```
000.001                1  .PIP.   EQU     1               Don't ASSEMBLE AS PIP
000.000                2  ONECOPY EQU     0               ASSEMBLE AS ONECOPY
                       3
000.001                4          IF      .PIP.
                       5          TITLE   'PIP - PERIPHERAL INTERCHANGE PROGRAM'
                       6          ELSE
                       8          ENDIF
                       9
                      10
                      11  ***     PIP - PERIHPERAL INTERCHANGE PROGRAM.
                      12  *
                      13  *           J. G. L., 11/1977 FOR *HEATH* COMPANY
                      14  *
                      15  *       COPYRIGHT 1977 BY HEATH COMPANY
                      16  *
                      17  *            G. C.,       78/09   Maintenence Release
                      18  *                         79/04
                      19  *
                      20  *                         79/11   50.05.00
                      21  *                         80      50.06.00
                      22  *                                  /2.0a/ = /80.09.sc/
                      23  *                                  /2.0b/ = /80.10.sc/
                      24  *


                      26  ***     USE:
                      27  *
                      28  *       DEST=SOURCE1 [,SOURCE2,...,SOURCEN] [/SWITCH1.../SWITCHN]
                      29  *
                      30  *       SWITCHES:
                      31  *
                      32  *       /ALL[OCATE]
                      33  *       /R[ENAME]                   RENAME
                      34  *       /DEL[ETE]           DELETE
                      35  *       /L[IST]             LIST
                      36  *       /B[RIEF]            BRIEF LIST
                      37  *       /SY[STEM]           ENCLUDE SYSTEM FILES
                      38  *       /V[ERSION]          PIP VERSION NUMBER
                      39  *       /MOU[NT]            MOUNT DEVICE
                      40  *       /DIS[MOUNT]         DISMOUNT DEVICE
                      41  *       /RES[ET]            RESET DEVICE
                      42  *
                      43  *       /SU[PRESS]          SUPRESS
                      44  *       /JGL                WHO?
```

```
                        46  **      SYSTEM EQUIVALENCES
                        47
000.000                 48  CN.SOU  EQU     0               SOURCE CHANNEL NUMBER
000.001                 49  CN.DES  EQU     1               DESTINATION CHANNEL NUMBER
000.002                 50  CN.DIR  EQU     2               DIRECTORY CHANNEL NUMBER
                        51
                        52  **      PROGRAM ERROR CODES
                        53
000.200                 54  PEC.DF  EQU     200Q            DEVICE FORMAT ERROR
000.201                 55  PEC.DNC EQU     201Q            DEVICES NOT CONSISTANT
000.203                 56  PEC.TFI EQU     203Q            TARGET FILE ILLEGAL
000.204                 57  PEC.CS  EQU     204Q            CONTRADICTORY SWITCHES
000.205                 58  PEC.IUW EQU     205Q            ILLEGAL USE OF WILDCARD
000.206                 59  PEC.IDF EQU     206Q            ILLEGAL DESTINATION FILE FORMAT
000.207                 60  PEC.SFI EQU     207Q            SOURCE FILE ILLEGAL
000.000                 61          IF      ONECOPY
000.210                 62  PEC.FCI EQU     210Q            FILE CONCATINATION ILLEGAL
                        63          ENDIF
                        64
000.000                 65          XTEXT   U8250


                        67X **      8250 UART CONTROL AND BIT DEFINITIONS.
                        68X
000.350                 69X SC.ACE  EQU     350Q            SYSTEM CONSOLE PORT IF 8250 ACE
000.156                 70X AC.DLY  EQU     110             220 MIL. SEC. DELAY FOR 8250
                        71X
000.000                 72X UR.RBR  EQU     0               RECEIVER BUFFER REGISTER (READ ONLY)
                        73X
000.000                 74X UR.THR  EQU     0               TRANSMITTER HOLDING REGISTER (WRITE ONLY)
                        75X
000.000                 76X UR.DLL  EQU     0               DIVISOR LATCH (LEAST SIGNIFICANT)
                        77X
000.001                 78X UR.DLM  EQU     1               DIVISOR LATCH (MOST SIGNIFICANT)
                        79X
000.001                 80X UR.IER  EQU     1               INTERRUPT ENABLE REGISTER
000.001                 81X UC.EDA  EQU     00000001B        ENABLE RECEIVED DATA AVAILABLE INTERRUPT
000.002                 82X UC.TRE  EQU     00000010B        ENABLE TRANSMIT HOLD REGISTER EMPTY INTERRUPT
000.004                 83X UC.RSI  EQU     00000100B        ENABLE RECEIVE STATUS INTERRUPT
000.010                 84X UC.MSI  EQU     00001000B        ENABLE MODEM STATUS INTERRUPT
                        85X
000.002                 86X UR.IIR  EQU     2               INTERRUPT IDENTIFICATION REGISTER
000.001                 87X UC.IIP  EQU     00000001B        INVERTED INTERRUPT PENDING (0 MEANS PENDING)
000.006                 88X UC.IID  EQU     00000110B        INTERRUPT ID
                        89X
000.003                 90X UR.LCR  EQU     3               LINE CONTROL REGISTER
000.000                 91X UC.5BW  EQU     00000000B        5 BIT WORDS
000.001                 92X UC.6BW  EQU     00000001B        6 BIT WORDS
000.002                 93X UC.7BW  EQU     00000010B        7 BIT WORDS
000.003                 94X UC.8BW  EQU     00000011B        8 BIT WORDS
000.004                 95X UC.2SB  EQU     00000100B        TWO STOP BITS SELECTED
000.010                 96X UC.PEN  EQU     00001000B        PARITY COMPUTATION ENABLED
000.020                 97X UC.EPS  EQU     00010000B        EVEN PARITY SELECT
000.040                 98X UC.SKP  EQU     00100000B        STICK PARITY
```

```
000.100                 99X UC.SB   EQU     01000000B       SET BREAK
000.200                100X UC.DLA  EQU     10000000B       DIVISOR LATCH ACCESS
                       101X
000.004                102X UR.MCR  EQU     4               MODEM CONTROL REGISTER
000.001                103X UC.DTR  EQU     00000001B       DATA TERMINAL READY
000.002                104X UC.RTS  EQU     00000010B       REQUEST TO SEND
000.004                105X UC.OU1  EQU     00000100B       OUT 1
000.010                106X UC.OU2  EQU     00001000B       OUT 2
000.020                107X UC.LOO  EQU     00010000B       LOOP
                       108X
000.005                109X UR.LSR  EQU     5               LINE STATUS REGISTER
000.001                110X UC.DR   EQU     00000001B       DATA READY
000.002                111X UC.OR   EQU     00000010B       OVERRUN
000.004                112X UC.PE   EQU     00000100B       PARITY ERROR
000.010                113X UC.FE   EQU     00001000B       FRAMING ERROR
000.020                114X UC.BI   EQU     00010000B       BREAK INTERRUPT
000.040                115X UC.THE  EQU     00100000B       TRANSMITTER HOLDING REGISTER EMPTY
000.100                116X UC.TSE  EQU     01000000B       TRANSMITTER SHIFT REGISTER EMPTY
                       117X
000.006                118X UR.MSR  EQU     6               MODEM STATUS REGISTER
000.001                119X UC.DCS  EQU     00000001B       DELTA CLEAR TO SEND
000.002                120X UC.DDR  EQU     00000010B       DELTA DATA SET READY
000.004                121X UC.TER  EQU     00000100B       TRAILING EDGE OF RING
000.010                122X UC.DRL  EQU     00001000B       DELTA RECEIVE LINE SIGNAL DETECT
000.020                123X UC.CTS  EQU     00010000B       CLEAR TO SEND
000.040                124X UC.DSR  EQU     00100000B       DATA SET READY
000.100                125X UC.RI   EQU     01000000B       RING INDICATOR
000.200                126X UC.RLS  EQU     10000000B       RECEIVED LINE SIGNAL DETECT
000.000                127          XTEXT   U8251
```

```
                    130X **     8251 USART BIT DEFINITIONS.
                    131X *
                    132X
                    133X **     PORT ADDRESSES
                    134X
000,000             135X UDR    EQU     0           DATA REGISTER IS EVEN
000.001             136X USR    EQU     1           STATUS REGISTER IS NEXT
                    137X
000.372             138X SC.UART EQU    372Q        CONSOLE USART ADDRESS (IFF 8251)
                    139X
                    140X
                    141X **     MODE INSTRUCTION CONTROL BITS.
                    142X
000.100             143X UMI.1B EQU     01000000B   1 STOP BIT
000.200             144X UMI.HB EQU     10000000B   1 1/2 STOP BITS
000.300             145X UMI.2B EQU     11000000B   2 STOP BITS
000.040             146X UMI.PE EQU     00100000B   EVEN PARITY
000.020             147X UMI.PA EQU     00010000B   USE PARITY
000.000             148X UMI.L5 EQU     00000000B   5 BIT CHARACTERS
000.004             149X UMI.L6 EQU     00000100B   6 BIT CHARACTERS
000.010             150X UMI.L7 EQU     00001000B   7 BIT CHARACTERS
000.014             151X UMI.L8 EQU     00001100B   8 BIT CHARACTERS
000.001             152X UMI.1X EQU     00000001B   CLOCK X 1
000.002             153X UMI.16X EQU    00000010B   CLOCK X 16
000.003             154X UMI.64X EQU    00000011B   CLOCK X 64
                    155X
                    156X **     COMMAND INSTRUCTION BITS.
                    157X
000.100             158X UCI.IR EQU     01000000B   INTERNAL RESET
000.040             159X UCI.RO EQU     00100000B   READER-ON CONTROL FLAG
000.020             160X UCI.ER EQU     00010000B   ERROR RESET
000.004             161X UCI.RE EQU     00000100B   RECEIVE ENABLE
000.002             162X UCI.IE EQU     00000010B   ENABLE INTERRUPTS FLAG
000.001             163X UCI.TE EQU     00000001B   TRANSMIT ENABLE
                    164X
                    165X **     STATUS READ COMMAND BITS.
                    166X
000.100             167X USR.BD EQU     01000000B   Break Detect                /80.08.sc/
000.040             168X USR.FE EQU     00100000B   FRAMING ERROR
000.020             169X USR.OE EQU     00010000B   OVERRUN ERROR
000.010             170X USR.PE EQU     00001000B   PARITY ERROR
000.004             171X USR.TXE EQU    00000100B   TRANSMITTER EMPTY
000.002             172X USR.RXR EQU    00000010B   RECEIVER READY
000.001             173X USR.TXR EQU    00000001B   TRANSMITTER READY
000.000             174         XTEXT   DIRDEF


                    176X **     DIRECTORY ENTRY FORMAT.
