TSKSHO -- Keyboard SHOW Command MACRO V05.04 Monday 21-Dec-87 13:04
Table of contents

```
                                        .TITLE  TSKSHO -- Keyboard SHOW Command Routines
                                        .ENABL  LC
                                        .DSABL  GBL
000000                                  .CSECT  TSKSHO
000000                          TSKSHO:
                                ;
                                ;  TSKSHO is the portion of TSKMON that contains the code
                                ;  to implement the SHOW command.
                                ;
                                ;  Copyright 1978, 1979, 1980, 1981, 1982, 1983, 1984,1985.
                                ;  S&H Computer Systems, Inc.
                                ;  Nashville, Tennessee
                                ;
                                ;  Macro calls
                                ;
                                        .MCALL  .CSISPC,.TTOUTR,.SRESET
                                        .MCALL  .READW,.TTYIN,.TTYOUT,.PURGE
                                        .MCALL  .CSIGEN,.SAVEST,.REOPEN
                                        .MCALL  .GTLIN,.GTIM,.DATE,.SPFUN
                                        .MCALL  .PRINT,.CLOSE,.LOOKUP
                                        .MCALL  .WRITW,.ENTER,.EXIT
                                        .MCALL  .SERR,.HERR,.FPROT,.GVAL,.PVAL
                                ;
                                ;  Global definitions
                                ;
                                        .GLOBL  TSKSHO,CMDHD,CMDOFF,KDOCIN,SKPSPC,UCLCMD
                                        .GLOBL  DORUN,CMDFRM,CMDDSN,STLGCN,DATTIM,PRGALL
                                        .GLOBL  DLCEMT,ALCDEV,CMDSHO,CMDSET,CMDWHO,CMDMEM,CMDUSE
                                ;
                                ;  Global references
                                ;
                                        .GLOBL  AF$DUP,AF$IND,AF$UCL,AF$SET,AF$CCA,TSXVRS,AF$NPW
                                        .GLOBL  CKCLUS,SHOKEY,HANIOC,SYPSWD,TM$NSP,TM$SL1,TM$OFF,TM$ON
                                        .GLOBL  TM$KED,TM$TTY,TM$SUB,TM$NO,VMXMSG,VMAXMC,VMXMRB
                                        .GLOBL  PEKEMT,PEKADR,PEKSIZ,TM$NNR,CDBUF,CDGET,TM$IN1,TM$IN2
                                        .GLOBL  CLRPRV,OPTLST,PFSO,PFCO,PVNPW,TSXVER
                                        .GLOBL  PO$SYS,PO$SPV,PRIVSO,PRVOPT
                                        .GLOBL  TM$PVA,TM$PVC,PRIVAO,EM$CNO,EM$CPO,EM$CAP,RSTPRV
                                        .GLOBL  CHKEQ,CKACOJ,PO$OPR,CKSYPV,AF$BYA,TM$PVL,DMYDEV,AF$TPO
                                        .GLOBL  INSTBL,INGADR,INGEMT,IIBUF,II$NAM,II$FLG,II$$SZ,EM$NAD
                                        .GLOBL  INSTBN,AF$SCA,AF$NOW,AF$MEM,PO$DBG,PRIVCO,PRVLST
                                        .GLOBL  ABRTAD,ABRTCD,CINFLG,$VNOTT,VPRILO,VPRIHI,II$PRV,II$NPV
                                        .GLOBL  TM$RD1,TM$RD2,TM$LCL,TM$GBL,SPACE1,RC$OWN,RC$CNT
                                        .GLOBL  RC$EXC,RC$AGE,RC$AEP,RC$USE,RC$FLG,RC$GBL,RC$NAM
                                        .GLOBL  RC$LEN,RC$PVT,RCBBAS,RCBEND,RC$$SZ,SHRRCB,SHRRCN
                                        .GLOBL  LP$SPD,LP$PAR,LP$ODD,LP$7BT,EM$ICL,PROSLT,RC$LCG
                                        .GLOBL  EM$NPD,EM$ILN,EM$CIP,EM$NSF,EM$IUN,EM$CLN
                                        .GLOBL  EM$ILN,EM$ACL,EM$TSL,EM$CLB,EM$NSL,EM$SLT,EM$SLW
                                        .GLOBL  SLKDON,SLKDOF,EM$UIO,TM$PR1,TM$PR2,TM$LPR,TM$HPR
                                        .GLOBL  TM$HPE,TM$CNG,TM$CDS,TM$CEN,TRMHD1,TRMHD2
                                        .GLOBL  OPRTXT,CLLINE,LCLTXT,REMTXT,TM$AUT,CLFREE,CLUNIT,CLVERS
                                        .GLOBL  TM$CL0,TM$CL1,TM$CL2,TM$CL3,TM$CL4,TM$CL5,TM$CL6,TM$CL7
                                        .GLOBL  TM$CL8,QHDMS1,QHDMS2,DVSHH1,DVSHH2,DVSHH3,SYASHD,DKASHD
                                        .GLOBL  TM$NAD,ALCHD1,ALCHD2,TM$NSD,TM$SDN,LNAME,TM$C13
                                        .GLOBL  CORUSR,LSW,$CTRLO,SERFLG,IOABFL,$CHACT,$STSNG
                                        .GLOBL  LSTHL,LCLUNT,FSTIOL,LSTIOL,CL$LIX,CW$PRO,CONFG2
                                        .GLOBL  CL$RQH,CL$WQH,MAXALC,ALCTBL,ALCEND
```

```
                .GLOBL  AD$DVU, AD$JOB, AD$$SZ, UCIDEF, HANCHN
                .GLOBL  NEDCHR, LOUTIR, LINIR, LINRTS, CLOTIR
                .GLOBL  CO$DEF, CL$COL, LCDTYP, SOPALC, SOPDAT, SOPTIM
                .GLOBL  UTRPAD, JSWLOC, ERRLOC, MAXMEM, MAXPRI
                .GLOBL  USRSTK, $KINIT, CFSTK, MXJMEM, DFJMEM, EM$HNI
                .GLOBL  SPUBUF, SXBPNT, VSWPFL, MXJADR, CLSFCH
                .GLOBL  TMTOTH, TMTOTL, TMUSRH, TMIOWH, LDMNT, EM$CSE
                .GLOBL  TMSWTH, TMIDLH, TMIOH, TMSWPH, LDCLEN
                .GLOBL  WILDFL, $NOIN, $NOWTT, $HITTY
                .GLOBL  TECO, EDIT, KED, K52, $1STLG, $DIBOL
                .GLOBL  SH$VAL, SH$NAM, SH$$SZ, SH$RTN, SH$FLG
                .GLOBL  SO$NVL, SO$OCT, SO$NO, HANENT, HANSIZ
                .GLOBL  H.CSR, H.VEC, DVSTAT, SFID, ACRSPD, HANPAR, LSTSPL
                .GLOBL  HAZEL, HAZLFL, HAZLNO, $MLOCK, MDT, GETKCH
                .GLOBL  LINBUF, LINNXT, LSTACT, PRGTOP, PRGSIZ, KMNHI
                .GLOBL  KMNTOP, KMNPGS, KMNSTK, KMNSTR, CXTPAG, FSTIOL
                .GLOBL  LINPNT, LINCNT, LACTIV, LRDTIM, CS$RON
                .GLOBL  LOTBUF, LOTNXT, LOTPNT, $VTESC
                .GLOBL  LOTSIZ, LOTSPC, LCOL, $SLKED, ESC
                .GLOBL  LAFSIZ, LFWLIM, LINCUR, NUMON, ILSW2
                .GLOBL  VPRIDF, VPR1VR, $DBKMN
                .GLOBL  $CARUP, DOASGN, UKMNAM, $UKMON, LSW9
                .GLOBL  LSUCF, $CCLRN, EM$NUK, S$QMIO, S$$RUN
                .GLOBL  KL3CLR, $PRGLK, LSW5, PVON, S$SPND, $AUTO
                .GLOBL  S$TWFN, S$TTFN, S$OTFN, S$IOFN, S$OTLO
                .GLOBL  LSTDL, FSTDL, $DETCH, UMSYTP, S$TTSC
                .GLOBL  $DISCN, LPROJ, LPROG, LUNAME, S$RT, S$LOW
                .GLOBL  LCPUHI, LCPULO, LCONTM, $CTRLS, $SPLJB, TXTCL
                .GLOBL  STPFLG, TOTON, USPLCH, SPLCHN, S$HICP, TXTC1
                .GLOBL  S$INWT, S$OTWT, S$TMWT, S$SFWT, S9600
                .GLOBL  S$MSWT, CFBUF, CFEND, CCLSAV, KMNCHN
                .GLOBL  MINTIM, LSECPT, MAXSEC, $EMTTR, VCSHNB
                .GLOBL  OKFILE, OKFEND, $CLTST, UCISPC, MHNSIZ
                .GLOBL  CASTBR, CASCBR, CASTBW, CASCUP, MHNSMS
                .GLOBL  CASTRO, CASTWO, CLTOTL, CO$DTR, CLSFSP
                .GLOBL  CO$CR, CO$FF, CO$FFO, CO$LC, CO$TAB, CO$CTL
                .GLOBL  CO$LFI, CO$LFO, CO$BNI, CO$BNO, CL$OPT
                .GLOBL  CL$LEN, CL$SKP, CL$WID, CL$LIN, PHYMEM
                .GLOBL  LJSW, CTRLTT, NEWJSW, JSTKND, VIMAGE
                .GLOBL  USTART, GENTOP, BOTDEV, BOTUNI, CSHALC
                .GLOBL  $CTRLC, LSW2, $INKMN, CHAIN, UFORM
                .GLOBL  $SGQ0, $SGQ3, LITIME
                .GLOBL  MAXASN, AT$$SZ, $CFABT, INDSTA, INDERR
                .GLOBL  RUNDEV, LNBLKS, CXTBAS, CXTWDS, UHIMEM
                .GLOBL  ASNTBL, $DILUP, CSHDEV, CSHDVN, LNSBLK
                .GLOBL  ASNEND, LSW3, LSW2S, $DUPRN
                .GLOBL  $FORM, $TAB, LSCCA, $CFSOT, LOFSPC, R50COM
                .GLOBL  $PAGE, $SCOPE, $ECHO, $LC, $8BIT, CHKALC
                .GLOBL  UCHAN, $FORMO, $CFALL, $CFDCC, $CFCCL
                .GLOBL  LNPRIM, LNMAP, CW$50H, CONFIG, $SUCF
                .GLOBL  $DOOFF, NUCHN, LRBFIL, CFIND, TALEMT
                .GLOBL  C.CSW, C.DEVG, C.SBLK, NLINES, CO$8BT
                .GLOBL  CD$NAM, CD$DVU, CD$BAS, CD$JOB, CD$$SZ, CD$$UB
                .GLOBL  LTSCMD, LNSPAC, CFNEST, UCLNAM
                .GLOBL  $CFOPN, CFSEND, PBFEND, CFSP, $TTGAG
                .GLOBL  UFPTRP, SDSFCB, SD$DEL, CFLFL4, $UCLCF
                .GLOBL  SDFLAG, SD$FLK, SD$WFM, SDFORM, $UCLRN
```

```
                .GLOBL  SDBUF1,SDBLK,NSPLDV,LD$RON,$UCLCM,$UCLCL
                .GLOBL  LDNAME,LDSIZE,LDFLAG,LDBASE,LDPDEV
                .GLOBL  LSW8,$SGQ1,$SGQ1A,$SGQ1B,$SGQ1C,$SGQ2,$SGIIO,$SGHIO
                .GLOBL  $DEFER,CFCHAN,SCHAIN,LDDEVX,$SGALL
                .GLOBL  CFPNT,CFBLK,$QUIET,DIABFL
                .GLOBL  DIABNO,VT52NO,LA36NO,LA36FL
                .GLOBL  LSW4,KL4CLR,SDSKIP,SDBU,SD$BAK
                .GLOBL  $INCOR,$KED,VQUN1B,VINTIO,VQUN1C
                .GLOBL  SF$BSY,SFFORM,SD$SNG,SFNMBL,NFRESB
                .GLOBL  SD$HLD,SF$HLD,CURPRM,PRMPNT,SF$1ST
                .GLOBL  LSTPRM,PRMBUF,PRMEND,CFSPND
                .GLOBL  SDFHD,SFFLAG,SFQLNK,CFHOLD,LOGDVU,LOGBAS
                .GLOBL  LCOL,$QTSET,$TECO,CD$TOP,LOGCHK
                .GLOBL  $WILD,ERRSEV,UERSEV,PASLIN,LOGBAS,LOGDVU
                .GLOBL  LSTPL,SDCB,SDCBND,VQUAN0,VQUAN3
                .GLOBL  VQUAN1,VQUN1A,VQUAN2,VHIPCT,VQUAN0,VQUAN3
                .GLOBL  DCTRD,DCCRD,DCTWR,DCCWR
                .GLOBL  AT$LOG,AT$SIZ,AT$DEV,AT$FIL,AT$EXT,AT$$SZ
                .GLOBL  VCORTM,NUMDCD,KMPRMT,MXPRMT,C1DEVX
                .GLOBL  RDB,RDBEND,RT$DEV,RT$NAM,RT$$SZ,CLDEVX,SDDVU
                .GLOBL  SDNAME,SDCBSZ,LSTSL,LSTATE
                .GLOBL  TK1VAL,CINDAT,SYSDAT,SYTIMH,SYTIML
                .GLOBL  BASMAP,LOMAP,HIMAP,JCXPGS
                .GLOBL  SMRSIZ,SRTSIZ,CSHSIZ,TK1SEC
                .GLOBL  TSXLN,TSXSIT,GRT1,TRGRET,LICTXT,SUPCOD,NAMTOP,SUMS,SUCS
                .GLOBL  LPRG1,LPRG2,S$QUSR,S$IOWT
                .GLOBL  S$SPDB,S$SPCB,SFUSER,SFFILE,VT200,VT2007,VT2008
                .GLOBL  LCBIT,LA36,LA120,VT52,VT100,DIABLO,QUME
                .GLOBL  ADM3A,LTRMTP,LA12FL,LA12NO,VT52FL
                .GLOBL  VT10FL,VT10NO,QUMEFL,QUMENO,ADM3FL
                .GLOBL  VT20FL,VT20NO
                .GLOBL  ADM3NO,SYINDX,SYUNIT,NUMDEV,PNAME
                .GLOBL  OF$DEV,OF$UNT,OF$FIL,OF$FLG,SYNAME
                .GLOBL  OF$$SZ,OT$RON,RESDEV,$TAPE
                .GLOBL  KMNBAS,ODTBAS,$CTRLD
                .GLOBL  LSW6,$SNWTT,PF$SYS,PF$IOW,$DEBUG
                .GLOBL  RSR,TSR,LMXNUM,LSTMX,MXDTR,ZCLR,MXCSR
                .GLOBL  $INDDF,$INDRN,IN$ACT,IN$CNT,IN$CMD,INDSAV
                .GLOBL  $PHONE,INVEC,LMXLN,MXVEC,$INIT,$DEAD,$HARD
                .GLOBL  ITRMTP,LMXPRM,LSW7,$INDAB,CFSTS,CF$IND,CF$QUT
                .GLOBL  CFABLV,MONVEC,LBSPRI,MAXPRI,MXJPRI,LPRI
                .GLOBL  LOGCHN,LOGFLG,LOGPTR,LOGBUF,LOGBLK
                .GLOBL  LF$OPN,LF$WRT,UCLBLK,UCLDAT
                .GLOBL  CSHHD,FC$CDX,FC$LNK,FD$NAM,UC$NDC,UC$MDC,CVTUC
                .GLOBL  CMDBUF,PAUMSG,RDCMD,DKSAV,SYSAV,CVTTAB,RUNHD,SEARCH
                .GLOBL  FKILL,ABRTCF,ACRFN,XAREA,FILNAM,NOPRG,FPRINT
                .GLOBL  PUSHCF,TRMSTR,FILNAM,R50DIR,R50SY,R50IND,R50SAV
                .GLOBL  INDACT,R50DUP,R50PIP,R50KED,R50K52,R50KEX,R50TSX,R50UCL
                .GLOBL  BLKO,RDERM,R50VIR,NOSTRT,RUNEMT,OVRCOR
                .GLOBL  BADSAV,LDNAM,NOPRG,NOCIN,SIZVAL,ASKLNM,BADCMD,KCSIBF
                .GLOBL  ASDEX,KCSIMS,ASNOVF,GTRD50,R50BUF,R50LDO,MNTDEV,DMTARG
                .GLOBL  DEADEV,CHKMNT,CHKMTX,INFOMT,NOFLAG,MTOPHD,ILLCMD
                .GLOBL  R50LD,INVLDN,R50DSK,ACRFIL,BDFNAM,LOGASN,MNTFUL,R50LD7
                .GLOBL  TBLOVF,SETHD,CSIMS2,CKPRIV,R50NO,AMBOPT,ACRDEC
                .GLOBL  MAXAVL,PRTDEC,DEVUNT,PNAME,HNBUF
                .GLOBL  ACROCT,HANBSY,CSIMS1,MISSEQ,NOIND,POPCF
                .GLOBL  BADPMT,BADPRI,TOTXT,CRLF,HIPRI,STLGHD,LOGCLS,R50LOG
```

```
                        .GLOBL  BDLGOP,SPLHLA,NOCCL,LDOPHD,PRTFIX,PRTSPC
                        .GLOBL  DLTXT,OCTFIX,PRTTTP,NATXT,SPDTX1,NOTXT,YESTXT,NINTXT
                        .GLOBL  PRTUNM,SYHD1,SYHD2,PRTLN,SPACE2,DETTXT,SPACE3,RNMS,WTMS
                        .GLOBL  SWPTX,LOCKTX,SPACE5,PRTDC3,KBMSG,DIVIDE,PRTDC2
                        .GLOBL  COLOO,CPUAH,CPUAL,PRTTMV,NOFIL,CMDBUF,CALUCL
                        .GLOBL  NOUDC,DEVHD1,ASNHD1,ASNHD2,SHMTH1,SHMTH2,PRTTMD
                        .GLOBL  CVDVNM,SPACE6,PRTBUF,PRTFNM,NONEMS,NODAT,NOLDMT
                        .GLOBL  SUBARO,EDTFIL,RONTXT,NOTAVL,KBTX,MNFLGS,MNBPC
                        .GLOBL  DELSPC,MNBASE,MNTOP,MONHD,MONAR1,NOPMGN,PMBUSY,MONAR2
                        .GLOBL  NSWPMS,MAXMTX,CURMTX,CHKDLM,SPLHD,AMBOPT,INVOPT
                        .GLOBL  DEVIDL,COAL,ALDEX,COAD,SPACTV,SPWFM,DEVIDL,SPSNG
                        .GLOBL  COAL,ALDEX,ALDBLK,COAD,SPACTV,SPWFM,DEVIDL
                        .GLOBL  SPSNG,SPFUL,SPCF,SPFLK,NOFIL,SPGEMT,NOOPTT
                        .GLOBL  BDLIN,MSGBUF,MSGEND,NOTON,GAGMSG
                        .GLOBL  LINFRE,DJABMS,DLMSG,INVTIM,DMTALL
                        .GLOBL  SHTMSG,AUTHFN,SPLACT,DOSTOP,OFFEMT,KILEMT,UPTMMS
                        .GLOBL  TMTOTH,DIVSOR,TMTOTL,PRTPCT,SUM1,SUM2,SUM3,SUM4
                        .GLOBL  SUM5,SUM6,SUM7,OTHRON,SPLPND,STPASK,SRTSMS
                        .GLOBL  SIZEMT,ASNOVF,INVLDM,CSIMS4,MNTARG,HUPARG,R50TT
                        .GLOBL  KMNNAM,NOKMON,CCLNAM,OTRMNT,CHKDEV,DMTSUB,CMDCCL
                        .GLOBL  SHOHD,SUBTXT,MNTTXT,SRTTXT,TOTMMS,UMSSMS,SSRMAP
                        .GLOBL  TSXSMS,USRMMS,JCXSMS,DZTXT,OCTPRT
                        .GLOBL  PRTR50,PRTDAT,PRTTOD,PRTTIM,INVDEV,ALFN,R50DK
                        .GLOBL  DETHD,DETARG,RUNMS,NOFRDL,R50MON,INVDAT,MUL32,COAF
                        .GLOBL  AR$PRJ,AR$PRG,AR$CON,AR$CNT,AR$CPH,AR$CPL,AR$UNM
                        .GLOBL  AR$DMY,AR$$SZ,ARNRPB,$SLON,$SLTTY,$SLLET
                        .GLOBL  PRTWRN,SLMXLN,$LOFCF,CSHMSG,$CARMN
                        .GLOBL  AF$HIE,AF$NOI,$NOINT,AF$PLK,AF$DBG
                        .GLOBL  AF$IOP,$RNIOP,SHVTX1,SHVTX2,SHVTX3,SHVTX4,SJSPPN
```

