ;Tell user that he must reboot the system
25 002040 000421 BR 2$
26 002042 012700 0000000 4$: MOV #HUPARG,R0 ;POINT TO ARG BLOCK FOR UPDATE INFO
27 002046 010260 0000004 MOV R2,4(R0) ;SET DEVICE INDEX NUMBER
28 002052 005067 0010066 CLR BLKO+1006 ;Clear LQE in handler image
29 002056 005067 0010106 CLR BLKO+1010 ;Clear CQE in handler image
30 002062 012760 0010000 0000006 MOV #BLKO+1000,6(R0);SET ADDRESS OF START OF NEW HANDLER CODE
31 002070 104375 ENT 375 ;TRY TO UPDATE RUNNING HANDLER
32 002072 103004 BCC 2$ ;BR IF SUCCESSFUL
33 002074 FABORT #HANBSY ;HANDLER WAS ACTIVE
34 002104 000167 0000000 2$: JMP RDCMD ;FINISHED
35 ;
36 ; Invalid command
37 ;
38 002110 SETIVS: .PURGE #HANCHN ;MAKE SURE CHANNEL IS CLOSED
39 002116 FABORT #CSIMS1

```

Real device

```

1          ;
2          ; Process options which only involve setting/resetting flags.
3          ;
4          ; Set a bit in a user table
5          ;
6 002126 004767 0000000 PRVSET: CALL   CKPRIV           ; REQUIRE PRIVILEGE
7 002132 012405          SETBIT: MOV    (R4)+,R5         ; POINT TO TABLE
8 002134 060105          ADD    R1,R5           ; POINT TO TABLE ENTRY FOR THIS USER
9 002136 051415          BIS    @R4,@R5          ; SET THE DESIRED FLAG
10 002140 000167 0000000          JMP    RDCMD
11         ;
12         ; Reset a bit in a user table
13         ;
14 002144 004767 0000000 PRVCLR: CALL   CKPRIV           ; REQUIRE PRIVILEGE
15 002150 012405          CLRBIT: MOV   (R4)+,R5         ; POINT TO TABLE
16 002152 060105          ADD    R1,R5           ; POINT TO TABLE ENTRY FOR THIS USER
17 002154 041415          BIC    @R4,@R5          ; RESET THE DESIRED FLAG
18 002156 000167 0000000          JMP    RDCMD
19         ;
20         ; Routine to store a value into a word cell
21         ;
22 002162 012405          STRWRD: MOV   (R4)+,R5         ; POINT TO WORD
23 002164 011415          MOV   (R4),(R5)          ; SET VALUE IN WORD
24 002166 000167 0000000          JMP    RDCMD
25         ;
26         ; Routine to accrue a byte value and store it into a line table.
27         ;
28 002172 122327 000075          SETBYT: CMPB  (R3)+,#'=         ; EQUAL SIGN SHOULD BE NEXT CHAR
29 002176 001014          BNE   NOEQL                 ; BR IF ERROR
30 002200 122327 000040          2$:  CMPB  (R3)+,#'         ; SKIP INTERVENING SPACES
31 002204 001775          BEQ   2$
32 002206 005303          DEC   R3
33 002210 004767 0000000          CALL   ACRDEC                ; ACCRUE THE DECIMAL VALUE
34 002214 116702 0000000          MOVB  CORUSR,R2             ; GET LINE INDEX #
35 002220 061402          ADD   @R4,R2                ; POINT TO LINE ENTRY IN TABLE
36 002222 110112          MOVB  R1,@R2                ; STORE VALUE INTO TABLE
37 002224 000167 0000000          JMP   RDCMD
38         ;
39 002230          NOEQL: FABORT #MISSEQ          ; MISSING EQUAL SIGN
40         ;
41         ; Set EDIT EDIT
42         ;
43 002240 042761 0000000 0000000 SEEDIT: BIC   #$TECO!$KED,LSW5(R1)
44 002246 000167 0000000          JMP   RDCMD
45         ;
46         ; Set EDIT TECO
47         ;
48 002252 042761 0000000 0000000 SETECO: BIC   #$KED,LSW5(R1)
49 002260 052761 0000000 0000000          BIS   #$TECO,LSW5(R1)
50 002266 000167 0000000          JMP   RDCMD
51         ;
52         ; SET EDIT KED/K52
53         ;
54 002272          SEK52:
55 002272 042761 0000000 0000000 SEKED:  BIC   #$TECO,LSW5(R1)
56 002300 052761 0000000 0000000          BIS   #$KED,LSW5(R1)
57 002306 000167 0000000          JMP   RDCMD

```

Real device

```

58 ;
59 ; SET UCL FIRST
60 ;
61 002312 052761 0000000 0000000 STUCLF: BIS    #$UCLCF,LSW7(R1); Set UCL-first flag
62 002320 042761 0000000 0000000          BIC    #<#$UCLCM!$UCLCL>,LSW7(R1) ; Clear call-middle & call-last flags
63 002326 000167 0000000          JMP    RDCMD
64 ;
65 ; SET UCL MIDDLE
66 ;
67 002332 052761 0000000 0000000 STUCLM: BIS    #$UCLCM,LSW7(R1); Set UCL-middle flag
68 002340 042761 0000000 0000000          BIC    #<#$UCLCF!$UCLCL>,LSW7(R1) ; Clear call-first & call-last flags
69 002346 000167 0000000          JMP    RDCMD
70 ;
71 ; SET UCL LAST
72 ;
73 002352 052761 0000000 0000000 STUCLL: BIS    #$UCLCL,LSW7(R1); Set UCL-last flag
74 002360 042761 0000000 0000000          BIC    #<#$UCLCF!$UCLCM>,LSW7(R1) ; Clear call-first & call-middle flag
75 002366 000167 0000000          JMP    RDCMD
76 ;
77 ; SET UCL NONE
78 ;
79 002372 042761 0000000 0000000 STUCLN: BIC    #<#$UCLCF!$UCLCM!$UCLCL>,LSW7(R1) ; Clear all UCL-call flags
80 002400 000167 0000000          JMP    RDCMD
81 ;
82 ; SET KMON IND
83 ;
84 002404 005767 0000000          SETIND: TST    INDSAV          ; IS IND AVAILABLE?
85 002410 001432          BEQ    1$          ; BR IF NOT
86 002412 052761 0000000 0000000          BIS    #$INDDF,LSW5(R1); SAY WE SHOULD USE IND
87 002420          .GVAL  #XAREA,#<CFSTS-MONVEC> ; GET CURRENT COMMAND FILE FLAGS
88 002440 010002          MOV    R0,R2
89 002442 052702 0000000          BIS    #CF$IND,R2          ; SET IND FLAG
90 002446          .PVAL  #XAREA,#<CFSTS-MONVEC>,R2; STORE UPDATED FLAGS
91 002472 000167 0000000          JMP    RDCMD
92 002476          1$: FABORT #NOIND
93 ;
94 ; SET KMON NO IND
95 ;
96 002506 042761 0000000 0000000 RSTIND: BIC    #$INDDF,LSW5(R1); CLEAR IND FLAG
97 002514          .GVAL  #XAREA,#<CFSTS-MONVEC> ; GET CURRENT COMMAND FILE FLAGS
98 002534 010002          MOV    R0,R2
99 002536 042702 0000000          BIC    #CF$IND,R2          ; RESET IND FLAG
100 002542          .PVAL  #XAREA,#<CFSTS-MONVEC>,R2; STORE UPDATED FLAGS
101 002566 000167 0000000          JMP    RDCMD

```

```

Prompt
1
2 ;----- .SBTTL Prompt -----
3 ; SET PROMPT string
4 ;
5 002572 SETPRT:
6 ;
7 ; Skip any leading spaces in front of prompt string
8 ;
9 002572 004767 000000G CALL SKPSPC
10 002576 012704 000000G MOV #KMPRMT,R4 ;Point to prompt holder in context block
11 ;
12 ; Determine if prompt string is enclosed in quotes
13 ;
14 002602 111300 MOV (R3),R0 ;Get first char of prompt string
15 002604 120027 000047 CMPB R0,#47 ;Is string enclosed in " ' " characters?
16 002610 001400 BEQ 2$ ;Br if yes
17 002612 120027 000042 CMPB R0,#42 ;How about " " " ?
18 002616 001007 BNE 3$ ;Br if not
19 ;
20 ; String is enclosed in quotes
21 ;
22 002620 004767 000000G 2$: CALL ACRSTR ;Accrue the string
23 002624 010002 MOV R0,R2 ;Get length of the string
24 002626 001417 BEQ 6$ ;Br if string is empty
25 002630 012703 000000G MOV #BLKO,R3 ;Point to buffer with string
26 002634 000403 BR 10$
27 ;
28 ; String is not enclosed in quotes
29 ;
30 002636 020302 3$: CMP R3,R2 ;Anything in prompt string?
31 002640 103012 BHIS 6$ ;Br if not
32 002642 160302 SUB R3,R2 ;Determine length of prompt string
33 ;
34 ; Make sure string is not too long
35 ;
36 002644 020227 000000G 10$: CMP R2,#MXPRMT ;Is string too long?
37 002650 101411 BLOS 5$ ;Br if ok
38 002652 FERR #BADPMT ;Prompt string too long
39 002666 112724 000056 6$: MOV (R4)+ ;Set prompt to "."
40 002672 000402 BR 8$
41 ;
42 ; Set prompt string
43 ;
44 002674 112324 5$: MOV (R3)+,(R4)+ ;Move prompt string to KMDN context block
45 002676 077202 SOB R2,5$
46 002700 112714 000200 8$: MOV #200,(R4) ;Terminate the prompt string
47 ;
48 ; Finished
49 ;
50 002704 000167 000000G 9$: JMP RDCMD ;Go get next command

```

Syspassword

```

1          .SBTTL      Syspassword
2          ;-----
3          ; SET SYSPASSWORD command
4          ;
5 002710   SETSYP:
6          ;
7          ; User must have SYSPRV privilege
8          ;
9 002710   004767   000000G      CALL      CKSYPV      ;Make sure user has SYSPRV privilege
10         ;
11        ; Accrue the password string
12        ;
13 002714   004767   000000G      CALL      ACRTXT      ;Accrue the text string
14        ;
15        ; Make sure it's not too long
16        ;
17 002720   020027   000024      CMP      R0,#20.      ;Compare with max legal length
18 002724   101404      BLOS      1$          ;Br if ok
19 002726      FABORT   #EM$SPL      ;Password string is too long
20        ;
21        ; Move password to password buffer
22        ;
23 002736   012702   000000G      1$:      MOV      #BLK0,R2      ;Point to accrued password
24 002742   012703   000000G      MOV      #SYPSWD,R3      ;Point to system buffer
25 002746   112223      2$:      MOVB     (R2)+,(R3)+      ;Move password string
26 002750   001374      BNE      2$          ;Loop if more to move
27        ;
28        ; Finished
29        ;
30 002752   000167   000000G      JMP      RDCMD

```

Endstartup