                    177X
000.000             178X        ORG     0
                    179X
                    180X
000.377             181X DF.EMP EQU     377Q        FLAGS ENTRY EMPTY
000.376             182X DF.CLR EQU     376Q        FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
                    183X
```

```
000.000              184X DIR.NAM DS      8              NAME
000.010              185X DIR.EXT DS      3              EXTENSION
000.013              186X DIR.PRO DS      1              PROJECT
000.014              187X DIR.VER DS      1              VERSION
000.015              188X DIRIDL  EQU     *              FILE IDENTIFICATION LENGTH
                     189X
000.015              190X DIR.CLU DS      1              CLUSTER FACTOR
000.016              191X DIR.FLG DS      1              FLAGS
000.017              192X         DS      1              RESERVED
000.020              193X DIR.FGN DS      1              FIRST GROUP NUMBER
000.021              194X DIR.LGN DS      1              LAST GROUP NUMBER
000.022              195X DIR.LSI DS      1              LAST SECTOR INDEX (IN LAST GROUP)
000.023              196X DIR.CRD DS      2              CREATION DATE
000.025              197X DIR.ALD DS      2              LAST ALTERATION DATE
                     198X
000.027              199X DIRELEN EQU     *              DIRECTORY ENTRY LENGTH
000.027              200          XTEXT   DIFDEF


                     202X **      DIRECTORY FILE FLAGS.
                     203X
000.200              204X DIF.SYS EQU     10000000B              SYSTEM FILE
000.100              205X DIF.LOC EQU     01000000B              LOCKED FOR CHANGE
000.040              206X DIF.WP  EQU     00100000B              WRITE PROTECTED
000.020              207X DIF.CNT EQU     00010000B              CONTIGUOUS FILE
                     208X
000.027              209          XTEXT   OVLDEF


                     211X **      OVERLAY TABLE ENTRYS.
                     212X
000.000              213X         ORG     0
                     214X
000.000              215X OVL.COD DS      2              FIRST SECTOR OF OVERLAY CODE
000.002              216X OVL.SIZ DS      2              OVERLAY SIZE
000.004              217X OVL.ENT DS      2              OVERLAY ENTRY POINT
000.006              218X OVL.FLB DS      1              OVERLAY FLAG BYTE
000.007              219X         DS      1              DUMMY BYTE TO ROUND TABLE SIZE UP TO 8
000.010              220X OVL.ENS EQU     *              OVERLAY ENTRY SIZE
                     221X
                     222X *       OVERLAY INDICES
                     223X
000.000              224X         ORG     0
                     225X
000.000              226X OVL0    DS      1
000.001              227X OVL1    DS      1
000.002              228          XTEXT   DEVDEF
```

```
                        230X **      DEVICE TABLE ENTRYS.
                        231X
000.000                 232X         ORG     0
                        233X
000.000                 234X DEV.NAM DS      2              DEVICE NAME
000.000                 235X DV.EL   EQU     00000000B      END OF DEVICE LIST FLAG
000.001                 236X DV.NU   EQU     00000001B      DEVICE ENTRY NOT IN USE
                        237X
000.002                 238X DEV.RES DS      1              DRIVER RESIDENSE CODE
000.001                 239X DR.IM   EQU     00000001B      DRIVER IN MEMORY
000.002                 240X DR.PR   EQU     00000010B      DRIVER PERMINANTLY RESIDENT
                        241X
000.003                 242X DEV.JMP DS      1              JMP TO PROCESSOR
000.004                 243X DEV.DDA DS      2              DRIVER ADDRESS
000.006                 244X DEV.FLG DS      1              FLAG BYTE
000.001                 245X DT.DD   EQU     00000001B      DIRECTORY DEVICE
000.002                 246X DT.CR   EQU     00000010B      CAPABLE OF READ OPERATION
000.004                 247X DT.CW   EQU     00000100B      CAPABLE OF WRITE OPERATION
000.010                 248X DT.RN   EQU     00001000B      Capable of random access        /80.02.SC/
000.020                 249X DT.CH   EQU     00010000B      Capable of Character mode       /80.02.SC/
                        250X
000.007                 251X DEV.MUM DS      1              MOUNTED UNIT MASK
000.010                 252X DEV.MNU DS      1              MAXIMUM NUMBER OF UNITS
000.011                 253X DEV.UNT DS      2              ADDRESS OF UNIT SPECIFIC DATA TABLE
                        254X
000.013                 255X DEV.DVL DS      2              DRIVER BYTE LENGTH
000.015                 256X DEV.DVG DS      1              DRIVER ROUTINE GROUP ADDRESS
                        257X
000.016                 258X DEVELEN EQU     *              DEVICE TABLE ENTRY LENGTH



                        260X **      UNIT SPECIFIC DEVICE DATA TABLE ENTRIES
                        261X
000.000                 262X         ORG     0
                        263X
000.000                 264X UNT.FLG DS      1              UNIT SPECIFIC  *DEV.FLG*
000.001                 265X UNT.SPG DS      1              Sectors Per Group               /80.04.GC/
000.002                 266X UNT.GRT DS      2              ADDRESS OF GROUP RESERVATION TABLE (IF DT.DD)
000.004                 267X UNT.GTS DS      2              GRT SECTOR NUMBER
000.006                 268X UNT.DIS DS      2              DIRECTORY FIRST SECTOR NUMBER
                        269X
000.010                 270X UNT.SIZ EQU     *              SIZE OF UNIT SPECIFIC DATA TABLE PER UNIT
000.010                 271          XTEXT   IOCDEF



                        273X **      I/O CHANNEL DEFINITIONS.
                        274X
000.000                 275X         ORG     0
                        276X
000.000                 277X IOC.LNK DS      2              ADDRESS OF NEXT CHANNEL; =0 IF LAST
000.002                 278X IOC.DDA DS      2              THREAD JUMP TO DEVICE DRIVER (VIA DEV TABLE)
                        279X
000.004                 280X IOC.FLG DS      1              FILE TYPE FLAGS
```

```
000.001                281X FT.DD   EQU     00000001B       =1 IF DIRECTORY DEVICE
000.002                282X FT.OR   EQU     00000010B       =1 IF OPEN FOR READ
000.004                283X FT.OW   EQU     00000100B       =1 IF OPEN FOR WRITE
000.010                284X FT.OU   EQU     00001000B       =1 IF OPEN FOR UPDATE
000.020                285X FT.OC   EQU     00010000B       =1 IF OPEN FOR CHARACTER MODE   /80.02.GC/
000.003                286X IOC.SQL EQU     *-IOC.DDA       LENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
                       287X
000.005                288X IOC.GRT DS      2               ADDRESS OF GROUP RESERVATION TABLE
000.007                289X IOC.SPG DS      1               SECTORS PER GROUP, THIS DEVICE
000.010                290X IOC.CGN DS      1               CURRENT GROUP NUMBER
000.011                291X IOC.CSI DS      1               CURRENT SECTOR INDEX (IN CURRENT GROUP)
000.012                292X IOC.LGN DS      1               LAST GROUP NUMBER
000.013                293X IOC.LSI DS      1               LAST SECTOR INDEX (IN LAST GROUP)
000.010                294X IOC.DRL EQU     *-IOC.FLG       LENGTH OF INFO NORMALLY COPIED BACK TO
                       295X *                               THE CHANNEL TABLE
000.014                296X IOC.DTA DS      2               DEVICE TABLE ADDRESS FOR THIS DEVICE
000.016                297X IOC.DES DS      2               SECTOR NUMBER OF DIRECTORY ENTRY
000.020                298X IOC.DEV DS      2               DEVICE CODE
000.022                299X IOC.UNI DS      1               UNIT NUMBER (0-9)
000.021                300X IOC.DIL EQU     *-IOC.DDA       LENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)
                       301X
000.023                302X IOC.DIR DS      DIRELEN         DIRECTORY ENTRY
                       303X
000.052                304X IOCELEN EQU     *               IOC ENTRY LENGTH
                       305X
000.001                306X IOCCTD  EQU     1               INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)
000.052                307          XTEXT   DISDEF


                       309X **      DIRECTORY BLOCK FORMAT.
                       310X
000.000                311X         ORG     0
                       312X
000.000                313X DIS.ENT EQU     *               FIRST ENTRY ADDRESS
000.000                314X         DS      22*DIRELEN      22 DIRECTORY ENTRYS PER BLOCK
001.372                315X         DS      1               0 BYTE = END OF ENTRYS IN THIS BLOCK
                       316X
001.373                317X         ORG     512-5           AT END OF BLOCK
001.373                318X DIS.ENL DS      1               LENGTH OF EACH ENTRY (=DIRELEN)
001.374                319X DIS.SEC DS      2               BLOCK # OF THIS BLOCK,
001.376                320X DIS.LNK DS      2               BLOCK # OF NEXT BLOCK, =0 IF THIS IS LAST
002.000                321          XTEXT   FBDEF


                       323X **      FILE BLOCK DEFINITIONS.
                       324X
000.000                325X         ORG     0
000.000                326X FB.CHA  DS      1               CHANNEL NUMBER
000.001                327X FB.FLG  DS      1               FLAGS
000.002                328X FB.FWA  DS      2               BUFFER FWA
000.004                329X FB.PTR  DS      2               BUFFER POINTER
000.006                330X FB.LIM  DS      2               LIMIT OF DATA IN BUFFER (READ OPERATIONS)
000.010                331X FB.LWA  DS      2               LWA OF BUFFER
```

```
000.012                 332X FB.NAM  DS      4+8+4+1         NAME OF FILE
000.021                 333X FB.NAML EQU     *-FB.NAM
000.033                 334X FBENL   EQU     *               ENTRY LENGTH
000.033                 335          XTEXT   ECDEF


                        337X **      ERROR CODE DEFINITIONS.
                        338X
000.000                 339X         ORG     0
000.000                 340X         DS      1               NO ERROR #0
000.001                 341X EC.EOF  DS      1               END OF FILE
000.002                 342X EC.EOM  DS      1               END OF MEDIA
000.003                 343X EC.ILC  DS      1               ILLEGAL SYSCALL CODE
000.004                 344X EC.CNA  DS      1               CHANNEL NOT AVAILABLE
000.005                 345X EC.DNS  DS      1               DEVICE NOT SUITABLE
000.006                 346X EC.IDN  DS      1               ILLEGAL DEVICE NAME
000.007                 347X EC.IFN  DS      1               ILLEGAL FILE NAME
000.010                 348X EC.NRD  DS      1               NO ROOM FOR DEVICE DRIVER
000.011                 349X EC.FNO  DS      1               CHANNEL NOT OPEN
000.012                 350X EC.ILR  DS      1               ILLEGAL REQUEST
000.013                 351X EC.FUC  DS      1               FILE USAGE CONFLICT
000.014                 352X EC.FNF  DS      1               FILE NAME NOT FOUND
000.015                 353X EC.UND  DS      1               UNKNOWN DEVICE
000.016                 354X EC.ICN  DS      1               ILLEGAL CHANNEL NUMBER
000.017                 355X EC.DIF  DS      1               DIRECTORY FULL
000.020                 356X EC.IFC  DS      1               ILLEGAL FILE CONTENTS
000.021                 357X EC.NEM  DS      1               NOT ENOUGH MEMORY
000.022                 358X EC.RF   DS      1               READ FAILURE
000.023                 359X EC.WF   DS      1               WRITE FAILURE
000.024                 360X EC.WPV  DS      1               WRITE PROTECTION VIOLATION
000.025                 361X EC.WP   DS      1               DISK WRITE PROTECTED
000.026                 362X EC.FAP  DS      1               FILE ALREADY PRESENT
000.027                 363X EC.DDA  DS      1               DEVICE DRIVER ABORT
000.030                 364X EC.FL   DS      1               FILE LOCKED
000.031                 365X EC.FAO  DS      1               FILE ALREADY OPEN
000.032                 366X EC.IS   DS      1               ILLEGAL SWITCH
000.033                 367X EC.UUN  DS      1               UNKNOWN UNIT NUMBER
000.034                 368X EC.FNR  DS      1               FILE NAME REQUIRED
000.035                 369X EC.DIW  DS      1               DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036                 370X EC.UNA  DS      1               UNIT NOT AVAILABLE
000.037                 371X EC.ILV  DS      1               ILLEGAL VALUE
000.040                 372X EC.ILO  DS      1               ILLEGAL OPTION
000.041                 373X EC.VPM  DS      1               VOLUME PRESENTLY MOUNTED ON DEVICE
000.042                 374X EC.NVM  DS      1               NO VOLUME PRESENTLY MOUNTED
000.043                 375X EC.FOD  DS      1               FILE OPEN ON DEVICE
000.044                 376X EC.NPM  DS      1               NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045                 377X EC.DNI  DS      1               DISK NOT INITIALIZED
000.046                 378X EC.DNR  DS      1               DISK IS NOT READABLE
000.047                 379X EC.DSC  DS      1               DISK STRUCTURE IS CORRUPT
000.050                 380X EC.NCV  DS      1               NOT CORRECT VERSION OF HDOS
000.051                 381X EC.NOS  DS      1               NO OPERATING SYSTEM MOUNTED
000.052                 382X EC.IOI  DS      1               ILLEGAL OVERLAY INDEX
000.053                 383X EC.OTL  DS      1               OVERLAY TO LARGE
000.054                 384          XTEXT   HOSEQU
```

```
                    386X **     HDOS SYSTEM EQUIVALENCES.