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

```
                        ;-----------------------------------------------------
                        ;  Macro to cause a fatal error message to be printed.
                        ;
                              .MACRO  FERR    MSG
                              MOV     R5,-(SP)
                              MOV     MSG,R5
                              CALL    FPRINT
                              MOV     (SP)+,R5
                              .ENDM   FERR
                        ;
                        ;-----------------------------------------------------
                        ;  Macro to print a fatal error message, clean up
                        ;  and then jump to RDCMD.
                        ;
                              .MACRO  FABORT  MSG
                              MOV     MSG,R5
                              JMP     FKILL
                              .ENDM   FABORT
                        ;
                        ;-----------------------------------------------------------
                        ;  Macro to print a warning message
                        ;
                              .MACRO  FWARN   MSG
                              MOV     R5,-(SP)
                              MOV     MSG,R5
                              CALL    PRTWRN
                              MOV     (SP)+,R5
                              .ENDM   FWARN
                        ;
                        ;-----------------------------------------------------------
                        ;  Macro to start a standard option table.
                        ;  Name = 1 to 4 character table name.
                        ;  NA = Number of arguments per table entry.
                        ;
                              .MACRO  TBLDEF  NAME,NA
                        NARGS =       NA
                              .CSECT  CMDVSH
                        NAME'HD: .WORD  2*NA
                              .ENDM   TBLDEF
                        ;
                        ;-----------------------------------------------------------
                        ;  Macro to enter an option text name and a set of parameters
                        ;  into the currently open table.
                        ;  STRNG = Ascii name
                        ;  A,B,C = Set of option parameters to store in table with name.
                        ;
                              .MACRO  CMDDEF  STRNG,A,B,C
                              .CSECT  NAMESH
                        L     =       .
                              .ASCIZ  /STRNG/
                              .CSECT  CMDVSH
                              .WORD   L               ;POINTER TO NAME STRING
                              .WORD   A
                        .IIF  GE,<NARGS-2>    .WORD   B
                        .IIF  GE,<NARGS-3>    .WORD   C
                              .ENDM   CMDDEF
                        ;
```

```
                        ;------------------------------------------------------------
                        ;  Macro to end a set of table entries.
                        ;
                                .MACRO  TBLEND
                                .CSECT  CMDVSH
                                .WORD   0
                                .CSECT  TSKSHO
                                .ENDM   TBLEND
                        ;------------------------------------------------------------------------
                        ;  Define options for SHOW command
                        ;
000000                          TBLDEF  SHO,1
000002                          CMDDEF  ALL,SOPALL
000006                          CMDDEF  ALLO*CATE,SOPALC
000012                          CMDDEF  ALLO*CATIONS,SOPALC
000016                          CMDDEF  AS*SIGNS,SOPASN
000022                          CMDDEF  CA*CHE,SOPCSH
000026                          CMDDEF  C*ONFIGURATION,SOPCON
000032                          CMDDEF  CL,SOPCL
000036                          CMDDEF  COM*MANDS,SOPCMD
000042                          CMDDEF  COR*TIM,SOPCTM
000046                          CMDDEF  C1,SOPCL
000052                          CMDDEF  D*EVICES,SOPDEV
000056                          CMDDEF  DA*TE,SOPDAT
000062                          CMDDEF  E*RRORS,RDCMD
000066                          CMDDEF  H*IPRCT,SOPHIO
000072                          CMDDEF  INS*TALL,SOPINS
000076                          CMDDEF  INT*IOC,SOPIIO
000102                          CMDDEF  J*OBS,SOPJOB
000106                          CMDDEF  KEY*S,SHOKEY
000112                          CMDDEF  LD,SOPSUB
000116                          CMDDEF  M*EMORY,SOPMEM
000122                          CMDDEF  MAXMC,SOPMC
000126                          CMDDEF  MAXMRB,SOPMR
000132                          CMDDEF  MAXMSG,SOPMB
000136                          CMDDEF  MO*UNTS,SOPMNT
000142                          CMDDEF  N*UMDC,SOPNDC
000146                          CMDDEF  PRI*ORITY,SOPPRI
000152                          CMDDEF  PRIL*OW,SOPPLO
000156                          CMDDEF  PRIH*I,SOPPHI
000162                          CMDDEF  PRID*EF,SOPPDF
000166                          CMDDEF  PRIV*ILEGES,SOPPRV
000172                          CMDDEF  PRIVIR,SOPPVR
000176                          CMDDEF  Q*UEUE,SOPQUE
000202                          CMDDEF  QUAN0,SOPQ0
000206                          CMDDEF  QUAN1,SOPQ1
000212                          CMDDEF  QUAN1A,SOPQ1A
000216                          CMDDEF  QUAN1B,SOPQ1B
000222                          CMDDEF  QUAN1C,SOPQ1C
000226                          CMDDEF  QUAN2,SOPQ2
000232                          CMDDEF  QUAN3,SOPQ3
000236                          CMDDEF  REG*IONS,SOPREG
000242                          CMDDEF  RUN*-TIMES,SOPSRT
000246                          CMDDEF  S*UBSETS,SOPSUB
000252                          CMDDEF  SL*E,SOPSLE
000256                          CMDDEF  SP*OOL,SOPSPL
000262                          CMDDEF  SYSP*ASSWORD,SOPSYP
```

```
    115 000266                          CMDDEF  T*ERMINALS,SOPTRM
    116 000272                          CMDDEF  TI*ME,SOPTIM
    117 000276                          CMDDEF  TT*Y,SOPTRM
    118 000302                          CMDDEF  USE,SOPUSE
    119 000306                          CMDDEF  USERS,SOPJOB
    120 000312                          CMDDEF  VE*RSION,SOPVER
    121 000316                          CMDDEF  VM,SOPVM
    122 000322                          TBLEND
```

```
                                        .SBTTL  Data areas
                                ;-------------------------------------------------------------------------------
                                ;  Data areas
                                ;
000000  075250  014644  000000  HANNAM: .RAD50  /SY ddd   TSX/  ;File spec for device handler
000006  100020
000010  000000                  DVEC:   .WORD   0
000012  000000                  DCSR:   .WORD   0
000014  012276                  R50CLO: .RAD50  /CLO/
000016  000000                  CDBASE: .WORD   0
000020  000000                  CDDVU:  .WORD   0
000022  000000                  CDBAS5: .WORD   0
                                ;
                                ;  Table of terminal device type names.
                                ;  Note, the order of the entries in this table must match the order
                                ;  of the values of the CDX$xx entries as defined in TSGEN.
                                ;
000024  015340                  CTLNAM: .RAD50  /DL /           ;CDX$DL
000026  016420                          .RAD50  /DZ /           ;CDX$DZ
000030  015100                          .RAD50  /DH /           ;CDX$DZ
000032  015126                          .RAD50  /DHV/           ;CDX$VH
000034  062170                          .RAD50  /PC /           ;CDX$PI
000036  012500                          .RAD50  /CP /           ;CDX$PC
000040  063200                          .RAD50  /PP /           ;CDX$PP
000042  066300                          .RAD50  /QP /           ;CDX$QP
000044  114700                          .RAD50  /XX /
```

```
                                        .SBTTL  SHOW command
                                ;-----------------------------------------------------------------------------
                                ;  Process the SHOW command
                                ;
000046  000240                  CMDSHO: NOP
000050  004767  0000000               CALL    CVTTAB          ;CONVERT TABS AND FF'S TO SPACES
000054  105713                        TSTB    @R3             ;IF NO OPTION WAS SPECIFED WITH COMMAND,
000056  001555                        BEQ     SOPASN          ;TREAT COMMAND LIKE "SHOW ASSIGNS"
000060  012704  000000'              MOV     #SHOHD,R4       ;POINT TO TABLE OF SHOW OPTIONS
000064  004767  0000000               CALL    SEARCH          ;LOOK UP OPTION
000070  103401                        BCS     BDSO            ;BR IF INVALID OPTION
000072  000134                        JMP     @(R4)+          ;JUMP TO PROCESSING ROUTINE
                                ;
                                ;  Invalid option
                                ;
000074  005704                  BDSO:   TST     R4              ;Invalid or ambiguous option?
000076  001404                        BEQ     1$              ;Br if invalid
000100                                  FABORT  #AMBOPT         ;Ambigious option
000110                          1$:     FABORT  #INVOPT         ;Invalid option
```

```
.       ALL

                                                .SBTTL  .       ALL
                                        ;-----------------------------------------------------------------------
                                        ;  SHOW ALL
                                        ;
 000120  004767  007610                 SOPALL: CALL    SHOVER          ;SHOW VERSION
 000124                                         .PRINT  #CRLF
 000132  004767  005012                         CALL    SHODEV          ;SHOW DEVICES
 000136                                         .PRINT  #CRLF
 000144  004767  006034                         CALL    SHOASN          ;SHOW ASSIGNS
 000150                                         .PRINT  #CRLF
 000156  004767  006370                         CALL    SHOALC          ;SHOW ALLOCATIONS
 000162                                         .PRINT  #CRLF
 000170  004767  003474                         CALL    SHOJOB          ;SHOW JOBS
 000174  004767  001414                         CALL    SHOTRM          ;SHOW TERMINALS
 000200                                         .PRINT  #CRLF
 000206  004767  002570                         CALL    SHOCL           ;SHOW CL
 000212                                         .PRINT  #CRLF
 000220  004767  001050                         CALL    SHOMEM          ;SHOW MEMORY
 000224                                         .PRINT  #CRLF
 000232  004767  011556                         CALL    SHOSPL          ;SHOW SPOOL
 000236                                         .PRINT  #CRLF
 000244                                         .PRINT  #SUBTXT
 000252  004767  011646                         CALL    SHOSUB          ;SHOW SUBSETS
 000256                                         .PRINT  #CRLF
 000264                                         .PRINT  #MNTTXT
 000272  004767  006472                         CALL    SHOMNT          ;SHOW MOUNTS
 000276                                         .PRINT  #CRLF
 000304                                         .PRINT  #SRTTXT
 000312  004767  011374                         CALL    SHOSRT          ;SHOW RUN-TIMES
 000316                                         .PRINT  #CRLF
 000324  004767  010346                         CALL    SHOREG          ;SHOW REGIONS
 000330                                         .PRINT  #CRLF
 000336  032767  0000000 0000000                BIT     #PO$SYS,PRIVCO          ;System privilege?
 000344  001405                                 BEQ     1$                      ;Branch if not
 000346  004767  007444                         CALL    SHOINS          ;SHOW INSTALL
 000352                                         .PRINT  #CRLF
 000360  004767  011030                 1$:     CALL    SHOPRV          ;SHOW PRIVILEGES
 000364                                         .PRINT  #CRLF
 000372  004767  011136                         CALL    SHOSLE          ;SHOW SL
 000376                                         .PRINT  #CRLF
 000404  004767  011764                         CALL    SHOWVM          ;SHOW VM (Base, Top and Size)
 000410  000426                                 BR      SOPJMP
                                        ;
                                        ;  SHOW ASSIGNS
                                        ;
 000412  004767  005566                 SOPASN: CALL    SHOASN          ;SHOW ASSIGNS
 000416  000423                                 BR      SOPJMP
                                        ;
                                        ;  SHOW COMMANDS
                                        ;
 000420                                 UCLCMD:
 000420  004767  004466                 SOPCMD: CALL    SHOCMD          ;SHOW COMMANDS
 000424  000420                                 BR      SOPJMP
                                        ;
                                        ;  SHOW CONFIGURATION
                                        ;
 000426  004767  004516                 SOPCON: CALL    SHODEV          ;SHOW DEVICES
```

```
000432  004767  001156                   CALL    SHOTRM          ;SHOW TERMINALS
000436  000410                           BR      SOPJMP
                                ;
                                ;   SHOW DATE
                                ;
000440  004767  007216          SOPDAT:  CALL    SHODAT          ;SHOW DATE
000444  000410                           BR      SOPJMP
                                ;
                                ;   SHOW DEVICES
                                ;
000446  004767  004476          SOPDEV:  CALL    SHODEV          ;SHOW DEVICES
000452  000405                           BR      SOPJMP
                                ;
                                ;   SHOW JOBS
                                ;
000454  004767  003210          SOPJOB:  CALL    SHOJOB          ;SHOW JOBS
000460  000402                           BR      SOPJMP
                                ;
                                ;   SHOW MEMORY
                                ;
000462  004767  000606          SOPMEM:  CALL    SHOMEM          ;SHOW MEMORY
000466  000167  000000G         SOPJMP:  JMP     RDCMD
                                ;
                                ;   SHOW ALLOCATIONS
                                ;
000472  004767  006054          SOPALC:  CALL    SHOALC          ;SHOW ALLOCATIONS
000476  000167  000000G                  JMP     RDCMD
                                ;
                                ;   SHOW CL
                                ;
000502  004767  002274          SOPCL:   CALL    SHOCL           ;SHOW CL
000506  000767                           BR      SOPJMP
                                ;
                                ;   SHOW INSTALL
                                ;
000510  004767  007302          SOPINS:  CALL    SHOINS          ;SHOW INSTALL
000514  000764                           BR      SOPJMP
                                ;
                                ;   SHOW MOUNTS
                                ;
000516  004767  006246          SOPMNT:  CALL    SHOMNT          ;SHOW MOUNTS
000522  000761                           BR      SOPJMP
                                ;
                                ;   SHOW PRIVILEGES
                                ;
000524  004767  010664          SOPPRV:  CALL    SHOPRV          ;SHOW PRIVILEGES
000530  000756                           BR      SOPJMP
                                ;
                                ;   SHOW QUEUE
                                ;
000532  004767  004256          SOPQUE:  CALL    SHOQUE          ;SHOW QUEUE
000536  000753                           BR      SOPJMP
                                ;
                                ;   SHOW RUN-TIMES
                                ;
000540  004767  011146          SOPSRT:  CALL    SHOSRT          ;SHOW RUN-TIMES
000544  000750                           BR      SOPJMP
```

. ALL

```
                                        ;
                                        ;   SHOW SL
                                        ;
000546  004767  010762          SOPSLE: CALL    SHOSLE          ;SHOW SL
000552  000745                          BR      SOPJMP
                                        ;
                                        ;   SHOW VM
                                        ;
000554  004767  011614          SOPVM:  CALL    SHOWVM          ;SHOW VM
000560  000742                          BR      SOPJMP
                                        ;
                                        ;   SHOW REGIONS
                                        ;
000562  004767  010110          SOPREG: CALL    SHOREG          ;SHOW REGIONS
000566  000737                          BR      SOPJMP
                                        ;
                                        ;   SHOW SUBSETS
                                        ;
000570  004767  011330          SOPSUB: CALL    SHOSUB          ;SHOW SUBSETS
000574  000734                          BR      SOPJMP
                                        ;
                                        ;   SHOW SPOOL
                                        ;
000576  004767  011212          SOPSPL: CALL    SHOSPL          ;SHOW SPOOL
000602  000731                          BR      SOPJMP
                                        ;
                                        ;   SHOW SYSPASSWORD
                                        ;
000604  004767  000432          SOPSYP: CALL    SHOSYP          ;SHOW SYSPASSWORD
000610  000726                          BR      SOPJMP
                                        ;
                                        ;   SHOW TERMINALS
                                        ;
000612  004767  000776          SOPTRM: CALL    SHOTRM          ;SHOW TERMINALS
000616  000723                          BR      SOPJMP
                                        ;
                                        ;   SHOW TIME
                                        ;
000620  004767  007074          SOPTIM: CALL    SHOTIM          ;SHOW TIME
000624  000720                          BR      SOPJMP
                                        ;
                                        ;   SHOW USE
                                        ;
000626  004767  007156          SOPUSE: CALL    SHOUSE          ;SHOW USE
000632  000715                          BR      SOPJMP
                                        ;
                                        ;   SHOW VERSION
                                        ;
000634  004767  007074          SOPVER: CALL    SHOVER          ;SHOW VERSION
000640  000712                          BR      SOPJMP
```

```
.       ALL

                                        ;
                                        ;   SHOW MAXMSG
                                        ;
000642  016705  0000000         SOPMB:  MOV     VMXMSG,R5
000646  000427                         BR      SHOVAL
                                        ;
                                        ;   SHOW MAXMC
                                        ;
000650  016705  0000000         SOPMC:  MOV     VMAXMC,R5
000654  000424                         BR      SHOVAL
                                        ;
                                        ;   SHOW MAXMRB
                                        ;
000656  016705  0000000         SOPMR:  MOV     VMXMRB,R5
000662  000421                         BR      SHOVAL
                                        ;
                                        ;   SHOW QUAN0
                                        ;
000664  016705  0000000         SOPQ0:  MOV     VQUAN0,R5       ;Get value
000670  000416                         BR      SHOVAL
                                        ;
                                        ;   SHOW QUAN1
                                        ;
000672  016705  0000000         SOPQ1:  MOV     VQUAN1,R5       ;GET VALUE
000676  000413                         BR      SHOVAL
                                        ;
                                        ;   SHOW QUAN1A
                                        ;
000700  016705  0000000         SOPQ1A: MOV     VQUN1A,R5
000704  000410                         BR      SHOVAL
                                        ;
                                        ;   SHOW QUAN1B
                                        ;
000706  016705  0000000         SOPQ1B: MOV     VQUN1B,R5
000712  000405                         BR      SHOVAL
                                        ;
                                        ;   SHOW QUAN1C
                                        ;
000714  016705  0000000         SOPQ1C: MOV     VQUN1C,R5
000720  000402                         BR      SHOVAL
                                        ;
                                        ;   SHOW QUAN2
                                        ;
000722  016705  0000000         SOPQ2:  MOV     VQUAN2,R5
000726                          SHOVAL: .PRINT  #SPACE3         ;Print 3 spaces
000734  004767  0000000                 CALL    PRTDEC
000740                                  .PRINT  #CRLF
000746  000167  0000000                 JMP     RDCMD
                                        ;
                                        ;   SHOW QUAN3
                                        ;
000752  016705  0000000         SOPQ3:  MOV     VQUAN3,R5       ;Get QUAN3 value
000756  000763                         BR      SHOVAL          ;Display it
                                        ;
                                        ;   SHOW CORTIM
                                        ;
000760  016705  0000000         SOPCTM: MOV     VCORTM,R5
```

```
        ALL

000764  000760                          BR      SHOVAL
                                ;
                                ;  SHOW INTIOC
                                ;
000766  016705  0000000         SOPIIO: MOV     VINTIO,R5
000772  000750                          BR      SHOVAL
                                ;
                                ;  SHOW HIPRCT
                                ;
000774  016705  0000000         SOPHIO: MOV     VHIPCT,R5
001000  000752                          BR      SHOVAL
                                ;
                                ;  SHOW NUMDC
                                ;
001002  016705  0000000         SOPNDC: MOV     NUMDCD,R5
001006  000747                          BR      SHOVAL
                                ;
                                ;  SHOW PRILOW
                                ;
001010  116705  0000000         SOPPLO: MOVB    VPRILO,R5
001014  000744                          BR      SHOVAL
                                ;
                                ;  SHOW PRIHI
                                ;
001016  116705  0000000         SOPPHI: MOVB    VPRIHI,R5
001022  000741                          BR      SHOVAL
                                ;
                                ;  SHOW PRIDEF
                                ;
001024  116705  0000000         SOPPDF: MOVB    VPRIDF,R5
001030  000736                          BR      SHOVAL
                                ;
                                ;  SHOW PRIVIR
                                ;
001032  116705  0000000         SOPPVR: MOVB    VPRIVR,R5
001036  000733                          BR      SHOVAL
                                ;
                                ;  SHOW PRIORITY
                                ;
001040                          SOPPRI: .PRINT  #TM$PR1         ;"Current priority = "
001046  116105  0000000                 MOVB    LPRI(R1),R5     ;Get current priority
001052  004767  0000000                 CALL    PRTDEC          ;Print it
001056                                  .PRINT  #TM$PR2         ;"; maximum authorized priority = "
001064  116705  0000000                 MOVB    MXJPRI,R5       ;Get max authorized priority
001070  004767  0000000                 CALL    PRTDEC          ;Print it
001074                                  .PRINT  #CRLF           ;End of line
001102                                  .PRINT  #TM$LPR         ;"Low priority range = 0 to "
001110  116705  0000000                 MOVB    VPRILO,R5       ;Get top of low priority range
001114  004767  0000000                 CALL    PRTDEC          ;Print the value
001120                                  .PRINT  #CRLF
001126                                  .PRINT  #TM$HPR         ;"High priority range = "
001134  116705  0000000                 MOVB    VPRIHI,R5       ;Get base of high priority range
001140  004767  0000000                 CALL    PRTDEC          ;Print it
001144                                  .PRINT  #TM$HPE         ;Finish message
001152  000167  0000000                 JMP     RDCMD
                                ;
                                ;  SHOW CACHE
```

```
.       ALL

                                        ;
001156  016705  0000000         SOPCSH: MOV     CSHALC,R5       ;Was caching genned into system?
001162  001004                         BNE     1$              ;Br if yes
001164                                  .PRINT  #TM$CNG         ;Caching not genned into system
001172  000421                         BR      9$
001174  005767  0000000         1$:     TST     VCSHNB          ;Is caching enabled?
001200  001004                         BNE     2$              ;Br if yes
001202                                  .PRINT  #TM$CDS         ;Caching is disabled
001210  000412                         BR      9$
001212                          2$:     .PRINT  #TM$CEN         ;Data cache size is ...
001220  016705  0000000                 MOV     VCSHNB,R5       ;Get current number of blocks for cache
001224  004767  0000000                 CALL    PRTDEC          ;Print size of cache
001230                                  .PRINT  #CRLF           ;Terminate print line
001236  000167  0000000         9$:     JMP     RDCMD

                                        ;
                                        ;   SHOW SYSPASSWORD
                                        ;
001242  004767  0000000         SHOSYP: CALL    CKSYPV          ;Make sure user has SYSPRV privilege
001246  105767  0000000                 TSTB    SYPSWD          ;Is there a system password?
001252  001004                         BNE     1$              ;Br if yes
001254                                  .PRINT  #TM$NSP         ;No system password
001262  000403                         BR      9$
001264                          1$:     .PRINT  #SYPSWD         ;Print the password
001272  000207                  9$:     RETURN
```

```
                                        .SBTTL  .       MEMORY
                                ;-----------------------------------------------------------------------
                                ;  SHOW MEMORY
                                ;
001274  010546                  SHOMEM: MOV     R5,-(SP)
                                ;  Total installed memory
001276                                  .PRINT  #TOTMMS         ;TOTAL MEMORY
001304  016705  0000000                 MOV     PHYMEM,R5       ;Get total # 64-byte blocks of memory
001310  072527  177775                  ASH     #-3,R5          ;Convert to # 512-byte pages
001314  042705  160000                  BIC     #160000,R5      ;Kill possible sign extension
001320  004767  012002                  CALL    PRTKB           ;DISPLAY THE VALUE
                                ;  Size of unmapped system & unmapped handlers
001324                                  .PRINT  #UMSSMS         ;SIZE OF UNMAPPED PORTION OF SYSTEM
001332  016705  0000000                 MOV     UMSYTP,R5       ;GET ADDRESS OF TOP OF TSX
001336  062705  001777                  ADD     #1777,R5        ;BOUND UP TO KB
001342  000241                          CLC                     ;CVT TO # PAGES
001344  006005                          ROR     R5
001346  072527  177770                  ASH     #-8.,R5
001352  004767  011750                  CALL    PRTKB           ;DISPLAY VALUE
                                ;  Size of mapped system regions
001356                                  .PRINT  #SSRMAP         ;SIZE OF SYSTEM MAPPED REGION
001364  016705  0000000                 MOV     SMRSIZ,R5       ;# 64-byte blocks for mapped regions
001370  062705  000007                  ADD     #7,R5           ;Round up
001374  072527  177775                  ASH     #-3,R5          ;Convert to 512-byte block units
001400  004767  011722                  CALL    PRTKB
001404  010546                          MOV     R5,-(SP)        ;SAVE MAPPED REGION SIZE
                                ;  Total TSX-Plus size
001406                                  .PRINT  #TSXSMS         ;TSX SIZE
001414  016705  0000000                 MOV     LOMAP,R5        ;PAGE AT TOP OF TSX
001420  166705  0000000                 SUB     BASMAP,R5       ;# PAGES FOR TSX AND HANDLERS
001424  005305                          DEC     R5
001426  062605                          ADD     (SP)+,R5        ;ADD MAPPED REGION SIZE
001430  004767  011672                  CALL    PRTKB
                                ;  Mapped handlers
001434                                  .PRINT  #MHNSMS         ;Size of mapped handlers
001442  016705  0000000                 MOV     MHNSIZ,R5       ;# 64-byte blocks for mapped handlers
001446  062705  000007                  ADD     #7,R5           ;Round up
001452  072527  177775                  ASH     #-3,R5          ;Convert to 512-byte block units
001456  004767  011644                  CALL    PRTKB
                                ;  Shared run-time systems
001462                                  .PRINT  #SRTSMS         ;SIZE OF SHARABLE RUN-TIME SYSTEMS
001470  016705  0000000                 MOV     SRTSIZ,R5       ;# 64-byte blocks for shared run-times
001474  062705  000007                  ADD     #7,R5           ;Round up
001500  072527  177775                  ASH     #-3,R5          ;Convert to 512-byte block units
001504  004767  011616                  CALL    PRTKB
                                ;  Data cache buffers
001510                                  .PRINT  #CSHMSG         ;Size of data cache
001516  016705  0000000                 MOV     CSHSIZ,R5       ;# 64-byte blocks for data cache
001522  062705  000007                  ADD     #7,R5           ;Round up
001526  000241                          CLC                     ;Clear carry to convert to 512 byte
001530  006005                          ROR     R5              ;  block units without sign extension
001532  072527  177776                  ASH     #-2,R5          ;  in case cache >=2Mb (100000 64bytes)
001536  004767  011564                  CALL    PRTKB
                                ;  User memory space
001542                                  .PRINT  #USRMMS         ;USER MEMORY SPACE
001550  016705  0000000                 MOV     HIMAP,R5
001554  166705  0000000                 SUB     LOMAP,R5
```