```
1 .SBTTL . Endstartup
2 ;-----
3 ; SET ENDSTARTUP command
4 ; (Signal end of privileged portion of start-up command file)
5 ;
6 002756 042761 0000000 0000000 SETESU: BIC ##NOIN,LSW3(R1) ;Allow input to be accepted for line
7 002764 042761 0000000 0000000 BIC ##SUCF,LSW9(R1) ;Say not in start-up command file
8 002772 000167 0000000 JMP RDCMD
```

Window

```

1          .SBTTL      Window
2          ;-----
3          ; SET WINDOW command
4          ;
5 002776   SETWIN:
6          ;
7          ; User must be authorized to create global regions in order to use windows
8          ;
9 002776   032767   0000000 0000000   BIT      #P2$CGR.PRIVC2   ;May job create global regions?
10 003004   001004   BNE      10$           ;Br if yes
11 003006   FABORT  #EM$WCM           ;Not privileged to do this
12          ;
13          ; Initialize some cells
14          ;
15 003016   112767   000120   175046 10$:   MOV     #80.,WINMAK+4   ;Set window width to 80 columns
16 003024   112767   000020   175041   MOV     #16.,WINMAK+5   ;Set default max scroll lines
17 003032   005067   174760   CLR     WINFLG         ;No flags yet
18          ;
19          ; Do command parsing
20          ;
21 003036   012704   001420'   MOV     #WINHD,R4      ;Point to option driver list
22 003042   004767   0000000   CALL   SCNOPS         ;Process all command options
23          ;
24          ; Now see which options were specified
25          ;
26 003046   016705   174744   12$:   MOV     WINFLG,R5   ;Pick up option flags
27          ;
28          ; See if OFF was specified
29          ;
30 003052   032705   000001   BIT     #WFDEL,R5     ;Was OFF option specified?
31 003056   001404   BEQ     1$           ;Br if not
32 003060   012700   000104'   MOV     #WINDEL,R0    ;Delete the window
33 003064   104375   ENT     375
34 003066   000455   BR     20$
35          ;
36          ; If WIDE option was specified, set window width to 132 columns
37          ;
38 003070   032705   000010   1$:   BIT     #WF132,R5    ;Was WIDE option specified?
39 003074   001403   BEQ     2$           ;Br if not
40 003076   112767   000204   174766   MOV     #132.,WINMAK+4 ;Set window width to 132 columns
41          ;
42          ; Try to make the window
43          ;
44 003104   032761   0000000 0000000 2$:   BIT     #<VT100!VT200!VT52>,LTRMTP(R1) ;VT100, VT200 or VT52
45 003112   001004   BNE     13$         ;Br if yes
46 003114   FABORT  #EM$SLT     ;Invalid terminal type
47 003124   012700   000066' 13$:   MOV     #WINMAK,R0    ;Point to window-make arg block
48 003130   104375   ENT     375         ;Try to create the window
49 003132   103435   BCS     30$         ;Br if error on window creation
50          ;
51          ; Select the window
52          ;
53 003134   012700   000100'   MOV     #WINMAP,R0    ;EMT to map to the window
54 003140   104375   ENT     375
55          ;
56          ; IF WIDE was specified, send string to put term in 132 col mode
57          ;

```

```

58 003142 032705 000010          BIT    #WF132,R5      ;Was WIDE option specified?
59 003146 001403          BEQ    3$             ;Br if not
60 003150          .PRINT #TC132        ;Put terminal in 132 column mode
61          ;
62          ; If NARROW was specified, send string to put term in 80 col mode
63          ;
64 003156 032705 000020 3$:    BIT    #WF80,R5      ;Was NARROW option specified?
65 003162 001403          BEQ    4$             ;Br if not
66 003164          .PRINT #TC80        ;Put terminal in 80 column mode
67          ;
68          ; If REVERSE was specified, put terminal in reverse video mode
69          ;
70 003172 032705 000002 4$:    BIT    #WFREV,R5     ;Was REVERSE option specified?
71 003176 001403          BEQ    5$             ;Br if not
72 003200          .PRINT #TCREV       ;Put terminal in reverse video mode
73          ;
74          ; If NORMAL was specified, put terminal in normal video mode
75          ;
76 003206 032705 000004 5$:    BIT    #WFNORM,R5    ;Was NORMAL option specified?
77 003212 001403          BEQ    20$            ;Br if not
78 003214          .PRINT #TCNORM      ;Put terminal in normal video mode
79          ;
80          ; Finished
81          ;
82 003222 000167 000000G 20$:   JMP    RDCMD
83          ;
84          ; Error occurred which making window
85          ;
86 003226 113702 000000G 30$:   MOVB   @#ERRLOC,R2    ;Get EMT error code
87 003232 020227 000004          CMP    R2,#MAXWEM      ;Is it too large?
88 003236 103401          BLD    31$            ;Br if not
89 003240 005002          CLR    R2             ;Use message for error 0 if too big
90 003242 006302 31$:   ASL    R2             ;Get word table index
91 003244 016202 003256'  MOV    WEM(R2),R2        ;Get address of error message
92 003250          FABORT R2        ;Print the error message
93          ;
94          ; Table of window-creation error messages
95          ;
96 003256 000000G WEM:   .WORD  EM$WC0
97 003260 000000G          .WORD  EM$WC1
98 003262 000000G          .WORD  EM$WC2
99 003264 000000G          .WORD  EM$WC3
100          MAXWEM = <. -WEM>??

```

Window

```

1          ; -----
2          ; Process COLUMN=n option of SET WINDOW command
3          ;
4 003266 010146 WINCOL: MOV      R1, -(SP)
5          ;
6          ; Accrue the column parameter value
7          ;
8 003270 004767 0000000 CALL    ACRDEC
9          ;
10         ; Store into EMT argument block
11         ;
12 003274 110167 174572        MOVB   R1, WINMAK+4      ;Set column width in EMT arg block
13         ;
14         ; Finished
15         ;
16 003300 012601        MOV    (SP)+, R1
17 003302 000207        RETURN
18         ;
19         ; -----
20         ; Process SCROLL[=n] option which specifies the maximum number
21         ; of scroll lines.
22         ;
23 003304 010146 WINSCR: MOV      R1, -(SP)
24         ;
25         ; If no parameter is specified, allow unlimited scrolling
26         ;
27 003306 004767 0000000 CALL    SKPSPC      ;Skip over spaces
28 003312 121327 000075    CMPB   (R3), #'=    ;Value specified?
29 003316 001000        BNE    3$          ;If not, allow unlimited
30         ;
31         ; Accrue the parameter value
32         ;
33 003320 004767 0000000 1$:    CALL    ACRDEC      ;Accrue decimal parameter
34 003324 020127 000177    CMP    R1, #127.   ;Asked for more than max?
35 003330 101402        BLOS   2$          ;If not, set the requested number
36 003332 012701 177777    3$:    MOV    #-1, R1  ;Allow unlimited scrolling
37         ;
38         ; Store value into EMT argument block
39         ;
40 003336 110167 174531    2$:    MOVB   R1, WINMAK+5      ;Set max scroll lines
41         ;
42         ; Finished
43         ;
44 003342 012601        MOV    (SP)+, R1
45 003344 000207        RETURN
46         ;
47         ; -----
48         ; SET WINDOW/NOSCROLL
49         ;
50 003346 105067 174521 WINNOS: CLRB   WINMAK+5      ;0 scroll lines ==> No scrolling
51 003352 000207        RETURN
52         ;
53         ; -----
54         ; Process a SET WINDOW option that just sets a flag bit
55         ;
56 003354 051467 174436 WINBIT: BIS    (R4), WINFLG  ;Set appropriate flag bit
57 003360 000207        RETURN

```

Printwindow

```

1          .SBTTL      Printwindow
2          ;-----
3          ; SET PRINTWINDOW/DEVICE=device/TYPE=type
4          ;
5 003362   SETPRG:
6          ;
7          ; Process qualifiers specified with command
8          ;
9 003362   012704   001534'   MOV      #PRSHD,R4      ;Point to option driver list
10 003366   004767   000000G   CALL     SCNOPS      ;Process all of the command options
11          ;
12          ; Finished
13          ;
14 003372   000167   000000G   JMP      RDCMD
15          ;
16          ; Process the /DEVICE=device qualifer
17          ;
18 003376   010246   PRSDEV: MOV      R2,-(SP)
19 003400   004767   000000G   CALL     CHKEQ      ;Equal sign should follow qualifier name
20 003404   004767   000000G   CALL     GTRD50     ;Accrue the device name
21 003410   122327   000072   CMPB    (R3)+,#'    ;Colon specified with device name?
22 003414   001401   BEQ     1$         ;Br if yes
23 003416   005303   DEC     R3         ;Backup pointer
24 003420   016767   000000G 000000G 1$: MOV     R50BUF,JPWDEV ;Store into job context cell
25 003426   012602   MOV     (SP)+,R2
26 003430   000207   RETURN
27          ;
28          ; Process the /TYPE=type qualifier
29          ;
30 003432   010446   PRSTYP: MOV     R4,-(SP)
31 003434   004767   000000G   CALL     CHKEQ      ;Make sure we have an equal sign
32 003440   012704   001610'   MOV     #PWTHD,R4  ;Point to table of printer types
33 003444   004767   000000G   CALL     SEARCH     ;Accrue and look up printer name
34 003450   103406   BCS    1$         ;Br if invalid printer type
35 003452   111467   000000G   MOVB   (R4),JPWTYP ;Set printer type code
36 003456   016467   000002 000000G   MOV     2(R4),JPWFLG ;Set printer attribute flags
37 003464   000417   BR     9$
38 003466   005704   1$: TST    R4      ;Printer unrecognized or ambiguous?
39 003470   001407   BEQ    2$         ;Br if not recognized
40 003472   FERR   #EM$PTA   ;Printer type ambiguous
41 003506   000406   BR     9$
42 003510   2$: FERR  #EM$PTU   ;Printer type unrecognized
43          ;
44          ; Finished
45          ;
46 003524   012604   9$: MOV    (SP)+,R4
47 003526   000207   RETURN
48          ;
49          ; Process the /LETTERQUALITY qualifier
50          ;
51 003530   052767   000000G 000000G PRSLET: BIS    #PA$LET,JPWFLG ;Set letter-quality flag
52 003536   000207   RETURN
53          ;
54          ; Process the /DRAFT qualifier
55          ;
56 003540   042767   000000G 000000G PRSDRF: BIC    #PA$LET,JPWFLG ;Clear letter-quality flag
57 003546   000207   RETURN

```