                    387X *
                    388X
024,000             389X S.GRT0 EQU     24000A          SYSTEM AREA FOR  GRT0
025.000             390X S.GRT1 EQU     25000A          SYSTEM AREA FOR  GRT1
026.000             391X S.GRT2 EQU     26000A          SYSTEM AREA FOR  GRT2
                    392X
030.000             393X ROMBOOT EQU    30000A          ROM BOOT ENTRY
                    394X
040.100             395X        ORG     40100A          FREE SPACE FROM PAM-8
                    396X
040.100             397X        DS      8               JUMP TO SYSTEM EXIT
040.110             398X D.CON  DS      16              DISK CONSTANTS
040.130             399X SYDD   EQU     *               SYSTEM DISK ENTRY POINT
040.130             400X D.VEC  DS      24*3            SYSTEM ROM ENTRY VECTORS
040.240             401X D.RAM  DS      31              SYSTEM ROM WORK AREA
040.277             402X S.VAL  DS      36              SYSTEM VALUES
040.343             403X S.INT  DS      115             SYSTEM INTERNAL WORK AREAS
041.126             404X        DS      16
041.146             405X S.SOVR DS      2               STACK OVERFLOW WARNING
041.150             406X        DS      42200A-*        SYSTEM STACK
001.032             407X STACKL EQU     *-S.SOVR        STACK SIZE
                    408X
042.200             409X STACK  EQU     *               LWA+1 SYSTEM STACK
042.200             410X USERFWA EQU    *               USER FWA
042.200             411         XTEXT   HOSDEF


                    413X **     HOSDEF - DEFINE HOS PARAMETER.
                    414X *
                    415X
                    416X
000.040             417X VERS   EQU     2*16+0          VERSION 2.0
                    418X
000.377             419X SYSCALL EQU    377Q            SYSCALL INSTRUCTION
                    420X
                    421X
000.000             422X        ORG     0
                    423X
                    424X *      RESIDENT FUNCTIONS
                    425X
000.000             426X .EXIT  DS      1               EXIT (MUST BE FIRST)
000.001             427X .SCIN  DS      1               SCIN
000.002             428X .SCOUT DS      1               SCOUT
000.003             429X .PRINT DS      1               PRINT
000.004             430X .READ  DS      1               READ
000.005             431X .WRITE DS      1               WRITE
000.006             432X .CONSL DS      1               SET/CLEAR CONSOLE OPTIONS
000.007             433X .CLRCO DS      1               CLEAR CONSOLE BUFFER
000.010             434X .LOADO DS      1               LOAD AN OVERLAY
000.011             435X .VERS  DS      1               RETURN HDOS VERSION NUMBER
000.012             436X .SYSRES DS     1               PRECEDING FUNCTIONS ARE RESIDENT
                    437X
                    438X
                    439X *      *HDOSOVL0.SYS*  FUNCTIONS
```

```
                    440X
000.040             441X        ORG     40A
                    442X
000.040             443X .LINK  DS      1           LINK  (MUST BE FIRST)
000.041             444X .CTLC  DS      1           CTL-C
000.042             445X .OPENR DS      1           OPENR
000.043             446X .OPENW DS      1           OPENW
000.044             447X .OPENU DS      1           OPENU
000.045             448X .OPENC DS      1           OPENC
000.046             449X .CLOSE DS      1           CLOSE
000.047             450X .POSIT DS      1           POSITION
000.050             451X .DELET DS      1           DELETE
000.051             452X .RENAM DS      1           RENAME
000.052             453X .SETTP DS      1           SETTOP
000.053             454X .DECODE DS     1           NAME DECODE
000.054             455X .NAME  DS      1           GET FILE NAME FROM CHANNEL
000.055             456X .CLEAR DS      1           CLEAR CHAN
000.056             457X .CLEARA DS     1           CLEAR ALL CHANS
000.057             458X .ERROR DS      1           LOOKUP ERROR
000.060             459X .CHFLG DS      1           CHANGE FLAGS
000.061             460X .DISMT DS      1           FLAG SYSTEM DISK DISMOUNTED
000.062             461X .LOADD DS      1           LOAD DEVICE DRIVER
000.063             462X .OPEN  DS      1           Parametrized Open
                    463X
                    464X
                    465X *      *HDOSOVL1.SYS*  FUNCTIONS
                    466X
000.200             467X        ORG     200Q
                    468X
000.200             469X .MOUNT DS      1           MOUNT  (MUST BE FIRST)
000.201             470X .DMOUN DS      1           DISMOUNT
000.202             471X .MONMS DS      1           MOUNT/NO MESSAGE
000.203             472X .DMNMS DS      1           DISMOUNT/NO MESSAGE
000.204             473X .RESET DS      1           RESET = DISMOUNT/MOUNT OF UNIT
000.205             474X .CLEAN DS      1           Clean device
000.206             475X .DAD   DS      1           Dismount All Disks          /80.08.sc/
000.207             476         XTEXT   ASCII

                    478X **     ASCII CHARACTER EQUIVALENCES.
                    479X
000.015             480X CR     EQU     13          CARRIAGE RETURN
000.012             481X LF     EQU     10          LINE FEED
000.200             482X NULL   EQU     200Q        PAD CHARACTER
000.000             483X NUL2   EQU     0
000.007             484X BELL   EQU     7           BELL CHARACTER
000.177             485X RUBOUT EQU     177Q
000.010             486X BKSP   EQU     10Q         CTL-H
000.026             487X C.SYN  EQU     26Q         SYNC
000.002             488X C.STX  EQU     2           STX
000.047             489X QUOTE  EQU     47Q
000.011             490X TAB    EQU     11Q
000.033             491X ESC    EQU     33Q
000.012             492X NL     EQU     12Q         NEW LINE (HDOS SYSTEMS)
000.212             493X ENL    EQU     NL+200Q     NL + END-OF-LINE-FLAG
000.014             494X FF     EQU     14Q         FORM FEED
```

```
000.001                 495X CTLA    EQU     01Q             CTL-A
000.002                 496X CTLB    EQU     02Q             CTL-B
000.003                 497X CTLC    EQU     03Q             CTL-C
000.004                 498X CTLD    EQU     04Q             CTL-D
000.017                 499X CTLO    EQU     17Q             CTL-O
000.020                 500X CTLP    EQU     20Q             CTL-P
000.021                 501X CTLQ    EQU     21Q             CTL-Q
000.023                 502X CTLS    EQU     23Q             CTL-S
000.032                 503X CTLZ    EQU     32Q             CTL-Z
000.207                 504          XTEXT   ESINT


                        506X **      S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
                        507X *
                        508X *       THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
                        509X *       MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
                        510X
                        511X
040.343                 512X         ORG     S.INT
                        513X
                        514X **      CONSOLE STATUS FLAGS
                        515X
040.343                 516X S.CDB   DS      1               CONSOLE DESCRIPTOR BYTE
000.000                 517X CDB.H85 EQU     00000000B
000.001                 518X CDB.H84 EQU     00000001B       =0 IF H8-5, =1 IF H8-4
040.344                 519X S.BAUD  DS      2               [0-14]  H8-4 BAUD RATE, =0 IF H8-5
                        520X *                               [15]    =1 IF BAUD RATE => 2 STOP BITS
                        521X
                        522X **      TABLE ADDRESS WORDS
                        523X
040.346                 524X S.DLINK DS      2               ADDRESS OF DATA IN HDOS CODE
040.350                 525X S.OFWA  DS      2               FWA  OVERLAY  TABLE
040.352                 526X S.CFWA  DS      2               FWA  CHANNEL  TABLE
040.354                 527X S.DFWA  DS      2               FWA  DEVICE   TABLE
040.356                 528X S.RFWA  DS      2               FWA  RESIDENT HDOS CODE
                        529X
                        530X **      DEVICE DRIVER DELAYED LOAD FLAGS
                        531X
040.360                 532X S.DDLDA DS      2               DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
040.362                 533X S.DDLEN DS      2               CODE LENGTH IN BYTES
040.364                 534X S.DDGRP DS      1               GROUP NUMBER FOR DRIVER
040.365                 535X         DS      1               HOLD PLACE
                        536X *S.DDSEC        DS      2               SECTOR NUMBER FOR DRIVER ( * OBSOLETE ! * )
040.366                 537X S.DDDTA DS      2               DEVICE'S ADDRESS IN DEVLST +DEV.RES
040.370                 538X S.DDOPC DS      1               OPEN OPCODE PENDEDING
                        539X
                        540X **      OVERLAY MANAGEMENT FLAGS
                        541X
000.001                 542X OVL.IN  EQU     00000001B       IN MEMORY
000.002                 543X OVL.RES EQU     00000010B       PERMINANTLY RESIDENT
000.014                 544X OVL.NUM EQU     00001100B       OVERLAY NUMBER MASK
000.200                 545X OVL.UCS EQU     10000000B       USER CODE SWAPPED FOR OVERLAY
                        546X
040.371                 547X S.OVLFL DS      1               OVERLAY FLAG
```

```
040.372          548X S.UCSF  DS    2          FWA SWAPPED USER CODE
040.374          549X S.UCSL  DS    2          LENGTH SWAPPED USER CODE
040.376          550X S.OVLS  DS    2          SIZE OF OVERLAY CODE
041.000          551X S.OVLE  DS    2          ENTRY POINT OF OVERLAY CODE
                 552X
041.002          553X S.SSN   DS    2          SWAP AREA SECTOR NUMBER
041.004          554X S.OSN   DS    2          OVERLAY SECTOR NUMBER
                 555X
                 556X *       SYSCALL PROCESSING WORK AREAS
                 557X
041.006          558X S.CACC  DS    1          (ACC) UPON SYSCALL
041.007          559X S.CODE  DS    1          SYSCALL INDEX IN PROGRESS
                 560X
                 561X *       JUMPS TO ROUTINES IN RESIDENT HDOS CODE
                 562X
041.010          563X S.JUMPS DS    0          START OF DUMP VECTORS
041.010          564X S.SDD   DS    3          JUMP TO STAND-IN DEVICE DRIVER
041.013          565X S.FASER DS    3          JUMP TO FATSERR (FATAL SYSTEM ERROR)
041.016          566X S.DIREA DS    3          JUMP TO DIREAD (DISK FILE READ)
041.021          567X S.FCI   DS    3          JUMP TO FCI (FETCH CHANNEL INFO)
041.024          568X S.SCI   DS    3          JUMP TO SCI (STORE CHANNEL INFO)
041.027          569X S.GUP   DS    3          JUMP TO GUP (GET UNIT POINTER)
                 570X
041.032          571X S.MOUNT DS    1          <>0 IF THE SYSTEM DISK IS MOUNTED
041.033          572X S.DCS   DS    1          DEFAULT CLUSTER SIZE-1
                 573X
041.034          574X S.BOOTF DS    1          BOOT FLAGS
000.001          575X BOOT.P  EQU   00000001B   EXECUTE PROLOGUE UPON BOOTUP
                 576X
                 577X *       STACK VALUE SAVED FOR OVERLAY SYSCALLS
                 578X
041.035          579X S.OVSTK DS    2          VALUE OF SP UPON SYSCALLS USING OVERLAY
                 580X
041.037          581X         DS    1          RESERVED



                 583X **      ACTIVE I/O AREA.