. MEMORY

```
      58 001560  004767  011542                    CALL    PRTKB
      59                                ;  Job context area size
      60 001564                                    .PRINT  #JCXSMS         ;SIZE OF JOB CONTEXT AREA
      61 001572  016705  0000000                   MOV     JCXPGS,R5
      62 001576  005205                            INC     R5              ;ROUND UP
      63 001600  004767  011522                    CALL    PRTKB
      64 001604  004767  011416                    CALL    DSPMEM          ;SHOW JOB MEMORY LIMITS
      65 001610  012605                            MOV     (SP)+,R5
      66 001612  000207                            RETURN
```

```
                                        .SBTTL  .       TERMINALS
                                ;-------------------------------------------------------------------------------
                                ;  Display information about terminals.
                                ;
001614  010146                  SHOTRM: MOV     R1,-(SP)
001616  010246                          MOV     R2,-(SP)
001620  010346                          MOV     R3,-(SP)
001622  010446                          MOV     R4,-(SP)
001624  010546                          MOV     R5,-(SP)
                                ;
                                ;  Print heading lines
                                ;
001626                                  .PRINT  #CRLF           ;Print a blank line
001634                                  .PRINT  #TRMHD1         ;Heading line 1
001642                                  .PRINT  #TRMHD2         ;Heading line 2
                                ;
                                ;  Begin loop to print info for each terminal
                                ;
001650  012701  000002                  MOV     #2,R1           ;Get first terminal index number
                                ;
                                ;  If we are on a pro, do not display info about uninstalled lines
                                ;
001654  032761  0000006 0000006 1$:     BIT     #$DEAD,LSW3(R1) ;Is this line installed?
001662  001406                          BEQ     33$             ;Br if yes
001664  032767  0000006 0000006         BIT     #CW$PRO,CONF62  ;Are we running on a pro?
001672  001402                          BEQ     33$             ;Br if not
001674  000167  001036                  JMP     30$             ;Go check next line
                                ;
                                ;  Display unit number
                                ;
001700  010105                  33$:    MOV     R1,R5           ;Get terminal index #
001702  006205                          ASR     R5              ;Convert to unit #
001704  012703  000002                  MOV     #2,R3           ;Print using 2 columns
001710  004767  0000000                 CALL    PRTFIX          ;Print unit number
                                ;
                                ;  Print "*" if this is the current line
                                ;
001714  116700  0000006                 MOVB    CORUSR,R0       ;Get current job index #
001720  116000  0000006                 MOVB    LNPRIM(R0),R0   ;Get primary line #
001724  012703  000004                  MOV     #4,R3           ;Set to print 4 spaces
001730  020100                          CMP     R1,R0           ;Is this our line?
001732  001005                          BNE     15$             ;Br if not
001734                                  .TTYOUT #'*             ;Flag our line
001744  005303                          DEC     R3              ;Print only 3 more spaces
001746  004767  0000000         15$:    CALL    PRTSPC          ;Print spaces
                                ;
                                ;  Display type of terminal
                                ;
001752  012703  0000000                 MOV     #OPRTXT,R3      ;Assume this is operator's console
001756  120167  0000000                 CMPB    R1,CTRLTT       ;Is this operator's console?
001762  001425                          BEQ     2$              ;Br if yes
001764  012703  0000000                 MOV     #CLLINE,R3      ;Assume this is a CL line
001770  020127  0000000                 CMP     R1,#LSTPL       ;Is this a CL line?
001774  101020                          BHI     2$              ;Br if yes
001776  012703  0000000                 MOV     #LCLTXT,R3      ;Assume this is a local terminal
002002  032761  0000000 0000006         BIT     #$PHONE,ILSW2(R1);Is this a dial-up terminal?
002010  001412                          BEQ     2$              ;Br if not dial-up
```

```
002012  032761  000000G 000000G          BIT     #$INIT,LSW(R1)  ;Is line in use now?
002020  001404                           BEQ     17$             ;Print "PHONE" if inactive dial-up line
002022  032761  000000G 000000G          BIT     #$CARMN,LSW5(R1) ;Is line being used as dial-up?
002030  001402                           BEQ     2$              ;Print "LOCAL" if active but no carrier
002032  012703  000000G          17$:    MOV     #REMTXT,R3      ;This is a remote terminal
002036                           2$:     .PRINT  R3              ;Print type
                                 ;
                                 ;  Display vector address and DL/DZ type
                                 ;
002042  016100  000000G                  MOV     LCDTYP(R1),R0   ;Get device type code for this line
002046  016000  000024'                  MOV     CTLNAM(R0),R0   ;Get RAD50 name of controlling device
002052  004767  000000G                  CALL    PRTR50          ;Print the device name
002056  016105  000000G                  MOV     LMXLN(R1),R5    ;Get line # within mux
002062  016104  000000G                  MOV     LMXNUM(R1),R4   ;Get mux index number
002066  001007                           BNE     3$              ;Br if this is a mux line
                                 ;  This is not a mux line
002070  012703  000004                   MOV     #4,R3           ;Print 4 spaces
002074  004767  000000G                  CALL    PRTSPC
002100  016105  000000G                  MOV     INVEC(R1),R5    ;Get input vector address
002104  000420                           BR      4$
                                 ;  This is a mux line
002106                           3$:     .TTYOUT #'-             ;Put in hyphen
002116  016105  000000G                  MOV     LMXLN(R1),R5    ;Get line # within the mux
002122  012703  000002                   MOV     #2,R3           ;Print in 2 column field
002126  004767  000000G                  CALL    PRTFIX          ;Print line number
002132  012703  000001                   MOV     #1,R3           ;Print 1 space
002136  004767  000000G                  CALL    PRTSPC
002142  016405  000000G                  MOV     MXVEC(R4),R5    ;Get DZ-11 interrupt vector address
002146  012703  000003           4$:     MOV     #3,R3           ;Print 3 digits
002152  004767  000000G                  CALL    OCTFIX          ;Print vector address
002156  012703  000002                   MOV     #2,R3           ;Print 2 spaces
002162  004767  000000G                  CALL    PRTSPC
                                 ;
                                 ;  Print address of Control Status Register (CSR)
                                 ;
002166  016104  000000G                  MOV     LMXNUM(R1),R4   ;Get mux index number
002172  001003                           BNE     10$             ;Br if DZ-11
                                 ;  This is a DL-11 line
002174  016102  000000G                  MOV     RSR(R1),R2      ;Get address of receiver status register
002200  000402                           BR      11$
                                 ;  This is a DZ-11 line
002202  016402  000000G          10$:    MOV     MXCSR(R4),R2    ;Get address of CSR
002206  004767  000000G          11$:    CALL    OCTPRT          ;Print the address
002212  012703  000002                   MOV     #2,R3           ;Print 2 spaces
002216  004767  000000G                  CALL    PRTSPC
                                 ;
                                 ;  Print information about terminal type
                                 ;
002222  004767  000000G                  CALL    PRTTTP          ;Print terminal type
002226  012703  000002                   MOV     #2,R3           ;Print 2 spaces
002232  004767  000000G                  CALL    PRTSPC
                                 ;
                                 ;  Print information about speed
                                 ;
002236  032761  000000G 000000G          BIT     #$AUTO,ILSW2(R1);Is autobaud specified for this line?
002244  001413                           BEQ     24$             ;Br if not
002246  005761  000000G                  TST     LCLUNT(R1)      ;Is this line being used as a CL unit?
```

```
     115 002252  002010                                BGE     24$             ;Br if yes -- No autobaud for CL
     116 002254  032761  0000000 0000000               BIT     #$DILUP,LSW(R1) ;Is a job active on this line?
     117 002262  001004                                BNE     24$             ;Br if yes -- speed is known if job on
     118 002264                                        .PRINT  #TM$AUT         ;Print "auto"
     119 002272  000424                                BR      7$
     120 002274  016103  0000000         24$:          MOV     LMXPRM(R1),R3   ;Get line parameters for mux line
     121 002300  000303                                SWAB    R3              ;Right justify speed code
     122 002302  042703  177760                        BIC     #177760,R3      ;Clear all but speed code
     123 002306  001004                                BNE     5$              ;Br if we know the speed
     124 002310                                        .PRINT  #NATXT          ;Print "N/A"
     125 002316  000412                                BR      7$
     126 002320  070327  000005          5$:           MUL     #5,R3           ;Values are stored 5 characters each
     127 002324  062703  0000000                       ADD     #SPDTX1,R3      ;Point to text string
     128 002330  012704  000005                        MOV     #5,R4           ;Print 5 characters
     129 002334  112300                  6$:           MOVB    (R3)+,R0        ;Get next char of speed
     130 002336                                        .TTYOUT                 ;Print it
     131 002342  077404                                SOB     R4,6$           ;Loop till all printed
     132 002344                          7$:           .TTYOUT #40             ;Print a space
     133                                               ;
     134                                               ;  Print information about character length and parity
     135                                               ;
     136 002354  116103  0000010                       MOVB    LMXPRM+1(R1),R3 ;Get line parameter flags
     137 002360  032703  0000000                       BIT     #LP$7BT,R3      ;7 bit characters wanted?
     138 002364  001005                                BNE     27$             ;Br if yes
     139 002366                                        .TTYOUT #'8             ;Say length is 8 bits
     140 002376  000404                                BR      28$
     141 002400                          27$:          .TTYOUT #'7             ;Say length is 7 bits
     142 002410  032703  0000000         28$:          BIT     #LP$PAR,R3      ;Parity wanted?
     143 002414  001005                                BNE     29$             ;Br if yes
     144 002416                                        .TTYOUT #'N             ;Say no parity
     145 002426  000414                                BR      31$
     146 002430  032703  0000000         29$:          BIT     #LP$ODD,R3      ;Odd parity wanted?
     147 002434  001005                                BNE     32$             ;Br if yes
     148 002436                                        .TTYOUT #'E             ;Say even parity
     149 002446  000404                                BR      31$
     150 002450                          32$:          .TTYOUT #'O             ;Say odd parity
     151 002460  012703  000004          31$:          MOV     #4,R3           ;Print 4 spaces
     152 002464  004767  0000000                       CALL    PRTSPC
     153                                               ;
     154                                               ;  Indicate if line is active now
     155                                               ;
     156 002470  012703  0000000                       MOV     #NOTXT,R3       ;Assume line is not active
     157 002474  032761  0000000 0000000               BIT     #$DEAD,LSW3(R1) ;Is line installed?
     158 002502  001035                                BNE     12$             ;Br if not installed
     159 002504  016105  0000000                       MOV     LCLUNT(R1),R5   ;Is a CL unit assigned to this line?
     160 002510  002012                                BGE     16$             ;Br if yes
     161 002512  020127  0000000                       CMP     R1,#LSTPL       ;Is this a real or CL line?
     162 002516  101027                                BHI     12$             ;Br if CL line
     163 002520  032761  0000000 0000000               BIT     #$INIT,LSW(R1)  ;Is line active now?
     164 002526  001423                                BEQ     12$             ;Br if not
     165 002530  012703  0000000                       MOV     #YESTXT,R3      ;Point to "Yes" text
     166 002534  000420                                BR      12$             ;Print yes
     167 002536  006205                  16$:          ASR     R5              ;Convert CL unit index to unit number
     168 002540  020527  000007                        CMP     R5,#7.          ;Units 0-7 go with CL device
     169 002544  101406                                BLOS    34$
     170 002546  162705  000010                        SUB     #8.,R5          ;Remove C1 unit bias
     171 002552                                        .PRINT  #TXTC1          ;Print "C1"
```

```
                        BR      35$
                34$:    .PRINT  #TXTCL          ;Print "CL"
000000G 004767  35$:    CALL    PRTDEC          ;Print CL unit number
                        BR      25$
                12$:    .PRINT  R3              ;Print Yes/No
000003  012703  25$:    MOV     #3,R3           ;Print 3 spaces
000000G                 CALL    PRTSPC
                ;
                ;  Print name of user who is using this line
                ;
000000G 000000G         BIT     #$DEAD,LSW3(R1) ;Is this line installed?
                        BEQ     26$             ;Br if line is installed
                        .PRINT  #NINTXT         ;Print Not installed
                        BR      14$
000000G         26$:    MOV     LNAME(R1),R5    ;Is there a descriptive name for this line?
                        BEQ     13$             ;Br if not
                        TSTB    (R5)            ;Is the line blank?
                        BEQ     13$             ;Branch if no line name
                        .PRINT  R5              ;Print asciz name string
                        BR      30$
000000G         13$:    MOV     LCLUNT(R1),R5   ;Is this line used as a CL unit?
                        BGE     21$             ;Br if yes
000000G                 CMP     R1,#LSTPL       ;Is this a real or CL line?
                        BHI     23$             ;Br if CL line
000000G 000000G         BIT     #$KINIT,LSW(R1) ;Is job logged on?
                        BEQ     14$             ;Br if not
000000G                 CALL    PRTUNM          ;Print user's name
                        BR      14$
                23$:    .PRINT  #CLFREE         ;This is a free CL line
                        BR      14$
                21$:    .PRINT  #CLUNIT         ;Print "CL line "
                        ASR     R5              ;Convert to unit number
000000G                 CALL    PRTDEC          ;Print decimal value
                14$:
                ;
                ;  Terminate print line and loop if more lines to print
                ;
                        .PRINT  #CRLF           ;Terminate print line
000002  062701  30$:    ADD     #2,R1           ;Advance line index number
000000G 020127          CMP     R1,#LSTHL       ;Are there more lines?
                        BHI     20$             ;Br if not
000020  020127          CMP     R1,#LSTPL+2     ;Have we stepped past last real line?
                        BNE     22$             ;Br if not
000000G 012701          MOV     #FSTIOL,R1      ;Skip up to 1st CL line
176666  000167  22$:    JMP     1$              ;Go display info for next line
                ;
                ;  Finished
                ;
                20$:    MOV     (SP)+,R5
                        MOV     (SP)+,R4
                        MOV     (SP)+,R3
                        MOV     (SP)+,R2
                        MOV     (SP)+,R1
                        RETURN
```

```
                        .SBTTL  .       CL
                ;-----------------------------------------------------------------------
                ;  Display information about CL units.
                ;
003002  010146  SHOCL:  MOV     R1,-(SP)
003004  010346          MOV     R3,-(SP)
003006  010446          MOV     R4,-(SP)
003010  010546          MOV     R5,-(SP)
                ;
                ;  See if there are any CL units
                ;
003012  005727  000000  TST     #CLTOTL         ;Are there any CL units?
003016  001005          BNE     11$             ;Br if yes
003020                  .PRINT  #TM$CLO         ;There are no CL units
003026  000167  000466  JMP     9$
                ;
                ;  There are some CL units.
                ;  Print version number.
                ;
003032          11$:    .PRINT  #TM$CL7         ;"CL emulating XL version "
003040  116705  000000  MOVB    CLVERS,R5       ;Set number to be printed
003044  004767  000000  CALL    PRTDEC          ;Print the decimal value
003050                  .PRINT  #TM$CL8         ;". "
                ;
                ;  Print heading lines.
                ;
003056          1$:     .PRINT  #TM$CL1         ;Heading line 1
003064                  .PRINT  #TM$CL2         ;Heading line 2
                ;
                ;  Begin loop to print info about each unit.
                ;
003072  005001          CLR     R1              ;Init CL unit index number
                ;
                ;  Print CL unit name
                ;
003074  010105  2$:     MOV     R1,R5           ;Get CL unit index
003076  006205          ASR     R5              ;Convert to unit number
003100  020527  000007  CMP     R5,#7.          ;Unit # in range 0-7 get name CL
003104  101406          BLOS    19$
003106                  .PRINT  #TM$C13         ;Print " C1"
003114  162705  000010  SUB     #8.,R5          ;Remove C1 unit number bias
003120  000403          BR      20$
003122          19$:    .PRINT  #TM$CL3         ;Print " CL"
003130  004767  000000  20$:    CALL    PRTDEC  ;Print unit #
003134  012703  000002  MOV     #2,R3           ;Print 2 spaces
003140  004767  000000  CALL    PRTSPC
                ;
                ;  Print the number of the associated line
                ;
003144  016105  000000  MOV     CL$LIX(R1),R5   ;Get associated line number
003150  001412          BEQ     3$              ;Br if there is no associated line
003152  006205          ASR     R5              ;Get line #
003154  012703  000003  MOV     #3.,R3          ;Get print field width
003160  004767  000000  CALL    PRTFIX          ;Print the line number
003164  012703  000003  MOV     #3.,R3          ;Print 3 spaces
003170  004767  000000  CALL    PRTSPC
003174  000407          BR      4$
```

```
003176                          3$:     .PRINT  #TM$CL4         ;Print "none"
003204  012703  000002                  MOV     #2,R3           ;Print 2 spaces
003210  004767  000000G                 CALL    PRTSPC
                                ;
                                ;  Print number of any job that is using this CL unit
                                ;
003214  010100                  4$:     MOV     R1,R0           ;Get CL index number
003216  004767  000000G                 CALL    CKCLUS          ;See if any job is using CL unit
003222  010005                          MOV     R0,R5           ;Get # of job that is using the device
003224  001411                          BEQ     18$             ;Br if device is free
003226  012703  000003                  MOV     #3,R3           ;Print in 3 character field
003232  004767  000000G                 CALL    PRTFIX          ;Print the job number
003236  012703  000002                  MOV     #2,R3           ;Print 2 spaces
003242  004767  000000G                 CALL    PRTSPC
003246  000407                          BR      17$
003250                          18$:    .PRINT  #TM$CL4         ;Print "none"
003256  012703  000001                  MOV     #1,R3           ;Print 1 space
003262  004767  000000G                 CALL    PRTSPC
                                ;
                                ;  See if this CL unit is spooled
                                ;
003266  016705  000000G         17$:    MOV     CLDEVX,R5       ;Get CL device index number
003272  010100                          MOV     R1,R0           ;Get CL unit # index
003274  006200                          ASR     R0              ;Convert to unit #
003276  020027  000007                  CMP     R0,#7           ;CL or C1 type unit?
003302  101404                          BLOS    21$             ;Br if CL
003304  162700  000010                  SUB     #8.,R0          ;Remove C1 unit bias
003310  016705  000000G                 MOV     C1DEVX,R5       ;Get C1 device index number
003314  000300                  21$:    SWAB    R0              ;Put unit number in high order byte
003316  050005                          BIS     R0,R5           ;Combine with device index #
003320  012700  000000G                 MOV     #SDCB,R0        ;Point to first SDCB
003324  020027  000000G         5$:     CMP     R0,#SDCBND      ;Have we checked all spooled devices?
003330  103011                          BHIS    7$              ;Br if yes -- This CL unit is not spooled
003332  020560  000000G                 CMP     R5,SDDVU(R0)    ;Is this SDCB for this CL unit?
003336  001403                          BEQ     6$              ;Br if yes -- CL unit is spooled
003340  062700  000000G                 ADD     #SDCBSZ,R0      ;Point to next SDCB
003344  000767                          BR      5$
003346                          6$:     .PRINT  #TM$CL5         ;Print "(spooled)"
                                ;
                                ;  Print which options are selected for this line
                                ;
003354  005761  000000G         7$:     TST     CL$LIX(R1)      ;Is this CL unit assigned to a line?
003360  001445                          BEQ     10$             ;Br if not -- Don't display options
003362  016105  000000G                 MOV     CL$OPT(R1),R5   ;Get option flags for this CL unit
003366  012703  003532'                 MOV     #CLOPTB,R3      ;Point to option name table
003372  005004                          CLR     R4              ;Say no option names printed yet
003374  032305                  15$:    BIT     (R3)+,R5        ;Is this option selected?
003376  001420                          BEQ     16$             ;Br if not
003400  005704                          TST     R4              ;Is this the 1st option?
003402  001004                          BNE     12$             ;Br if not
003404                                  .PRINT  #TM$CL6         ;Print " ["
003412  000404                          BR      13$
003414                          12$:    .TTYOUT #54             ;Print comma
003424  010304                  13$:    MOV     R3,R4           ;Point to option name string
003426  112400                  14$:    MOVB    (R4)+,R0        ;Get next char from option name string
003430  001403                          BEQ     16$             ;Br if hit end of name string
003432                                  .TTYOUT                 ;Print next char of option name
```

```
    115 003436  000773                        BR      14$             ;Loop to print rest of option name
    116 003440  105723                16$:    TSTB    (R3)+           ;Search for end of option name
    117 003442  001376                        BNE     16$             ;Loop till end found
    118 003444  005203                        INC     R3              ;Point to next word
    119 003446  042703  000001                BIC     #1,R3
    120 003452  020327  003670'               CMP     R3,#CLOPND      ;Checked all options?
    121 003456  103746                        BLO     15$             ;Loop if not
    122 003460  005704                        TST     R4              ;Were any options printed?
    123 003462  001404                        BEQ     10$             ;Br if not
    124 003464                                .TTYOUT #']             ;Close option list
    125                               ;
    126                               ;  Terminate the print line
    127                               ;
    128 003474                        10$:    .PRINT  #CRLF           ;End of print line
    129                               ;
    130                               ;  See if there are more CL units
    131                               ;
    132 003502  062701  000002                ADD     #2,R1           ;Advance CL unit index
    133 003506  020127  000000C               CMP     R1,#2*CL.TOTL   ;Have we done all CL units?
    134 003512  103002                        BHIS    9$              ;Br if yes
    135 003514  000167  177354                JMP     2$              ;Go show info for next unit
    136                               ;
    137                               ;  Finished
    138                               ;
    139 003520  012605                9$:     MOV     (SP)+,R5
    140 003522  012604                        MOV     (SP)+,R4
    141 003524  012603                        MOV     (SP)+,R3
    142 003526  012601                        MOV     (SP)+,R1
    143 003530  000207                        RETURN
    144                               ;
    145                               ;  Table of CL option flags and option names
    146                               ;
    147                                       .MACRO  CLOP    FLAG,NAME
    148                                       .WORD   FLAG
    149                                       .ASCIZ  /'NAME'/
    150                                       .EVEN
    151                                       .ENDM   CLOP
    152                               ;
    153 003532                        CLOPTB:
    154 003532                                CLOP    CO$FF,FORM
    155 003542                                CLOP    CO$TAB,TAB
    156 003550                                CLOP    CO$LC,LC
    157 003556                                CLOP    CO$8BT,EIGHTBIT
    158 003572                                CLOP    CO$LFO,LFOUT
    159 003602                                CLOP    CO$LFI,LFIN
    160 003612                                CLOP    CO$FFO,FORMO
    161 003622                                CLOP    CO$BNO,BINOUT
    162 003634                                CLOP    CO$BNI,BININ
    163 003644                                CLOP    CO$CR,CR
    164 003652                                CLOP    CO$CTL,CTRL
    165 003662                                CLOP    CO$DTR,DTR
    166 003670                        CLOPND:
```

```
                                        .SBTTL  .       JOBS
                                ;-----------------------------------------------------------------------
                                ;  Show information about all jobs.
                                ;
                                ;  Define /PPN option for SHOW JOBS command
                                ;
003670                                  TBLDEF  SJO,1
000326                                  CMDDEF  P*PN,0
000332                                  TBLEND
                                ;
003670  010146                  SHOJOB: MOV     R1,-(SP)
003672  010246                          MOV     R2,-(SP)
003674  010346                          MOV     R3,-(SP)
003676  010446                          MOV     R4,-(SP)
003700  010546                          MOV     R5,-(SP)
                                ;
                                ;  See if we should display PPN instead of User names
                                ;
003702  105713                          TSTB    (R3)            ;ANY COMMAND OPTIONS?
003704  001420                          BEQ     40$             ;BRANCH IF NOT
003706  121327  000057                  CMPB    (R3),#'/        ;VALID SWITCH LEADIN?
003712  001015                          BNE     40$             ;IGNORE IF NOT
003714  005203                          INC     R3              ;SKIP /
003716  012704  000324'                MOV     #SJOHD,R4       ;POINT TO SHOW JOBS OPTION TABLE
003722  004767  0000000                 CALL    SEARCH          ;VALID OPTION?
003726  103004                          BCC     41$             ;REMEMBER /PPN IF SO
003730                                  FABORT  #INVOPT
003740  112767  000001  0000000 41$:    MOVB    #1,SJSPPN       ;/PPN ONLY VALID OPTION AT THIS TIME
                                ;
                                ;  Print system usage statistics
                                ;
003746  004767  006726          40$:    CALL    PRTUSE          ;PRINT SYSTEM USAGE STATISTICS
                                ;
                                ;  Print heading lines
                                ;
003752                                  .PRINT  #SYHD1          ;PRINT FIRST LINE OF JOB HEADING
003760                                  .PRINT  #SYHD2          ;AND UNDERLINE IT
                                ;
                                ;  Begin loop to display information about each active line
                                ;
003766  012701  000002                  MOV     #2,R1           ;GET 1ST LINE INDEX #
                                ;
                                ;  See if line is active
                                ;
003772  032761  0000000 0000000 1$:     BIT     #$KINIT,LSW(R1) ;HAS LINE BEEN INITIALIZED?
004000  001002                          BNE     20$             ;BR IF YES
004002  000167  000624                  JMP     30$             ;TRY NEXT LINE
                                ;
                                ;  Print line number
                                ;
004006  010105                  20$:    MOV     R1,R5           ;GET LINE INDEX #
004010  004767  0000000                 CALL    PRTLN           ;PRINT THE LINE #
                                ;
                                ;  Print '*' if this is our line
                                ;
004014  112700  000040                  MOVB    #' ,R0          ;ASSUME THAT IT'S NOT OUR LINE
004020  120167  0000000                 CMPB    R1,CORUSR       ;IS THIS OUR LINE?
```

```
 004024  001002                          BNE     2$               ;BR IF NOT
 004026  112700  000052                  MOVB    #'*,R0           ;PRINT * IF YES
 004032                          2$:     .TTYOUT
 004036                                  .PRINT  #SPACE2          ;PRINT 2 SPACES
                                 ;
                                 ;  Print information about owner jobs
                                 ;
 004044  020127  000000G                 CMP     R1,#LSTPL        ;IS THIS A PRIMARY LINE?
 004050  101407                          BLOS    3$               ;BR IF PRIMARY LINE
 004052  020127  000000G                 CMP     R1,#LSTDL        ;DETACHED JOB?
 004056  101004                          BHI     3$               ;BR IF NOT
 004060                                  .PRINT  #DETTXT          ;PRINT "Detached"
 004066  000443                          BR      18$
 004070  116105  000000G         3$:     MOVB    LNPRIM(R1),R5    ;GET OWNER LINE NUMBER
 004074  004767  000000G                 CALL    PRTLN            ;PRINT OWNER LINE NUMBER
 004100                                  .TTYOUT #'(              ;ENCLOSE VIRTUAL LINE # IN ( )
 004110  020127  000000G                 CMP     R1,#LSTPL        ;IS THIS A PRIMARY LINE?
 004114  101412                          BLOS    19$              ;BR IF YES
 004116  016504  000000G                 MOV     LSECPT(R5),R4    ;POINT TO VIRTUAL LINE # TABLE FOR OWNER JOB
 004122  012700  000000G                 MOV     #MAXSEC,R0       ;GET HIGHEST VIRTUAL LINE #
 004126  012705  000001                  MOV     #1,R5            ;START WITH VIRTUAL LINE #1
 004132  120124                  21$:    CMPB    R1,(R4)+         ;SEARCH FOR LINE WITHIN TABLE
 004134  001403                          BEQ     22$              ;BR IF FOUND RIGHT ONE
 004136  005205                          INC     R5               ;INC VIRTUAL LINE #
 004140  077004                          SOB     R0,21$           ;LOOP TILL FOUND
 004142  005005                  19$:    CLR     R5               ;SAY THIS IS LINE 0
 004144  004767  000000G         22$:    CALL    PRTDEC           ;PRINT VIRTUAL LINE #
 004150                                  .TTYOUT #')              ;CLOSE VIRTUAL LINE #
 004160  020527  000011                  CMP     R5,#9.           ;1 OR 2 DIGIT VIRTUAL LINE #?
 004164  101004                          BHI     18$              ;BR IF 2 DIGIT
 004166                                  .TTYOUT #40              ;PRINT A SPACE
 004176                          18$:    .PRINT  #SPACE2          ;PRINT 2 SPACES
                                 ;
                                 ;  Print job priority
                                 ;
 004204  116105  000000G                 MOVB    LPRI(R1),R5      ;GET JOB'S PRIORITY VALUE
 004210  012703  000003                  MOV     #3,R3            ;PRINT IN 3 SPACES
 004214  004767  000000G                 CALL    PRTFIX
 004220                                  .PRINT  #SPACE2          ;PRINT 2 SPACES
                                 ;
                                 ;  Print current line state
                                 ;
 004226  005003                          CLR     R3
 004230  016102  000000G                 MOV     LSTATE(R1),R2    ;GET USER'S CURRENT EXECUTION STATE
 004234  020263  004670'         4$:     CMP     R2,STBIT(R3)     ;LOOK UP THE STATE
 004240  001416                          BEQ     6$               ;BR WHEN FOUND
 004242  062703  000002                  ADD     #2,R3            ;TRY NEXT FLAG
 004246  020327  000050                  CMP     R3,#NUMST        ;DONE ALL?
 004252  103770                          BLO     4$               ;BR IF NOT
 004254  012700  000000G                 MOV     #RNMS,R0         ;ASSUME 'RN' STATE
 004260  020227  000000G                 CMP     R2,#S$$RUN       ;IS JOB EXECUTABLE?
 004264  101402                          BLOS    25$              ;USE 'RN' IF SO
 004266  012700  000000G                 MOV     #WTMS,R0         ;GENERIC WAIT IF STATE NOT RN OR IN TABLE
 004272                          25$:    .PRINT                   ;PRINT 'RN' OR 'WT' AS STATE
 004274  000406                          BR      5$
 004276  016367  004740' 000504  6$:     MOV     STNAM(R3),STPBUF;MOVE STATE NAME TO PRINT BUFFER
 004304                                  .PRINT  #STPBUF          ;PRINT STATE NAME
```

```
                                        ;
                                        ;  Print "SWP" if job is outswapped
                                        ;
004312  032761  0000000 0000000 5$:     BIT     #$INCOR,LSW(R1) ;IS JOB IN MEMORY NOW?
004320  001004                          BNE     12$             ;BR IF YES
004322                                  .PRINT  #SWPTX          ;PRINT "SWP"
004330  000413                          BR      13$
                                        ;
                                        ;  Print "Lock" if job is locked in memory
                                        ;
004332  032761  0000000 0000000 12$:    BIT     #$MLOCK,LSW6(R1);IS JOB LOCKED IN MEMORY?
004340  001404                          BEQ     14$             ;BR IF NOT