```
58 ;  
59 ; Process the /BELL qualifier  
60 ;  
61 003550 052767 0000000 0000000 PRSBEL: BIS #PA$BEL,JPWFLG ;Set bell flag  
62 003556 000207 RETURN  
63 ;  
64 ; Process the /NOBELL qualifier  
65 ;  
66 003560 042767 0000000 0000000 PRSNBL: BIC #PA$BEL,JPWFLG ;Clear bell flag  
67 003566 000207 RETURN  
68 ;  
69 ; Process the /WIDTH qualifier  
70 ;  
71 003570 042767 0000000 0000000 PRSWID: BIC #PA$NWD,JPWFLG ;Clear nowidth flag  
72 003576 000207 RETURN  
73 ;  
74 ; Process the /NOWIDTH qualifier  
75 ;  
76 003600 052767 0000000 0000000 PRSNWD: BIS #PA$NWD,JPWFLG ;Set nowidth flag  
77 003606 000207 RETURN  
78 ;  
79 ; Process the /KEYPRINT qualifier  
80 ;  
81 003610 116701 0000000 PRSKEY: MOVB CORUSR,R1 ;Get job index number  
82 003614 052761 0000000 0000000 BIS #PWKEY,LSW11(R1) ;Set KEYPRINT flag  
83 003622 000207 RETURN  
84 ;  
85 ; Process the /NOKEYPRINT qualifier  
86 ;  
87 003624 116701 0000000 PRSNKY: MOVB CORUSR,R1 ;Get job index number  
88 003630 042761 0000000 0000000 BIC #PWKEY,LSW11(R1) ;Clear KEYPRINT flag  
89 003636 000207 RETURN
```

Priority

			.SBTTL	Priority
1				
2				
3			;	-----
4			;	SET PRIORITY value
5	003640	004767	0000000	SETPRI: CALL ACRDEC ;Accrue decimal value
6	003644	020127	0000000	CMP R1,#MAXPRI ;Is priority too large?
7	003650	101425		BLOS 3# ;Br if ok
8	003652			5#: FERR #BADPRI ;Invalid priority value
9	003666	012705	0000000	MOV #0,R5 ;Print minimum priority
10	003672	004767	0000000	CALL PRTDEC
11	003676			.PRINT #TOTXT ;Print " to "
12	003704	116705	0000000	MOVB MXJPRI,R5 ;Print maximum priority
13	003710	004767	0000000	CALL PRTDEC
14	003714			.PRINT #CRLF ;End print line
15	003722	000427		BR 9#
16	003724	120167	0000000	3#: CMPB R1,MXJPRI ;Does it exceed max allowed for job?
17	003730	101417		BLOS 2# ;Br if ok
18	003732			FERR #HIPRI ;Priority is too high
19	003746	116705	0000000	MOVB MXJPRI,R5 ;Get max allowed value
20	003752	004767	0000000	CALL PRTDEC ;Display max priority
21	003756			.PRINT #CRLF
22	003764	116701	0000000	MOVB MXJPRI,R1 ;Get max allowed for job
23	003770	012700	000062'	2#: MOV #PRIEMT,R0 ;Point to EMT arg block to set job priority
24	003774	010160	000002	MOV R1,2(R0) ;Store prio value into EMT arg block
25	004000	104375		EMT 375 ;Set job priority
26	004002	000167	0000000	9#: JMP RDCMD ;Go get next command

Maxpriority

			. SBTTL	Maxpriority
1				
2				
3			; SET MAXPRIORITY value	
4				
5	004006	004767	0000000	SETMPR: CALL ACRDEC ;Accrue decimal value
6	004012	020127	0000000	CMP R1,#MAXPRI ;Larger than max allowed?
7	004016	101425		BLOS 1\$;Br if ok
8	004020			FERR #BADPRI ;Invalid priority value
9	004034	005000		CLR R0 ;Get minimum value
10	004036	004767	0000000	CALL PRTDEC ;Print minimum value
11	004042			.PRINT #TOTXT ;Print " to "
12	004050	012705	0000000	MOV #MAXPRI,R5 ;Get maximum value
13	004054	004767	0000000	CALL PRTDEC ;Print it
14	004060			.PRINT #CRLF
15	004066	116701	0000000	MOVB VPRIDF,R1 ;Set to default value
16	004072	120167	0000000	1\$: CMPB R1,MXJPRI ;Is he attempting to increase the max pri?
17	004076	101404		BLOS 2\$;Br if not
18	004100			FABORT #EM\$CIP ;Cannot increase maxpri
19	004110	110167	0000000	2\$: MOVB R1,MXJPRI ;Set maximum job priority
20	004114	116702	0000000	MOVB CORUSR,R2 ;Get current job index number
21	004120	126201	0000000	CMPB LBSPRI(R2),R1 ;Is current priority too high?
22	004124	101405		BLOS 3\$;Br if ok
23	004126	012700	000062'	MOV #PRIEMT,R0 ;Point to EMT arg block to set prio
24	004132	010160	0000002	MOV R1,2(R0) ;Set new priority for job
25	004136	104375		ENT 375
26	004140	000167	0000000	3\$: JMP RDCMD ;Finished

Error

```

1          .SBTTL      Error
2          ;-----
3          ; ROUTINE TO SET ERROR SEVERITY ABORT LEVEL
4          ;
5 004144 111467 0000000  SETSEV: MOVB    @R4,ERRSEV    ;SET ERROR SEVERITY FOR USER
6 004150 111404          MOVB    @R4,R4      ;GET ERROR SEVERITY LEVEL
7 004152 042704 177400   BIC    #177400,R4      ;CLEAR HIGH ORDER BYTE
8 004156 052704 001400   BIS    #3*400,R4      ;SET COMMAND FILE NESTING DEPTH
9 004162          .PVAL  #XAREA,#<<CFABLV-MONVEC>,R4 ;SET ABORT LEVEL IN SIMUL RMON
10 004206 000167 0000000  JMP    RDCMD
11
12         .SBTTL      Shutdown
13         ;-----
14         ; Set the flag that says a system shutdown is being done.
15         ;
16 004212 004767 0000000  SETSHT: CALL   CKPRIV      ;User must have privilege
17 004216 112767 177777 0000000  MOVB    #-1,STPFLG      ;Say system shutdown taking place
18 004224 005067 0000000  CLR    BOTDEV          ;Reboot from system disk
19 004230 000167 0000000  JMP    RDCMD
20
21         ;-----
22         ; Clear the flag that says a shutdown is taking place.
23         ;
24 004234 004767 0000000  SETNSH: CALL   CKPRIV      ;User must have privilege
25 004240 105067 0000000  CLR    STPFLG          ;Say we are not doing a shutdown
26 004244 000167 0000000  JMP    RDCMD

```

Log

```

1          .SBTTL      Log
2          ;-----
3          ; Process the SET LOG command.
4          ;
5 004250   SETLOG:
6          ;
7          ; Accrue option keyword
8          ;
9 004250   012704   002270'      MOV      #STLGHD,R4      ;Point to keyword table
10 004254   004767   0000000     CALL     SCNOP5      ;Process command qualifiers
11 004260   000167   0000000     JMP      RDCMD      ;Finished with command
12          ;
13          ; SET LOG CLOSE
14          ;
15 004264   004767   0000000     STLOCL: CALL    LOGCLS      ;Close the log file
16 004270   000207
17          ;
18          ; SET LOG CLEAN
19          ;
20 004272   010344   STLOCN: MOV      R3,-(SP)
21 004274   032767   0000000 0000000     BIT      #LF#OPN,LOGFLG ;Is the log file open now?
22 004302   001427           BEQ      9$           ;Br if not
23 004304   012767   0000000 0000000     MOV      #LOGBUF,LOGPTR ;Reset buffer pointer
24 004312   005067   0000000           CLR     LOGBLK      ;Start writing at block 0
25 004316   012703   0000000           MOV     #LOGCHN,R3   ;Get channel # used for log file
26 004322   016700   0000000           MOV     CXTRMN,R0    ;Point to simulated RMDN area
27 004326   062700   0000000           ADD     #R#CHN,R0    ;Point to channel block for channel 0
28 004332   020327   000021           CMP     R3,#17.     ;Is this channel in extended chan area?
29 004336   103404           BLO     1$           ;Br if not
30 004340   162703   000021           SUB     #17.,R3     ;Get channel # relative to extended channels
31 004344   062700   0000000           ADD     #R#XCHN-R#CHN,R0;Point to 1st extended channel block
32 004350   070327   0000000   1$:    MUL     #CHNSIZ,R3 ;Get offset to block of interest
33 004354   060003           ADD     R0,R3       ;Point to our channel block
34 004356   005063   0000000           CLR     C,USED(R3)  ;Say no data has been written to file
35 004362   012603   9$:    MOV     (SP)+,R3
36 004364   000207           RETURN
37          ;
38          ; SET LOG WRITE
39          ;
40 004366   052767   0000000 0000000     STLGWR: BIS     #LF#WRT,LOGFLG ;Enable writes to log file
41 004374   000207           RETURN
42          ;
43          ; SET LOG NOWRITE
44          ;
45 004376   042767   0000000 0000000     STLGNW: BIC     #LF#WRT,LOGFLG ;Disable log file
46 004404   000207           RETURN
47          ;
48          ; SET LOG INPUT
49          ;
50 004406   042767   0000000 0000000     STLGIN: BIC     #LF#OUT,LOGFLG ;Clear output-logging flag
51 004414   052767   0000000 0000000           BIS     #LF#IN,LOGFLG ;Set input-logging flag
52 004422   000207           RETURN
53          ;
54          ; SET LOG OUTPUT
55          ;
56 004424   042767   0000000 0000000     STLGOT: BIC     #LF#IN,LOGFLG ;Clear input-logging flag
57 004432   052767   0000000 0000000           BIS     #LF#OUT,LOGFLG ;Set output-logging flag

```

Log

```
58 004440 000207          RETURN
59          ;
60          ; SET LOG ALL
61          ;
62 004442 052767 000000C 000000G STL GAL: BIS #LF$IN!LF$OUT,LOGFLG ;Log both input and output
63 004450 000207          RETURN
```

Log