                 584X *
                 585X *       THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION
                 586X *       CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM
                 587X *       THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.
                 588X *
                 589X *       NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY
                 590X *       FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE
                 591X *       8080 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY
                 592X *       COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND
                 593X *       BACKDATED AFTER PROCESSING.
                 594X
041.040          595X AIO.VEC DS    3          JUMP INSTRUCTION
041.041          596X AIO.DDA EQU   *-2         DEVICE DRIVER ADDRESS
041.043          597X AIO.FLG DS    1          FLAG BYTE
041.044          598X AIO.GRT DS    2          ADDRESS OF GROUP RESERV TABLE
041.046          599X AIO.SPG DS    1          SECTORS PER GROUP
041.047          600X AIO.CGN DS    1          CURRENT GROUP NUMBER
```

```
041.050                601X AIO.CSI DS      1              CURRENT SECTOR INDEX
041.051                602X AIO.LGN DS      1              LAST GROUP NUMBER
041.052                603X AIO.LSI DS      1              LAST SECTOR INDEX
041.053                604X AIO.DTA DS      2              DEVICE TABLE ADDRESS
041.055                605X AIO.DES DS      2              DIRECTORY SECTOR
041.057                606X AIO.DEV DS      2              DEVICE CODE
041.061                607X AIO.UNI DS      1              UNIT NUMBER (0-9)
                       608X
041.062                609X AIO.DIR DS      DIRELEN        DIRECTORY ENTRY
                       610X
041.111                611X AIO.CNT DS      1              SECTOR COUNT
041.112                612X AIO.EOM DS      1              END OF MEDIA FLAG
041.113                613X AIO.EOF DS      1              END OF FILE FLAG
041.114                614X AIO.TFP DS      2              TEMP FILE POINTERS
041.116                615X AIO.CHA DS      2              ADDRESS OF CHANNEL BLOCK (IOC.DDA)



041.120                617X S.BDA   DS      1              Boot Device Address (Setup by ROM) /80.09.sc/
041.121                618X S.SCR   DS      2              SYSTEM SCRATCH AREA ADDRESS
041.123                619          XTEXT   ESVAL



                       621X **      S.VAL - SYSTEM VALUE DEFINTIONS.
                       622X *
                       623X *       THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.
                       624X *
                       625X *       THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.
                       626X
                       627X
040.277                628X         ORG     S.VAL
                       629X
040.277                630X S.DATE  DS      9              SYSTEM DATE (IN ASCII)
040.310                631X S.DATC  DS      2              CODED DATE
040.312                632X S.TIME  DS      4              TIME FROM MIDNIGHT (IN TICS)
040.316                633X S.HIMEM DS      2              HARDWARE HIGH MEMORY ADRESS+1
                       634X
040.320                635X S.SYSM  DS      2              FWA RESIDENT SYSTEM
                       636X
040.322                637X S.USRM  DS      2              LWA USER MEMORY
                       638X
040.324                639X S.OMAX  DS      2              MAX OVERLAY SIZE FOR SYSTEM
                       640X
                       641X
                       642X **      THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL
                       643X
000.200                644X CSL.ECH EQU     10000000B      SUPPRESS ECHO
000.004                645X CSL.RAW EQU     00000100B      Raw Mode I/O                    /80.09.sc/
000.002                646X CSL.WRP EQU     00000010B      WRAP LINES AT WIDTH
000.001                647X CSL.CHR EQU     00000001B      OPERATE IN CHARACTER MODE
                       648X
000.000                649X I.CSLMD EQU     0              S.CSLMD IS FIRST BYTE
040.326                650X S.CSLMD DS      1              CONSOLE MODE
```

```
                    651X
000.200             652X CTP.BKS EQU     10000000B      TERMINAL PROCESSES BACKSPACES
000.100             653X CTP.FF  EQU     01000000B      Terminal Processes Form-Feed    /80.09.sc/
000.040             654X CTP.MLI EQU     00100000B      MAP LOWER CASE TO UPPER ON INPUT
000.020             655X CTP.MLO EQU     00010000B      MAP LOWER CASE TO UPPER ON OUTPUT
000.010             656X CTP.2SB EQU     00001000B      TERMINAL NEEDS TWO STOP BITS
000.002             657X CTP.BKM EQU     00000010B      MAP BKSP (UPON INPUT) TO RUBOUT
000.001             658X CTP.TAB EQU     00000001B      TERMINAL SUPPORTS TAB CHARACTERS
                    659X
000.001             660X I.CONTY EQU     1              S.CONTY IS 2ND BYTE
000.000             661X         ERRNZ   *-S.CSLMD-I.CONTY
040.327             662X S.CONTY DS      1              CONSOLE TYPE FLAGS
000.002             663X I.CUSOR EQU     2              S.CUSOR IS 3RD BYTE
000.000             664X         ERRNZ   *-S.CSLMD-I.CUSOR
040.330             665X S.CUSOR DS      1              CURRENT CURSOR POSITION
000.003             666X I.CONWI EQU     3              S.CONWI IS 4TH BYTE
000.000             667X         ERRNZ   *-S.CSLMD-I.CONWI
040.331             668X S.CONWI DS      1              CONSOLE WIDTH
                    669X
000.001             670X CO.FLG  EQU     00000001B      CTL-O FLAG
000.200             671X CS.FLG  EQU     10000000B      CTL-S FLAG
                    672X
000.004             673X I.CONFL EQU     4              S.CONFL IS 5TH BYTE
000.000             674X         ERRNZ   *-S.CSLMD-I.CONFL
040.332             675X S.CONFL DS      1              CONSOLE FLAGS
                    676X
040.333             677X S.CAADR DS      2              ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040.335             678X S.CCTAB DS      6              ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
040.343             679          XTEXT   DDDEF


                    681X **      DEVICE DRIVER COMMUNICATION FLAGS.
                    682X *
                    683X
000.000             684X         ORG     0
                    685X
000.000             686X DC.REA  DS      1              READ
000.001             687X DC.WRI  DS      1              WRITE
000.002             688X DC.RER  DS      1              READ REGARDLESS
000.003             689X DC.OPR  DS      1              OPEN FOR READ
000.004             690X DC.OPW  DS      1              OPEN FOR WRITE
000.005             691X DC.OPU  DS      1              OPEN FOR UPDATE
000.006             692X DC.CLO  DS      1              CLOSE
000.007             693X DC.ABT  DS      1              ABORT
000.010             694X DC.MOU  DS      1              MOUNT DEVICE
000.011             695X DC.LOD  DS      1              LOAD DEVICE DRIVER
000.012             696X DC.RDY  DS      1              Device Ready                    /80.04.GC/
000.013             697X DC.MAX  DS      1              MAXIMUM ENTRY INDEX
000.014             698          XTEXT   MTR
```

```
                        701X **     MTR - PAM/8 EQUIVALENCES.
                        702X *
                        703X *      THIS DECK CONTAINS SYMBOLIC DEFINITIONS USED TO
                        704X *      MAKE USE OF THE PAM/8 CODE AND CONTROL BYTES.


                        706X **     IO PORTS
                        707X
    000.360             708X IP.PAD  EQU    360Q          PAD INPUT PORT
    000.360             709X OP.CTL  EQU    360Q          CONTROL OUTPUT PORT
    000.360             710X OP.DIG  EQU    360Q          DIGIT SELECT OUTPUT PORT
    000.361             711X OP.SEG  EQU    361Q          SEGMENT SELECT OUTPUT PORT
    000.362             712X IP.CON  EQU    362Q          H-88/H-89/HA-8-8 Configuration  /80.07.sc/
    000.362             713X OP2.CTL EQU    362Q          H-88/H-89/HA-8-8 Control Port   /80.07.sc/


                        715X **     FRONT PANEL CONTROL BITS.                           /80.07.sc/
                        716X *
                        717X *      CB.*  set in OP.CTL
                        718X *      CB2.* set in OP2.CTL
                        719X *
                        720X
    000.020             721X CB.SSI  EQU    00010000B     SINGLE STEP INTERRUPT
    000.040             722X CB.MTL  EQU    00100000B     MONITOR LIGHT
    000.100             723X CB.CLI  EQU    01000000B     CLOCK INTERRUPT ENABLE
    000.200             724X CB.SPK  EQU    10000000B     SPEAKER ENABLE
                        725X
    000.001             726X CB2.SSI EQU    00000001B     Single Step Interrupt
    000.002             727X CB2.CLI EQU    00000010B     Clock Interrupt Enable
    000.040             728X CB2.ORG EQU    00100000B     ORG 0 Select
    000.100             729X CB2.SID EQU    01000000B     Side 1 Select


                        731X **     Secondary Control Bits
                        732X


                        734X **     MONITOR MODE FLAGS.
                        735X
    000.000             736X DM.MR   EQU    0             MEMORY READ
    000.001             737X DM.MW   EQU    1             MEMORY WRITE
    000.002             738X DM.RR   EQU    2             REGISTER READ
    000.003             739X DM.RW   EQU    3             REGISTER WRITE
```

```
                        741X **     USER OPTION BITS.
                        742X *
                        743X *      THESE BITS ARE SET IN CELL .MFLAG.
                        744X
    000.200             745X UO.HLT EQU     10000000B       DISABLE HALT PROCESSING
    000.100             746X UO.NFR EQU     CR.CLI          NO REFRESH OF FRONT PANEL
    000.002             747X UO.DDU EQU     00000010B       DISABLE DISPLAY UPDATE
    000.001             748X UO.CLK EQU     00000001B       ALLOW PRIVATE INTERRUPT PROCESSING


                        750X **     MONITOR IDENTIFICATION FLAGS
                        751X *
                        752X *      THESE BYTES IDENTIFY THE ROM MONITOR.
                        753X *      THEY ARE THE VARIOUS VALUES OF LOCATION .IDENT
                        754X
    000.021             755X M.PAM8 EQU     021Q            'LXI' INSTRUCTION AT 000.000 IN PAM-8
    000.303             756X M.FOX  EQU     303Q            'JMP' INSTRUCTION AT 000.000 IN FOX ROM


                        758X **     Confisuration Flass                                 /80.07.sc/
                        759X *
                        760X *      These bits are read in IP.CON.