004342                                  .PRINT  #LOCKTX         ;PRINT "LOCK"
004350  000403                          BR      13$
004352                                  .PRINT  #SPACE5         ;PRINT SPACES
                                        ;
                                        ;  Pring current memory size
                                        ;
004360                                  .TTYOUT #40             ;PRINT A SPACE
004370                                  .TTYOUT #40             ;PRINT ANOTHER SPACE
004400  016105  0000000                 MOV     LNBLKS(R1),R5   ;GET # 256-WORD BLOCKS ASSIGNED TO JOB
004404  066105  0000000                 ADD     LNSBLK(R1),R5   ;ADD # BLOCKS USED FOR PLAS REGIONS
004410  005205                          INC     R5              ;ROUND UP
004412  006205                          ASR     R5              ;CVT TO # K BYTES
004414  004767  0000000                 CALL    PRTDC3          ;PRINT # K-BYTES OF MEMORY USED
004420                                  .PRINT  #KBMSG          ;PRINT "Kb"
004426                                  .PRINT  #SPACE2         ;PUT IN 2 SPACES
                                        ;
                                        ;  Print job connect time
                                        ;
004434  016705  0000000                 MOV     MINTIM,R5       ;GET CURRENT MINUTE TIMER VALUE
004440  166105  0000000                 SUB     LCONTM(R1),R5   ;CALCULATE CONNECT TIME FOR LINE
004444  005205                          INC     R5              ;CHARGE A MINIMUM OF 1 MINUTE
004446  005004                          CLR     R4              ;CLEAR HIGH-ORDER FOR DIVIDE
004450  012703  000074                  MOV     #60.,R3         ;SET TO DIVIDE BY 60.
004454  004767  0000000                 CALL    DIVIDE          ;DIVIDE BY 60 TO GET HOURS AND MINUTES
004460  010046                          MOV     R0,-(SP)        ;SAVE NUMBER OF MINUTES
004462  020527  000144                  CMP     R5,#100.        ;DO WE NEED MORE THAN 2 DIGITS FOR HOURS?
004466  103405                          BLO     23$             ;BR IF NOT
004470  012703  000003                  MOV     #3,R3           ;PRINT 3 DIGITS
004474  004767  0000000                 CALL    PRTFIX          ;PRINT # HOURS
004500  000406                          BR      24$
004502                                23$:      .TTYOUT #40     ;PRINT A SPACE
004512  004767  0000000                 CALL    PRTDC2          ;PRINT # HOURS WITH 2 DIGITS
004516  012605                        24$:      MOV     (SP)+,R5        ;GET # MINUTES CONNECTED
004520                                          .TTYOUT #':             ;PRINT COLON AFTER HOURS
004530  004767  0000000                         CALL    PRTDC2          ;PRINT # MINUTES CONNECTED
004534                                          .PRINT  #COL00          ;PRINT ':00' SECONDS
                                        ;
                                        ;  Print cpu time
                                        ;
004542                                  .PRINT  #SPACE2         ;PRINT 2 SPACES
004550  016104  0000000                 MOV     LCPUHI(R1),R4   ;GET HIGH ORDER CPU TIME (CLOCK TICKS)
004554  016105  0000000                 MOV     LCPULO(R1),R5   ;GET LOW ORDER CPU TIME (CLOCK TICKS)
004560  004767  0000000                 CALL    PRTTMV          ;PRINT TIME VALUE
                                        ;
                                        ;  Print name of running program
```

```
                                        ;
004564                                          .PRINT  #SPACE2
004572  016100  0000000                         MOV     LPRG1(R1),R0    ;GET 1ST 3 CHARS OF NAME (RAD50)
004576  004767  0000000                         CALL    PRTR50          ;PRINT THEM
004602  016100  0000000                         MOV     LPRG2(R1),R0    ;PRINT 2ND 3 CHARS
004606  004767  0000000                         CALL    PRTR50
                                        ;
                                        ;  Print user name or PPN
                                        ;
004612                                          .PRINT  #SPACE3         ;PRINT 3 SPACES
004620  004767  0000000                         CALL    PRTUNM          ;PRINT USER NAME
004624                                          .PRINT  #CRLF
                                        ;
                                        ;  Do next line
                                        ;
004632  062701  000002          30$:    ADD     #2,R1           ;GET NEXT LINE INDEX #
004636  020127  0000000                 CMP     R1,#LSTSIZ      ;CHECKED ALL?
004642  101002                          BHI     99$             ;BR IF YES
004644  000167  177122                  JMP     1$              ;DO NEXT LINE
                                        ;
                                        ;  Finished
                                        ;
004650  105067  0000000         99$:    CLRB    SJSPPN          ;TURN OFF /PPN SWITCH
004654  012600                          MOV     (SP)+,R5
004656  012604                          MOV     (SP)+,R4
004660  012603                          MOV     (SP)+,R3
004662  012602                          MOV     (SP)+,R2
004664  012601                          MOV     (SP)+,R1
004666  000207                          RETURN

                                ;
                                ;-----------------------------
                                ;  JOB STATE CODES
                                ;
004670  0000000                 STBIT:  .WORD   S$RT            ;REAL-TIME
004672  0000000                         .WORD   S$TTSC          ;TT INPUT DONE AND SINGLE CHAR ACTIVATION
004674  0000000                         .WORD   S$TTFN          ;TT INPUT DONE
004676  0000000                         .WORD   S$OTFN          ;TT OUTPUT BUFFER EMPTY
004700  0000000                         .WORD   S$HICP          ;INTERACTIVE JOB COMPUTATION
004702  0000000                         .WORD   S$TWFN          ;TIMED WAIT FINISH
004704  0000000                         .WORD   S$OTLO          ;OUTPUT BUFFER ALMOST EMPTY
004706  0000000                         .WORD   S$IOFN          ;I/O WAIT COMPLETION
004710  0000000                         .WORD   S$LOW           ;LOW PRIORITY COMPUTATION
004712  0000000                         .WORD   S$INWT          ;INPUT WAIT
004714  0000000                         .WORD   S$OTWT          ;OUTPUT WAIT
004716  0000000                         .WORD   S$TMWT          ;TIMED WAIT
004720  0000000                         .WORD   S$SPND          ;JOB HAS DONE A .SPND
004722  0000000                         .WORD   S$MSWT          ;MESSAGE WAIT
004724  0000000                         .WORD   S$QUSR          ;WAITING FOR USR
004726  0000000                         .WORD   S$IOWT          ;WAITING FOR I/O TO FINISH
004730  0000000                         .WORD   S$SFWT          ;WAITING FOR SHARED FILE RECORD
004732  0000000                         .WORD   S$QMIO          ;WAITING FOR MAPPED I/O BUFFER
004734  0000000                         .WORD   S$SPDB          ;WAITING FOR SPOOLED FILE SPACE
004736  0000000                         .WORD   S$SPCB          ;WAITING FOR SPOOLED FILE
000050                          NUMST   =       <.-STBIT>/2
                                ;
                                ;  PARALLEL VECTOR OF STATE NAMES
```

```
                                          ;
    004740     122       124             STNAM:    .ASCII   /RT/               ;S$RT
    004742     111       116                       .ASCII   /IN/               ;S$TISC
    004744     111       116                       .ASCII   /IN/               ;S$TTFN
    004746     111       116                       .ASCII   /IN/               ;S$OTFN
    004750     111       116                       .ASCII   /IN/               ;S$HICP
    004752     110       111                       .ASCII   /HI/               ;S$TWFN
    004754     110       111                       .ASCII   /HI/               ;S$OTLO
    004756     110       111                       .ASCII   /HI/               ;S$IOFN
    004760     114       117                       .ASCII   /LO/               ;S$LOW
    004762     124       111                       .ASCII   /TI/               ;S$INWT
    004764     124       117                       .ASCII   /TO/               ;S$OTWT
    004766     123       114                       .ASCII   /SL/               ;S$TMWT
    004770     123       114                       .ASCII   /SL/               ;S$SPND
    004772     115       123                       .ASCII   /MS/               ;S$MSWT
    004774     125       123                       .ASCII   /US/               ;S$QUSR
    004776     111       117                       .ASCII   /IO/               ;S$IOWT
    005000     123       106                       .ASCII   /SF/               ;S$SFWT
    005002     115       111                       .ASCII   /MI/               ;S$QMIO
    005004     123       120                       .ASCII   /SP/               ;S$SPDB
    005006     123       120                       .ASCII   /SP/               ;S$SPCB
                                          ;
    005010     130       130       200   STPBUF:   .ASCII   /XX/<200>
                                                   .EVEN
```

```
                                                .SBTTL  .       QUEUE
                                        ;-----------------------------------------------------------------------
                                        ;
                                        ;   SHOW QUEUE
                                        ;
005014  010146                  SHOQUE: MOV     R1,-(SP)
005016  012701  0000000                 MOV     #SDCB,R1        ;POINT TO FIRST SDCB
005022  020127  0000000         4$:     CMP     R1,#SDCBND      ;CHECKED ALL SDCB'S?
005026  103024                          BHIS    3$              ;BR IF YES -- THERE ARE NO QUEUED FILES
005030  005761  0000000                 TST     SDFHD(R1)       ;ANY FILES QUEUED FOR THIS DEVICE?
005034  001003                          BNE     1$              ;BR IF YES
005036  062701  0000000                 ADD     #SDCBSZ,R1      ;POINT TO NEXT SDCB
005042  000767                          BR      4$              ;GO CHECK IT
005044                          1$:     .PRINT  #QHDMS1         ;PRINT QUEUE HEAD MESSAGE
005052                                  .PRINT  #QHDMS2         ;Underline the heading
005060  004767  0000000         5$:     CALL    LSTSPL          ;LIST INFO ABOUT FILES QUEUED FOR THIS DEV
005064  062701  0000000                 ADD     #SDCBSZ,R1      ;POINT TO NEXT SDCB
005070  020127  0000000                 CMP     R1,#SDCBND      ;DONE ALL SDCB'S?
005074  103771                          BLO     5$              ;BR IF MORE TO DO
005076  000403                          BR      2$              ;FINISHED
005100                          3$:     .PRINT  #NOFIL          ;THERE ARE NO PENDING SPOOL FILES
005106  012601                  2$:     MOV     (SP)+,R1
005110  000207                          RETURN
```

```
                                        .SBTTL  .       COMMANDS
                                ;-----------------------------------------------------------------------
                                ;  SHOW COMMANDS Calls TSXUCL to display specific or all command definitions.
                                ;
                                ;  Inputs:
                                ;       R3      Points to next keyword in command buffer or end of command
                                ;       R4      Points to next word in SHOW command table
                                ;
                                ;  Outputs:
                                ;       R3      Points to end of input command in CMDBUF
                                ;       R4      Points to end of string to be passed to TSXUCL in CMDBUF
                                ;
005112                          SHOCMD:
                                ;
                                ;  See if there are any user defined commands
                                ;
005112  005767  0000000                 TST     UCLBLK          ;Any user defined commands?
005116  001410                         BEQ     1$              ;Br if not
                                ;
                                ;  There may be user defined commands.
                                ;  Pass any specific command to be displayed to TSXUCL.
                                ;  Enter TSXUCL to show the commands.
                                ;
005120  012704  0000000                 MOV     #CMDBUF,R4      ;Point to beginning of command buffer
                                                                ;Probably holds "SHOW ... " at this point
005124  112724  000077                  MOVB    #'?,(R4)+       ;Tell TSXUCL we want to show commands
005130  112324                  3$:     MOVB    (R3)+,(R4)+     ;Pass next character from command buffer
005132  001376                         BNE     3$              ;Until remainder of command line passed
005134  000167  0000000                 JMP     CALUCL          ;Call TSXUCL
                                ;
                                ;  There are no user defined commands
                                ;
005140                          1$:     .PRINT  #NOUDC          ;No user defined commands
005146  000207                         RETURN
```

```
                                        .SBTTL  .       DEVICES
                                ;-----------------------------------------------------------------------
                                ;  SHOW DEVICES
                                ;
005150  010246                  SHODEV: MOV     R2,-(SP)
                                ;
                                ;  Print heading lines
                                ;
005152                                  .PRINT  #DVSHH1
005160                                  .PRINT  #DVSHH2
005166                                  .PRINT  #DVSHH3
                                ;
                                ;  List status of each device that is installed in the system
                                ;
005174  005002                          CLR     R2              ;Start with the first device
005176  004767  000016          1$:     CALL    DEVDSP          ;Display status about this device
005202  062702  000002                  ADD     #2,R2           ;Get next device index
005206  020267  0000000                 CMP     R2,NUMDEV       ;Finished all devices?
005212  101771                          BLOS    1$              ;Loop if not
                                ;
                                ;  Finished
                                ;
005214  012602                          MOV     (SP)+,R2
005216  000207                          RETURN
```

```
                                ;-----------------------------------------------------------------------------
                                ;  Display a line of status information for a device.
                                ;
                                ;  Inputs:
                                ;   R2 = Device index number.
                                ;
005220  010346                  DEVDSP: MOV     R3,-(SP)
005222  010546                          MOV     R5,-(SP)
                                ;
                                ;  Print the device name
                                ;
005224  012703  000002                  MOV     #2,R3           ;Print 2 spaces
005230  004767  0000000                 CALL    PRTSPC
005234  016200  0000000                 MOV     PNAME(R2),R0    ;Get RAD50 device name
005240  004767  0000000                 CALL    PRTR50          ;Print it
                                ;
                                ;  Print the current I/O count
                                ;
005244  012703  000003                  MOV     #3,R3           ;Print spaces
005250  004767  0000000                 CALL    PRTSPC
005254  016205  0000000                 MOV     HANIOC(R2),R5   ;Get # pending I/O operations for dev
005260  012703  000005                  MOV     #5,R3           ;Print in 5 column field
005264  004767  0000000                 CALL    PRTFIX          ;Print it
                                ;
                                ;  Print the device status word
                                ;
005270  012703  000003                  MOV     #3,R3           ;Print spaces
005274  004767  0000000                 CALL    PRTSPC
005300  016205  0000000                 MOV     DVSTAT(R2),R5   ;Get status word value
005304  012703  000006                  MOV     #6,R3           ;Print 6 digits
005310  004767  0000000                 CALL    OCTFIX          ;Print octal value
                                ;
                                ;  Print the handler virtual base address
                                ;
005314  012703  000003                  MOV     #3,R3           ;Print spaces
005320  004767  0000000                 CALL    PRTSPC
005324  016205  0000000                 MOV     HANENT(R2),R5   ;Get base address of handler
005330  020527  000100                  CMP     R5,#100         ;Is this a real or pseudo handler?
005334  103440                          BLO     9$              ;Branch of pseudo handler
005336  162705  000006                  SUB     #6,R5           ;Get actual handler base address
005342  012703  000006                  MOV     #6,R3           ;Print 6 digits
005346  004767  0000000                 CALL    OCTFIX          ;Print octal value
                                ;
                                ;  Print the handler physical base address
                                ;
005352  012703  000003                  MOV     #3,R3           ;Print spaces
005356  004767  0000000                 CALL    PRTSPC
005362  016205  0000000                 MOV     HANPAR(R2),R5   ;Get physical base address of handler
005366  012703  000006                  MOV     #6,R3           ;Print 6 digits
005372  004767  0000000                 CALL    OCTFIX          ;Print octal value
                                ;
                                ;  Print the handler size
                                ;
005376  012703  000002                  MOV     #2,R3           ;Print 2 spaces
005402  004767  0000000                 CALL    PRTSPC
005406  016205  0000000                 MOV     HANSIZ(R2),R5   ;Get the handler size
005412  012703  000005                  MOV     #5,R3           ;Print 5 digits
```

```
      58 005416  004767  0000000                 CALL    PRTFIX          ;Print decimal value
      59 005422  012703  000002                  MOV     #2,R3           ;Print spaces
      60 005426  004767  0000000                 CALL    PRTSPC
      61                                ;
      62                                ;  Print CSR and Vector addresses for this device
      63                                ;
      64 005432  004767  000014                  CALL    DSPCSR
      65                                ;
      66                                ;  Terminate this print line
      67                                ;
      68 005436                         9$:      .PRINT  #CRLF           ;End the print line
      69                                ;
      70                                ;  Finished
      71                                ;
      72 005444  012605                          MOV     (SP)+,R5
      73 005446  012603                          MOV     (SP)+,R3
      74 005450  000207                          RETURN
```

```
                                ;-------------------------------------------------------------------------------
                                ;  Extract CSR and Vector addresses from a device handler and display them.
                                ;
                                ;  Inputs:
                                ;   R2 = Device index number.
                                ;
005452  010146                  DSPCSR: MOV     R1,-(SP)
005454  010246                          MOV     R2,-(SP)
005456  010346                          MOV     R3,-(SP)
005460  010446                          MOV     R4,-(SP)
005462  010546                          MOV     R5,-(SP)
                                ;
                                ;  Build file spec for device handler
                                ;
005464  016767  0000000 172306          MOV     SYNAME,HANNAM   ;Set physical device name for SY device
005472  016267  0000000 172302          MOV     PNAME(R2),HANNAM+2 ;Set RAD50 device name
                                ;
                                ;  Try to open the device handler file
                                ;
005500                                  .SERR                   ;Don't abort on lookup errors
005506                                  .LOOKUP #XAREA,#1,#HANNAM ;Try to open handler file
005526  103547                          BCS     9$              ;Br if unable to open handler
                                ;
                                ;  Read block 0 of handler and save information about CSR address
                                ;
005530                                  .READW  #XAREA,#1,#BLK0,#256.,#0 ;Read block 0 of handler
005566  103527                          BCS     9$
005570  016767  000000C 172214          MOV     BLK0+H.CSR,DCSR ;Save CSR info
                                ;
                                ;  Read blocks 1 and 2 of handler
                                ;
005576                                  .READW  #XAREA,#1,#BLK0,#512.,#1 ;Read blocks 1 and 2 of handler
005636  016767  000000C 172144          MOV     BLK0-1000+H.VEC,DVEC    ;Save vector info
                                ;
                                ;  If both the CSR and vector are zero, don't display anything
                                ;
005644  016700  172142                  MOV     DCSR,R0         ;Get CSR value
005650  056700  172134                  BIS     DVEC,R0         ;Add Vector value
005654  001474                          BEQ     9$              ;Br if both are zero
                                ;
                                ;  Determine if this is a handler for a PRO device with floating
                                ;  vector and CSR addresses.
                                ;
005656  005001                          CLR     R1              ;Assume this is not a PRO device
005660  012704  000000C                 MOV     #BLK0-1000+H.VEC,R4     ;Point to handler vector cell
005664  012400                          MOV     (R4)+,R0        ;Get address of vector
005666  002013                          BGE     3$              ;Br if not pointer to vector list
005670  006300                          ASL     R0              ;Get byte offset to vector list
005672  060004                          ADD     R0,R4           ;Point to start of vector list
005674  005714                          TST     (R4)            ;Is this a PRO device with floating addresses?
005676  002007                          BGE     3$              ;Br if not
005700  005724                          TST     (R4)+           ;Point to word with PRO device ID
005702  012401                          MOV     (R4)+,R1        ;Get PRO device ID
005704  010100                          MOV     R1,R0           ;Get PRO device ID
005706  004767  000226                  CALL    PIDCSR          ;Get CSR address for PRO device
005712  010005                          MOV     R0,R5           ;Get to R5 for OCTFIX
005714  000402                          BR      4$              ;Go display it
```

```
                                        ;
                                        ;  Display the CSR value
                                        ;
005716  016705  172070        3$:     MOV     DCSR,R5         ;Get CSR value
005722  012703  000002        4$:     MOV     #2,R3           ;Print 2 spaces
005726  004767  0000000               CALL    PRTSPC
005732  012703  000006               MOV     #6,R3           ;Print 6 digits
005736  004767  0000000               CALL    OCTFIX          ;Print octal value
                                        ;
                                        ;  Display vector addresses
                                        ;
005742  012703  000002               MOV     #2,R3           ;Print 2 spaces
005746  004767  0000000               CALL    PRTSPC
005752  016705  172032               MOV     DVEC,R5         ;Single vector or vector list?
005756  001433                       BEQ     9$              ;Br if no vector
005760  002405                       BLT     1$              ;Br if multiple vector
005762  012703  000003               MOV     #3,R3           ;Print 3 digits
005766  004767  0000000               CALL    OCTFIX          ;Print octal value
005772  000425                       BR      9$
                                        ;
                                        ;  We have multiple vectors (and possibly floating PRO vectors)
                                        ;
005774  005701              1$:     TST     R1              ;Do we have floating vectors for PRO device?
005776  001404                       BEQ     5$              ;Br if not
006000  010100                       MOV     R1,R0           ;Get PRO device ID
006002  004767  000154               CALL    PIDVEC          ;Get base vector based on device ID
006006  010001                       MOV     R0,R1           ;Save base vector location for device
006010  011405              5$:     MOV     (R4),R5         ;Get address of next vector
006012  060105                       ADD     R1,R5           ;Add base vector address
006014  012703  000003               MOV     #3,R3           ;Print 3 digits
006020  004767  0000000               CALL    OCTFIX          ;Print octal value
006024  062704  000006               ADD     #6,R4           ;Point to next vector entry
006030  005714                       TST     (R4)            ;Are there more vectors?
006032  003405                       BLE     9$              ;Br if not
006034  012703  000001               MOV     #1,R3           ;Print 1 space
006040  004767  0000000               CALL    PRTSPC
006044  000761                       BR      5$              ;Print next vector
                                    ;
                                    ;  Finished
                                    ;
006046                      9$:     .CLOSE  #1              ;Close handler file
006054                              .HERR                   ;Reset error aborts
006062  012605                       MOV     (SP)+,R5
006064  012604                       MOV     (SP)+,R4
006066  012603                       MOV     (SP)+,R3
006070  012602                       MOV     (SP)+,R2
006072  012601                       MOV     (SP)+,R1
006074  000207                       RETURN
```

```
                                ;-----------------------------------------------------------------------------
                                ;  Convert a PRO device ID number into the option slot # for the device.
                                ;
                                ;  Inputs:
                                ;   R0 = PRO device ID.
                                ;
                                ;  Outputs:
                                ;   C-flag set ==> Could not find device ID.
                                ;   R0 = Option slot number.
                                ;
006076  010146          PIDSLT: MOV     R1,-(SP)
006100  012701  0000000         MOV     #PROSLT,R1      ;Point to entry for slot 0
006104  021127  177777  1$:     CMP     (R1),#-1        ;Checked all entries?
006110  001410                  BEQ     2$              ;Br if yes
006112  020021                  CMP     R0,(R1)+        ;Search for correct entry
006114  001373                  BNE     1$              ;Loop if this is not it
006116  162701  0000020         SUB     #PROSLT+2,R1    ;Get byte index for slot
006122  006201                  ASR     R1              ;Get word index
006124  010100                  MOV     R1,R0           ;Return in R0
006126  000241                  CLC                     ;Signal success on return
006130  000401                  BR      9$
006132  000261          2$:     SEC                     ;Signal failure on return
006134  012601          9$:     MOV     (SP)+,R1
006136  000207                  RETURN

                                ;-----------------------------------------------------------------------------
                                ;  Determine the address of the CSR for a PRO device given the device ID.
                                ;
                                ;  Inputs:
                                ;   R0 = PRO device ID
                                ;
                                ;  Outputs:
                                ;   C-flag set ==> Do not recognize the device.
                                ;   R0 = CSR address
                                ;
006140  004767  177732  PIDCSR: CALL    PIDSLT          ;Determine slot where device is installed
006144  103405                  BCS     9$              ;Br if don't recognize device
006146  072027  000007          ASH     #7.,R0          ;CSR addresses are 200 apart per slot
006152  062700  174000          ADD     #174000,R0      ;CSR for slot 0 is here
006156  000241                  CLC                     ;Signal success on return
006160  000207          9$:     RETURN

                                ;-----------------------------------------------------------------------------
                                ;  Determine the base vector for a PRO device given the device ID.
                                ;
                                ;  Inputs:
                                ;   R0 = PRO device ID
                                ;
                                ;  Outputs:
                                ;   C-flag set ==> Did not recognize device.
                                ;   R0 = Base vector location for device.
                                ;
006162  004767  177710  PIDVEC: CALL    PIDSLT          ;Determine what slot has controller
006166  103405                  BCS     9$              ;Br if don't recognize device
006170  072027  000003          ASH     #3.,R0          ;Vectors are 8 bytes apart per slot
006174  062700  000300          ADD     #300,R0         ;Vector for slot 0 is at 300
006200  000241                  CLC                     ;Signal success on return
```

```
      58 006202  000207                    9$:      RETURN
```

```
                                        .SBTTL  .       ASSIGNS
                                ;-------------------------------------------------------------------------------
                                ;  Display assignments
                                ;
006204  010246                  SHOASN: MOV     R2,-(SP)
006206                                  .PRINT  #ASNHD1         ;PRINT HEADING
                                ;
                                ;  If there is no assignment for SY, print default assignment
                                ;
006214  012702  0000000                 MOV     #ASNTBL,R2      ;Point to assign table
006220  026762  0000000 0000000 5$:     CMP     R50SY,AT$LOG(R2);Is this entry for SY?
006226  001423                          BEQ     6$              ;Br if yes
006230  062702  0000000                 ADD     #AT$$SZ,R2      ;Point to next entry
006234  020227  0000000                 CMP     R2,#ASNEND      ;Checked all assign entries?
006240  103767                          BLO     5$              ;Loop if not
006242                                  .PRINT  #SYASHD         ;Print "SY --> "
006250  016700  0000000                 MOV     SYNAME,R0       ;Get default device
006254  004767  0000000                 CALL    PRTR50
006260                                  .TTYOUT #':
006270                                  .PRINT  #CRLF
                                ;
                                ;  If there is no assignment for DK, print default assignment
                                ;
006276  012702  0000000         6$:     MOV     #ASNTBL,R2      ;Point to assign table
006302  026762  0000000 0000000 7$:     CMP     R50DK,AT$LOG(R2);Is this entry for DK?
006310  001423                          BEQ     8$              ;Br if yes
006312  062702  0000000                 ADD     #AT$$SZ,R2      ;Point to next entry
006316  020227  0000000                 CMP     R2,#ASNEND      ;Checked all assign entries?
006322  103767                          BLO     7$              ;Loop if not
006324                                  .PRINT  #DKASHD         ;Print "DK --> "
006332  016700  0000000                 MOV     SYNAME,R0       ;Get default device
006336  004767  0000000                 CALL    PRTR50          ;Print it
006342                                  .TTYOUT #':
006352                                  .PRINT  #CRLF
                                ;
                                ;  Now print user assign table entries
                                ;
006360  012702  0000000         8$:     MOV     #ASNTBL,R2      ;POINT TO ASSIGN TABLE
006364  005762  0000000         2$:     TST     AT$LOG(R2)      ;IS THIS ASSIGN TABLE ENTRY IN USE?
006370  001461                          BEQ     3$              ;BR IF NOT
006372  016200  0000000                 MOV     AT$LOG(R2),R0           ;GET LOGICAL DEVICE NAME
006376  004767  0000000                 CALL    PRTR50          ;DISPLAY LOGICAL DEVICE NAME
006402                                  .PRINT  #ASNHD2         ;PRINT ARROW
006410  016200  0000000                 MOV     AT$DEV(R2),R0   ;GET PHYSICAL DEVICE NAME
006414  004767  0000000                 CALL    PRTR50          ;PRINT IT
006420                                  .TTYOUT #':             ;PUT IN COLON
006430  016200  0000000                 MOV     AT$FIL(R2),R0   ;GET 1ST 3 CHARS OF FILE NAME
006434  001434                          BEQ     4$              ;BR IF NO NAME
006436  004767  0000000                 CALL    PRTR50          ;PRINT 1ST PART OF NAME
006442  016200  0000020                 MOV     AT$FIL+2(R2),R0 ;GET 2ND 3 CHARS OF FILE NAME
006446  004767  0000000                 CALL    PRTR50          ;PRINT THE NAME
006452                                  .TTYOUT #'.             ;PUT IN A PERIOD
006462  016200  0000000                 MOV     AT$EXT(R2),R0   ;GET EXTENSION
006466  001417                          BEQ     4$              ;BR IF NONE SPECIFIED
006470  004767  0000000                 CALL    PRTR50          ;DISPLAY THE EXTENSION
006474  016205  0000000                 MOV     AT$SIZ(R2),R5   ;GET FILE SIZE
006500  001412                          BEQ     4$              ;BR IF NONE SPECIFIED
```

```
      58 006502                                          .TTYOUT #'[             ;START SIZE SPEC
      59 006512  004767  0000000                        CALL    PRTDEC          ;DISPLAY THE FILE SIZE
      60 006516                                          .TTYOUT #']             ;TERMINATE THE SIZE SPEC
      61 006526                                  4$:     .PRINT  #CRLF           ;TERMINATE THE LINE
      62 006534  062702  0000000                3$:     ADD     #AT$$SZ,R2      ;POINT TO NEXT ASSIGN ENTRY
      63 006540  020227  0000000                        CMP     R2,#ASNEND      ;HAVE WE REACHED END OF TABLE?
      64 006544  103707                                 BLO     2$              ;BR IF NOT
      65 006546  012602                                 MOV     (SP)+,R2
      66 006550  000207                                 RETURN
```

```
                                        .SBTTL  .       ALLOCATIONS
                                ;-------------------------------------------------------------------------------
                                ;  Show allocations
                                ;
006552  010146                  SHOALC: MOV     R1,-(SP)
006554  010246                          MOV     R2,-(SP)
006556  010346                          MOV     R3,-(SP)
006560  010546                          MOV     R5,-(SP)
                                ;
                                ;  Make a fast scan to see if there are any allocated devices