```

1
2 ; SET LOG FILE=file-spec
3 ;
4 004452 010446 STLQFL: MOV R4, -(SP)
5 004454 010546 MOV R5, -(SP)
6 004456 004767 0000000 CALL LOGCLS ;Close current log file
7 004462 004767 0000000 CALL SKPSPC ;Skip over spaces
8 004466 122327 0000075 CMPB (R3)+, #'= ;See if equal sign was specified
9 004472 001401 BEQ 1$ ;Br if yes (skip over it)
10 004474 005303 DEC R3 ;Point back to first char of file spec
11 ;
12 ; Accrue the file spec
13 ;
14 004476 012704 0000000 1$: MOV #R5OLOG, R4 ;Point to default extension ("LOG")
15 004502 012705 0000001 MOV #1, R5 ;Tell ACRFIL this is an output file
16 004506 004767 0000000 CALL ACRFIL ;Accrue the file spec
17 004512 103475 BCS 10$ ;Br if invalid file spec
18 ;
19 ; Translate logical device name to physical
20 ;
21 004514 012705 0000000 MOV #FILNAM, R5 ;Point to file name buffer
22 004520 004767 0000000 CALL LOGASN ;Perform logical name translation
23 ;
24 ; Don't allow log output to go to "TT:"
25 ;
26 004524 016700 0000000 MOV FILNAM, R0 ;Get the log device name
27 004530 004767 0000000 CALL CHKTTD ;Is the log device TT?
28 004534 103464 BCS 10$ ;Br if yes -- Error
29 ;
30 ; Try to open the file
31 ;
32 004536 . ENTER #XAREA, #LOGCHN, #FILNAM, FILNAM+B.
33 004564 103004 BCC 2$ ;Br if enter ok
34 004566 FABORT #BDLGOP ;Cannot open log file
35 ;
36 ; Save information about the log file physical device
37 ;
38 004576 016705 0000000 2$: MOV FILNAM, R5 ;Get log file device name
39 004602 004767 0000000 CALL CHKDEV ;Convert to dev # and unit #
40 004606 103421 BCS 3$ ;Br if invalid device
41 004610 020467 0000000 CMP R4, LDDEVX ;Is log file on a logical disk?
42 004614 001407 BEQ 4$ ;Br if yes
43 004616 110467 0000000 MOVB R4, LOGDVU ;Save log file device #
44 004622 110067 0000010 MOVB R0, LOGDVU+1 ;Save log file unit #
45 004626 005067 0000000 CLR LOGBAS ;Say not on a logical disk
46 004632 000407 BR 3$
47 004634 006300 4$: ASL R0 ;Convert unit # to word table index
48 004636 016067 0000000 0000000 MOV LDPDEV(R0), LOGDVU ;Save physical device and unit #
49 004644 016067 0000000 0000000 MOV LDBASE(R0), LOGBAS ;Save base block # of logical disk
50 ;
51 ; Set "hold" mode for spooled log files
52 ;
53 004652 012700 0000000 3$: MOV #SPLHLA, R0 ;Point to emt argument block
54 004656 104375 ENT 375 ;Set hold mode for log file
55 ;
56 ; Initialize the log file
57 ;

```

Log

```
58 004660 052767 000000C 000000G      BIS      #<LF#OPN!LF#WRT!LF#IN!LF#OUT>,LOGFLG ;Say log file is open
59 004666 012767 000000G 000000G      MOV      #LOGBUF,LOGPTR ;Init pointer to log buffer
60 004674 005067 000000G      CLR      LOGBLK      ;Write to block 0
61 ;
62 ; Finished
63 ;
64 004700 012605      MOV      (SP)+,R5
65 004702 012604      MOV      (SP)+,R4
66 004704 000207      RETURN
67 ;
68 ; Error -- Invalid file spec
69 ;
70 004706      10#:      FABORT #BDFNAM      ;Invalid file name
```

Logoff

```

1          .SBTTL      Logoff
2          ;-----
3          ; Process the SET LOGOFF command.
4          ; This command is used to declare a logoff command file.
5          ;
6 004716   SETLOF:
7          ;
8          ; Accrue option keyword
9          ;
10 004716  012704  002334'      MOV      #STLOHD,R4      ;Point to keyword table
11 004722  004767  000000G     CALL     SEARCH      ;Identify keyword
12 004726  103002                BCC     1$           ;Br if identified keyword
13 004730  000167  173512     JMP      BDSO        ;Invalid option
14          ;
15          ; Enter keyword processing routine
16          ;
17 004734  000134     1$:      JMP      @(R4)+      ;Enter keyword processing routine
18          ;
19          ; SET LOGOFF FILE=file-spec
20          ;
21 004736  032761  000000G 000000G STLOFL: BIT     ##SUCF,LSW9(R1) ;Are we in a startup command file?
22 004744  001010                BNE     1$           ;Br if yes
23 004746  032767  000000G 000000G     BIT     #P0$SYS,PRIVCO ;Do we have SYSPRV privilege?
24 004754  001004                BNE     1$           ;Br if yes
25 004756                FABORT  #EM$NSF      ;Not in a startup command file
26          ;
27          ; Accrue the file spec
28          ;
29 004766  004767  000000G     1$:      CALL     SKPSPC      ;Skip over spaces
30 004772  122327  000075     CMPB    (R3)+,#/=    ;Should have equal sign
31 004776  001401                BEQ     3$           ;Br if yes (skip over it)
32 005000  005303                DEC     R3           ;Point to 1st char of file name
33 005002  012704  000000G     3$:      MOV     #R50COM,R4    ;Set "COM" as default extension
34 005006  005005                CLR     R5           ;Tell ACRFIL this is an input file
35 005010  004767  000000G     CALL     ACRFIL      ;Accrue the file spec
36 005014  103416                BCS    10$          ;Br if invalid file spec
37          ;
38          ; Translate logical file device name to physical device
39          ;
40 005016  012705  000000G     MOV     #FILNAM,R5    ;Point to file spec
41 005022  004767  000000G     CALL     LOGASN      ;Perform any logical device assignment
42          ;
43          ; Move file spec to job context block
44          ;
45 005026  012705  000000G     MOV     #FILNAM,R5    ;Point to file spec
46 005032  012704  000000G     MOV     #LGFSPC,R4    ;Point to job context area
47 005036  012700  000004     MOV     #4,R0         ;Get # words to move
48 005042  012524     2$:      MOV     (R5)+,(R4)+    ;Move file spec to job context block
49 005044  077002                SOB    R0,2$
50          ;
51          ; Finished
52          ;
53 005046  000167  000000G     JMP     RDCMD
54          ;
55          ; Invalid file spec
56          ;
57 005052     10$:     FABORT  #BDFNAM      ;Invalid file name

```

Subprocess

```

1          .SBTTL      Subprocess
2          ;-----
3          ; Process the SET SUBPROCESS command.
4          ; This command is used to declare a command file to be executed when
5          ; a sub process (virtual line) is initiated.
6          ;
7 005062   SETSBP:
8          ;
9          ; Accrue option keyword
10         ;
11 005062   012704   002344'   MOV      #SBPHD,R4      ;Point to keyword table
12 005066   004767   0000000   CALL    SCNOPS        ;Process the options
13 005072   000167   0000000   JMP     RDCMD
14         ;
15         ; SET SUBPROCESS/FILE=file-spec
16         ;
17 005076   010446   SBPFIL: MOV      R4,-(SP)
18 005100   010546   MOV      R5,-(SP)
19         ;
20         ; This command is only legal within a start-up command file
21         ;
22 005102   032761   0000000 0000000   BIT     ##$UCF,LSW7(R1) ;Are we in a startup command file?
23 005110   001010   BNE     1$          ;Br if yes
24 005112   032767   0000000 0000000   BIT     #PO$SYS,PRIVCO ;Do we have SYSPRV privilege?
25 005120   001004   BNE     1$          ;Br if yes
26 005122   FABORT   #EM$NSF      ;Not in a startup command file
27         ;
28         ; Accrue the file spec
29         ;
30 005132   004767   0000000   1$:    CALL    ACRTXT      ;Accrue the file string
31 005136   020027   000017   CMP     RO,#15.      ;Compare with max legal length
32 005142   101404   BLOS   3$          ;Br if ok
33 005144   FABORT   #EM$STL      ;File spec is too long
34         ;
35         ; Move file spec to job context block
36         ;
37 005154   012705   0000000   3$:    MOV     #BLKO,R5      ;Point to file spec
38 005160   012704   0000000   MOV     #SBPSUF,R4     ;Point to job context area
39 005164   112524   2$:    MOVB   (R5)+,(R4)+  ;Move file spec to job context block
40 005166   001376   BNE    2$          ;Loop till all moved
41         ;
42         ; Finished
43         ;
44 005170   012605   MOV     (SP)+,R5
45 005172   012604   MOV     (SP)+,R4
46 005174   000207   RETURN
47         ;
48         ; Invalid file spec
49         ;
50 005176   10$:    FABORT   #BDFNAM      ;Invalid file name

```

Kmon

```

1          .SBTTL      Kmon
2          ;-----
3          ; Process the SET KMON USER command.
4          ; Verify that a user-written command interface program exists and if so,
5          ; set a flag saying to use it for this job.
6          ;
7 005206   SETUKM:
8          ;
9          ; See if user specified a file spec
10         ;
11 005206   004767   000000G      CALL      SKPSPC      ;Skip over any spaces
12 005212   105713          TSTB      (R3)      ;Anything specified following "USER"?
13 005214   001417          BEQ       2$        ;Br if not
14 005216   122327   000075      CMPB      (R3)+,#/=    ;Equal sign?
15 005222   001053          BNE       3$        ;Error if not
16         ;
17         ; Accrue the user file name
18         ;
19 005224   012704   000000G      MOV       #R5OSAV,R4   ;Set default extension
20 005230   005005          CLR       R5          ;Tell ACRFIL this is an input file
21 005232   004767   000000G      CALL      ACRFIL      ;Accrue the file spec
22 005236   103451          BCS      4$          ;Br if invalid file spec
23 005240   012704   000000G      MOV       #FILNAM,R4   ;Point to area with file spec
24 005244   010405          MOV       R4,R5
25 005246   004767   000000G      CALL      LOGASN      ;Convert logical device name to physical
26 005252   000404          BR        6$
27         ;
28         ; User did not specify a file name. Default to "SY:UKMON.SAV".
29         ;
30 005254   012704   000000G      2$:      MOV       #UCIDEF,R4   ;Point to default file spec
31 005260   016714   000000G      MOV       SYNAME,(R4)   ;Set physical SY device
32         ;
33         ; Move file spec to UCISPC area in context block
34         ;
35 005264   012700   000004      6$:      MOV       #4,R0        ;Get # words to move
36 005270   012705   000000G      MOV       #UCISPC,R5    ;Store UCI file spec here
37 005274   012425          5$:      MOV       (R4)+,(R5)+    ;Move file spec to job context block
38 005276   077002          SOB      R0,5$
39         ;
40         ; Try to lookup the specified file
41         ;
42 005300          .LOOKUP #XAREA,#1,#UCISPC;Try to find command processor program
43 005320   103410          BCS      1$          ;Br if not available
44 005322          .CLOSE #1          ;Close the program file
45 005330   052761   000000G 000000G      BIS      #UKMON,LSW7(R1);Set flag saying to use user-written program
46 005336   000167   000000G      JMP      RDCMD
47         ;
48         ; Error processing
49         ; Cannot find specified file
50         ;
51 005342          1$:      FABORT #EM#NUK      ;User-written command processor not available
52         ;
53         ; Invalid syntax
54         ;
55 005352          3$:      FABORT #MISSEQ
56         ;
57         ; Invalid file spec

```

58
59 005362

;
4*: FABORT #BDFNAM

Kmon