                        761X *
                        762X
    000.003             763X CN.174M EQU    00000011B       Port 174Q Device-Type Mask
    000.014             764X CN.170M EQU    00001100B       Port 170Q Device-Type Mask
    000.020             765X CN.PRI EQU     00010000B       Primary/Secondary:  1=>Primary == 170Q
    000.040             766X CN.MEM EQU     00100000B       Memory Test/Normal Switch:  0=>Test; 1=>Normal
    000.100             767X CN.BAU EQU     01000000B       Baud Rate:  0=>9600; 1=>19,200
    000.200             768X CN.ABO EQU     10000000B       Auto-Boot:  1=>Auto-Boot
                        769X
    000.000             770X CND.H17 EQU    00B             H-17 Disk,              Valid only in CN.174M
    000.000             771X CND.NDI EQU    00B             No Device Installed,    Valid only in CN.170M
    000.001             772X CND.H47 EQU    01B             H-47 Disk


                        774X **     ROUTINE ENTRY POINTS.
                        775X *
                        776X
    000.000             777X .IDENT EQU     0000A           IDENTIFICATION LOCATION
    000.053             778X .DLY   EQU     0053A           DELAY
    001.267             779X .LOAD  EQU     1267A           TAPE LOAD
    001.374             780X .DUMP  EQU     1374A           TAPE DUMP
    002.136             781X .ALARM EQU     2136A           ALARM ROUTINE
    002.140             782X .HORN  EQU     2140A           HORN
    002.172             783X .CTC   EQU     2172A           CHECK TAPE CHECKSUM
    002.205             784X .TPERR EQU     2205A           TAPE ERROR ROUTINE
    002.264             785X .PCHL  EQU     2264A           PCHL INSTRUCTION
    002.265             786X .SRS   EQU     2265A           SCAN RECORD START
    002.325             787X .RNP   EQU     2325A           READ NEXT PAIR
    002.331             788X .RNB   EQU     2331A           READ NEXT BYTE
```

```
002.347                 789X .CRC    EQU     2347A           CRC-16 CALCULATOR
003.017                 790X .WNP    EQU     3017A           WRITE NEXT PAIR
003.024                 791X .WNB    EQU     3024A           WRITE NEXT BYTE
003.122                 792X .DOD    EQU     3122A           DECODE FOR OCTAL DISPLAY
003.260                 793X .RCK    EQU     3260A           READ CONSOLE KEYSET
003.356                 794X .DODA   EQU     3356A           SEGMENT CODE TABLE


                        796X **      RAM CELLS USED BY H8MTR.
                        797X *
                        798X
040.000                 799X .START  EQU     40000A          START DUMP ADDRESS
040.002                 800X .IOWRK  EQU     40002A          IN OR OUT INSTRUCTION
040.005                 801X .REGI   EQU     40005A          DISPLAYED REGISTER INDEX
040.006                 802X .DSPROT EQU     40006A          PERIOD FLAG BYTE
040.007                 803X .DSPMOD EQU     40007A          DISPLAY MODE
040.010                 804X .MFLAG  EQU     40010A          USER OPTION BYTE
040.011                 805X .CTLFLG EQU     40011A          PANEL CONTROL BYTE
040.013                 806X .ALEDS  EQU     40013A          ABUSS LEDS
040.021                 807X .DLEDS  EQU     40021A          DBUSS LEDS
040.024                 808X .ABUSS  EQU     40024A          ABUSS REGISTER
040.027                 809X .CRCSUM EQU     40027A          CRCSUM WORD
040.031                 810X .TPERRX EQU     40031A          TAPE ERROR EXIT VECTOR
040.033                 811X .TICCNT EQU     40033A          CLOCK TICK COUNTER
040.035                 812X .REGPTR EQU     40035A          REGISTER POINTER
040.037                 813X .UIVEC  EQU     40037A          USER INTERRUPT VECTORS
040.064                 814X .NMIRET EQU     40064A          H88/H89 NMI Return Address      /80.07.sc/
040.066                 815X .CTL2FL EQU     40066A          OP2.CTL Control Byte            /80.07.sc/
000.014                 816          XTEXT   DDFDEF


                        818X **      DIRECTORY DEVICE FORMAT DEFINITION.                     /80.09.sc/
                        819X *
                        820X *       Modified:       Sep-80
                        821X *                        No longer require 2 sectors per group
                        822X *                        Reserved Group Table dynamically allocated
                        823X *
                        824X
000.000                 825X         ORG     0
                        826X
000.000                 827X DDF.BOO DS      9               2K BOOT PROGRAM
000.011                 828X DDF.BOL EQU     *               LENGTH OF BOOT
000.011                 829X DDF.LAB DS      1               LABEL SECTOR
000.012                 830X DDF.USR DS      0               BEGINNING OF OPEN SPACE
000.012                 831          XTEXT   LABDEF
```

```
                        833X **      DISK LABEL SECTOR FORMATS.
                        834X
000.000                 835X         ORG     0
000.000                 836X LAB.SER DS      1              SERIAL NUMBER OF VOLUME
000.001                 837X LAB.IND DS      2              INITIALIZATION DATE
000.003                 838X LAB.DIS DS      2              SECTOR NUMBER OF 1ST DIRECTORY SECTOR
000.005                 839X LAB.GRT DS      2              INDEX OF GRT SECTOR
000.007                 840X LAB.SPG DS      1              SECTORS PER GROUP
                        841X
000.000                 842X LAB.DAT EQU     0              DATA VOLUME ONLY
000.001                 843X LAB.SYS EQU     1              SYSTEM VOLUME
000.002                 844X LAB.NOD EQU     2              => LAB.NOD MEANS VOLUME HAS NO DIRECTORY
                        845X
000.010                 846X LAB.VLT DS      1              VOLUME TYPE
000.011                 847X LAB.VER DS      1              VERSION OF INIT17 THAT INITED DISK
                        848X
000.012                 849X LAB.RGT DS      2              RGT sector number                /80.06.sc/
                        850X
000.014                 851X LAB.VPR EQU     *              Volume dependant data            /80.05.sc/
000.014                 852X LAB.SIZ DS      2               Volume Size (Bytes/256)         /80.05.sc/
000.016                 853X LAB.PSS DS      2               Physical Sector Size            /80.05.sc/
000.020                 854X LAB.VFL DS      1               Volume dependant Flags          /80.09.sc/
000.001                 855X VFL.NSD EQU     00000001B        Number of Sides:  1 => 2       /80.09.sc/
000.005                 856X LAB.VFL EQU     *-LAB.VPR      Lensth of volume dependant data  /80.05.sc/
                        857X
000.000                 858X         ERRMI   5-LAB.VPL                                       /80.05.sc/
000.021                 859X         DS      5-LAB.VPL      Reserved                         /80.05.sc/
                        860X
000.021                 861X LAB.LAB DS      60             LABEL
000.074                 862X LAB.LBL EQU     *-LAB.LAB      LABEL LENGTH
000.115                 863X         DS      2              Reserved for 0 bytes             /80.09.sc/
                        864X
000.117                 865X LAB.AUX EQU     *              Auxiliary Data                   /80.09.sc/
000.117                 866X LAB.SPT DS      1               Sectors per Track               /80.09.sc/
000.001                 867X LAB.AXL EQU     *-LAB.AUX      Lensth of Aux. Data              /80.09.sc/
000.120                 868          XTEXT   FILDEF


                        870X **      FILDEF - FILE TYPE DEFINITIONS.
                        871X *
                        872X *       DB      377Q,FT.XXX
                        873X
                        874X
000.000                 875X FT.ABS  EQU     0              ABSOLUTE BINARY
000.001                 876X FT.PIC  EQU     1              POSITION INDEPENDANT CODE
000.002                 877X FT.REL  EQU     2              RELOCATABLE CODE
000.003                 878X FT.BAC  EQU     3              COMPILED BASIC CODE
000.120                 879          XTEXT   ABSDEF
```

```
                        881X **      ABS FORMAT EQUIVALENCES.
                        882X
        000.000         883X         ORG     0
                        884X
        000.000         885X ABS.ID  DS      1          377Q = BINARY FILE FLAG
        000.001         886X         DS      1          FILE TYPE (FT.ABS)
        000.002         887X ABS.LDA DS      2          LOAD ADDRESS
        000.004         888X ABS.LEN DS      2          LENGTH OF ENTIRE RECORD
        000.006         889X ABS.ENT DS      2          ENTRY POINT
                        890X
        000.010         891X ABS.COD DS      0          CODE STARTS HERE
```

```
042.170                  894          ORG     USERFWA-ABS.COD
042.170  377 000         895          DB      377Q,FT.ABS
042.172  200 042         896          DW      USERFWA          LOAD ADDRESS
042.174  346 022         897          DW      MEML-USERFWA     SIZE
042.176  272 063         898          DW      ENTRY            ENTRY
                         899
000.000                  900          IF      ONECOPY
                         901
                         902  *       Since this code overlays PRS, it is included here              /2.0a/
                         903
042.200  315 056 047     904  PRS3    CALL    GETLAB           Get Label
042.203  330             905          RC
042.204  001 000 001     906          LXI     B,256
042.207  021 321 064     907          LXI     D,LABEL
042.212  041 321 063     908          LXI     H,SLABEL
042.215  315 252 030     909          CALL    $MOVE            Save Current Label
                         910
042.220  315 040 047     911          CALL    MND              Mount New Disk
042.223  332 026 053     912          JC      ERROR
042.226  072 321 064     913          LDA     LABEL+LAB.SER
042.231  062 113 063     914          STA     VOLSER           Set Current Volume Number
042.234  303 246 042     915          JMP     START
                         916
                         917          ENDIF
                         918
042.237                  919  PIP     EQU     *
                         920
                         921  *       COMMAND INTERPRETATION COMES HERE
                         922
042.237                  923  RESTART EQU     *
                         924
042.237  072 202 063     925          LDA     MODE
042.242  247             926          ANA     A
042.243  302 001 043     927          JNZ     EXIT             ENTERED WITH COMMAND, WILL NOW EXIT
042.246  061 200 042     928  START   LXI     SP,STACK         CLEAN STACK
042.251  315 257 042     929          CALL    PIP1             EXECUTE COMMAND
                         930
                         931  *       COMMANDS EXIT HERE IF NO ERRORS FOUND
                         932
042.254  303 237 042     933          JMP     RESTART
                         934
                         935  *       GET READY TO PROCESS COMMAND
                         936
042.257  315 042 057     937  PIP1    CALL    SDD              SET DEFAULT DEFAULT
                         938
                         939  *       CLEAR CHANNELS AND FILE BUFFER
                         940
042.262  377 056         941          DB      SYSCALL,.CLEARA  CLEAR CHANNELS
042.264  257             942          XRA     A
042.265  062 232 063     943          STA     DESTFB+FB.FLG    FLAG FILE NOT OPEN
                         944
                         945  *       CLEAR DYNAMIC BUFFERS
                         946
042.270  041 000 000     947          LXI     H,0
042.273  042 227 063     948          SHLD    BUFSIZ           EMPTY BUFFER
042.276  042 264 063     949          SHLD    NAMTLEN          CLEAR NAMTAB
```

```
042.301  042 266 063   950          SHLD    NAMTMAX         CLEAR NAMTAB AREA
042.304  041 114 066   951          LXI     H,BUFF
042.307  042 225 063   952          SHLD    BUFPTR          SET BUFFER AGAINST END OF NAMTAB
                       953
                       954  *       INPUT COMMAND LINE
                       955
042.312  315 172 057   956          CALL    $CCO            CLEAR CONTROL-O
042.315  072 202 063   957          LDA     MODE
042.320  247           958          ANA     A
042.321  314 256 043   959          CZ      ACL             ACCEPT COMMAND LINE (UNLESS WAS PASSED ONE BY CALLER)
042.324  332 001 043   960          JC      EXIT            EOF
042.327  041 374 065   961          LXI     H,LINE          (HL) = COMMAND ADDRESS
042.332  021 016 043   962          LXI     D,PIPA          (DE) = SWITCH LIST
000.000                963          ERRNZ   I.COP
042.335  257           964          XRA     A               (A) = $I.COP
042.336  062 201 063   965          STA     COMAND          ASSUME COPY COMMAND
042.341  062 204 063   966          STA     SUPRES          CLEAR /SUP FLAG
042.344  062 200 063   967          STA     ALLOCA          Clear /ALL flag                 /80.06.sc/
042.347  074           968          INR     A               FLAG NO /S FLAG
042.350  062 205 063   969          STA     SYSTEM          CLEAR /S FLAG
042.353  315 111 061   970          CALL    $DRS            DETECT AND REMOVE SWITCHES
042.356  332 026 053   971          JC      ERROR           ERROR
042.361  072 201 063   972          LDA     COMAND
042.364  315 061 031   973          CALL    $TJMP           PROCESS COMMAND
```

```
                                    975  **      COMMAND LIST
                                    976
042.367                             977  PIPB    DS      0               COMMAND PROCESSOR TABLE
000.000                             978  I.COP   EQU     *-PIPB/2                COMMAND INDEX
042.367  347 043                    979          DW      COPY
000.001                             980  I.LIS   EQU     *-PIPB/2                COMMAND INDEX
042.371  106 047                    981          DW      LIST
000.002                             982  I.BRE   EQU     *-PIPB/2                COMMAND INDEX
042.373  114 047                    983          DW      BRIEF           /BR
000.003                             984  I.VER   EQU     *-PIPB/2                COMMAND INDEX
042.375  116 052                    985          DW      VERSN           /V
000.004                             986  I.MOU   EQU     *-PIPB/2        /MOU,/M
042.377  302 043                    987          DW      MOUNT
000.001                             988          IF      .PIP.