                                ;
006562  012702  0000000         1$:     MOV     #ALCTBL,R2      ;Point to start of allocation table
006566  105762  0000000                 TSTB    AD$JOB(R2)      ;Is this entry used?
006572  001011                          BNE     2$              ;Br if yes
006574  062702  0000000                 ADD     #AD$$SZ,R2      ;Point to next entry
006600  020227  0000000                 CMP     R2,#ALCEND      ;Checked all entries?
006604  103770                          BLO     1$              ;Loop if not
                                ;
                                ;  There are no allocated devices
                                ;
006606                                  .PRINT  #TM$NAD         ;No allocated devices
006614  000460                          BR      9$              ;Finished
                                ;
                                ;  There are some allocated devices.
                                ;  Print heading lines.
                                ;
006616                          2$:     .PRINT  #CRLF           ;Print a blank line
006624                                  .PRINT  #ALCHD1         ;Print heading line 1
006632                                  .PRINT  #ALCHD2         ;Print heading line 2
                                ;
                                ;  Begin loop to print information about each allocated device
                                ;
006640  012702  0000000                 MOV     #ALCTBL,R2      ;Point to start of allocation table
006644  116205  0000000         3$:     MOVB    AD$JOB(R2),R5   ;Is this entry used?
006650  001435                          BEQ     4$              ;Br if not
                                ;
                                ;  Print the device name
                                ;
006652                                  .TTYOUT #40             ;Put a space in front of the device name
006662  016200  0000000                 MOV     AD$DVU(R2),R0   ;Get device and unit number
006666  004767  0000000                 CALL    CVDVNM          ;Convert dev and unit # to device name
006672  004767  0000000                 CALL    PRTR50          ;Print the device name
006676  012703  000004                  MOV     #4.,R3          ;Print 4 spaces
006702  004767  0000000                 CALL    PRTSPC
                                ;
                                ;  Print the number of the job to which the device is allocated
                                ;
006706  012703  000002                  MOV     #2.,R3          ;Print job number in 2 col field
006712  010501                          MOV     R5,R1           ;Save job index number in R1
006714  006205                          ASR     R5              ;Convert job index number to job number
006716  004767  0000000                 CALL    PRTFIX          ;Print the job number
006722  012703  000003                  MOV     #3.,R3          ;Space over 3 columns
006726  004767  0000000                 CALL    PRTSPC
                                ;
                                ;  Print the user's name
                                ;
006732  004767  0000000                 CALL    PRTUNM          ;Print the user's name
```

```
                                        ;
                                        ;  Terminate this print line
                                        ;
006736                                          .PRINT  #CRLF           ;End of line
                                        ;
                                        ;  See if there are more allocated devices to display
                                        ;
006744  062702  0000000                 4$:     ADD     #AD$$SZ,R2      ;Point to next allocation entry
006750  020227  0000000                         CMP     R2,#ALCEND      ;Finished all entries?
006754  103733                                  BLO     3$              ;Br if more to do
                                        ;
                                        ;  Finished
                                        ;
006756  012605                          9$:     MOV     (SP)+,R5
006760  012603                                  MOV     (SP)+,R3
006762  012602                                  MOV     (SP)+,R2
006764  012601                                  MOV     (SP)+,R1
006766  000207                                  RETURN
```

```
                        .SBTTL  MOUNTS
                ;-------------------------------------------------------------------------------
                ;  Show mounts
                ;
006770  010146  SHOMNT:  MOV     R1,-(SP)
006772  010246          MOV     R2,-(SP)
006774  010346          MOV     R3,-(SP)
006776  010446          MOV     R4,-(SP)
007000  010546          MOV     R5,-(SP)
                ;
                ;  Set up a table of pointers to active mount entries in BLKO area.
                ;
007002  016704  000000G         MOV     CSHDEV,R4       ;Point to start of mount table
007006  012705  000000G         MOV     #BLKO,R5        ;Point to area where we will build table
007012  010400  20$:    MOV     R4,R0           ;Get address of next mount entry
007014  004767  000000G         CALL    CDGET           ;Get mount entry into CDBUF
007020  005767  000000C         TST     CDBUF+CD$DVU    ;Is this mount entry in use?
007024  001404                  BEQ     21$             ;Br if not
007026  020527  001774G         CMP     R5,#BLKO+1020.  ;Is table full?
007032  103001                  BHIS    21$             ;Br if yes
007034  010425                  MOV     R4,(R5)+        ;Save pointer to active mount entry
007036  062704  000000G 21$:    ADD     #CD$$SZ,R4      ;Point to next entry in mount table
007042  020467  000000G         CMP     R4,CSHDVN       ;Have we checked all mount entries?
007046  103761                  BLO     20$             ;Loop if not
007050  005015                  CLR     (R5)            ;Put null pointer at end of table
                ;
                ;  Now sort the list of mount table pointers using two keys:
                ;  key 1 -- Device # & unit #;  key 2 -- base block number of logical disks
                ;
007052  012705  000000G         MOV     #BLKO,R5        ;Point to first entry in table
007056  012503  24$:    MOV     (R5)+,R3        ;Get pointer to a mount entry
007060  001437                  BEQ     26$             ;Br if finished sorting table
007062  010504                  MOV     R5,R4           ;Get pointer to next entry in table
007064  010300  23$:    MOV     R3,R0           ;Get address of mount entry
007066  004767  000000G         CALL    CDGET           ;Copy entry into CDBUF
007072  016702  000000C         MOV     CDBUF+CD$DVU,R2 ;Get device # and unit # of mounted device
007076  000302                  SWAB    R2              ;Put dev # in high-order, unit # in low-order
007100  016767  000000C 170710  MOV     CDBUF+CD$BAS,CDBASE ;Save base block number
007106  012401  25$:    MOV     (R4)+,R1        ;Get pointer to another mount entry
007110  001762                  BEQ     24$             ;Br if checked all beyond reference entry
007112  010100                  MOV     R1,R0           ;Get address of mount entry
007114  004767  000000G         CALL    CDGET           ;Copy entry into CDBUF
007120  016700  000000C         MOV     CDBUF+CD$DVU,R0 ;Get device # and unit # of this entry
007124  000300                  SWAB    R0              ;Rearrange dev # and unit # order
007126  020002                  CMP     R0,R2           ;Look for smallest dev # and unit #
007130  101366                  BHI     25$             ;Br if this device is higher
007132  103404                  BLO     22$             ;Br if this device is smaller than ref one
007134  026767  000000C 170654  CMP     CDBUF+CD$BAS,CDBASE;Same device, now check base block #'s
007142  103361                  BHIS    25$             ;Br if not smaller
007144  010364  177776  22$:    MOV     R3,-2(R4)       ;Swap table entries
007150  010165  177776          MOV     R1,-2(R5)
007154  010103                  MOV     R1,R3           ;Set new reference pointer
007156  000742                  BR      23$             ;Continue sort
                ;
                ;  If there are any mounted devices, print table heading
                ;
007160  005767  000000G 26$:    TST     BLKO            ;Are there any mounted devices?
```

```
     58 007164  001406                        BEQ     30$             ;Br if not
     59 007166                                .PRINT  #SHMTH1         ;Print heading line
     60 007174                                .PRINT  #SHMTH2         ;Underline it
     61                                ;
     62                                ;  Now print information about each device.
     63                                ;
     64 007202  012702  0000000       30$:    MOV     #BLK0,R2        ;Get pointer to table of sorted pointers
     65 007206  012201                1$:     MOV     (R2)+,R1        ;Get next mount table pointer
     66 007210  001002                        BNE     27$             ;Br if got another entry
     67 007212  000167  000414                JMP     2$              ;Finished all devices
     68 007216  010100                27$:    MOV     R1,R0           ;Get address of mount entry
     69 007220  004767  0000000               CALL    CDGET           ;Move entry into CDBUF
     70 007224  016700  000000C               MOV     CDBUF+CD$DVU,R0 ;GET UNIT # / DEVICE # OF MOUNTED DEVICE
     71 007230  001576                        BEQ     8$              ;BR IF NULL -- THIS ENTRY IS FREE
     72                                ;
     73                                ;  Found a mounted device.
     74                                ;  Print device name and unit number
     75                                ;
     76 007232  004767  0000000               CALL    CVDVNM          ;CONVERT DEVICE & UNIT #'S TO DEVICE NAME
     77 007236  004767  0000000               CALL    PRTR50          ;PRINT THE DEVICE NAME
     78 007242                                .TTYOUT #':             ;PRINT ":"
     79                                ;
     80                                ;  If this is a logical disk, print the LD file name
     81                                ;
     82 007252  005767  000000C               TST     CDBUF+CD$BAS    ;IS THIS A LOGICAL DISK?
     83 007256  001005                        BNE     29$             ;BR IF YES
     84 007260                                .PRINT  #SPACE6         ;SPACE OVER 6 COLUMNS
     85 007266  000167  000234                JMP     4$
     86 007272  010146                29$:    MOV     R1,-(SP)        ;SAVE ORIGINAL MOUNT TABLE INDEX
     87 007274  005046                        CLR     -(SP)           ;PUT NULL ON STACK TO SIGNAL END
     88 007276  010146                12$:    MOV     R1,-(SP)        ;SAVE POINTER TO INNER-MOST LOGICAL DISK
     89 007300  016703  000000C               MOV     CDBUF+CD$BAS,R3 ;GET BASE BLOCK NUMBER OF THIS LD
     90 007304  005005                        CLR     R5              ;SAY NO ENCLOSING LD FOUND YET
     91 007306  016767  000000C  170504       MOV     CDBUF+CD$DVU,CDDVU ;Save device and unit info
     92 007314  016704  0000000               MOV     CSHDEV,R4       ;SEARCH THROUGH MOUNTED DEVICE TABLE
     93 007320  010400                13$:    MOV     R4,R0           ;Get address of mount entry
     94 007322  004767  0000000               CALL    CDGET           ;Move into CDBUF
     95 007326  026767  170466  000000C       CMP     CDDVU,CDBUF+CD$DVU; LOOKING FOR ONE WITH SAME PHYS DEVICE
     96 007334  001023                        BNE     14$             ;BR IF NOT THIS ONE
     97 007336  005767  000000C               TST     CDBUF+CD$BAS    ;IS THIS ALSO A LOGICAL DISK?
     98 007342  001420                        BEQ     14$             ;BR IF NOT
     99 007344  020367  000000C               CMP     R3,CDBUF+CD$BAS ;SEE IF THIS LD ENCLOSES INNER LD
    100 007350  101415                        BLOS    14$             ;BR IF NOT
    101 007352  020367  000000C               CMP     R3,CDBUF+CD$TOP ;CHECK UPPER RANGE
    102 007356  103012                        BHIS    14$             ;BR IF NOT ENCLOSING
    103 007360  005705                        TST     R5              ;ANY OTHER ENCLOSING LD'S FOUND SO FAR?
    104 007362  001404                        BEQ     15$             ;BR IF NOT
    105 007364  026767  000000C  170430       CMP     CDBUF+CD$BAS,CDBAS5; SAVE ONE WITH HIGHEST BASE
    106 007372  101404                        BLOS    14$
    107 007374  010405                15$:    MOV     R4,R5           ;THIS IS AN ENCLOSING LOGICAL DISK
    108 007376  016767  000000C  170416       MOV     CDBUF+CD$BAS,CDBAS5
    109 007404  062704  0000000       14$:    ADD     #CD$$SZ,R4      ;POINT TO NEXT MOUNT TABLE ENTRY
    110 007410  020467  0000000               CMP     R4,CSHDVN       ;CHECKED ALL?
    111 007414  103741                        BLO     13$             ;LOOP IF NOT
    112 007416  010501                        MOV     R5,R1           ;DID WE FIND AN ENCLOSING LOGICAL DISK?
    113 007420  001404                        BEQ     16$             ;BR IF NOT
    114 007422  010500                        MOV     R5,R0           ;WE FOUND AN OUTER LD
```

```
007424  004767  000000G            CALL    CDGET             ;GET OUTER LD DATA INTO CDBUF
007430  000722                     BR      12$               ;KEEP SCANNING OUTWARD
007432  012605            16$:     MOV     (SP)+,R5          ;GET POINTER TO OUTER-MOST LOGICAL DISK
007434  012703  000000G            MOV     #PRTBUF,R3        ;GET POINTER TO TEXT BUFFER FOR PRTFNM
007440  010500            17$:     MOV     R5,R0             ;Get address of mount entry
007442  004767  000000G            CALL    CDGET             ;Get mount entry into CDBUF
007446  012700  000000C            MOV     #CDBUF+CD$NAM,R0;GET POINTER TO FILE NAME
007452  004767  000000G            CALL    PRTFNM            ;FORMAT THE FILE NAME
007456  112723  000072             MOVB    #':,(R3)+         ;PUT COLON AFTER FILE NAME
007462  012605                     MOV     (SP)+,R5          ;IS THERE ANOTHER NAME?
007464  001365                     BNE     17$               ;BR IF YES
007466  005303                     DEC     R3                ;POINT BACK TO LAST COLON
007470  020327  000006G   18$:     CMP     R3,#PRTBUF+6.     ;TAB UP TO COLUMN 6
007474  103003                     BHIS    19$
007476  112723  000040             MOVB    #40,(R3)+         ;FILL WITH TRAILING SPACES
007502  000772                     BR      18$
007504  112713  000200    19$:     MOVB    #200,(R3)         ;PUT IN END OF STRING MARKER
007510                             .PRINT  #PRTBUF           ;PRINT THE LOGICAL DISK SPECIFICATION
007516  012601                     MOV     (SP)+,R1          ;GET BACK MOUNT TABLE INDEX
007520  010100                     MOV     R1,R0             ;Get address of mount entry
007522  004767  000000G            CALL    CDGET             ;Move entry into CDBUF
                          ;
                          ;  Print numbers of jobs that have this device mounted
                          ;
007526  012705  000001    4$:      MOV     #1,R5             ;START WITH JOB # 1
007532  012703  000001             MOV     #1,R3             ;GET MOUNT FLAG FOR JOB # 1
007536  012704  000000C            MOV     #CDBUF+CD$JOB,R4;POINT TO TABLE WITH JOB BIT FLAGS
007542  130314            7$:      BITB    R3,(R4)           ;IS DEVICE MOUNTED BY THIS JOB?
007544  001415                     BEQ     5$                ;BR IF NOT
007546                             .TTYOUT #40               ;PRINT A SPACE
007556  020527  000012             CMP     R5,#10.           ;IS THIS A 2 DIGIT JOB NUMBER?
007562  103004                     BHIS    10$               ;BR IF YES
007564                             .TTYOUT #40               ;PRINT EXTRA SPACE FOR COLUMN ALIGNMENT
007574  004767  000000G   10$:     CALL    PRTDEC            ;PRINT JOB NUMBER
007600  106303            5$:      ASLB    R3                ;SHIFT OVER MOUNT FLAG
007602  103002                     BCC     6$                ;BR IF DID NOT SHIFT OUT OF BYTE
007604  005204                     INC     R4                ;POINT TO NEXT BYTE
007606  006103                     ROL     R3                ;RESET FLAG BIT
007610  005205            6$:      INC     R5                ;ADVANCE JOB NUMBER
007612  020527  000000G            CMP     R5,#NLINES        ;CHECKED ALL JOBS?
007616  101751                     BLOS    7$                ;LOOP IF MORE TO CHECK
                          ;
                          ;  Finished with this mount entry
                          ;
007620                             .PRINT  #CRLF             ;END LINE
007626  000167  177354    8$:      JMP     1$                ;Go back and print info for next device
                          ;
                          ;  Finished checking all entries in mount table
                          ;
007632  005767  000000G   2$:      TST     BLKO              ;WERE THERE ANY MOUNTED DEVICES?
007636  001003                     BNE     3$                ;BR IF YES
007640                             .PRINT  #NONEMS           ;PRINT "(NONE)"
                          ;
                          ;  Finished
                          ;
007646  012605            3$:      MOV     (SP)+,R5
007650  012604                     MOV     (SP)+,R4
```

```
    172 007652  012603                         MOV     (SP)+,R3
    173 007654  012602                         MOV     (SP)+,R2
    174 007656  012601                         MOV     (SP)+,R1
    175 007660  000207                         RETURN
```

```
                                        .SBTTL  .       DATE
                                ;-------------------------------------------------------------------------------
                                ;  Display current date.
                                ;
007662                          SHODAT: .DATE                   ;GET CURRENT DATE
007670  005700                          TST     R0              ;IS SYSTEM DATE KNOWN?
007672  001406                          BEQ     1$              ;BR IF NO DATE ENTERED
007674  004767  0000000                 CALL    PRTDAT          ;DISPLAY DATE
007700                                  .PRINT  #CRLF
007706  000403                          BR      9$
                                ;  Date is unknown.
007710                          1$:     .PRINT  #NODAT          ;NO DATE
007716  000207                  9$:     RETURN

                                        .SBTTL  .       TIME
                                ;-------------------------------------------------------------------------------
                                ;  Display current time of day.
                                ;
007720  004767  0000000         SHOTIM: CALL    PRTTOD          ;DISPLAY CURRENT TIME
007724                                  .PRINT  #CRLF
007732  000207                          RETURN

                                        .SBTTL  .       VERSION
                                ;-------------------------------------------------------------------------------
                                ;  Display the TSX-Plus system version number.
                                ;
007734                          SHOVER: .PRINT  #TSXVER         ;"TSX-Plus Version="
007742  012703  0000000                 MOV     #TSXVRS,R3      ;Get system version number
007746  005002                          CLR     R2              ;Clear high-order for divide
007750  071227  000144                  DIV     #100.,R2        ;Divide R2-R3 by 100
007754  010205                          MOV     R2,R5           ;Get major version number
007756  004767  0000000                 CALL    PRTDEC          ;Print major version number
007762                                  .TTYOUT #'.             ;Print decimal point
007772  010305                          MOV     R3,R5           ;Get fractional version number
007774  004767  0000000                 CALL    PRTDC2          ;Print fractional version number
010000                                  .PRINT  #CRLF           ;Terminate print line
010006  000207                          RETURN

                                        .SBTTL  .       USE
                                ;-------------------------------------------------------------------------------
                                ;  Display computer usage information for current job.
                                ;
010010  004767  0000000         SHOUSE: CALL    PRTTIM          ;PRINT CONNECT AND CPU TIME FOR JOB
010014  000207                          RETURN
```

```
                                        .SBTTL  .       INSTALL
                                ;-----------------------------------------------------------------------
                                ;  Show information about installed programs
                                ;
010016  010546                  SHOINS: MOV     R5,-(SP)
010020  004767  0000000                 CALL    CKSYPV          ;Require SYSPRV privilege to do this
                                ;
                                ;  Print title lines
                                ;
010024                                  .PRINT  #TM$IN1
010032                                  .PRINT  #TM$IN2
                                ;
                                ;  Begin loop to print information about each installed program
                                ;
010040  016705  0000000                 MOV     INSTBL,R5       ;Point to 1st install table entry
010044  010567  0000000         1$:     MOV     R5,INGADR       ;Set address of entry to get
010050  012700  0000000                 MOV     #INGEMT,R0      ;Get EMT arg block
010054  104375                          EMT     375             ;Get the entry to IIBUF
010056  005767  000000C                 TST     IIBUF+II$NAM    ;Is this entry in use?
010062  001402                          BEQ     2$              ;Br if not
010064  004767  000016                  CALL    INSPRT          ;Print info about this program
010070  062705  0000000         2$:     ADD     #II$$SZ,R5      ;Point to next install table entry
010074  020567  0000000                 CMP     R5,INSTBN       ;Done all entries?
010100  103761                          BLO     1$              ;Loop if not
                                ;
                                ;  Finished
                                ;
010102  012605                  MOV     (SP)+,R5
010104  000207                  RETURN
```

```
                                      ;----------------------------------------------------------------------------
                                      ;  Print information about the program whose install entry is in IIBUF.
                                      ;
                                      ;  Inputs:
                                      ;   IIBUF contains install entry for program.
                                      ;
010106  010146                        INSPRT: MOV     R1,-(SP)
010110  010246                                MOV     R2,-(SP)
010112  010346                                MOV     R3,-(SP)
010114  010446                                MOV     R4,-(SP)
                                      ;
                                      ;  Print the file spec for the program.
                                      ;
010116  012704  000000C                       MOV     #IIBUF+II$NAM,R4;Point to file spec to be converted
010122  012703  000000G                       MOV     #BLKO,R3        ;Point to result area
010126  004767  000000G                       CALL    EDTFIL          ;Convert file spec to asciz form
010132  112723  000040                1$:     MOVB    #' ,(R3)+       ;Store space following file spec
010136  020327  000021G                       CMP     R3,#BLKO+17.    ;Filled out to flag area?
010142  103773                                BLO     1$              ;Loop if not
010144  112713  000200                        MOVB    #200,(R3)       ;Store null at end of name
010150                                        .PRINT  #BLKO           ;Print the name
                                      ;
                                      ;  Print names of attribute flags
                                      ;
010156  012704  177756                        MOV     #-18.,R4        ;No attributes printed yet
010162  012703  010406'                       MOV     #INSANT,R3      ;Point to attribute name table
010166  032367  000000C               2$:     BIT     (R3)+,II$FLG+IIBUF ;Is this attribute set?
010172  001422                                BEQ     4$              ;Br if not
010174  005704                                TST     R4              ;Is this the first attribute for program?
010176  002406                                BLT     8$              ;Br if yes
010200                                        .TTYOUT #'/             ;Print slash
010210  005204                                INC     R4              ;Count column used by slash
010212  000402                                BR      3$
010214  012704  000022                8$:     MOV     #18.,R4         ;Initialize column number
010220  011301                        3$:     MOV     (R3),R1         ;Get address of name string
010222                                        .PRINT  R1              ;Print attribute name
010226  122127  000200                10$:    CMPB    (R1)+,#200      ;End of keyword?
010232  001402                                BEQ     4$              ;Br if yes
010234  005204                                INC     R4              ;Count columns
010236  000773                                BR      10$
010240  005723                        4$:     TST     (R3)+           ;Skip name pointer
010242  005713                                TST     (R3)            ;Any more attributes to check?
010244  001350                                BNE     2$              ;Loop if yes
                                      ;
                                      ;  See if program has any associated privileges
                                      ;
010246  012702  000000C                       MOV     #II$PRV+IIBUF,R2;Point to privilege flag info
010252  012703  000000C                       MOV     #II$NPV+IIBUF,R3;Point to negative privilege flags
010256  012700  000000G                       MOV     #PVNPW,R0       ;Get # privilege words
010262  005722                        12$:    TST     (R2)+           ;Any privilege flags set?
010264  001004                                BNE     13$             ;Br if yes
010266  005723                                TST     (R3)+           ;Any negative privilege flags set?
010270  001002                                BNE     13$             ;Br if yes
010272  077005                                SOB     R0,12$          ;Check all privilege words
010274  000434                                BR      11$             ;No privileges specified for program
                                      ;
                                      ;  Print information about special privileges associated with program
```

```
                                        ;
010276                          13$:    .PRINT  #TM$PVL         ;Print "/PRIV=("
010304  005704                          TST     R4              ;Any attributes printed?
010306  002401                          BLT     14$             ;Br if not
010310  005404                          NEG     R4              ;Make R4 negative to suppress leading comma
010312  162704  000007        14$:    SUB     #7.,R4          ;Increase column count
010316  012702  00000C        15$:    MOV     #II$PRV+IIBUF,R2;Point to words with privilege flags
010322  012703  000001                MOV     #+1,R3          ;Select positive flags this time
010326  012700  000022                MOV     #18.,R0         ;Wrap-around to column 18
010332  004767  000000                CALL    PRVLST          ;Print selected privileges
010336  012702  00000C                MOV     #II$NPV+IIBUF,R2;Point to words with deselected priv flags
010342  012703  177777                MOV     #-1,R3          ;Select negative flags
010346  012700  000022                MOV     #18.,R0         ;Set wrap-around column
010352  004767  000000                CALL    PRVLST          ;Print deselected privileges
010356                                .TTYOUT #')             ;Terminate privilege list
                                        ;
                                        ;  Terminate the print line
                                        ;
010366                          11$:    .PRINT  #CRLF           ;End line
                                        ;
                                        ;  Finished
                                        ;
010374  012604                          MOV     (SP)+,R4
010376  012603                          MOV     (SP)+,R3
010400  012602                          MOV     (SP)+,R2
010402  012601                          MOV     (SP)+,R1
010404  000207                          RETURN
                                        ;
                                        ;  Table of install program attributes and names
                                        ;
010406  000000  010510'         INSANT: .WORD   AF$SCA,1$
010412  000000  010523'                 .WORD   AF$NOW,2$
010416  000000  010532'                 .WORD   AF$HIE,3$
010422  000000  010537'                 .WORD   AF$NOI,4$
010426  000000  010556'                 .WORD   AF$IOP,5$
010432  000000  010565'                 .WORD   AF$MEM,6$
010436  000000  010575'                 .WORD   AF$PLK,7$
010442  000000  010602'                 .WORD   AF$DBG,8$
010446  000000  010610'                 .WORD   AF$BYA,9$
010452  000000  010617'                 .WORD   AF$TPO,10$
010456  000000  010633'                 .WORD   AF$DUP,11$
010462  000000  010637'                 .WORD   AF$IND,12$
010466  000000  010643'                 .WORD   AF$UCL,13$
010472  000000  010652'                 .WORD   AF$SET,14$
010476  000000  010660'                 .WORD   AF$CCA,15$
010502  000000  010665'                 .WORD   AF$NPW,16$
010506  000000                          .WORD   0
                                        ;
010510      123     111     116 1$:     .ASCII  /SINGLECHAR/<200>
010513      107     114     105
010516      103     110     101
010521      122     200
010523      116     117     127 2$:     .ASCII  /NOWAIT/<200>
010526      101     111     124
010531      200
010532      110     111     107 3$:     .ASCII  /HIGH/<200>
010535      110     200
```

```
    109 010537      116     117     116  4$:     .ASCII  /NONINTERACTIVE/<200>
        010542      111     116     124
        010545      105     122     101
        010550      103     124     111
        010553      126     105     200
    110 010556      111     117     120  5$:     .ASCII  /IOPAGE/<200>
        010561      101     107     105
        010564      200
    111 010565      115     105     115  6$:     .ASCII  /MEMLOCK/<200>
        010570      114     117     103
        010573      113     200
    112 010575      114     117     103  7$:     .ASCII  /LOCK/<200>
        010600      113     200
    113 010602      104     105     102  8$:     .ASCII  /DEBUG/<200>
        010605      125     107     200
    114 010610      102     131     120  9$:     .ASCII  /BYPASN/<200>
        010613      101     123     116
        010616      200
    115 010617      124     122     101  10$:    .ASCII  /TRANSPARENT/<200>
        010622      116     123     120
        010625      101     122     105
        010630      116     124     200
    116 010633      104     125     120  11$:    .ASCII  /DUP/<200>
        010636      200
    117 010637      111     116     104  12$:    .ASCII  /IND/<200>
        010642      200
    118 010643      124     123     130  13$:    .ASCII  /TSXUCL/<200>
        010646      125     103     114
        010651      200
    119 010652      123     105     124  14$:    .ASCII  /SETUP/<200>
        010655      125     120     200
    120 010660      123     103     103  15$:    .ASCII  /SCCA/<200>
        010663      101     200
    121 010665      116     117     127  16$:    .ASCII  /NOWINDOW/<200>
        010670      111     116     104
        010673      117     127     200
    122                                  .EVEN
```

```
                                                .SBTTL          REGIONS
                                        ;-----------------------------------------------------------------
                                        ;  Show named regions
                                        ;
 010676  010246                         SHOREG: MOV     R2,-(SP)
 010700  010446                                 MOV     R4,-(SP)
 010702  005002                                 CLR     R2              ;Clear count of # regions shown
                                        ;
                                        ;  First show information about all private regions
                                        ;
 010704  012704  000000G                        MOV     #RCBBAS,R4      ;Point to first private Region Control Blk
 010710  004767  000106                 1$:     CALL    RGNDSP          ;Display info about this region
 010714  062704  000000G                        ADD     #RC$$SZ,R4      ;Point to next RCB
 010720  020427  000000G                        CMP     R4,#RCBEND      ;Checked all RCB's?
 010724  103771                                 BLO     1$              ;Br if not
                                        ;
                                        ;  Now show information about all shared regions
                                        ;
 010726  016704  000000G                        MOV     SHRRCB,R4       ;Point to 1st shared RCB
 010732  020467  000000G                2$:     CMP     R4,SHRRCN       ;Checked all shared RCB's?
 010736  103021                                 BHIS    3$              ;Br if yes
 010740  010467  000000G                        MOV     R4,PEKADR       ;Set address of RCB
 010744  012767  000000G 000000G                MOV     #RC$$SZ,PEKSIZ  ;Get # bytes to fetch
 010752  012700  000000G                        MOV     #PEKEMT,R0      ;Point to EMT arg block
 010756  104375                                 EMT     375             ;Move RCB to BLKO buffer
 010760  012704  000000G                        MOV     #BLKO,R4        ;Point to RCB in our buffer
 010764  004767  000032                         CALL    RGNDSP          ;Display info about the region
 010770  016704  000000G                        MOV     PEKADR,R4       ;Recover real RCB address
 010774  062704  000000G                        ADD     #RC$$SZ,R4      ;Point to next RCB
 011000  000754                                 BR      2$
                                        ;
                                        ;  If there were no named regions, print a message.
                                        ;
 011002  005702                         3$:     TST     R2              ;Were there any named regions?
 011004  001003                                 BNE     9$              ;Br if yes
 011006                                         .PRINT  #TM$NNR         ;No named regions
                                        ;
                                        ;  Finished
                                        ;
 011014  012604                         9$:     MOV     (SP)+,R4
 011016  012602                                 MOV     (SP)+,R2
 011020  000207                                 RETURN
```

```
                                ;-------------------------------------------------------------------------------
                                ;  Display a line of information about a named region.
                                ;
                                ;  Inputs:
                                ;   R4 = Pointer to Region Control Block.
                                ;   R2 = Count of number of active regions so far.
                                ;
                                ;  Outputs:
                                ;   R2 = Incremented if this is an active region.
                                ;
011022                          RGNDSP:
                                ;
                                ;  See if this RCB is for an active, named region
                                ;
011022  032764  0000000 0000000         BIT     #RC$USE,RC$FLG(R4) ;Is this an active RCB?
011030  001410                          BEQ     1$                 ;Br if not
011032  032764  0000000 0000000         BIT     #RC$LCG,RC$FLG(R4) ;Is this a local copy of a global RCB?
011040  001004                          BNE     1$                 ;Br if yes -- We will list global RCB
011042  032764  0000000 0000000         BIT     #RC$GBL,RC$FLG(R4) ;Is this a named region?
011050  001001                          BNE     2$                 ;Br if yes
011052  000207                  1$:     RETURN
                                ;
                                ;  This is an RCB for an active named region.
                                ;
011054  010146                  2$:     MOV     R1,-(SP)
011056  010346                          MOV     R3,-(SP)
011060  010546                          MOV     R5,-(SP)
                                ;
                                ;  See if this is the 1st region
                                ;
011062  005702                          TST     R2                 ;Is this the first region?
011064  001006                          BNE     3$                 ;Br if not
011066                                  .PRINT  #TM$RD1            ;Display title line 1
011074                                  .PRINT  #TM$RD2            ;Display title line 2
011102  005202                  3$:     INC     R2                 ;Count another region displayed
                                ;
                                ;  Display name of the region
                                ;
011104  016400  0000000                 MOV     RC$NAM(R4),R0      ;Get 1st 3 chars of name
011110  004767  0000000                 CALL    PRTR50             ;Print them
011114  016400  0000020                 MOV     RC$NAM+2(R4),R0    ;Get 2nd 3 chars of name
011120  004767  0000000                 CALL    PRTR50             ;Print them
                                ;
                                ;  Display size of region
                                ;
011124  016401  0000000                 MOV     RC$LEN(R4),R1      ;Get # 64-byte blocks allocated for region
011130  005000                          CLR     R0                 ;Clear high-order for divide
011132  071027  000020                  DIV     #16.,R0            ;Conver to # Kb
011136  010005                          MOV     R0,R5              ;Get # whole K
011140  012703  000006                  MOV     #6.,R3             ;Print in 6 digit field
011144  004767  0000000                 CALL    PRTFIX             ;Print # whole Kb
011150  116105  011374'                 MOVB    FRAC64(R1),R5      ;Convert to decimal Kb fraction
011154                                  .TTYOUT #'.                ;Print decimal point
011164  004767  0000000                 CALL    PRTDEC             ;Print decimal digit
                                ;
                                ;  Display type of region
                                ;
```

```
     58 011170                                          .PRINT  #SPACE2          ;Print 2 spaces
     59 011176  032764  0000000 0000000                BIT     #RC$PVT,RC$FLG(R4) ;Is this a private or shared region?
     60 011204  001407                                  BEQ     4$               ;Br if shared
     61 011206                                          .PRINT  #TM$LCL          ;Say region is local to job
     62 011214                                          .PRINT  #SPACE1          ;Print extra space
     63 011222  000403                                  BR      5$
     64 011224                                  4$:     .PRINT  #TM$GRL          ;Say region is global
     65                                                 ;
     66                                                 ;  Print number of job that created region
     67                                                 ;
     68 011232  116405  0000000                 5$:     MOVB    RC$OWN(R4),R5    ;Get # of job that created region
     69 011236  006205                                  ASR     R5               ;Convert to #
     70 011240  012703  000005                          MOV     #5.,R3           ;Print 5 digit field
     71 011244  004767  0000000                         CALL    PRTFIX           ;Print job #
     72                                                 ;
     73                                                 ;  Print attachment (use) count
     74                                                 ;
     75 011250  116405  0000000                         MOVB    RC$CNT(R4),R5    ;Get attachment count
     76 011254  012703  000005                          MOV     #5.,R3           ;Print in 5 digit field
     77 011260  004767  0000000                         CALL    PRTFIX           ;Print use count
     78                                                 ;
     79                                                 ;  Show if shared
     80                                                 ;
     81 011264  012703  000004                          MOV     #4.,R3           ;Print 4 spaces
     82 011270  004767  0000000                         CALL    PRTSP0
     83 011274  032764  0000000 0000000                 BIT     #RC$EXC,RC$FLG(R4) ;Is this region sharable?
     84 011302  001404                                  BEQ     6$               ;Br if yes
     85 011304                                          .PRINT  #NOTXT           ;Print 'No'
     86 011312  000403                                  BR      7$
     87 011314                                  6$:     .PRINT  #YESTXT          ;Print 'Yes'
     88                                                 ;
     89                                                 ;  Show if AGE is set
     90                                                 ;
     91 011322                                  7$:     .PRINT  #SPACE3          ;Print 3 spaces
     92 011330  032764  0000000 0000000                 BIT     #RC$AGE!RC$AEP,RC$FLG(R4) ;Is AGE enabled?
     93 011336  001004                                  BNE     8$               ;Br if yes
     94 011340                                          .PRINT  #NOTXT           ;Print 'No'
     95 011346  000403                                  BR      9$
     96 011350                                  8$:     .PRINT  #YESTXT          ;Print 'Yes'
     97 011356                                  9$:
     98                                                 ;
     99                                                 ;  Terminate print line
    100                                                 ;
    101 011356                                          .PRINT  #CRLF
    102                                                 ;
    103                                                 ;  Finished
    104                                                 ;
    105 011364  012605                                  MOV     (SP)+,R5
    106 011366  012603                                  MOV     (SP)+,R3
    107 011370  012601                                  MOV     (SP)+,R1
    108 011372  000207                                  RETURN
    109                                                 ;
    110                                                 ;  Values to convert # fractional 64-byte blocks to tenths of Kb
    111                                                 ;
    112 011374      000       001       001   FRAC64:  .BYTE   0.,1.,1.,2.,2.,3.,4.,4.,5.,6.,6.,7.,7.,8.,9.,9.
        011377      002       002       003
        011402      004       004       005
```

```
.       REGIONS

        011405      006     006     007
        011410      007     010     011
        011413      011
    113                                         .EVEN
```

```
                                        .SBTTL  .       PRIVILEGES
                                ;-----------------------------------------------------------------------
                                ;  Show privileges
                                ;
011414  010246                  SHOPRV: MOV     R2,-(SP)
011416  010346                          MOV     R3,-(SP)
011420  010446                          MOV     R4,-(SP)
                                ;
                                ;  List authorized privileges
                                ;
011422                                  .PRINT  #TM$PVA         ;Authorized privileges
011430  012702  0000000                 MOV     #PRIVAO,R2      ;Point to priv flag vector
011434  012703  000001                  MOV     #+1,R3          ;Show positive privileges only
011440  012704  177750                  MOV     #-24.,R4        ;Start at column 24
011444  012700  000030                  MOV     #24.,R0         ;Wrap around to col 24
011450  004767  0000000                 CALL    PRVLST          ;List privileges
011454                                  .PRINT  #CRLF           ;Terminate last line
011462                                  .PRINT  #CRLF           ;Put in a blank line
                                ;
                                ;  List current privileges
                                ;
011470                                  .PRINT  #TM$PVC         ;Current privileges
011476  012702  0000000                 MOV     #PRIVCO,R2      ;Current privilege flags
011502  012704  177750                  MOV     #-24.,R4        ;Start at column 24
011506  012700  000030                  MOV     #24.,R0         ;Wrap around to col 24
011512  004767  0000000                 CALL    PRVLST          ;List privileges
011516                                  .PRINT  #CRLF
                                ;
                                ;  Finished
                                ;
011524  012604                          MOV     (SP)+,R4
011526  012603                          MOV     (SP)+,R3
011530  012602                          MOV     (SP)+,R2
011532  000207                          RETURN
```

```
                                        .SBTTL  .       SL
                                ;-----------------------------------------------------------------------------
                                ;  Display single line editor status.
                                ;
011534  010246                  SHOSLE: MOV     R2,-(SP)
011536  010346                          MOV     R3,-(SP)
011540  010446                          MOV     R4,-(SP)
                                ;
                                ;  Print message heading
                                ;
011542                                  .PRINT  #TM$SL1         ;"SL status:  "
                                ;
                                ;  Print ON or OFF
                                ;
011550  032761  0000000 0000000         BIT     #$SLON,LSW7(R1) ;Is SL turned on?
011556  001003                          BNE     1$              ;Br if yes
011560  012702  0000000                 MOV     #TM$OFF,R2      ;Point to OFF message
011564  000402                          BR      2$
011566  012702  0000000          1$:    MOV     #TM$ON,R2       ;Point to ON message
011572                           2$:    .PRINT  R2              ;Print text
                                ;
                                ;  Check for KED mode
                                ;
011576  012704  0000000                 MOV     #$SLKED,R4      ;Get Ked flag
011602  012703  0000000                 MOV     #TM$KED,R3
011606  004767  000046                  CALL    SLCKFL
                                ;
                                ;  Check for TTY mode
                                ;
011612  012704  0000000                 MOV     #$SLTTY,R4      ;Get flag
011616  012703  0000000                 MOV     #TM$TTY,R3      ;Point to text string
011622  004767  000032                  CALL    SLCKFL
                                ;
                                ;  Check for SUBSTITUTE mode
                                ;
011626  012704  0000000                 MOV     #$SLLET,R4      ;Get flag
011632  012703  0000000                 MOV     #TM$SUB,R3      ;Point to text string
011636  004767  000016                  CALL    SLCKFL
                                ;
                                ;  Terminate the print line
                                ;
011642                                  .PRINT  #CRLF           ;Terminate print line
                                ;
                                ;  Finished
                                ;
011650  012604                          MOV     (SP)+,R4
011652  012603                          MOV     (SP)+,R3
011654  012602                          MOV     (SP)+,R2
011656  000207                          RETURN
```

```
                                        ;------------------------------------------------------------------------
                                        ;  Check of a SL option flag is set in LSW7, and if it is print that the
                                        ;  option is on; otherwise, print that the option is off.
                                        ;
                                        ;  Inputs:
                                        ;   R1 = Job index number
                                        ;   R3 = Pointer to text string for option name
                                        ;   R4 = Flag bit to be tested in LSW7
                                        ;
011660                                  SLCKFL:
                                        ;
                                        ;  Print a leading comma
                                        ;
011660                                          .TTYOUT #54             ;Print comma
                                        ;
                                        ;  If option flag is not set, print "NO"
                                        ;
011670  030461  0000000                         BIT     R4,LSW7(R1)     ;Is the option flag set?
011674  001003                                  BNE     1$              ;Br if yes
011676                                          .PRINT  #TM$NO          ;Print NO
                                        ;
                                        ;  Print the name of the option
                                        ;
011704                                  1$:     .PRINT  R3              ;Print the option name
                                        ;
                                        ;  Finished
                                        ;
011710  000207                                  RETURN
```

```
                                        .SBTTL  .       RUN-TIMES
                                ;-----------------------------------------------------------------------------
                                ;  Show run-times
                                ;
011712  010146                  SHOSRT: MOV     R1,-(SP)
011714  012701  0000000                 MOV     #RDB,R1         ;POINT TO FIRST RUN-TIME DESCRIPTOR BLOCK
011720  005005                          CLR     R5              ;Count run-times in R5
011722  020127  0000000         1$:     CMP     R1,#RDBEND      ;ANY RUN-TIMES?
011726  103023                          BHIS    2$              ;BR IF FINISHED
011730  026127  0000000 0000000         CMP     RT$DEV(R1),#DMYDEV ;Is this dummy entry for patching?
011736  001414                          BEQ     4$              ;Br if yes
011740  016100  0000000                 MOV     RT$NAM(R1),R0   ;GET 1ST 3 CHARS OF RUN-TIME NAME
011744  005200                          INC     R5              ;Count another run-time
011746  004767  0000000                 CALL    PRTR50          ;PRINT THEM
011752  016100  0000020                 MOV     RT$NAM+2(R1),R0 ;GET 2ND 3 CHARS OF NAME
011756  004767  0000000                 CALL    PRTR50          ;PRINT THEM
011762                                  .PRINT  #CRLF           ;END LINE
011770  062701  0000000         4$:     ADD     #RT$$SZ,R1      ;POINT TO NEXT DESCRIPTOR BLOCK
011774  000752                          BR      1$
011776  005705                  2$:     TST     R5              ;WERE THERE ANY RUN-TIMES?
012000  001003                          BNE     3$              ;BR IF YES
012002                                  .PRINT  #NONEMS         ;PRINT NONE
012010  012601                  3$:     MOV     (SP)+,R1
012012  000207                          RETURN
```

```
                                .SBTTL  .       SPOOL
                        ;-----------------------------------------------------------------------
                        ;  SHOW SPOOL
                        ;  List the names of the spooled devices.
                        ;
012014  010246          SHOSPL: MOV     R2,-(SP)
                        ;
                        ;  See if there are any spooled devices
                        ;
012016  105767  0000000         TSTB    NSPLDV          ;Are there any spooled devices?
012022  001405                  BEQ     3$              ;Br if not
012024  012702  0000000         MOV     #SDCB,R2        ;Point to first spooled device control block
012030  020227  0000000         CMP     R2,#SDCBND      ;Are there any spooled devices?
012034  103404                  BLO     1$              ;Br if there are spooled devices
012036                  3$:     .PRINT  #TM$NSD         ;There are no spooled devices
012044  000425                  BR      9$
                        ;
                        ;  There are spooled devices, print their names
                        ;
012046                  1$:     .PRINT  #TM$SDN         ;Print heading
012054  016200  0000000 2$:     MOV     SDNAME(R2),R0   ;Get RAD50 name of spooled device
012060  020027  0000000         CMP     R0,#DMYDEV      ;Uninstalled device?
012064  001405                  BEQ     4$              ;Br if yes
012066  004767  0000000         CALL    PRTR50          ;Print the name
012072                  .PRINT  #SPACE2         ;Print 2 spaces
012100  062702  0000000 4$:     ADD     #SDCBSZ,R2      ;Point to next SDCB
012104  020227  0000000         CMP     R2,#SDCBND      ;Are there more?
012110  103761                  BLO     2$              ;Br if yes
012112                  .PRINT  #CRLF           ;Terminate the line
                        ;
                        ;  Finished
                        ;
012120  012602  9$:     MOV     (SP)+,R2
012122  000207                  RETURN
```

```
                                        .SBTTL  .       SUBSET
                                ;-------------------------------------------------------------------------------
                                ;  SHOW SUBSET
                                ;  Display information about logical disks.
                                ;
012124  010146                  SHOSUB: MOV     R1,-(SP)
012126  010246                          MOV     R2,-(SP)
012130  010346                          MOV     R3,-(SP)
012132  010446                          MOV     R4,-(SP)
012134  010546                          MOV     R5,-(SP)
                                ;
                                ;  First do a SET LD CLEAN to update logical disk information
                                ;
012136  004767  0000000                 CALL    LDCLEN          ;DO SET LD CLEAN
                                ;
                                ;  Make a fast scan and see if any logical disks are mounted
                                ;
012142  012703  000010                  MOV     #8.,R3          ;GET # LOGICAL DISK ENTRIES
012146  012704  0000000                 MOV     #LDNAME,R4      ;POINT TO FILE NAME TABLE
012152  005714                  5$:     TST     (R4)            ;IS THIS DISK ASSIGNED TO A FILE?
012154  001007                          BNE     6$              ;BR IF YES
012156  062704  000010                  ADD     #8.,R4          ;POINT TO NEXT ENTRY IN TABLE
012162  077305                          SOB     R3,5$           ;LOOP IF MORE TO CHECK
012164                                  .PRINT  #NOLDMT         ;NO LOGICAL DISKS MOUNTED
012172  000472                          BR      9$
                                ;
                                ;  Now begin to display logical disk information
                                ;
012174  016702  0000000         6$:     MOV     R50LDO,R2       ;GET "LDO" NAME
012200  005003                          CLR     R3              ;INIT DISK TABLE INDEX
012202  012704  0000000                 MOV     #LDNAME,R4      ;POINT TO FILE NAME TABLE
                                ;
                                ;  See if next logical disk unit is assigned
                                ;
012206  005714                  2$:     TST     (R4)            ;IS DISK ASSIGNED TO A FILE?
012210  001453                          BEQ     1$              ;BR IF NOT
                                ;
                                ;  Print logical disk name
                                ;
012212  010200                          MOV     R2,R0           ;GET DISK NAME
012214  004767  0000000                 CALL    PRTR50          ;PRINT THE DISK NAME
012220                                  .PRINT  #SUBARO         ;" ---> "
                                ;
                                ;  Print the file name
                                ;
012226  010346                          MOV     R3,-(SP)        ;SAVE R3
012230  012703  0000000                 MOV     #BLKO,R3        ;EDIT FILE NAME INTO BLKO
012234  004767  0000000                 CALL    EDTFIL          ;EDIT FILE NAME
012240  112723  000133                  MOVB    #133,(R3)+      ;"[" START OF FILE SIZE
012244  112713  000200                  MOVB    #200,(R3)       ;TERMINATE NAME STRING
012250                                  .PRINT  #BLKO           ;PRINT NAME
012256  012603                          MOV     (SP)+,R3
012260  016305  0000000                 MOV     LDSIZE(R3),R5   ;GET FILE SIZE
012264  004767  0000000                 CALL    PRTDEC          ;PRINT IT
012270                                  .TTYOUT #135            ;"]"
                                ;
                                ;  Print "read only" if that is the case
```

```
                                ;
012300  032763  0000000 0000000         BIT     #LD$RON,LDFLAG(R3) ;IS IS MOUNTED READ-ONLY
012306  001403                          BEQ     3$                 ;BR IF NOT
012310                                  .PRINT  #RONTXT            ;PRINT "(read only)"
                                ;
                                ;  Print "not available" if that is the case
                                ;
012316  005763  0000000         3$:     TST     LDPDEV(R3)         ;IS FILE CURRENTLY ACTIVE?
012322  001003                          BNE     4$                 ;BR IF YES
012324                                  .PRINT  #NOTAVL            ;NOT AVAILABLE
012332                          4$:     .PRINT  #CRLF              ;PRINT THE LINE
                                ;
                                ;  Advance to next logical disk
                                ;
012340  005207                  1$:     INC     R2                 ;ADVANCE LOGICAL DISK NAME
012342  062704  000010                  ADD     #8.,R4             ;ADVANCE NAME POINTER
012346  062703  000002                  ADD     #2,R3              ;ADVANCE TABLE POINTER
012352  020327  000016                  CMP     R3,#14.            ;DONE ALL?
012356  101713                          BLOS    2$                 ;BR IF NOT
                                ;
                                ;  Finished
                                ;
012360  012605                  9$:     MOV     (SP)+,R5
012362  012604                          MOV     (SP)+,R4
012364  012603                          MOV     (SP)+,R3
012366  012602                          MOV     (SP)+,R2
012370  012601                          MOV     (SP)+,R1
012372  000207                          RETURN
```

```
                                        .SBTTL  VM
                                ;-----------------------------------------------------------
                                ; SHOW CURRENT VM BASE, TOP AND DEVICE SIZE
                                ;
012374  012767  105646  0000000 SHOWVM: MOV     #^RVMO,FILNAM   ;Set device name VM:
012402  005067  0000020         CLR     FILNAM+2        ;No file name
012406                          .SERR                   ;Trap .LOOKUP errors
012414                          .LOOKUP #XAREA,#1,#FILNAM ;Get channel to VMO:
012434  103004                  BCC     1$              ;Branch if we got VM
012436                          .PRINT  #SHVTX4         ;"VM not installed"
012444  000476                  BR      9$              ;Exit if no VM
                                ;
                                ;  Get current base and top
                                ;
012446                  1$:     .SPFUN  #XAREA,#1,#372,#BLKO,#0,#0 ;Request VM base and top
012510  103451                  BCS     8$              ;Ignore command on error
012512                          .PRINT  #SHVTX1         ;" VM Base="
012520  016705  0000000         MOV     BLKO,R5         ;Retrieve base
012524  012703  000006          MOV     #6,R3           ;Six digit display
012530  004767  0000000         CALL    OCTFIX          ;Display it
012534                          .PRINT  #SHVTX2         ;" Top="
012542  016705  0000020         MOV     BLKO+2,R5       ;Retrieve top
012546  004767  0000000         CALL    OCTFIX          ;Display it
                                ;
                                ;  Get current device size in blocks from handler
                                ;
012552                          .SPFUN  #XAREA,#1,#373,#BLKO,#0,#0 ;Request VM device size
012614  103407                  BCS     8$              ;Skip size if bad
012616                          .PRINT  #SHVTX3         ;" Size="
012624  016705  0000000         MOV     BLKO,R5         ;Retrieve size
012630  004767  0000000         CALL    PRTDEC          ;Display decimal size in blocks
                                ;
012634                  8$:     .CLOSE  #1              ;Only close if successful open
012642                  9$:     .PRINT  #CRLF           ;Format display
012650                          .HERR                   ;Give back error trapping
012656  000207                  RETURN
```

```
                                        .SBTTL  SYSTAT (& WHO) command
                                ;-------------------------------------------------------
                                ;  THE WHO COMMAND PRINTS OUT A LIST OF ALL LINE NUMBERS
                                ;  WHICH ARE LOGGED ON.
                                ;
 012660  004767  171004         CMDWHO: CALL    SHOJOB          ;SHOW JOB INFORMATION
 012664  000167  000000G                 JMP     RDCMD

                                        .SBTTL  USE command
                                ;-------------------------------------------------------------------------------
                                ;  Process the USE command.
                                ;
 012670  004767  000000G        CMDUSE: CALL    PRTTIM          ;PRINT CONNECT AND CPU TIME FOR JOB
 012674  000167  000000G                 JMP     RDCMD
```

```
                                        .SBTTL  PRTUSE -- Print system usage statistics
                                ;-----------------------------------------------------------------------
                                ;  PRTUSE is called to print the system usage statistics.
                                ;
012700  010146                  PRTUSE: MOV     R1,-(SP)
012702  010246                          MOV     R2,-(SP)
012704  010346                          MOV     R3,-(SP)
012706  010446                          MOV     R4,-(SP)
                                ;  Print system up-time.
012710                                  .PRINT  #UPTMMS         ;"UPTIME: "
012716  016704  0000000               MOV     TMTOTH,R4       ;GET TOTAL UP-TIME (OR TIME SINCE LAST RESET)
012722  016705  0000000               MOV     TMTOTL,R5
012726  000241                          CLC                     ;DIVIDE TIME VALUE BY 2 TO GET 1/10 SEC UNITS
012730  006004                          ROR     R4              ;SHIFT HIGH-ORDER PART
012732  006005                          ROR     R5              ;AND LOW-ORDER PART
012734  004767  0000000               CALL    PRTTMD          ;PRINT TIME VALUE
012740                                  .PRINT  #CRLF           ;TERMINATE LINE
                                ;  Print system usage statistics
012746  016767  0000000 0000000       MOV     TMTOTH,DIVSOR   ;SET TOTAL UP-TIME AS DIVISOR FOR PERCENTAGE
012754  016767  0000000 0000020       MOV     TMTOTL,DIVSOR+2
012762  012702  013020'               MOV     #SUMVEC,R2      ;POINT TO DRIVER VECTOR TABLE
012766  012200                  1$:     MOV     (R2)+,R0        ;GET ADDRESS OF ASCIZ STRING TO PRINT
012770  001406                          BEQ     2$              ;BR IF END OF LIST HIT
012772                                  .PRINT                  ;PRINT TEXT MESSAGE
012774  012201                          MOV     (R2)+,R1        ;GET ADDRESS OF TIME CELL TO PRINT
012776  001773                          BEQ     1$              ;BR IF NONE WITH THIS TEXT
013000  004767  0000000               CALL    PRTPCT          ;CONVERT TO PERCENTAGE AND PRINT VALUE
013004  000770                          BR      1$
                                ;
                                ;  Finished
                                ;
013006  012604                  2$:     MOV     (SP)+,R4
013010  012603                          MOV     (SP)+,R3
013012  012602                          MOV     (SP)+,R2
013014  012601                          MOV     (SP)+,R1
013016  000207                          RETURN
                                ;
                                ;  Driver vector for system usage printout.
                                ;  First entry of each pair is the address of an asciz string to print.
                                ;  Second entry is address of 32-bit time value to be printed as percentage.
                                ;
013020  0000000 0000000         SUMVEC: .WORD   SUM1,TMUSRH     ;RUN-TIME
013024  0000000 0000000                 .WORD   SUM2,TMIOWH     ;I/O WAIT
013030  0000000 0000000                 .WORD   SUM3,TMSWTH     ;SWAP WAIT
013034  0000000 0000000                 .WORD   SUM4,TMIDLH     ;IDLE TIME
013040  0000000 000000                  .WORD   SUM5,0          ;END OF LINE
013044  0000000 0000000                 .WORD   SUM6,TMIOH      ;USER I/O TIME
013050  0000000 0000000                 .WORD   SUM7,TMSWPH     ;SWAP TIME
013054  0000000 000000                  .WORD   SUM5,0          ;END OF LINE
013060  0000000 000000                  .WORD   SUM5,0
013064  000000                          .WORD   0               ;END OF LIST
```

```
                                        .SBTTL  MEMORY command
                                ;-----------------------------------------------------------------------
                                ;  The MEMORY command is used to set or display the maximum memory limit
                                ;  for the current job.
                                ;
013066  004767  0000000G        CMDMEM: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
013072  111300                          MOVB    (R3),R0         ;WAS A MEMORY LIMIT SPECIFIED WITH COMMAND?
013074  001004                          BNE     SETMEM          ;BR IF YES
                                ;
                                ;  Display current memory values
                                ;
013076  004767  000124                  CALL    DSPMEM          ;DISPLAY JOB MEMORY LIMITS
013102  000167  0000000G                JMP     RDCMD           ;GO GET NEXT COMMAND
                                ;
                                ;  Set a new memory limit for the job.
                                ;
013106  105767  0000000G        SETMEM: TSTB    VSWPFL          ;IS SWAPPING ALLOWED?
013112  001005                          BNE     4$              ;BR IF YES
013114                                  .PRINT  #NSWPMS         ;CAN'T CHANGE MEMORY SIZE OF NON-SWAP SYSTEM
013122  000167  0000000G                JMP     RDCMD
013126  004767  0000000G        4$:     CALL    ACRDEC          ;ACCRUE THE VALUE
013132  120027  000113                  CMPB    R0,#'K          ;DID HE SPECIFY K-SOMETHING?
013136  001001                          BNE     1$              ;BR IF NOT
013140  005203                          INC     R3              ;SKIP "K"
013142  121327  000127          1$:     CMPB    (R3),#'W        ;WAS IT "KW"?
013146  001001                          BNE     2$              ;BR IF NOT
013150  006301                          ASL     R1              ;DOUBLE MEMORY VALUE
                                ;  Compare request with max limit.
013152  020167  0000000G        2$:     CMP     R1,MXJMEM       ;IS REQUEST LARGET THAN MAX ALLOWED?
013156  101413                          BLOS    3$              ;BR IF NOT
013160  016701  0000000G                MOV     MXJMEM,R1       ;SET TO MAX ALLOWED
013164  010105                          MOV     R1,R5
013166                                  .PRINT  #MAXMTX         ;DISPLAY MAX ALLOWED
013174  004767  0000000G                CALL    PRTDEC
013200                                  .PRINT  #KBTX
013206  072127  000012          3$:     ASH     #10.,R1         ;CONVERT # KB TO ADDRESS
013212  001002                          BNE     5$              ;BR IF DIDN'T OVERFLOW 64KB
013214  012701  177774                  MOV     #177774,R1      ;SET TO 64KB
013220  010167  0000000G        5$:     MOV     R1,MAXMEM       ;SET AS MAX ADDRESS FOR JOB
013224                                  .EXIT                   ;EXIT TO ACTUALLY DO THE MEMORY SIZE CHANGE

                                ;-----------------------------------------------------------------------
                                ;  Display information about job memory limits.
                                ;
013226  010546                  DSPMEM: MOV     R5,-(SP)
013230                                  .PRINT  #CURMTX         ;CURRENT MEMORY =
013236  016705  0000000G                MOV     MAXMEM,R5       ;GET CURRENT HIGH-MEMORY LIMIT FOR JOB
013242  020527  177770                  CMP     R5,#177770      ;64KB?
013246  103403                          BLO     1$              ;BR IF NOT
013250  012705  000100                  MOV     #64.,R5         ;DO THIS TO AVOID OVERFLOW IN CONVERSION
013254  000403                          BR      2$
013256  000305                  1$:     SWAB    R5              ;CONVERT TO # KB
013260  072527  177776                  ASH     #-2,R5
013264  004767  0000000G        2$:     CALL    PRTDEC          ;DISPLAY THE VALUE
013270                                  .PRINT  #KBTX           ;PRINT "KB"
013276                                  .PRINT  #MAXMTX         ;MAX MEMORY =
013304  016705  0000000G                MOV     MXJMEM,R5       ;MAX SIZE ALLOWED
```

```
     58 013310  004767  0000000                  CALL    PRTDEC          ;DISPLAY THE VALUE
     59 013314                                   .PRINT  #KBTX           ;"KB"
     60 013322  012600                           MOV     (SP)+,R5
     61 013324  000207                           RETURN
```

```
                                  ;------------------------------------------------------------------------
                                  ;  PRTKB is called to convert a value from # of 256-word memory pages
                                  ;  to # k-bytes and print the value followed by "Kb<cr><lf>"
                                  ;
013326  010546                    PRTKB:  MOV     R5,-(SP)
013330  006205                            ASR     R5              ;CONVERT # PAGES TO # KB
013332  004767  000000                    CALL    PRTDEC          ;PRINT THE VALUE
013336                                    .PRINT  #KBTX           ;PRINT "KB<CR><LF>"
013344  012605                            MOV     (SP)+,R5
013346  000207                            RETURN
        000001                            .END
```