```

1 ; -----
2 ; Process the SET KMON NEW command.
3 ; Begin to use a new version of TSKMON.
4 ;
5 005372 004767 0000000 SETKNW: CALL CKSYPV ;Must have SYSPRV privilege
6 005376 .LOOKUP #XAREA,#1,#KMNNAM ;TRY TO LOOKUP SY:TSKMON.SAV
7 005416 103500 BCS 1# ;BR IF NOT THERE
8 ;
9 ; Get information about TSKMON out of block 0 of file
10 ;
11 005420 .READW #XAREA,#1,#BLKO,#256,#0 ;READ IN BLOCK 0
12 005456 016700 0000500 MOV BLKO+50,R0 ;GET TOP ADDRESS OF KMON
13 005462 062700 0000003 ADD #3,R0 ;BOUND UP TO NEXT WORD
14 005466 042700 0000001 BIC #1,R0 ;FORCE EVEN
15 005472 010067 0000000 MOV R0,KMNTOP
16 005476 162700 0000000 SUB #KMNBAS,R0 ;BASE ADDRESS OF KMON
17 005502 010067 0000000 MOV R0,KMNIHI ;TOP OF TSKMON-KMNBAS
18 005506 062700 0000777 ADD #511,R0 ;BOUND UP TO PAGE SIZE
19 005512 000241 CLC
20 005514 006000 ROR R0 ;CVT TO # WORDS
21 005516 000300 SWAB R0 ;CVT TO # PAGES
22 005520 042700 177400 BIC #^C377,R0
23 005524 062700 0000000 ADD #CXTPAG,R0 ;# PAGES NEEDED TO JOB CONTEXT AREA
24 005530 010067 0000000 MOV R0,KMNPGS ;# 256 WORD PAGES FOR KMON & CONTEXT
25 005534 016767 0000420 0000000 MOV BLKO+42,KMNSTK ;INITIAL STACK POINTER
26 005542 016767 0000400 0000000 MOV BLKO+40,KMNSTR ;STARTING ADDRESS
27 ;
28 ; Now do .SAVESTATUS so we can quickly reopen to kmon
29 ;
30 005550 .SAVEST #XAREA,#1,#KMNCHN ;SAVE STATUS
31 ;
32 ; Now purge and reopen channel 17 to the new file (for overlays).
33 ;
34 005570 .PURGE #17 ;Purge channel 17 (overlay channel)
35 005576 .REOPEN #XAREA,#17,#KMNCHN ;Open channel 17 to new file
36 ;
37 ; Exit to reload Kmon
38 ;
39 005616 .EXIT ;EXIT AND REENTER NEW VERSION OF TSKMON
40 ;
41 ; Error: Cannot find new Kmon file
42 ;
43 005620 1#: FABORT #NOKMON ;ERROR -- CAN'T FIND TSKMON
44 ;
45 ; -----
46 ; Process the SET CCL NEW command.
47 ; Begin to use a new version of CCL
48 ;
49 005630 004767 0000000 SETCNW: CALL CKSYPV ;Must have SYSPRV privilege
50 005634 .LOOKUP #XAREA,#1,#CCLNAM ;TRY TO LOOKUP SY:CCL.SAV
51 005654 103412 BCS 1# ;BR IF CAN'T FIND IT
52 005656 .SAVEST #XAREA,#1,#CCLSAV ;SAVE STATUS
53 005676 000167 0000000 JMP RDCMD ;FINISHED
54 005702 1#: FABORT #NOCCL ;CAN'T FIND CCL.SAV

```

LD

```

1
2
3
4
5 005712 005002
6 005714 020067 0000000
7 005720 001404
8 005722 010002
9 005724 166702 0000000
10 005730 006302
11
12
13
14 005732 012704 001224'
15 005736 004767 0000000
16 005742 103004
17 005744
18
19
20
21 005754 000134
22
23
24
25 005756 004767 0000000
26 005762 000167 0000000
27
28
29
30 005766 042762 0000000 0000000
31 005774 000167 0000000
32
33
34
35 006000 052762 0000000 0000000
36 006006 000167 0000000
37
38
39
40 006012 004767 000052
41 006016 004767 0000000
42 006022 000167 0000000

```

```

          .SBTTL      LD
          -----
          ; Process SET LDn command
          ;
          SETLD: CLR      R2          ; ASSUME UNIT # = 0
                CMP      R0,R5OLD    ; IS UNIT "LD"?
                BEQ      1$          ; BR IF YES
                MOV      R0,R2       ; GET DEVICE NAME
                SUB      R5OLD0,R2   ; SUBTRACT "LD0" TO GET UNIT # IN R2
                ASL      R2          ; CONVERT TO WORD TABLE INDEX
          ;
          ; Get the option word
          ;
          1$:  MOV      #LDOPHD,R4    ; POINT TO OPTION TABLE
                CALL     SEARCH      ; ACCRUE AND CHECK OPTION WORD
                BCC     2$          ; BR IF FOUND
                FABORT  #CSIMS1     ; INVALID OPTION WORD
          ;
          ; Jump off to option processing routine
          ;
          2$:  JMP      @(R4)+        ; ENTER PROCESSING ROUTINE
          ;
          ; SET LD CLEAN
          ;
          SLDCLN: CALL    LDCLEN      ; CLEAN UP LOGICAL DISKS
                JMP      RDCMD
          ;
          ; SET LDn WRITE
          ;
          SLDWRT: BIC     #LD*RON,LDFLAG(R2) ; CLEAR READ-ONLY FLAG
                JMP      RDCMD      ; FINISHED
          ;
          ; SET LDn NOWRITE
          ;
          SLDNWR: BIS     #LD*RON,LDFLAG(R2) ; SET READ-ONLY FLAG
                JMP      RDCMD      ; FINISHED
          ;
          ; SET LDn FREE
          ;
          SLDFRE: CALL    LDDMT       ; DISMOUNT THIS LD
                CALL    LDCLEN      ; NOW RESET ALL LOGICAL DISK ASSIGNMENTS
                JMP      RDCMD      ; FINISHED

```

LD

```

1      ;
2      ; SET LD EMPTY
3      ; (Dismount all logical disks)
4      ;
5 006026 SLDEMP:
6      ;
7      ; Begin loop to close log files and deassign any assignments for each
8      ; logical disk
9      ;
10 006026 005002          CLR      R2          ; Init index to 1st LD
11 006030 004767 000102 1$:    CALL    LOGDEA      ; Close any log files and make deassignments
12 006034 062702 000002          ADD     #2,R2      ; Get index for next LD
13 006040 020227 000016          CMP     R2,#14.     ; Have we done all?
14 006044 101771          BLOS    1$          ; Loop if not
15      ;
16      ; Dismount the all LD's
17      ;
18 006046 012700 000110'        MOV     #DMTALD,R0    ; Emt arg. block to dismount all LD's
19 006052 104375          ENT     375          ; Dismount all LD's
20 006054 103003          BCC     9$          ; Branch if all LD's dismounted
21 006056          .PRINT #EDMALD    ; "Unable to dismount all logical disks."
22      ;
23      ; Finished
24      ;
25 006064 000167 0000000 9$:   JMP     RDCMD

```

LD

```

1          ; -----
2          ; Subroutine to dismount a logical disk.
3          ;
4          ; Inputs:
5          ; R2 = LD unit index
6          ;
7 006070 010246 LDDMT:  MOV    R2, -(SP)
8 006072 004767 000040      CALL   LOGDEA      ;Close any log files and make deassignments
9          ;
10         ; Tell system to stop doing directory caching for device
11         ;
12 006076 112767 000001 0000000 MOVB   #1, SERFLO      ;Do .SERR to avoid abort for illegal device
13 006104 012700 0000000      MOV    #DMTARG, R0    ;Point to EMT arg block
14 006110 104375              EMT    375             ;Tell system to stop doing caching
15 006112 105067 0000000      CLRB   SERFLO        ;Do .HERR
16         ;
17         ; Clean out LD info tables
18         ;
19 006116 005062 0000000      CLR    LDPDEV(R2)    ;No physical device assignment
20 006122 072227 000002      ASH   #2, R2         ;Get index into LDNAME table
21 006126 005062 0000000      CLR    LDNAME(R2)   ;No file name
22         ;
23         ; Finished
24         ;
25 006132 012602          9$:  MOV    (SP)+, R2
26 006134 000207          RETURN

```

LD

```

1 ;-----
2 ;
3 ; Subroutine to check for open log files and deassign any assignments
4 ;
5 ; Inputs:
6 ; R1 = LD unit index
7 ;
8 006136 010246 LOGDEA: MOV R2, -(SP)
9 006140 010546 MOV R5, -(SP)
10 ;
11 ; See if this LD is in use
12 ;
13 006142 010200 MOV R2, R0 ;Get LD unit index
14 006144 072027 000002 ASH #2, R0 ;Cvt to index into name table
15 006150 005760 0000000 TST LDNAME(R0) ;Is this LD in use?
16 006154 001413 BEQ 9$ ;Br if not
17 ;
18 ; Build name of LD
19 ;
20 006156 010205 MOV R2, R5 ;Get LD unit index
21 006160 006205 ASR R5 ;Convert to unit number
22 006162 066705 0000000 ADD R5, LD0, R5 ;Add "LD0" to form unit name
23 006166 010567 0000000 MOV R5, MNTDEV ;Save name of device being dismantled
24 ;
25 ; Close the log file if it is on the device being dismantled
26 ;
27 006172 004767 0000000 CALL LOGCHK ;Close log file if it is on this LD
28 ;
29 ; Do DEASSIGN of anything assigned to this LD
30 ;
31 006176 010500 MOV R5, R0 ;Get LD name
32 006200 004767 0000000 CALL DEADEV ;Do deassign
33 ;
34 ; Finished
35 ;
36 006204 012605 9$: MOV (SP)+, R5
37 006206 012602 MOV (SP)+, R2
38 006210 000207 RETURN

```

SL