                                    989  I.DEL   EQU     *-PIPB/2
                                    990          DW      DELETE          /DEL
                                    991  I.REN   EQU     *-PIPB/2
                                    992          DW      RENAME          /RE
                                    993  I.DIS   EQU     *-PIPB/2
                                    994          DW      DISMOU          /DIS
                                    995  I.RES   EQU     *-PIPB/2
                                    996          DW      RESET           /RES
                                    997          ENDIF
                                    998
                                    999  *       CTL-D HIT
                                   1000
043.001  257                       1001  EXIT    XRA     A
043.002  377 000                   1002          DB      SYSCALL,.EXIT   EXIT


                                   1004  **      CCHIT - CTL-C HIT
                                   1005  *
                                   1006  *       ENTRY   FROM SYSTEM
                                   1007
                                   1008
043.004  315 136 031               1009  CCHIT   CALL    $TYPTX
043.007  136 303                   1010          DB      '^',’C’+200Q
043.011  377 007                   1011          DB      SYSCALL,.CLRCO  CLEAR CONSOLE TYPEAHEAD
043.013  303 237 042               1012          JMP     RESTART         GET NEW COMMAND
```

```
                                1015  ***     SWITCH PROCESSING TABLES AND ROUTINES.
                                1016  *
                                1017  *       COMMAND SWITCHES ARE PROCESSED VIA THE ROUTINE $DRS, 'DECODE AND
                                1018  *       REMOVE SWITCHES'. $DRS IS SUPPLIED WITH A SWITCH DESCRIPTION
                                1019  *       TABLE, WHICH CONTAINS THE ADDRESSES OF ROUTINES
                                1020  *       WHICH ARE ENVOKED WHEN THE SWITCHES ARE ENCOUNTERED.
                                1021
                                1022
                                1023  **      SWITCH TABLE
                                1024
     043.016                    1025  PIPA    DS      0               FWA SWITCH TABLE
     000.001                    1026          IF      .PIP.
                                1027          DB      'DEL'           /DELETE
                                1028          DB      'E'+200Q,'T'+200Q,'E'+200Q,200Q
                                1029          DW      SW.DEL          PROCESSING ROUTINES
                                1030
                                1031          DB      'R'             /RENAME
                                1032          DB      'E'+200Q,'N'+200Q,'A'+200Q,'M'+200Q,'E'+200Q,200Q
                                1033          DW      SW.REN          PROCESS RENAME
                                1034
                                1035          DB      'DIS'           /DISMOUNT
                                1036          DB      'M'+200Q,'O'+200Q,'U'+200Q,'N'+200Q,'T'+200Q,200Q
                                1037          DW      SW.DIS
                                1038
                                1039          DB      'RES'           /RESET
                                1040          DB      'E'+200Q,'T'+200Q,200Q
                                1041          DW      SW.RES
                                1042          ENDIF
                                1043
     043.016  101 114 114      1044          DB      'ALL'           /ALLOCATE                       /80.06,sc/
     043.021  317 303 301      1045          DB      'O'+200Q,'C'+200Q,'A'+200Q,'T'+200Q,'E'+200Q,200Q /.06,sc/
     043.027  142 043          1046          DW      SW.ALL                                          /80.06,sc/
                                1047
     043.031  114              1048          DB      'L'             /LIST
     043.032  311 323 324      1049          DB      'I'+200Q,'S'+200Q,'T'+200Q,200Q
     043.036  223 043          1050          DW      SW.LIS          PROCESS LIST
                                1051
     043.040  102              1052          DB      'B'             /BRIEF
     043.041  322 311 305      1053          DB      'R'+200Q,'I'+200Q,'E'+200Q,'F'+200Q,200Q
     043.046  200 043          1054          DW      SW.BRE          PROCESS BRIEF
                                1055
     043.050  126              1056          DB      'V'             /VERSION
     043.051  305 322 323      1057          DB      'E'+200Q,'R'+200Q,'S'+200Q,'I'+200Q,'O'+200Q,'N'+200Q,200Q
     043.060  244 043          1058          DW      SW.VER          PROCESS VERSION
                                1059
     043.062  115 117 125      1060          DB      'MOU'           /MOUNT
     043.065  316 324 200      1061          DB      'N'+200Q,'T'+200Q,200Q
     043.070  251 043          1062          DW      SW.MOU
                                1063
     043.072  123              1064          DB      'S'             /SYSTEM
     043.073  331 323 324      1065          DB      'Y'+200Q,'S'+200Q,'T'+200Q,'E'+200Q,'M'+200Q,200Q
     043.101  150 043          1066          DW      SW.SYS          PROCESS SYSTEM
                                1067
     043.103  123 125          1068          DB      'SU'            /SUPRESS
     043.105  320 322 305      1069          DB      'P'+200Q,'R'+200Q,'E'+200Q,'S'+200Q,'S'+200Q,200Q
     043.113  155 043          1070          DW      SW.SUP
```

```
                                1071
043.115 112 107 114             1072            DB      'JGL'           /JGL INTERNAL SWITCH
043.120 200                     1073            DB      200Q
043.121 163 043                 1074            DW      SW.JGL
                                1075
043.123 000                     1076            DB      0               END OF TABLE
```

```
000.001                 1078            IF      .PIP.
                        1079    SW.DEL  SPACE   3,10
                        1080    **      SW.DEL - /DELETE SWITCH DETECTED.
                        1081
                        1082    SW.DEL  MVI     A,I.DEL
                        1083            JMP     SWIT1           IS MAJOR FUNCTION
                        1084    SW.REN  SPACE   3,10
                        1085    **      SW.REN - /RENAME SWITCH DETECTED.
                        1086
                        1087    SW.REN  MVI     A,I.REN
                        1088            JMP     SWIT1           IS MAJOR FUNCTION
                        1089    SW.DIS  SPACE   3,10
                        1090    **      SW.DIS  -  /DISMOUNT SWITCH DETECTED
                        1091
                        1092    SW.DIS  MVI     A,I.DIS
                        1093            JMP     SWIT1           IS MAJOR FUNCTION
                        1094    SW.RES  SPACE   3,10
                        1095    **      SW.RES  -  /RESET SWITCH DETECTED.
                        1096
                        1097    SW.RES  MVI     A,I.RES
                        1098            JMP     SWIT1           IS MAJOR FUNCTION
                        1099            ENDIF


                        1101    *       SWIT1 - PROCESS MAJOR FUNCTION SWITCH.
                        1102    *
                        1103    *       SWIT1 IS ENTERED TO PROCESS SWITCHES WHICH DETERMINE THE FUNCTION
                        1104    *       PIP IS TO PERFORM. I.E. 'VERB' SWITCHES, SUCH
                        1105    *       AS /DELETE (AS OPOSED TO 'MODIFIER' SWITCHES, LIKE /SYSTEM)
                        1106
043.124  001 201 063    1107    SWIT1   LXI     B,COMAND
043.127  365            1108            PUSH    PSW             SAVE COMMAND
043.130  012            1109            LDAX    B               (A) = PREVIOUS COMMAND
043.131  247            1110            ANA     A
043.132  076 204        1111            MVI     A,PEC.CS        CONTRADICTORY SWITCHES
043.134  302 026 053    1112            JNZ     ERROR           IF SO
043.137  361            1113            POP     PSW             (A) = NEW CODE
043.140  002            1114            STAX    B               STORE IT
043.141  311            1115            RET


                        1117    **      SW.ALL  - /ALLOCATE Switch Detected                     /80.06.sc/
                        1118
043.142  076 001        1119    SW.ALL  MVI     A,1
043.144  062 200 063    1120            STA     ALLOCA
043.147  311            1121            RET
```

```
                                1123  **      SW.SYS - /SYSTEM SWITCH DETECTED.
                                1124
043.150  257                   1125  SW.SYS  XRA     A               SET /S FLAG
043.151  062 205 063           1126          STA     SYSTEM
043.154  311                   1127          RET


                                1129  **      SW.SUP - /SUPPRESS SWITCH.
                                1130
                                1131
043.155  076 001               1132  SW.SUP  MVI     A,1
043.157  062 204 063           1133          STA     SUPRES
043.162  311                   1134          RET


                                1136  **      SW.JGL - /JGL SYSTEM SWITCH.
                                1137
                                1138
043.163  076 001               1139  SW.JGL  MVI     A,1
043.165  062 203 063           1140          STA     JGL
043.170  076 103               1141          MVI     A,'C'
043.172  062 110 052           1142          STA     PFIB1           SET 'C' CHARACTER FOR FLAGS DISPLAY
043.175  303 150 043           1143          JMP     SW.SYS


                                1145  **      SW.BRE - /BRIEF SWITCH DETECTED.