Errors detected: 0

*** Assembler statistics

Work file reads: 0
Work file writes: 0
Size of work file: 11964 Words (47 Pages)
Size of core pool: 17920 Words (70 Pages)
Operating system: RT-11

Elapsed time: 00:01:23.05
DK:TSKSHO,LP:TSKSHO=DK:TSKSHO.MAC/C/N:SYM

Cross reference table (CREF V05.04)

```
$1STLC    1-27
$BRIT     1-105
$AUTO     1-91        9-112
$CARMN    1-198       9-60
$CARUP    1-79
$CCLRN    1-80
$CFABT    1-100
$CFALL    1-106
$CFCCL    1-106
$CFDCC    1-106
$CFOPN    1-112
$CFSOT    1-104
$CHACT    1-55
$CLTST    1-90
$CTRLC    1-98
$CTRLD    1-149
$CTRLO    1-55
$CTRLS    1-85
$DBKMN    1-78
$DEAD     1-153       9-70        9-157       9-182
$DEBUG    1-150
$DEFER    1-118
$DETCH    1-83
$DIBOL    1-67
$DILUP    1-102       9-116
$DISCN    1-84
$DOOFF    1-108
$DUPRN    1-103
$ECHO     1-105
$EMTTR    1-89
$FORM     1-104
$FORMO    1-106
$HARD     1-153
$HITTY    1-66
$INCOR    1-122       11-118
$INDAB    1-154
$INDDF    1-152
$INDRN    1-152
$INIT     1-153       9-58        9-163
$INKMN    1-98
$KFD      1-122
$KINIT    1-62        9-196       11-45
$LC       1-105
$LOFCF    1-198
$MLOCK    1-71        11-128
$NOIN     1-66
$NOINT    1-199
$NOWTT    1-66
$PAGE     1-105
$PHONE    1-153       9-66
$PRGLK    1-81
$QTSET    1-127
$QUIET    1-119
$RNIOP    1-200
$SCOPE    1-105
$SCALL    1-118
```

Cross reference table (CREF V05.04)

Cross reference table (CREF V05.04)

Cross reference table (CREF V05.04)

Cross reference table (CREF V05.04)

```
CFSEND    1-112
CFSP      1-112
CFSPND    1-125
CFSTK     1-62
CFSTS     1-154
CHAIN     1-98
CHKALC    1-105
CHKDEV    1-191
CHKDLM    1-181
CHKEQ     1-29
CHKMNT    1-166
CHKMTX    1-166
CINDAT    1-136
CINFLG    1-42
CKACOJ    1-39
CKCLUS    1-33      10-65
CKPRIV    1-168
CKSYPV    1-37       7-133     22-6
CL$COL    1-60
CL$LEN    1-95
CL$LIN    1-95
CL$LIX    1-56      10-50      10-99
CL$OPT    1-94      10-101
CL$RQH    1-57
CL$SKP    1-95
CL$WID    1-95
CL$WQH    1-57
CLDEVX    1-134     10-79
CLFREE    1-51       9-200
CLLINE    1-51       9-52
CLOPND   10-120     10-166#
CLOPTB   10-102     10-153#
CLOTIR    1-59
CLRPRV    1-36
CLSFCH    1-63
CLSFSP    1-92
CLTOTL    1-92      10-12      10-133
CLUNIT    1-51       9-202
CLVERS    1-51      10-21
CMDBUF    1-159      1-176     13-24
CMDCCL    1-191
CMDDSN    1-27
CMDFRM    1-27
CMDHD     1-26
CMDMEM    1-28      35-6#
CMDOFF    1-26
CMDSET    1-28
CMDSHO    1-28       5-5#
CMDUSE    1-28      33-13#
CMDWHO    1-28      33-6#
CO$BBT    1-109     10-157
CO$BNI    1-94      10-162
CO$BNO    1-94      10-161
CO$CR     1-93      10-163
CO$CTL    1-93      10-164
CO$DEF    1-60
```

Cross reference table (CREF V05.04)

```
CO$DTR    1-92     10-165
CO$FF     1-93     10-154
CO$FFO    1-93     10-160
CO$LC     1-93     10-156
CO$LFI    1-94     10-159
CO$LFO    1-94     10-158
CO$TAB    1-93     10-155
COAD      1-182     1-183
COAF      1-195
COAL      1-182     1-183
COLOO     1-176    11-162
COMMA     2-10#
CONFG2    1-56      9-25
CONFIG    1-107
CORUSR    1-55      9-38     11-57
CPUAH     1-176
CPUAL     1-176
CR        2-5#
CRLF      1-171     6-6      6-8      6-10     6-12     6-15     6-17     6-19     6-21     6-24     6-27     6-30
          6-32      6-36     6-38     6-40     7-47     7-103    7-107    7-127    9-13     9-209    10-128   11-183
         15-68     18-20    18-34    18-61    19-27    19-61    20-159   21-9     21-20    21-36    23-76    25-101
         26-17     26-18    26-27    27-42    29-17    30-29    31-68    32-34    34-17
CS$RON    1-74
CSHALC    1-97      7-116
CSHDEV    1-102    20-13    20-92
CSHDVN    1-102    20-23    20-110
CSHHD     1-158
CSHMSG    1-198     8-47
CSHSIZ    1-138     8-48
CSIMS1    1-170
CSIMS2    1-168
CSIMS4    1-190
CTLNAM    4-17#     9-68
CTRLTT    1-96      9-50
CURMTX    1-181    35-46
CURPRM    1-124
CVDVNM    1-178    19-41    20-76
CVTTAB    1-159     5-6      35-6
CVTUC     1-158
CW$5OH    1-107
CW$PRO    1-56      9-25
CXTBAS    1-101
CXTPAG    1-73
CXTWDS    1-101
DATTIM    1-27
DCCRD     1-131
DCCWR     1-131
DCSR      4-7#     16-28#    16-37    16-61
DCTRD     1-131
DCTWR     1-131
DEADEV    1-166
DELSPC    1-180
DETARG    1-195
DETHD     1-195
DETTXT    1-174    11-69
DEVDSP   14-16     15-78
```

Cross reference table (CREF V05.04)

```
DEVHD1    1-177
DEVIDL    1-162      1-182      1-183
DEVUNT    1-169
DFJMEM    1-62
DIABFL    1-119
DIABLO    1-142
DIABNO    1-120
DIVIDE    1-175     11-150
DIVSOR    1-188     34-19*     34-20*
DJABMS    1-186
DKASHD    1-53      18-30
DKSAV     1-159
DLCEMT    1-28
DLMSG     1-186
DLTXT     1-173
DMTALL    1-186
DMTARG    1-165
DMTSUB    1-191
DMYDEV    1-39      29-10      30-22
DOASGN    1-79
DORUN     1-27
DOSTOP    1-187
DSPCSR   15-64      16-7#
DSPMEM    8-64      35-12      35-45#
DVEC      4-6#      16-33*     16-38      16-71
DVSHH1    1-53      14-9
DVSHH2    1-53      14-10
DVSHH3    1-53      14-11
DVSTAT    1-70      15-29
DZTXT     1-193
EDIT      1-67
EDTFIL    1-179     23-16      31-48
EM$ACL    1-48
EM$CAP    1-38
EM$CIP    1-47
EM$CLB    1-48
EM$CLN    1-47
EM$CNO    1-38
EM$CPO    1-38
EM$CSE    1-64
EM$HNI    1-62
EM$ICL    1-46
EM$ILN    1-47       1-48
EM$IUN    1-47
EM$NAD    1-40
EM$NPD    1-47
EM$NSF    1-47
EM$NSL    1-48
EM$NUK    1-80
EM$SLT    1-48
EM$SLW    1-48
EM$TSL    1-48
EM$UIO    1-49
ERRLOC    1-61
ERRSEV    1-128
ESC       1-76
```

Cross reference table (CREF V05.04)

```
FC$CDX    1-158
FC$LNK    1-158
FD$NAM    1-158
FF        2-9#
FILNAM    1-160      1-161     32-5*     32-6*     32-8
FKILL     1-160      5-18       5-19     11-27
FPRINT    1-160
FRAC64   25-52      25-112#
FSTDL     1-83
FSTIOL    1-56       1-73       9-215
GAGMSG    1-185
GENTOP    1-97
GETKCH    1-71
GRT1      1-139
GTRD50    1-165
H.CSR     1-70      16-26
H.VEC     1-70      16-33     16-45
HANBSY    1-170
HANCHN    1-58
HANENT    1-69      15-37
HANIOC    1-33      15-21
HANNAM    4-5#      16-15*    16-16*    16-21
HANPAR    1-70      15-48
HANSIZ    1-69      15-56
HAZEL     1-71
HAZLFL    1-71
HAZLNO    1-71
HIMAP     1-137      8-06
HIPRI     1-171
HNBUF     1-169
HUPARG    1-190
II$$SZ    1-40      22-22
II$FLG    1-40      23-27
II$NAM    1-40      22-19     23-14
II$NPV    1-42      23-48     23-68
II$PRV    1-42      23-47     23-64
IIBUF     1-40      22-19     23-14     23-27     23-47     23-48     23-64     23-68
ILLCMD    1-166
ILSW2     1-77       9-56      9-112
IN$ACT    1-152
IN$CMD    1-152
IN$CNT    1-152
INDACT    1-162
INDERR    1-100
INDSAV    1-152
INDSTA    1-100
INFOMT    1-166
INGADR    1-40      22-16#
INGEMT    1-40      22-17
INSANT   23-26      23-88#
INSPRT   22-21      23-7#
INSTBL    1-40      22-15
INSTBN    1-41      22-23
INVDAT    1-195
INVDEV    1-194
INVEC     1-153      9-76
```

Cross reference table (CREF V05.04)

Cross reference table (CREF V05.04)

```
LDBASE    1-116
LDCLEN    1-65      31-14
LDDEVX    1-118
LDFLAG    1-116     31-59
LDMNT     1-64
LDNAM     1-164
LDNAME    1-116     31-19      31-31
LDOPHD    1-172
LDPDEV    1-116     31-65
LDSIZE    1-116     31-53
LF        2-4#
LF$OPN    1-157
LF$WRT    1-157
LFWLIM    1-77
LICTXT    1-139
LINBUF    1-72
LINCNT    1-74
LINCUR    1-77
LINFRE    1-186
LINIR     1-59
LINNXT    1-72
LINPNT    1-74
LINRTS    1-59
LITIME    1-99
LJSW      1-96
LMXLN     1-153      9-70       9-80
LMXNUM    1-151      9-71       9-93
LMXPRM    1-154      9-120      9-136
LNAME     1-54       9-186
LNBLKS    1-101     11-135
LNMAP     1-107
LNPRIM    1-107      9-39      11-71
LNSBLK    1-102     11-136
LNSPAC    1-111
LOCKTX    1-175     11-127
LOFSPC    1-104
LOGASN    1-167
LOGBAS    1-126      1-128
LOGBLK    1-156
LOGBUF    1-156
LOGCHK    1-127
LOGCHN    1-156
LOGCLS    1-171
LOGDVU    1-126      1-129
LOGFLG    1-156
LOGPTR    1-156
LOMAP     1-137      8-29       8-57
LOTBUF    1-75
LOTNXT    1-75
LOTPNT    1-75
LOTSIZ    1-76
LOTSPC    1-76
LOUTIR    1-59
LP$7BT    1-46       9-137
LP$ODD    1-46       9-146
LP$PAR    1-46       9-142
```

Cross reference table (CREF V05.04)

```
LP$SPD    1-46
LPRG1     1-140    11-174
LPRG2     1-140    11-176
LPRI      1-155     7-98     11-93
LPROG     1-84
LPROJ     1-84
LRBFIL    1-108
LRDTIM    1-74
LSCCA     1-104
LSECPT    1-89     11-76
LSTACT    1-72
LSTATE    1-135    11-101
LSTDL     1-83     11-67
LSTHL     1-56      9-211
LSTIOL    1-56
LSTMX     1-151
LSTPL     1-129     9-53     9-161     9-194     9-213     11-65     11-74
LSTPRM    1-125
LSTSL     1-135    11-188
LSTSPL    1-70     12-16
LSUCF     1-80
LSW       1-55      9-58     9-116     9-163     9-196     11-45     11-118
LSW2      1-98
LSW2S     1-103
LSW3      1-103     9-23     9-157     9-182
LSW4      1-121
LSW5      1-81      9-60
LSW6      1-150    11-125
LSW7      1-154    27-15     28-18
LSW8      1-117
LSW9      1-79
LTRMTP    1-143
LTSCMD    1-111
LUNAME    1-84
MAXALC    1-57
MAXASN    1-100
MAXAVL    1-169
MAXMEM    1-61     35-39*    35-47
MAXMTX    1-181    35-33     35-56
MAXPRI    1-61      1-155
MAXSEC    1-89     11-77
MDT       1-71
MHNSIZ    1-90      8-36
MHNSMS    1-91      8-35
MINTIM    1-89     11-145
MISSEQ    1-170
MNBASE    1-180
MNBPC     1-179
MNFLGS    1-179
MNTARG    1-190
MNTDEV    1-165
MNTFUL    1-167
MNTOP     1-180
MNTTXT    1-192     6-25
MONAR1    1-180
MONAR2    1-180
```