```

1          .SBTTL      SL
2          ;-----
3          ; SET SL
4          ;
5          ; Inputs:
6          ; R3 = Pointer to parameters following "SET SL".
7          ;
8 006212   SETSL:
9          ;
10         ; Process each of the command parameters
11         ;
12 006212   012704   001254'   MOV      #SLEHD,R4      ;Point to list of qualifiers
13 006216   004767   000000G   CALL     SCNOPS       ;Process the qualifiers
14         ;
15         ; Finished
16         ;
17 006222   000167   000000G   JMP      RDCMD
18
19
20         ;-----
21         ; SET SL ON
22         ;
23 006226   105767   000000G   SLOON:  TSTB      VSLEDT      ;Is SL available?
24 006232   001007           BNE      1$              ;Br if yes
25 006234           FWARN   #EM$NSL      ;SL not genned into system
26 006250   000425           BR       9$
27 006252   032761   000000C 000000G 1$:  BIT      #<VT100!VT200!VT52>,LTRMTP(R1) ;VT100, VT200 or VT52
28 006260   001007           BNE      2$              ;Br if yes
29 006262           FWARN   #EM$SLT      ;Invalid terminal type
30 006276   000412           BR       9$
31 006300   052761   000000G 000000G 2$:  BIS      ##SLON,LSW7(R1) ;Enable SL for this line
32 006306   032761   000000G 000000G   BIT      ##SLKED,LSW7(R1);Is Ked mode wanted?
33 006314   001403           BEQ     9$              ;Br if not
34 006316           .PRINT  #SLKDON      ;Enable terminal alternate keypad mode
35 006324   000207   9$:  RETURN
36         ;
37         ;-----
38         ; SET SL OFF
39         ;
40 006326           SLOFF: .PRINT  #SLKDOF      ;Return numeric keypad to normal mode
41 006334   042761   000000G 000000G   BIC     ##SLON,LSW7(R1) ;Disable SL for this line
42 006342   000207           RETURN
43         ;
44         ;-----
45         ; SET SL TTY
46         ;
47 006344   052761   000000G 000000G SLOTTY: BIS      ##SLTTY,LSW7(R1);Enable SL for .TTYIN
48 006352   000207           RETURN
49         ;
50         ;-----
51         ; SET SL NOTTY
52         ;
53 006354   042761   000000G 000000G SLONTT: BIC     ##SLTTY,LSW7(R1);Disable SL for .TTYIN
54 006362   000207           RETURN
55         ;
56         ;-----
57         ; SET SL WIDTH=n

```

SL