                                1146
043.200  072 201 063           1147  SW.BRE  LDA     COMAND          ALLOW TO SUPERCEDE /LIST
043.203  247                   1148          ANA     A
043.204  312 215 043           1149          JZ      SW.BRE1         NO OTHER COMMAND
000.000                         1150          ERRNZ   I.LIS-1
043.207  075                   1151          DCR     A
043.210  076 204               1152          MVI     A,PEC.CS        ASSUME CONTRADICTORY SWITCHES
043.212  302 026 053           1153          JNZ     ERROR
043.215  076 002               1154  SW.BRE1 MVI     A,I.BRE         IS /BREIF
043.217  062 201 063           1155          STA     COMAND
043.222  311                   1156          RET


                                1158  **      SW.LST - /LIST SWITCH DETECTED.
                                1159
043.223  072 201 063           1160  SW.LIS  LDA     COMAND
043.226  247                   1161          ANA     A
043.227  312 236 043           1162          JZ      SW.LIS1         NO FUNCTION
000.000                         1163          ERRNZ   I.BRE-2
000.000                         1164          ERRNZ   I.LIS-1
043.232  326 003               1165          SUI     3
043.234  077                   1166          CMC
043.235  320                   1167          RNC                     ALREADY HAVE ONE SPECIFIED, I.BRE OVERRULES
043.236  076 001               1168  SW.LIS1 MVI     A,I.LIS         /LIST
```

```
043.240  062 201 063  1169          STA     COMAND
043.243  311         1170          RET


                      1172  **      SW.VER  - /VERSION SWITCH DETECTED
                      1173
043.244  076 003     1174  SW.VER  MVI     A,I.VER
043.246  303 124 043 1175          JMP     SWIT1


                      1177  **      SW.MOU  - /MOUNT SWITCH DETECTED
                      1178
043.251  076 004     1179  SW.MOU  MVI     A,I.MOU
043.253  303 124 043 1180          JMP     SWIT1
```

```
                                1184  ***     ACL - ACCEPT COMMAND LINE.
                                1185  *
                                1186  *       ACL PROMPTS FOR AND READS A COMMAND LINE FROM
                                1187  *       THE CONSOLE.
                                1188  *
                                1189  *       ENTRY   NONE
                                1190  *       EXIT    'C' CLEAR, GOT LINE
                                1191  *                'LINE' = COMMAND LINE
                                1192  *                'C' SET IF EOF
                                1193  *       USES    ALL
                                1194
                                1195
043.256  315 207 057            1196  ACL     CALL    $GNL            GUARANTEE NEW LINE
043.261  315 136 031            1197          CALL    $TYPTX
000.001                         1198          IF      .PIP.
                                1199          DB      ':P',':'+200Q
                                1200          ELSE    ONECOPY
043.264  072 117 103            1201          DB      ':OC',':'+200Q
                                1202          ENDIF
043.270  257                    1203          XRA     A
043.271  062 326 040            1204          STA     S.CSLMD         CLEAR SPECIAL MODES
043.274  041 374 065            1205          LXI     H,LINE
043.277  303 254 057            1206          JMP     $RTL.           READ UPPER CASE LINE AND EXIT
```

```
000.001  1209          IF      .PIP.           PIP USES 'COPY'
         1210  ***     COPY - PROCESS COPY COMMAND.
         1211  *
         1212  *       SYNTAX:
         1213  *
         1214  *       DEST=SOURCE1,...,SOURCEN
         1215  *
         1216  *       D'DEST' IS THE DESTINATION FILE DESIGNATOR. IF NULL
         1217  *       (IN WHICH CASE THE '=' MAY BE OMITTED) IT DEFAULTS  TO
         1218  *               KB:PIPDEST.JGL
         1219  *
         1220  *       THE 'SOURCE' FIELDS ARE THE SOURCE FILE DESIGNATORS. WILDCARDS
         1221  *       MAY BE USED FOR FILE NAME AND EXTENSION.
         1222  *       IF NO WILDCARDS ARE USED IN THE DESTINATION, MULTIPLE SOURCE FILES
         1223  *       ARE CONCATINATED TOGETHER.
         1224  *
         1225  *       IF WILDCARDS ARE PRESENT IN THE DESTINATION FILE DESCRIPTION,
         1226  *       THE SOURCE FILES ARE COPIED TO INDIVIDUAL OUTPUT FILES. THE
         1227  *       NAMES OF THE OUTPUT FILES ARE CREATED BY FILLING
         1228  *       THE 'WILD' SPOTS IN THE DESTINATION NAME WITH THE CORRESPONDING
         1229  *       CHARACTERS IN THE SOURCE NAME.
         1230
         1231
         1232  COPY    EQU     *
         1233          XRA     A
         1234          STA     COPYC           CLEAR FILE COUNT
         1235          CALL    DDF             DECODE DESTINATION FILE
         1236          JC      ERROR           ERROR
         1237          STA     COPYA           SAVE DESTIONATION TYPE
         1238          CALL    SDD             RESET DEFAULT DEFAULTS
         1239          XRA     A               ALLOW *.*
         1240          CALL    BSL             BUILD SOURCE FILE LIST
         1241          JC      ERROR
         1242          CALL    $MOVEL
         1243          DW      COPYDL
         1244          DW      DESTFB+FB.NAM
         1245          DW      COPYD           SAVE WILDCARD DESTINATION
         1246
         1247  *       HAVE DESTINATION AND SOURCE FILE NAMES. DO THE COPYING.
         1248  *
         1249  *       IF NO DESTINATION WILD CARDS, THUS COPIING TO A SINGLE OUTPUT
         1250  *       FILE, OPEN THAT FILE NOW.
         1251
         1252          LDA     COPYA
         1253          ANA     A
         1254          JZ      COPY1           IS WILDCARDED
         1255          LXI     H,DESTFB+FB.NAM
         1256          MVI     A,CN.DES        (A) = DESTINATION CHANNEL
         1257          DB      SYSCALL,.OPENW          OPEN IT
         1258          LXI     H,DESTFB
         1259          JC      $FERROR         IF ERROR
         1260
         1261  *       OPEN NEXT SOURCE FILE
         1262
         1263  COPY1   LHLD    NAMTLEN
         1264          MOV     A,H
```

```
1265                    ORA     L
1266                    JZ      COPY5           NO MORE INPUT FILES
1267                    LXI     H,COPYC
1268                    INR     M               COUNT FILE
1269                    LXI     H,NAMTAB        (HL) = NAME ADDRESS
1270                    MVI     A,CN.SOU        SOURCE CHANNEL
1271                    DB      SYSCALL,.OPENR  OPEN FOR READ
1272                    JC      NAMERR          IF ERROR
1273
1274    *               OPEN DESTINATION FILE IFF WILDCARDS
1275
1276                    LDA     COPYA
1277                    ANA     A
1278                    JNZ     COPY2           NOT WILDCARDS
1279                    LXI     B,COPYD         (BC) = WILDCARD PATTERN ADDRESS
1280                    LXI     D,NAMTAB        (DE) = SOURCE NAME
1281                    LXI     H,DESTFB+FB.NAM (HL) = RESULT AREA
1282                    PUSH    H               SAVE POINTER TO RESULT AREA
1283                    CALL    MWN             MERGE WILDCARD NAME
1284                    POP     H               (HL) = $DESTFB+FB.NAM
1285                    MVI     A,CN.DES
1286                    DB      SYSCALL,.OPENW
1287                    LXI     H,DESTFB
1288                    JC      $FERROR         CANT GET FILE OPEN
1289
1290    *               INPUT AND OUTPUT FILES OPEN, COPY
1291
1292    COPY2   CALL    EBM             EXPAND BUFFER TO MAX SIZE
1293    COPY3   LHLD    BUFSIZ
1294                    MOV     B,H
1295                    MOV     C,L             (BC) = LENGTH OF BUFFER
1296                    LHLD    BUFPTR
1297                    XCHG                    (DE) = BUFFER FWA
1298                    MVI     A,CN.SOU
1299                    PUSH    D
1300                    DB      SYSCALL,.READ
1301                    POP     D               (DE) = BUFFER FWA
1302                    PUSH    PSW
1303                    JNC     COPY4           GOT IT ALL
1304                    CPI     EC.EOF
1305                    JE      COPY4           IS EOF
1306                    POP     PSW             RESTORE ERROR CODE
1307                    JMP     NAMERR
1308
1309    COPY4   LDA     BUFSIZ+1        (A) = # OF SECTORS IN BUFFER
1310                    SUB     B
1311                    MOV     B,A             (B) = SECTORS READ
1312                    MVI     C,0
1313                    MVI     A,CN.DES
1314                    DB      SYSCALL,.WRITE  WRITE IT OUT
1315                    LXI     H,DESTFB
1316                    JC      $FERROR         ERROR ON WRITE
1317                    POP     PSW             (PSW) = STATUS FROM READ
1318                    JNC     COPY3           NOT EOF
1319                    CALL    SBE             SHRINK BUFFER TO MINIMUM SIZE
1320                    MVI     A,CN.SOU
```

```
1321                    DB      SYSCALL,.CLOSE  CLOSE SOURCE
1322                    JC      NAMERR          ERROR ON CLOSE
1323                    CALL    REN             REMOVE ENTRY FROM NAMTAB
1324
1325    *       IF DOING INDIVIDUAL FILE COPIES, CLOSE OUTPUT FILE.
1326
1327                    LDA     COPYA
1328                    ANA     A
1329                    JNZ     COPY1           CONCATINATING
1330                    MVI     A,CN.DES
1331                    DB      SYSCALL,.CLOSE  CLOSE DESTINATION
1332                    LXI     H,DESTFB
1333                    JC      $FERROR         ERROR ON CLOSE
1334                    JMP     COPY1           GET NEXT FILE
1335
1336    **      ALL COPIES COMPLETE. CLOSE FILES AND CLEAN UP
1337
1338    COPY5   LDA     COPYC
1339                    ANA     A
1340                    JNZ     COPY6
1341
1342    *       NO FILES COPIED
1343
1344                    CALL    $TYPTX
1345                    DB      BELL,'No Files Copied',ENL
1346                    MVI     A,CN.DES
1347                    DB      SYSCALL,.CLEAR  CLEAR CHANNEL
1348                    RET
1349
1350    COPY6   MVI     B,0             (BC) = COUNT OF FILES COPIED
1351                    MOV     C,A
1352                    LDA     COPYA
1353                    ANA     A
1354                    JZ      COPY7           WILDCARDED
1355                    PUSH    B               SAVE COUNT
1356                    MVI     A,CN.DES
1357                    DB      SYSCALL,.CLOSE  CLOSE DESTINATION
1358                    POP     B               (BC) = FILES COPIED COUNT
1359                    LXI     H,DESTFB
1360                    JC      $FERROR         ERROR ON CLOSE
1361
1362    *       TYPE FILE COUNT
1363
1364    COPY7   LDA     SUPRES
1365                    ANA     A
1366                    RNZ                     SUPPRESS TRAIL MESSAGE
1367                    MVI     A,3
1368                    LXI     H,COPYE
1369                    CALL    $UDDN           UNPACK COUNT INTO MESSAGE
1370                    CALL    $TYPTX
1371                    DB      NL
1372    COPYE   DB      'XXX'
1373                    DB      ' Files Copied',ENL
1374                    RET
1375
1376    COPYA   DB      0               DESTINATION FILE WILDCARD FLAG (=0 IF WC)
```

```
1377  COPYC   DB      0               FILES COPIED COUNT
1378  COPYD   DS      FB.NAML         HOLD AREA FOR WILDCARD DESTINATION
1379  COPYDL  EQU     *-COPYD
1380          STL     'MOUNT  -  MOUNT A NEW DISK'
1381          EJECT
1382  ***     MOUNT   -  MOUNT A NEW DISK
1383  *
1384  *       MOUNT MOUNTS A NEW DISK ON THE SPECIFIED UNIT OF THE SELECTED
1385  *       DEVICE.