Cross reference table (CREF V05.04)

```
MONHD     1-180
MONVEC    1-165
MSGBUF    1-185
MSGEND    1-185
MTOPHD    1-166
MUL32     1-195
MXCSR     1-151     9-99
MXDTR     1-151
MXJADR    1-63
MXJMEM    1-62     35-29     35-31     35-57
MXJPRI    1-155     7-101
MXPRMT    1-133
MXVEC     1-153     9-85
NAMTOP    1-139
NARGS     3-69#     3-70      3-70      3-71      3-71      3-72      3-72      3-73      3-73      3-74      3-74      3-75
          3-75      3-76      3-76      3-77      3-77      3-78      3-78      3-79      3-79      3-80      3-80      3-81
          3-81      3-82      3-82      3-83      3-83      3-84      3-84      3-85      3-85      3-86      3-86      3-87
          3-87      3-88      3-88      3-89      3-89      3-90      3-90      3-91      3-91      3-92      3-92      3-93
          3-93      3-94      3-94      3-95      3-95      3-96      3-96      3-97      3-97      3-98      3-98      3-99
          3-99      3-100     3-100     3-101     3-101     3-102     3-102     3-103     3-103     3-104     3-104     3-105
          3-105     3-106     3-106     3-107     3-107     3-108     3-108     3-109     3-109     3-110     3-110     3-111
          3-111     3-112     3-112     3-113     3-113     3-114     3-114     3-115     3-115     3-116     3-116     3-117
          3-117     3-118     3-118     3-119     3-119     3-120     3-120     3-121     3-121    11-7#     11-8      11-8
NATXT     1-173     9-124
NEDCHR    1-59
NEWJSW    1-96
NFRESB    1-123
NINTXT    1-173     9-184
NLINES    1-109    20-154
NOCCL     1-172
NOCIN     1-164
NODAT     1-178    21-12
NOFIL     1-176     1-184     12-21
NOFLAG    1-166
NOFRDL    1-195
NOIND     1-170
NOKMON    1-191
NOLDMT    1-178    31-24
NONEMS    1-178    20-166     29-22
NOOPTT    1-184
NOPMGN    1-180
NOPRG     1-160     1-164
NOSTRT    1-163
NOTAVL    1-179    31-67
NOTON     1-185
NOTXT     1-173     9-156     25-85     25-94
NOUDC     1-177    13-33
NSPLDV    1-115    30-10
NSWPMS    1-181    35-19
NUCHN     1-108
NUMDCD    1-133     7-72
NUMDEV    1-146    14-18
NUMON      1-77
NUMST     11-105   11-226#
OCTFIX    1-173     9-87      15-31     15-42     15-50     16-65     16-75     16-88     32-20     32-23
OCTPRT    1-193     9-100
```

Cross reference table (CREF V05.04)

```
ODTBAS    1-149
OF$$SZ    1-148
OF$DEV    1-147
OF$FIL    1-147
OF$FLG    1-147
OF$UNT    1-147
OFFEMT    1-187
OKFEND    1-90
OKFILE    1-90
OPRTXT    1-51      9-49
OPTLST    1-36
OT$RON    1-148
OTHRON    1-189
OTRMNT    1-191
OVRCOR    1-163
PO$DBG    1-41
PO$OPR    1-39
PO$SPV    1-37
PO$SYS    1-37      6-33
PASLIN    1-128
PAUMSG    1-159
PBFEND    1-112
PEKADR    1-35     24-22*    24-28
PEKEMT    1-35     24-24
PEKSIZ    1-35     24-23*
PF$IOW    1-150
PF$SYS    1-150
PFCO      1-36
PFSO      1-36
PHYMEM    1-95      8-8
PIDCSR   16-55     17-36#
PIDSLT   17-11#    17-36     17-53
PIDVEC   16-83     17-53#
PMBUSY    1-180
PNAME     1-146     1-169     15-14     16-16
POPCF     1-170
PRGALL    1-27
PRGSIZ    1-72
PRGTOP    1-72
PRIVAO    1-38     26-12
PRIVCO    1-41      6-33     26-23
PRIVSO    1-37
PRMBUF    1-125
PRMEND    1-125
PRMPNT    1-124
PROSLT    1-46     17-12     17-17
PRTBUF    1-178    20-118    20-127    20-132
PRTDAT    1-194    21-8
PRTDC2    1-175    11-158    11-161    21-35
PRTDC3    1-175    11-139
PRTDEC    1-169     7-46      7-99      7-102     7-106     7-110     7-126     9-174     9-204     10-22     10-44     11-84
          18-59    20-148    21-32     25-54     31-54     32-31     35-34     35-54     35-58     36-7
PRTFIX    1-172     9-34      9-82      10-54     10-69     11-95     11-155    15-23     15-58     19-51     25-51     25-71
          25-77
PRTFNM    1-178    20-122
PRTKB     8-11      8-19      8-25      8-33      8-39      8-45      8-53      8-58      8-63      36-5#
```

Cross reference table (CREF V05.04)

```
PRTLN    1-174   11-52   11-72
PRTPCT   1-188   34-27
PRTR50   1-194    9-69   11-175  11-177  15-15   18-18   18-32   18-42   18-45   18-49   18-51   18-55
        19-42   20-77   25-40   25-42   29-14   29-16   30-24   31-41
PRTSPC   1-172    9-45    9-75    9-84    9-89    9-102   9-108   9-152   9-178   10-46   10-56   10-60
        10-71   10-75   15-13   15-20   15-28   15-36   15-47   15-55   15-60   16-63   16-70   16-93
        19-44   19-53   25-82
PRTTIM   1-194   21-43   33-13
PRTTMD   1-177   34-16
PRTTMV   1-176   11-169
PRTTOD   1-194   21-19
PRTTTP   1-173    9-106
PRTUNM   1-174    9-198   11-182  19-57
PRTUSE  11-32    34-5#
PRTWRN   1-198
PRVLST   1-41    23-67   23-71   26-16   26-26
PRVOPT   1-37
PUSHCF   1-161
PVNPW    1-36    23-49
PVON     1-81
QHDMS1   1-52    12-14
QHDMS2   1-53    12-15
QUME     1-142
QUMEFL   1-144
QUMENO   1-144
R50BUF   1-165
R50CLO   4-8#
R50COM   1-104
R50DIR   1-161
R50DK    1-194   18-25
R50DSK   1-167
R50DUP   1-162
R50IND   1-161
R50K52   1-162
R50KED   1-162
R50KEX   1-162
R50LD    1-167
R50LDO   1-165   31-29
R50LD7   1-167
R50LOG   1-171
R50MON   1-195
R50NO    1-168
R50PIP   1-162
R50SAV   1-161
R50SY    1-161   18-11
R50TSX   1-162
R50TT    1-190
R50UCL   1-162
R50VIR   1-163
RC$$SZ   1-45    24-13   24-23   24-29
RC$AEP   1-44    25-92
RC$AGE   1-44    25-92
RC$CNT   1-43    25-75
RC$EXC   1-44    25-83
RC$FLG   1-44    25-15   25-17   25-19   25-59   25-83   25-92
RC$GBL   1-44    25-19
```

Cross reference table (CREF V05.04)

```
RC$LCG    1-46      25-17
RC$LEN    1-45      25-46
RC$NAM    1-44      25-39     25-41
RC$OWN    1-43      25-68
RC$PVT    1-45      25-59
RC$USE    1-44      25-15
RCBBAS    1-45      24-11
RCBEND    1-45      24-14
RDB       1-134     29-6
RDBEND    1-134     29-9
RDCMD     1-159      3-82      6-79      6-84      7-48      7-112     7-128     33-7      33-14     35-13     35-20
RDERM     1-163
REMTXT    1-51       9-62
RESDEV    1-148
RGNDSP   24-12      24-27     25-11#
RNMS      1-174     11-107
RONTXT    1-179     31-61
RSR       1-151      9-76
RSTPRV    1-38
RT$$SZ    1-134     29-18
RT$DEV    1-134     29-10
RT$NAM    1-134     29-12     29-15
RUNDEV    1-101
RUNEMT    1-163
RUNHD     1-159
RUNMS     1-195
S$$RUN    1-80      11-108
S$HICP    1-86      11-210
S$INWT    1-87      11-215
S$IOFN    1-82      11-213
S$IOWT    1-140     11-221
S$LOW     1-84      11-214
S$MSWT    1-88      11-219
S$OTFN    1-82      11-209
S$OTLO    1-82      11-212
S$OTWT    1-87      11-216
S$QMIO    1-80      11-223
S$QUSR    1-140     11-220
S$RT      1-84      11-206
S$SFWT    1-87      11-222
S$SPCB    1-141     11-225
S$SPDB    1-141     11-224
S$SPND    1-81      11-218
S$TMWT    1-87      11-217
S$TTFN    1-82      11-208
S$TTSC    1-83      11-207
S$TWFN    1-82      11-211
S9600     1-87
SCHAIN    1-118
SD$BAK    1-121
SD$DEL    1-113
SD$FLK    1-114
SD$HLD    1-124
SD$SNG    1-123
SD$WFM    1-114
SDBLK     1-115
```

Cross reference table (CREF V05.04)

```
SDBU      1-121
SDBUF1    1-115
SDCB      1-129     10-88     12-7      30-12
SDCBND    1-129     10-89     12-8      12-18     30-13     30-27
SDCBSZ    1-135     10-93     12-12     12-17     30-26
SDDVU     1-134     10-91
SDFHD     1-126     12-10
SDFLAG    1-114
SDFORM    1-114
SDNAME    1-135     30-21
SDSFCB    1-113
SDSKIP    1-121
SEARCH    1-159      5-10     11-25
SERFLG    1-55
SETHD     1-168
SETMEM   35-8       35-17#
SF$1ST    1-124
SF$BSY    1-123
SF$HLD    1-124
SFFILE    1-141
SFFLAG    1-126
SFFORM    1-123
SFID      1-70
SFNMBL    1-123
SFQLNK    1-126
SFUSER    1-141
SH$$SZ    1-69
SH$FLG    1-68
SH$NAM    1-68
SH$RTN    1-68
SH$VAL    1-68
SHMTH1    1-177    20-59
SHMTH2    1-177    20-60
SHOALC    6-11      6-83     19-5#
SHOASN    6-9       6-46     18-5#
SHOCL     6-16      6-88     10-5#
SHOCMD    6-52     13-13#
SHODAT    6-63     21-5#
SHODEV    6-7       6-57      6-68     14-5#
SHOHD     1-192     3-69#     5-9
SHOINS    6-35      6-93     22-5#
SHOJOB    6-13      6-73     11-11#    33-6
SHOKEY    1-33      3-87
SHOMEM    6-18      6-78      8-5#
SHOMNT    6-26      6-98     20-5#
SHOPRV    6-37      6-103    26-5#
SHOQUE    6-108    12-6#
SHOREG    6-31      6-128    24-5#
SHOSLE    6-39      6-118    27-5#
SHOSPL    6-20      6-138    30-6#
SHOSRT    6-29      6-113    29-5#
SHOSUB    6-23      6-133    31-6#
SHOSYP    6-143     7-133#
SHOTIM    6-153    21-19#
SHOTRM    6-14      6-58      6-148     9-5#
SHOUSE    6-158    21-43#
```

Cross reference table (CREF V05.04)

```
SHOVAL   7-5     7-10    7-15    7-20    7-25    7-30    7-35    7-40    7-45#   7-53    7-58    7-63
         7-68    7-73    7-78    7-83    7-88    7-93
SHOVER   6-5     6-163   21-27#
SHOWVM   6-41    6-123   32-5#
SHRRCB   1-45    24-19
SHRRCN   1-45    24-20
SHTMSG   1-187
SHVTX1   1-200   32-17
SHVTX2   1-200   32-21
SHVTX3   1-200   32-29
SHVTX4   1-200   32-10
SIZEMT   1-190
SIZVAL   1-164
SJOHD    11-7#   11-24
SJSPPN   1-200   11-28*   11-194*
SKPSPC   1-26
SLCKFL   27-26   27-32    27-38   28-10#
SLKDOF   1-49
SLKDON   1-49
SLMXLN   1-198
SMRSIZ   1-138    8-22
SO$NO    1-69
SO$NVL   1-69
SO$OCT   1-69
SOPALC   1-60    3-71     3-72    6-83#
SOPALL   3-70    6-58#
SOPASN   3-73    5-8      6-46#
SOPCL    3-76    3-79     6-88#
SOPCMD   3-77    6-52#
SOPCON   3-75    6-57#
SOPCSH   3-74    7-116#
SOPCTM   3-78    7-57#
SOPDAT   1-60    3-81     6-63#
SOPDEV   3-80    6-68#
SOPHIO   3-83    7-67#
SOPIIO   3-85    7-62#
SOPINS   3-84    6-93#
SOPJMP   6-42    6-47     6-53    6-59    6-64    6-69    6-74    6-79#   6-89    6-94    6-99    6-104
         6-109   6-114    6-119   6-124   6-129   6-134   6-139   6-144   6-149   6-154   6-159   6-164
SOPJOB   3-86    3-119    6-73#
SOPMB    3-92    7-4#
SOPMC    3-90    7-9#
SOPMEM   3-89    6-78#
SOPMNT   3-93    6-98#
SOPMR    3-91    7-14#
SOPNDC   3-94    7-72#
SOPPDF   3-98    7-87#
SOPPHI   3-97    7-82#
SOPPLO   3-96    7-77#
SOPPRI   3-95    7-97#
SOPPRV   3-99    6-103#
SOPPVR   3-100   7-92#
SOPQ0    3-102   7-19#
SOPQ1    3-103   7-24#
SOPQ1A   3-104   7-29#
SOPQ1B   3-105   7-34#
```

Cross reference table (CREF V05.04)

```
SOPQ1C    3-106     7-39#
SOPQ2     3-107     7-44#
SOPQ3     3-108     7-52#
SOPQUE    3-101     6-108#
SOPREQ    3-109     6-128#
SOPSLE    3-112     6-118#
SOPSPL    3-113     6-138#
SOPSRT    3-110     6-113#
SOPSUB    3-88      3-111     6-133#
SOPSYP    2-114     6-143#
SOPTIM    1-60      3-116     6-153#
SOPTRM    3-115     3-117     6-148#
SOPUSE    3-118     6-158#
SOPVER    3-120     6-163#
SOPVM     3-121     6-123#
SPACE1    1-43     25-62
SPACE2    1-174    11-61     11-89     11-96     11-141    11-166    11-173    25-58     30-25
SPACE3    1-174     7-45     11-181    25-91
SPACE5    1-175    11-129
SPACE6    1-178    20-84
SPACTV    1-182     1-183
SPCF      1-184
SPDTX1    1-173     9-127
SPFLK     1-184
SPFUL     1-184
SPGEMT    1-184
SPLACT    1-187
SPLCHN    1-66
SPLHD     1-181
SPLHLA    1-172
SPLPND    1-189
SPSNG     1-182      1-184
SPUBUF    1-63
SPWFM     1-182      1-183
SRTSIZ    1-138      8-42
SRTSMS    1-189      8-41
SRTTXT    1-192      6-28
SSRMAP    1-192      8-21
STBIT    11-102    11-206#    11-226
STLGCN    1-27
STLGHD    1-171
STNAM    11-113     11-230#
STPASK    1-189
STPBUF   11-113*    11-114     11-251#
STPFLG    1-86
SUBARG    1-179     31-42
SUBTXT    1-192      6-42
SUCS      1-137
SUM1      1-188     34-42
SUM2      1-188     34-43
SUM3      1-188     34-44
SUM4      1-188     34-45
SUM5      1-189     34-46     34-49     34-50
SUM6      1-189     34-47
SUM7      1-189     34-48
SUMS      1-139
```

Cross reference table (CREF V05.04)

```
SUMVEC  34-21      34-42#
SUPCOD   1-139
SWPTX    1-175     11-120
SXBPNT   1-63
SYASHD   1-53      18-16
SYHD1    1-174     11-36
SYHD2    1-174     11-37
SYINDX   1-146
SYNAME   1-147     16-15     18-17     18-31
SYPSWD   1-33       7-134     7-138
SYSAV    1-159
SYSDAT   1-136
SYTIMH   1-136
SYTIML   1-136
SYUNIT   1-146
TAB      2-8#
TALEMT   1-108
TBLOVF   1-168
TECO     1-67
TK1SEC   1-138
TK1VAL   1-136
TM$AUT   1-51       9-118
TM$C13   1-54      10-40
TM$CDS   1-50       7-122
TM$CEN   1-50       7-124
TM$CLO   1-52      10-14
TM$CL1   1-52      10-27
TM$CL2   1-52      10-28
TM$CL3   1-52      10-43
TM$CL4   1-52      10-58     10-73
TM$CL5   1-52      10-95
TM$CL6   1-52      10-108
TM$CL7   1-52      10-20
TM$CL8   1-53      10-23
TM$CNG   1-50       7-118
TM$GBL   1-43      25-64
TM$HPE   1-50       7-111
TM$HPR   1-49       7-108
TM$IN1   1-35      22-10
TM$IN2   1-35      22-11
TM$KED   1-34      27-25
TM$LCL   1-43      25-61
TM$LPR   1-49       7-104
TM$NAD   1-54      19-21
TM$NNR   1-35      24-36
TM$NO    1-34      28-20
TM$NSD   1-54      30-15
TM$NSP   1-33       7-136
TM$OFF   1-33      27-17
TM$ON    1-33      27-19
TM$PR1   1-49       7-97
TM$PR2   1-49       7-100
TM$PVA   1-38      26-11
TM$PVC   1-38      26-22
TM$PVL   1-39      23-57
TM$RD1   1-43      25-33
```

Cross reference table (CREF V05.04)

```
TM$RD2    1-43      25-34
TM$SDN    1-54      30-20
TM$SL1    1-33      27-11
TM$SUB    1-34      27-37
TM$TTY    1-34      27-31
TMIDLH    1-65      34-45
TMIOH     1-65      34-47
TMIOWH    1-64      34-43
TMSWPH    1-65      34-48
TMSWTH    1-65      34-44
TMTOTH    1-64       1-188     34-11     34-19
TMTOTL    1-64       1-188     34-12     34-20
TMUSRH    1-64      34-42
TOTMMS    1-192      8-7
TOTON     1-86
TOTXT     1-171
TRGRET    1-139
TRMHD1    1-50       9-14
TRMHD2    1-50       9-15
TRMSTR    1-161
TSKSHO    1-5#       1-26
TSR       1-151
TSXLN     1-139
TSXSIT    1-139
TSXSMS    1-193      8-28
TSXVER    1-36      21-27
TSXVRS    1-32      21-28
TXTC1     1-86       9-171
TXTCL     1-85       9-173
UC$MDC    1-158
UC$NDC    1-158
UCHAN     1-106
UCIDEF    1-58
UCISPC    1-90
UCLBLK    1-157     13-17
UCLCMD    1-26       6-51#
UCLDAT    1-157
UCLNAM    1-111
UERSEV    1-128
UFORM     1-98
UFPTRP    1-113
UHIMEM    1-101
UKMNAM    1-79
UMSSMS    1-192      8-13
UMSYTP    1-83       8-14
UPTMMS    1-187     34-10
USPLCH    1-86
USRMMS    1-193      8-55
USRSTK    1-62
USTART    1-97
UTRPAD    1-61
VCORTM    1-133      7-57
VCSHNB    1-89       7-120     7-125
VHIPCT    1-130      7-67
VIMAGE    1-96
VINTIO    1-128      7-42
```

Cross reference table (CREF V05.04)

```
VMAXMC    1-34       7-9
VMXMRB    1-34       7-14
VMXMSG    1-34       7-4
VPRIDF    1-78       7-87
VPRIHI    1-42       7-82       7-109
VPRILO    1-42       7-77       7-105
VPRIVR    1-78       7-92
VQUAN0    1-129      1-130      7-19
VQUAN1    1-130      7-24
VQUAN2    1-130      7-44
VQUAN3    1-129      1-130      7-52
VQUN1A    1-130      7-29
VQUN1B    1-122      7-34
VQUN1C    1-122      7-39
VSWPFL    1-63      35-17
VT100     1-142
VT10FL    1-144
VT10N0    1-144
VT200     1-141
VT2007    1-141
VT2008    1-141
VT20FL    1-145
VT20N0    1-145
VT52      1-142
VT52FL    1-143
VT52N0    1-120
WILDFL    1-66
WLDNAM    2-12#
WTMS      1-174     11-110
XAREA     1-160     16-21      16-26      16-32      32-8       32-15      32-27
YESTXT    1-173      9-165     25-87      25-96
ZCLR      1-151
```

Cross reference table (CREF V05.04)

```
...CM1   16-21   16-26   16-32   32-8    32-15   32-27
...CM2   16-21   16-21   16-26   16-26   16-26   16-26   16-32   16-32   16-32   16-32   32-8    32-8
         32-15   32-15   32-15   32-15   32-15   32-27   32-27   32-27   32-27   32-27
...CM3   16-98   32-33
...CM5    6-6     6-8     6-10    6-12    6-15    6-17    6-19    6-21    6-22    6-24    6-25    6-27
          6-28    6-30    6-32    6-36    6-38    6-40    7-45    7-47    7-97    7-100   7-103   7-104
          7-107   7-108   7-111   7-118   7-122   7-124   7-127   7-136   7-138   8-7     8-13    8-21
          8-28    8-35    8-41    8-47    8-55    8-60    9-13    9-14    9-15    9-43    9-63    9-79
          9-118   9-124   9-130   9-132   9-139   9-141   9-144   9-148   9-150   9-171   9-173   9-176
          9-184   9-190   9-200   9-202   9-209   10-14   10-20   10-23   10-27   10-28   10-40   10-43
         10-58   10-73   10-95   10-108  10-110  10-114  10-124  10-128  11-36   11-37   11-60   11-61
         11-69   11-73   11-85   11-88   11-89   11-96   11-111  11-114  11-120  11-127  11-129  11-133
         11-134  11-140  11-141  11-157  11-160  11-162  11-166  11-173  11-181  11-183  12-14   12-15
         12-21   13-33   14-9    14-10   14-11   15-68   16-21   16-26   16-32   18-6    18-16   18-19
         18-20   18-30   18-33   18-34   18-43   18-46   18-52   18-58   18-60   18-61   19-21   19-27
         19-28   19-29   19-39   19-61   20-59   20-60   20-78   20-84   20-132  20-144  20-147  20-159
         20-166  21-9    21-12   21-20   21-27   21-33   21-36   22-10   22-11   23-21   23-31   23-36
         23-59   23-72   23-76   24-36   25-33   25-34   25-53   25-58   25-61   25-62   25-64   25-85
         25-87   25-91   25-94   25-96   25-101  26-11   26-17   26-18   26-22   26-27   27-11   27-20
         27-42   28-14   28-20   28-24   29-17   29-22   30-15   30-20   30-25   30-29   31-24   31-42
         31-51   31-55   31-61   31-67   31-68   32-8    32-10   32-15   32-17   32-21   32-27   32-29
         32-34   34-10   34-17   34-24   35-19   35-33   35-35   35-46   35-55   35-56   35-59   36-8
...CM7   16-26   16-32
.CLOSE    1-20#  16-98   32-33
.CSIGE    1-18#
.CSISP    1-16#
.DATE     1-19#  21-5
.ENTER    1-21#
.EXIT     1-21#  35-40
.FPROT    1-22#
.GTIM     1-19#
.GTLIN    1-19#
.GVAL     1-22#
.HERR     1-22#  16-99   32-35
.LOOKU    1-20#  16-21   32-8
.PRINT    1-20#   6-6     6-8     6-10    6-12    6-15    6-17    6-19    6-21    6-22    6-24    6-25
          6-27    6-28    6-30    6-32    6-36    6-38    6-40    7-45    7-47    7-97    7-100   7-103
          7-104   7-107   7-108   7-111   7-118   7-122   7-124   7-127   7-136   7-138   8-7     8-13
          8-21    8-28    8-35    8-41    8-47    8-55    8-60    9-13    9-14    9-15    9-63    9-118
          9-124   9-171   9-173   9-176   9-184   9-190   9-200   9-202   9-209   10-14   10-20   10-23
         10-27   10-28   10-40   10-43   10-58   10-73   10-95   10-108  10-128  11-36   11-37   11-61
         11-69   11-89   11-96   11-111  11-114  11-120  11-127  11-129  11-140  11-141  11-162  11-166
         11-173  11-181  11-183  12-14   12-15   12-21   13-33   14-9    14-10   14-11   15-68   18-6
         18-16   18-20   18-30   18-34   18-43   18-61   19-21   19-27   19-28   19-29   19-61   20-59
         20-60   20-84   20-132  20-159  20-166  21-9    21-12   21-20   21-27   21-36   22-10   22-11
         23-21   23-36   23-59   23-76   24-36   25-33   25-34   25-58   25-61   25-62   25-64   25-85
         25-87   25-91   25-94   25-96   25-101  26-11   26-17   26-18   26-22   26-27   27-11   27-20
         27-42   28-20   28-24   29-17   29-22   30-15   30-20   30-25   30-29   31-24   31-42   31-51
         31-61   31-67   31-68   32-10   32-17   32-21   32-29   32-34   34-10   34-17   34-24   35-19
         35-33   35-35   35-46   35-55   35-56   35-59   36-8
.PURGE    1-17#
.PVAL     1-22#
.READW    1-17#  16-26   16-32
.REOPE    1-18#
.SAVES    1-18#
.SERR     1-22#  16-20   32-7
```

Cross reference table (CREF V05.04)