```

58 ;
59 006364 SLOWID: FWARN #EM$SLW ;Print warning message
60 006400 012705 0000000 MOV #SLMXLN,R5 ;Get SL width
61 006404 004767 0000000 CALL PRTDEC ;Print the value
62 006410 .PRINT #CRLF ;Terminate the line
63 006416 000207 RETURN
64 ;
65 ;-----
66 ; SET SL LET
67 ;
68 006420 052761 0000000 0000000 SLOLET: BIS ##SLLET,LSW7(R1);Enable LET feature
69 006426 000207 RETURN
70 ;
71 ;-----
72 ; SET SL NOLET
73 ;
74 006430 042761 0000000 0000000 SLOHLT: BIC ##SLLET,LSW7(R1);Disable LET feature
75 006436 000207 RETURN
76 ;
77 ;-----
78 ; SET SL KED
79 ;
80 006440 052761 0000000 0000000 SLOKED: BIS ##SLKED,LSW7(R1);Remember SL is in KED mode
81 006446 .PRINT #SLKDON ;Enable terminal alternate keypad mode
82 006454 000207 RETURN
83 ;
84 ;-----
85 ; SET SL RT11
86 ;
87 006456 042761 0000000 0000000 SLORT: BIC ##SLKED,LSW7(R1);Remember SL is not in KED mode
88 006464 .PRINT #SLKDOF ;Return numeric keypad to normal mode
89 006472 000207 RETURN
90 ;
91 ;-----
92 ; Unimplemented SL option
93 ;
94 006474 SLOUNI: FWARN #EM$UIO ;Unimplemented option
95 006510 000207 RETURN
96 ;
97 ;-----
98 ; Ignored option
99 ;
100 006512 000207 SLDNOP: RETURN
101 ;
102 ;-----
103 ; SET RECALL NORMAL/REVERSE
104 ;
105 006514 012704 001404' SETREC: MOV #RCLHD,R4 ;Point to option table
106 006520 004767 0000000 CALL SEARCH ;Accrue and check options
107 006524 103006 BCC 1$ ;Br if valid option
108 006526 FWARN #INVOPT ;Invalid option
109 006542 111467 0000000 1$: MOVB @R4,RCLREV ;Set into context cell
110 006546 000167 0000000 JMP RDCMD
111 ;
112 000001 .END

```

Errors detected: 0

SL

*** Assembler statistics

Work file reads: 0

Work file writes: 0

Size of work file: 11952 Words (47 Pages)

Size of core pool: 17920 Words (70 Pages)

Operating system: RT-11

Elapsed time: 00:01:53.46

DK: TSKST1, LP: TSKST1=DK: TSKST1. MAC/C/N: SYM

\$1STLQ	1-73				
\$BRIT	1-111				
\$AUTO	1-87				
\$CARUP	1-85				
\$CCLRN	1-86				
\$CFABT	1-106				
\$CFALL	1-112				
\$CFCLL	1-112				
\$CFDCC	1-112				
\$CFOPN	1-118				
\$CFSOT	1-110				
\$CHACT	1-61				
\$CLTST	1-96	5-45	5-46		
\$CTRLC	1-104				
\$CTRLD	1-154	10-38	10-40		
\$CTRLO	1-61				
\$CTRLS	1-91				
\$DBKMN	1-84	5-38	5-39		
\$DEAD	1-158				
\$DEBUG	1-155				
\$DEFER	1-124				
\$DETCH	1-89				
\$DIBOL	1-73	5-163	5-164		
\$DILUP	1-108				
\$DISCN	1-90				
\$DOOFF	1-114				
\$DUPRN	1-109				
\$ECHO	1-111				
\$EMTTR	1-95	5-170	5-171	10-48	
\$FORM	1-110				
\$FORMO	1-112				
\$HARD	1-158				
\$HITTY	1-72				
\$INCOR	1-128				
\$INDAB	1-159	5-62	5-63		
\$INDDF	1-157	5-32	5-33	14-86	14-96
\$INDRN	1-157				
\$INIT	1-158				
\$INKMN	1-104				
\$KED	1-128	14-43	14-48	14-56	
\$KINIT	1-68				
\$LC	1-111				
\$LOFCF	1-203				
\$MLOCK	1-77				
\$NOIN	1-72	17-6			
\$NOINT	1-204				
\$NOWTT	1-72				
\$PAGE	1-111				
\$PHONE	1-158				
\$PRGLK	1-87				
\$PWKEY	1-32	20-87	20-88		
\$QTSET	1-133				
\$QUIET	1-125				
\$RNIOP	1-205				
\$SCOPE	1-111				
\$SGALL	1-124	5-202			

CPUAL	1-181						
CR	2-5#						
CRLF	1-176	9-27	10-18	21-14	21-21	22-14	34-62
CS\$RON	1-80						
CSHALC	1-103	10-9					
CSHDEV	1-108						
CSHDVN	1-108						
CSHHD	1-163						
CSHMSG	1-203						
CSHSIZ	1-143						
CSIMS1	1-175	13-39	30-17				
CSIMS2	1-173	7-44	11-36				
CSIMS4	1-195						
CTDHD	4-12	5-232#					
CTRLTT	1-102						
CURMTX	1-186						
CURPRM	1-130						
CVDVNM	1-183						
CVTTAB	1-164	7-5					
CVTUC	1-163						
CW\$50H	1-113						
CW\$PRO	1-62						
CXTBAS	1-107						
CXTPAG	1-79	29-23					
CXTRMN	1-41	24-26					
CXTWDS	1-107						
DATTIM	1-27						
DCCRD	1-137						
DCCWR	1-137						
DCHNEW	6-33#	10-26					
DCTRD	1-137						
DCTWR	1-137						
DEADEV	1-171	33-32					
DELSPC	1-185						
DETARG	1-200						
DETHD	1-200						
DETTXT	1-179						
DEVHD1	1-182						
DEVIDL	1-187	1-187	1-188				
DEVUNT	1-174	11-20*	12-85				
DFJMEM	1-68						
DIABFL	1-125						
DIABLO	1-147						
DIABNO	1-126						
DIVIDE	1-180						
DIVSOR	1-193						
DJABMS	1-191						
DKASHD	1-59						
DKSAV	1-164						
DLCEMT	1-28						
DLMSG	1-191						
DLTXT	1-178						
DMTALD	6-60#	31-18					
DMTALL	1-191						
DMTARG	1-170	32-13					
DMTSUB	1-196						

DDASGN	1-85		
DORUN	1-27		
DOSTOP	1-192		
DVFLAG	1-32	13-22	
DVSHH1	1-59		
DVSHH2	1-59		
DVSHH3	1-59		
DVSTAT	1-76		
DX#NST	1-32	13-22	
DZTXT	1-198		
EDIT	1-73		
EDITHD	4-13	5-4#	
EDMALD	1-42	31-21	
EDTFIL	1-184		
EM#ACL	1-54		
EM#CAP	1-44		
EM#CIP	1-53	22-18	
EM#CLB	1-54		
EM#CLN	1-53		
EM#CLX	1-36		
EM#CNO	1-44		
EM#CPO	1-44		
EM#CSE	1-70		
EM#HNI	1-68	13-16	
EM#ICL	1-52		
EM#IDR	1-42		
EM#ILN	1-53	1-54	
EM#IST	1-42		
EM#IUN	1-53		
EM#NAD	1-46	10-37	10-47
EM#NPD	1-53	10-34	
EM#NPR	1-45		
EM#NSF	1-53	26-25	27-26
EM#NSL	1-54	34-25	
EM#NUK	1-86	28-51	
EM#OPR	1-36	11-45	
EM#PTA	1-36	20-40	
EM#PTU	1-36	20-42	
EM#SLT	1-54	18-46	34-29
EM#SLW	1-54	34-59	
EM#SPL	1-39	16-19	
EM#STL	1-42	27-33	
EM#TSL	1-54		
EM#UID	1-55	34-94	
EM#WCO	1-40	18-96	
EM#WC1	1-40	18-97	
EM#WC2	1-40	18-98	
EM#WC3	1-40	18-99	
EM#WCM	1-40	18-11	
EMTHD	4-14	5-169#	
ERRHD	4-16	5-13#	
ERRLOC	1-67	18-86	
ERRSEV	1-134	23-5#	
ESC	1-82		
FC#CDX	1-163		
FC#LNK	1-163		

INVLDO	8-57#	12-26										
INVOPT	1-165	1-171	1-186	8-58	34-108							
INVSOP	8-55	8-58#										
INVTIM	1-191											
IDABFL	1-61	5-177	5-178									
IDABHD	4-22	5-176#										
ITRMTP	1-159											
JCXPFS	1-142											
JCXSMS	1-198											
JPWDEV	1-39	20-24*										
JPWFLG	1-39	20-36*	20-51*	20-56*	20-61*	20-66*	20-71*	20-76*				
JPWTYP	1-39	20-35*										
JSTKND	1-102											
JSWLOC	1-67											
K52	1-73											
KBMSG	1-180											
KBTX	1-184											
KCSIBF	1-169											
KCSIMS	1-170											
KDOCIN	1-26											
KED	1-73											
KILEMT	1-192											
KL3CLR	1-87											
KL4CLR	1-127											
KMNBAS	1-154	29-16										
KMNCHN	1-94	12-96	29-30	29-35								
KMNHI	1-78	29-17*										
KMNNAM	1-196	29-6										
KMNPFS	1-79	29-24*										
KMNSTK	1-79	29-25*										
KMNSTR	1-79	29-26*										
KMNTOP	1-79	29-15*										
KMONHD	4-23	5-31#										
KMPRMT	1-138	15-10										
L	4-9	4-9#	4-10	4-10#	4-11	4-11#	4-12	4-12#	4-13	4-13#	4-14	4-14#
	4-15	4-15#	4-16	4-16#	4-17	4-17#	4-18	4-18#	4-19	4-19#	4-20	4-20#
	4-21	4-21#	4-22	4-22#	4-23	4-23#	4-24	4-24#	4-25	4-25#	4-26	4-26#
	4-27	4-27#	4-28	4-28#	4-29	4-29#	4-30	4-30#	4-31	4-31#	4-32	4-32#
	4-33	4-33#	4-34	4-34#	4-35	4-35#	4-36	4-36#	4-37	4-37#	4-38	4-38#
	4-39	4-39#	4-40	4-40#	4-41	4-41#	4-42	4-42#	4-43	4-43#	4-44	4-44#
	4-45	4-45#	4-46	4-46#	4-47	4-47#	4-48	4-48#	4-49	4-49#	4-50	4-50#
	4-51	4-51#	4-52	4-52#	4-53	4-53#	4-54	4-54#	4-55	4-55#	4-56	4-56#
	4-57	4-57#	4-58	4-58#	4-59	4-59#	5-5	5-5#	5-6	5-6#	5-7	5-7#
	5-8	5-8#	5-14	5-14#	5-15	5-15#	5-16	5-16#	5-17	5-17#	5-18	5-18#
	5-19	5-19#	5-25	5-25#	5-26	5-26#	5-32	5-32#	5-33	5-33#	5-34	5-34#
	5-35	5-35#	5-36	5-36#	5-37	5-37#	5-38	5-38#	5-39	5-39#	5-45	5-45#
	5-46	5-46#	5-47	5-47#	5-53	5-53#	5-54	5-54#	5-55	5-55#	5-56	5-56#
	5-62	5-62#	5-63	5-63#	5-69	5-69#	5-70	5-70#	5-71	5-71#	5-72	5-72#
	5-73	5-73#	5-79	5-79#	5-80	5-80#	5-81	5-81#	5-82	5-82#	5-83	5-83#
	5-84	5-84#	5-85	5-85#	5-86	5-86#	5-87	5-87#	5-88	5-88#	5-89	5-89#
	5-90	5-90#	5-91	5-91#	5-92	5-92#	5-93	5-93#	5-94	5-94#	5-95	5-95#
	5-96	5-96#	5-97	5-97#	5-98	5-98#	5-99	5-99#	5-105	5-105#	5-106	5-106#
	5-112	5-112#	5-113	5-113#	5-114	5-114#	5-115	5-115#	5-116	5-116#	5-117	5-117#
	5-118	5-118#	5-119	5-119#	5-120	5-120#	5-121	5-121#	5-122	5-122#	5-123	5-123#
	5-129	5-129#	5-130	5-130#	5-131	5-131#	5-132	5-132#	5-133	5-133#	5-134	5-134#
	5-135	5-135#	5-136	5-136#	5-137	5-137#	5-138	5-138#	5-144	5-144#	5-145	5-145#

LMXLN	1-158									
LMXNUM	1-156									
LMXPRM	1-159									
LNAME	1-60									
LNBLKS	1-107									
LNMAP	1-113									
LNPRIM	1-113									
LNSBLK	1-108									
LNSPAC	1-117									
LOCKTX	1-180									
LOFSPC	1-110	26-46								
LOGASN	1-172	25-22	26-41	28-25						
LOGBAS	1-132	1-134	25-45*	25-49*						
LOGBLK	1-161	24-24*	25-60*							
LOGBUF	1-161	24-23	25-59							
LOGCHK	1-133	33-27								
LOGCHN	1-161	24-25	25-32	25-32						
LOGCLS	1-176	24-15	25-6							
LOGDEA	31-11	32-8	33-8#							
LOGDVU	1-132	1-134	25-43*	25-44*	25-48*					
LOGFLG	1-161	24-21	24-40*	24-45*	24-50*	24-51*	24-56*	24-57*	24-62*	25-58*
LOGPTR	1-161	24-23*	25-59*							
LOMAP	1-142									
LOTBUF	1-81									
LOTNXT	1-81									
LOTPNT	1-81									
LOTSIZ	1-82									
LOTSPC	1-82									
LOUTIR	1-65									
LP#7BT	1-52									
LP#ODD	1-52									
LP#PAR	1-52									
LP#SPD	1-52									
LPRG1	1-145									
LPRG2	1-145									
LPRI	1-160									
LPROG	1-90									
LPROJ	1-90									
LRBFIL	1-114									
LRDTIM	1-80									
LSCCA	1-110									
LSECPT	1-95									
LSPEMT	6-27#									
LSTACT	1-78									
LSTATE	1-140									
LSTDL	1-89									
LSTHL	1-62									
LSTIOL	1-62									
LSTMX	1-156									
LSTPL	1-135									
LSTPRM	1-131									
LSTSL	1-140									
LSTSPL	1-76									
LSUCF	1-86									
LSW	1-61									
LSW11	1-32	20-02*	20-88*							

LSW2	1-104												
LSW2S	1-109												
LSW3	1-109	17-6*											
LSW4	1-127												
LSW5	1-87	5-25	5-26	5-32	5-33	5-45	5-46	14-43*	14-48*	14-49*	14-55*	14-56*	
	14-86*	14-96*											
LSW6	1-155	5-163	5-164	5-170	5-171	10-48*							
LSW7	1-159	5-34	5-53	5-54	5-55	5-56	5-62	5-63	14-61*	14-62*	14-67*	14-68*	
	14-73*	14-74*	14-79*	28-45*	34-31*	34-32	34-41*	34-47*	34-53*	34-68*	34-74*	34-80*	
	34-87*												
LSW8	1-123	5-184	5-185	5-186	5-187	5-188	5-189	5-190	5-191	5-192	5-193	5-194	
	5-195	5-196	5-197	5-198	5-199	5-200	5-201	5-202					
LSW9	1-85	5-38	5-39	10-38*	10-40*	17-7*	26-21	27-22					
LTRMTP	1-148	18-44	34-27										
LTSCMD	1-117												
LUNAME	1-90												
MAXALC	1-63												
MAXASN	1-106												
MAXAVL	1-174	9-25	10-16										
MAXMEM	1-67												
MAXMTX	1-186												
MAXPRI	1-67	1-160	21-6	22-6	22-12								
MAXSEC	1-95												
MAXWEM	18-87	18-100#											
MDT	1-77												
MHNSIZ	1-96												
MHNSMS	1-97												
MINTIM	1-95												
MISSEQ	1-175	14-39	28-55										
MNBASE	1-185												
MNBPIC	1-184												
MNFLGS	1-184												
MNTARG	1-195												
MNTDEV	1-170	33-23*											
MNTFUL	1-172												
MNTOP	1-185												
MNTTXT	1-197												
MONAR1	1-185												
MONAR2	1-185												
MONHD	1-185												
MONVEC	1-160	14-87	14-90	14-97	14-100	23-9							
MSOBUF	1-190												
MSGEND	1-190												
MTOPHD	1-171												
MUL32	1-200												
MXCSR	1-156												
MXDTR	1-156												
MXJADR	1-69												
MXJMEM	1-68												
MXJPRI	1-160	21-12	21-16	21-19	21-22	22-16	22-19*						
MXPRMT	1-138	15-36											
MXVEC	1-158												
NAMTOP	1-144												
NARGS	4-8#	4-9	4-9	4-10	4-10	4-11	4-11	4-12	4-12	4-13	4-13	4-14	
	4-14	4-15	4-15	4-16	4-16	4-17	4-17	4-18	4-18	4-19	4-19	4-20	
	4-20	4-21	4-21	4-22	4-22	4-23	4-23	4-24	4-24	4-25	4-25	4-26	

RT#NAM	1-139					
RUNDEV	1-107					
RUNEMT	1-168					
RUNHD	1-164					
RUNMS	1-200					
S#HICP	1-92					
S#INWT	1-93					
S#IOFN	1-88					
S#IOWT	1-145					
S#LOW	1-90					
S#MSWT	1-94					
S#OTFN	1-88					
S#OTLO	1-88					
S#OTWT	1-93					
S#QMIO	1-86					
S#QUSR	1-145					
S#RT	1-90					
S#SFWT	1-93	1-145				
S#SPCB	1-146					
S#SPDB	1-146					
S#SPND	1-87					
S#TMWT	1-93					
S#TTFN	1-88					
S#TTSC	1-89					
S#TWFN	1-88					
S9600	1-93					
SBPFIL	5-227	27-17#				
SBPHD	5-226#	27-11				
SBPSUF	1-36	27-38				
SC#ERR	1-33	5-15				
SC#FTL	1-33	5-17				
SC#NON	1-33	5-19				
SC#SEV	1-33	5-16				
SC#SUC	1-33					
SC#UNC	1-33	5-18				
SC#WRN	1-33	5-14				
SCHAIN	1-124					
SCNOPS	1-37	18-22	20-10	24-10	27-12	34-13
SD#BAK	1-127					
SD#DEL	1-119					
SD#FLK	1-120					
SD#HLD	1-130					
SD#SNG	1-129					
SD#WFM	1-120					
SDBLK	1-121					
SDBU	1-127					
SDBUF1	1-121					
SDCB	1-135					
SDCBND	1-135					
SDCBSZ	1-140					
SDDVU	1-139					
SDFHD	1-132					
SDFLAG	1-120					
SDFORM	1-120					
SDNAME	1-140					
SDSFCB	1-119					

SPFUL	1-189		
SPGEMT	1-189		
SPLACT	1-192		
SPLCHN	1-92		
SPLHD	1-186		
SPLHLA	1-177	25-53	
SPLPND	1-194		
SPSNG	1-187	1-189	
SPUBUF	1-69		
SPWFM	1-187	1-189	
SRTSIZ	1-143		
SRTSMS	1-194		
SRTTXT	1-197		
SSRMAP	1-197		
STCDF	5-234	10-40#	
STCDON	5-233	10-32#	
STCHNB	4-9	10-4#	
STLGAL	5-208	24-62#	
STLGCL	5-210	24-15#	
STLGCN	1-27	5-209	24-20#
STLGFL	5-211	25-4#	
STLGHD	1-176	5-207#	24-9
STLGIN	5-212	24-50#	
STLGNW	5-215	24-45#	
STLGOT	5-213	24-56#	
STLGWR	5-214	24-40#	
STLOFL	5-221	26-21#	
STLOHD	5-220#	26-10	
STNOVR	1-34	4-56	
STPASK	1-194		
STPFLQ	1-92	23-17#	23-25#
STRWRD	5-177	5-178	14-22#
STUCLF	5-53	14-61#	
STUCLL	5-55	14-73#	
STUCLM	5-54	14-67#	
STUCLN	5-56	14-79#	
STVRFY	1-34	4-55	
SUBARO	1-184		
SUBTXT	1-197		
SUCS	1-144		
SUM1	1-193		
SUM2	1-193		
SUM3	1-193		
SUM4	1-193		
SUM5	1-194		
SUM6	1-194		
SUM7	1-194		
SUMS	1-144		
SUPCOD	1-144		
SWPTX	1-180		
SXBPNT	1-69		
SYASHD	1-59		
SYHD1	1-179		
SYHD2	1-179		
SYINDX	1-151		
SYNAME	1-152	28-31	

SYPSWD	1-39	16-24
SYSAV	1-164	
SYSDAT	1-141	
SYTIMH	1-141	
SYTIML	1-141	
SYUNIT	1-151	
TAB	2-8#	
TALEMT	1-114	
TBLQVF	1-173	
TC132	6-21#	18-60
TC80	6-22#	18-66
TCNORM	6-20#	18-78
TCREV	6-19#	18-72
TECO	1-73	
TK1SEC	1-143	
TK1VAL	1-141	
TM\$AUT	1-57	
TM\$CDS	1-56	
TM\$CEN	1-56	
TM\$CLO	1-58	
TM\$CL1	1-58	
TM\$CL2	1-58	
TM\$CL3	1-58	
TM\$CL4	1-58	
TM\$CL5	1-58	
TM\$CL6	1-58	
TM\$CNG	1-56	
TM\$GBL	1-49	
TM\$HPE	1-56	
TM\$HPR	1-55	
TM\$IN1	1-38	
TM\$IN2	1-38	
TM\$LCL	1-49	
TM\$LPR	1-55	
TM\$MRS	1-32	13-24
TM\$NAD	1-60	
TM\$NNR	1-38	
TM\$NSD	1-60	
TM\$PR1	1-55	
TM\$PR2	1-55	
TM\$PVA	1-44	
TM\$PVC	1-44	
TM\$RD1	1-49	
TM\$RD2	1-49	
TM\$SDN	1-60	
TM\$XBK	1-27	
TMIDLH	1-71	
TMIOH	1-71	
TMIOWH	1-70	
TMSWPH	1-71	
TMSWTH	1-71	
TMTOTH	1-70	1-193
TMTOTL	1-70	1-193
TMUSRH	1-70	
TOTMMS	1-197	
TOTON	1-92	

TOTXT	1-176	21-11	22-11
TRGRET	1-144		
TRMHD1	1-56		
TRMHD2	1-56		
TRMSTR	1-166		
TSKST1	1-5#	1-26	
TSR	1-156		
TSXLN	1-144		
TSXSIT	1-144		
TSXSMS	1-198		
TXTCL	1-91		
UC#MDC	1-163		
UC#NDC	1-163		
UCHAN	1-112		
UCIDEF	1-64	28-30	
UCISPC	1-96	28-36	28-42
UCLBLK	1-162		
UCLCMD	1-26		
UCLDAT	1-162		
UCLHD	4-53	5-52#	
UCLNAM	1-117		
UERSEV	1-134		
UFORM	1-104		
UFPTRP	1-119		
UHIMEM	1-107		
UKMNAM	1-85		
UMSSMS	1-197		
UNSYTP	1-89		
UPTMMS	1-192		
USPLCH	1-92		
USRMMMS	1-198		
USRSTK	1-68		
USTART	1-103		
UTRPAD	1-67		
VCORTM	1-138	4-11	
VCSHNB	1-95	10-22#	
VDBFLG	1-82	10-32	
VHIPCT	1-136	4-18	
VIMAGE	1-102		
VINTIO	1-128	4-21	
VLDSYS	1-86	7-56	
VNUMDC	1-138	9-42	
VOFFTM	1-205	4-29	
VPRIDF	1-42	22-15	
VQUANO	1-135	1-136	4-35
VQUAN1	1-136	4-36	
VQUAN2	1-136	4-40	
VQUAN3	1-136	1-136	4-41
VQUN1A	1-136	4-37	
VQUN1B	1-128	4-38	
VQUN1C	1-128	4-39	
VSLEDT	1-203	34-23	
VT100	1-147	18-44	34-27
VT10FL	1-149		
VT10ND	1-149		
VT200	1-146	18-44	34-27

... CM1	11-28 29-35	11-30 29-50	11-41 29-52	12-75	12-96	12-97	13-5	25-32	28-42	29-6	29-11	29-30
... CM2	11-28 13-5 25-32 29-50	11-28 13-5 28-42 29-52	11-30 14-87 28-42	11-30 14-90 29-6	11-30 14-90 29-6	11-30 14-97 29-11	11-41 14-100 29-11	12-75 14-100 29-11	12-96 23-9 29-11	12-97 23-9 29-30	13-5 25-32 29-35	13-5 25-32 29-50
... CM3	8-57	11-35	11-44	12-74	12-95	13-6	13-38	28-44	29-34			
... CM5	9-25 14-90 23-9 34-62	9-27 14-97 25-32 34-81	10-16 14-100 28-42 34-88	10-18 18-60 29-6	11-28 18-66 29-11	11-30 18-72 29-30	11-41 18-78 29-35	12-75 21-11 29-50	12-96 21-14 29-52	12-97 21-21 31-21	13-5 22-11 34-34	14-87 22-14 34-40
... CM6	14-87	14-90	14-97	14-100	23-9							
... CM7	11-30	13-5	29-11									
. CLOSE	1-20#	13-6	28-44									
. CSIGE	1-18#											
. CSISP	1-16#											
. DATE	1-19#											
. ENTER	1-21#	25-32										
. EXIT	1-21#	29-39										
. FPROT	1-22#											
. GTIM	1-19#											
. GTLIN	1-19#											
. GVAL	1-22#	14-87	14-97									
. HERR	1-22#											
. LOOKU	1-20#	11-28	28-42	29-6	29-50							
. PRINT	1-20# 22-11	9-25 22-14	9-27 31-21	10-16 34-34	10-18 34-40	18-60 34-62	18-66 34-81	18-72 34-88	18-78	21-11	21-14	21-21
. PURGE	1-17#	8-57	11-35	11-44	12-74	12-95	13-38	29-34				
. PVAL	1-22#	14-90	14-100	23-9								
. READW	1-17#	11-30	29-11									
. REOPE	1-18#	12-75	12-96	12-97	29-35							
. SAVES	1-18#	11-41	29-30	29-52								
. SERR	1-22#											
. SPFUN	1-19#											
. SRESE	1-16#											
. TTOUT	1-16#											
. TTYIN	1-17#											
. TTYOU	1-17#											
. WRITW	1-21#	13-5										
CMDDEF	3-47#	4-9	4-10	4-11	4-12	4-13	4-14	4-15	4-16	4-17	4-18	4-19
	4-20	4-21	4-22	4-23	4-24	4-25	4-26	4-27	4-28	4-29	4-30	4-31
	4-32	4-33	4-34	4-35	4-36	4-37	4-38	4-39	4-40	4-41	4-42	4-43
	4-44	4-45	4-46	4-47	4-48	4-49	4-50	4-51	4-52	4-53	4-54	4-55
	4-56	4-57	4-58	4-59	5-5	5-6	5-7	5-8	5-14	5-15	5-16	5-17
	5-18	5-19	5-25	5-26	5-32	5-33	5-34	5-35	5-36	5-37	5-38	5-39
	5-45	5-46	5-47	5-53	5-54	5-55	5-56	5-62	5-63	5-69	5-70	5-71
	5-72	5-73	5-79	5-80	5-81	5-82	5-83	5-84	5-85	5-86	5-87	5-88
	5-89	5-90	5-91	5-92	5-93	5-94	5-95	5-96	5-97	5-98	5-99	5-105
	5-106	5-112	5-113	5-114	5-115	5-116	5-117	5-118	5-119	5-120	5-121	5-122
	5-123	5-129	5-130	5-131	5-132	5-133	5-134	5-135	5-136	5-137	5-138	5-144
	5-145	5-146	5-147	5-148	5-149	5-150	5-151	5-152	5-153	5-154	5-155	5-156
	5-157	5-163	5-164	5-170	5-171	5-177	5-178	5-184	5-185	5-186	5-187	5-188
	5-189	5-190	5-191	5-192	5-193	5-194	5-195	5-196	5-197	5-198	5-199	5-200
	5-201	5-202	5-208	5-209	5-210	5-211	5-212	5-213	5-214	5-215	5-221	5-227
	5-233	5-234										
FABORT	3-15#	7-44	8-56	8-58	10-34	10-37	10-47	11-36	11-45	13-33	13-39	14-39

	14-92	16-19	18-11	18-46	18-92	22-18	25-34	25-70	26-25	26-57	27-26	27-33
	27-50	28-51	28-55	28-59	29-43	29-54	30-17					
FERR	3-4#	15-3R	20-40	20-42	21-8	21-18	22-8					
FWARN	3-23#	13-16	13-24	34-25	34-29	34-59	34-94	34-108				
TBLDEF	3-35#	4-8	5-4	5-13	5-24	5-31	5-44	5-52	5-61	5-68	5-78	5-104
	5-111	5-128	5-143	5-162	5-169	5-176	5-183	5-207	5-220	5-226	5-232	
TBLEND	3-61#	4-60	5-9	5-20	5-27	5-40	5-48	5-57	5-64	5-74	5-100	5-107
	5-124	5-139	5-158	5-165	5-172	5-179	5-203	5-216	5-222	5-228	5-235	