1386  *
1387  *       DEV:/MOUCNT]
1388  *
1389
1390  MOUNT   EQU     *
1391          MVI     A,.MOUNT
1392          CALL    MDR.            MOUNT/DISMOUNT/RESET
1393          RET
1394          STL     'DISMOU -  DISMOUNT CURRENT DISK'
1395          EJECT
1396  DISMOU  SPACE   4,10
1397  ***     DISMOU  -  DISMOUNT CURRENT DISK
1398  *
1399  *       DISMOU DISMOUNTS THE CURRENT DISK ON THE SPECIFIED UNIT OF THE
1400  *       SELECTED DEVICE.
1401  *
1402  *       DEV:/DISCMOUNT]
1403  *
1404
1405  DISMOU  EQU     *
1406          MVI     A,.DMOUN
1407          CALL    MDR.            MOUNT/DISMOUNT/RESET
1408          RET
1409          STL     'RESET  -  RESET CURRENT DISK'
1410          EJECT
1411  RESET   SPACE   4,10
1412  ***     RESET   -  RESET THE CURRENT DISK
1413  *
1414  *       RESET RESETS THE SPECIFIED UNIT OF THE SELECTED DEVICE BY ISSUING
1415  *       THE HDOS RESET CALL, WHICH IN TURN ISSUES A DISMOUNT AND MOUNT
1416  *       ASKING THE USER TO OPEN THE DRIVE IN BETWEEN THE TWO.
1417  *
1418  *       DEV:/RESCET]
1419  *
1420
1421  RESET   EQU     *
1422          MVI     A,.RESET
1423          CALL    MDR.            MOUNT/DISMOUNT/RESET
1424          RET
1425  MDR.    SPACE   4,10
1426  **      MDR.    -  MOUNT/DISMOUNT/RESET
1427  *
1428  *       MDR. PERFORMS THE SIMILAR FUNCTIONS OF MOUNT, DISMOUNT, AND RESET.
1429  *
1430  *
1431  *       ENTRY   (A)     =  SYSCALL CODE FOR OPERATION TO BE PERFORMED
1432  *
```

```
1433  *       EXIT    IF  NO ERROR
1434  *                   TO CALLER
1435  *                 ELSE
1436  *                   TO ERROR
1437  *
1438  *       USES    ALL
1439  *
1440
1441  MDR.    STA     MDRA            STORE SYSCALL VALUE
1442          CALL    CTS             CHECK FOR TARGET FILE SPECIFICATION
1443          STC
1444          JNZ     ERROR           THERE WAS A TARGET FILE
1445          LXI     H,LINE
1446          CALL    $DTB            DELETE TRAILING BLANKS
1447          CPI     1               (A) = LINE LENGTH INCLUDING <00> BYTE
1448          MVI     A,EC.DF         DEVICE FORMAT ERROR
1449          JZ      ERROR           NULL DEVICE IS ILLEGAL, ONLY BYTE IS NULL
1450  MDR1    PUSH    H               SAVE SPEC. ADDRESS FOR RETRY
1451          DB      SYSCALL,0
1452  MDRA    EQU     *-1             SYSCALL VALUE
1453          POP     H
1454          RNC                     NO ERROR
1455          PUSH    H               SAVE SPEC. ADDRESS
1456          CPI     EC.NPM          NO PROVISIONS MADE FOR REMOUNT
1457          STC
1458          JNZ     ERROR           ALL ERRORS BUT 'EC.NPM' CONSIDERED FATAL
1459          MVI     A,OVL0
1460          DB      SYSCALL,.LOADO  LOAD *HDOSOVL0.SYS*
1461          JC      ERROR
1462          MVI     A,OVL1
1463          DB      SYSCALL,.LOADO  LOAD *HDOSOVL1.SYS*
1464          JC      ERROR           SYSCALL ERROR
1465          POP     H               RESTORE SPEC. ADDRESS
1466          JMP     MDR1            TRY AGAIN
1467          ELSE
```

```
                          1471  ***     MOUNT - MOUNT A DIFFERENT DISK.
                          1472  *
                          1473  *       MOUNT CAUSES A NEW DISK TO BE MOUNTED.
                          1474  *
                          1475  *       INSERT THE DISK IN SY0, THEN TYPE
                          1476  *
                          1477  *       /MOUNT
                          1478
                          1479
                          1480
043.302  041 321 063      1481  MOUNT   LXI     H,SLABEL                                /2.0a/
043.305  006 000          1482          MVI     B,0             Count of 256            /2.0a/
043.307  315 212 031      1483          CALL    $ZERO           Zero the old label      /2.0a/
                          1484
043.312  021 323 043      1485          LXI     D,MOUNTA
043.315  006 377          1486          MVI     B,377Q          OFF PERIODS
043.317  315 224 046      1487          CALL    MAD             MOUNT ALTERNATE DISK
043.322  311              1488          RET
                          1489
043.323  244 306 307      1490  MOUNTA  DB      244Q,306Q,307Q
043.326  012 111 156      1491          DB      NL,'Insert New Disk',':'+200Q
```

```
                                1495  ***     ONECOPY - COPY FILES BETWEEN TWO VOLUMES, WITH ONLY ONE
                                1496  *       DRIVE.
                                1497  *
                                1498  *       (AND FOR MY NEXT TRICK...)
                                1499  *
                                1500  *       OPECOPY COPIES FILES BETWEEN TWO VOLUMES BY ALTERNATING BETWEEN
                                1501  *       TWO PHASES, THE READ PHASE AND THE WRITE PHASE.
                                1502  *
                                1503  *       READ PHASE:
                                1504  *
                                1505  *       DURING THE READ PHASE, THE SOURCE DISK IS MOUNTED. SOURCE FILES ARE
                                1506  *       OPENED IN THE ORDER OF THEIR APPEARANCE. FOR EACH OPENED
                                1507  *       FILE, A 'FILE DESCRIPTOR NODE' *FDN* IS ADDED  TO  THE ACTIVE
                                1508  *       CHAIN. THEN, AS MUCH AS THE FILE AS POSSIBLE IS READ INTO MEMORY.
                                1509  *
                                1510  *       THE PROCESS CONTINUES UNTIL
                                1511  *               1) THERE IS NO MORE FREE RAM
                                1512  *               2) OR, THERE ARE NO MORE FILE DESCRIPTOR NODES IN THE FREE CHAIN
                                1513  *               3) OR, THERE ARE NO MORE FILES IN NAMTAB (INPUT FILE LIST)
                                1514  *
                                1515  *
                                1516  *       WRITE PHASE
                                1517  *
                                1518  *       DURING THE WRITE PHASE, THE DESTINATION DISK IS MOUNTED. THE NODES
                                1519  *       ARE TAKEN FROM THE ACTIVE CHAIN, AND PROCESSED. IF THE FILE HAD
                                1520  *       BEEN PARTIALLY WRITTEN THE LAST PASS, IT IS  RE-OPENED AND POSITIONED.
                                1521  *       IF THERE IS NOT MORE DATA TO READ FOR A PROCESSED
                                1522  *       NODE, IT IS REMOVED, AND THE CORRESPONDING ENTRY IN NAMTAB IS DELETED.
                                1523  *
                                1524  *       WRITE PHASE CONTINUES UNTIL
                                1525  *
                                1526  *               1) THERE ARE NO MORE FILE NODES IN THE ACTIVE LIST
                                1527  *               2) OR, THE FIRST (AND ONLY) ENTRY IN THE LIST HAS NO
                                1528  *                  MORE DATA IN MEMORY, BUT HAS NOT BEEN COMPLETELY READ.
                                1529
                                1530
043.347                         1531  COPY    EQU     *               CALLED 'COPY' BY MAINLINE CODE
043.347                         1532  OCOPY   EQU     *
043.347  315 173 046            1533          CALL    IFL             INITIALIZE  FDN LISTS
043.352  257                    1534          XRA     A
043.353  062 204 044            1535          STA     OCOPYC          CLEAR FILE COUNT
043.356  062 112 063            1536          STA     VOLFLAG         FLAG SOURCE  VOLUME MOUNTED
043.361  072 321 064            1537          LDA     LABEL+LAB.SER   A  = Volume Label                    /2.0a/
043.364  062 113 063            1538          STA     VOLSER          SET VOLUME SERIAL NUMBER
043.367  315 022 054            1539          CALL    DDF             DECODE DESTINATION FILE
043.372  332 026 053            1540          JC      ERROR           ERROR
043.375  062 203 044            1541          STA     OCOPYA          SAVE DESTIONATION TYPE
044.000  315 042 057            1542          CALL    SDD             RESET DEFAULT DEFAULTS
044.003  257                    1543          XRA     A               ALLOW *.*
044.004  315 253 053            1544          CALL    BSL             BUILD SOURCE FILE LIST
044.007  332 026 053            1545          JC      ERROR
044.012  315 044 061            1546          CALL    $MOVEL
044.015  021 000                1547          DW      OCOPYDL
044.017  243 063                1548          DW      DESTFB+FB.NAM
044.021  205 044                1549          DW      OCOPYD          SAVE WILDCARD DESTINATION
044.023  315 146 055            1550          CALL    EBM             EXPAND BUFFER TO MAX
```

```
                                1551
                                1552  *       MAKE SURE HE'S NOT TRYING TO CONCATINATE
                                1553
044.026  072 203 044           1554          LDA     OCOPYA
044.031  247                   1555          ANA     A
044.032  312 053 044           1556          JZ      OCOPY1          HAVE WILDCARDS
044.035  052 264 063           1557          LHLD    NAMTLEN         NO WILDCARDS, ONLY LET HIM SPEFICY ONE SOURCE
044.040  021 357 377           1558          LXI     D,-FB.NAML
044.043  031                   1559          DAD     D
044.044  174                   1560          MOV     A,H
044.045  265                   1561          ORA     L
044.046  076 210               1562          MVI     A,PEC.FCI       FILE CONCATINATION IS ILLEGAL
044.050  302 026 053           1563          JNZ     ERROR
                                1564
                                1565  *       START READ PHASE
                                1566
044.053  072 226 063           1567  OCOPY1  LDA     BUFPTR+1        (A) = BUFFER FWA/256
044.056  074                   1568          INR     A               ROUND UP TO NEXT PAGE
044.057  062 115 063           1569          STA     OBUFPTR         SET SECTOR BUFFER  FWA/256
044.062  072 112 063           1570          LDA     VOLFLAG
044.065  247                   1571          ANA     A
044.066  312 100 044           1572          JZ      OCOPY2          SOURCE IS MOUNTED
044.071  021 226 044           1573          LXI     D,OCOPYF
044.074  107                   1574          MOV     B,A             (B) = 377Q = PERIODS MASK
044.075  315 224 046           1575          CALL    MAD             MOUNT ALTERNATE DISK
044.100  315 277 044           1576  OCOPY2  CALL    RPH             READ PHASE
044.103  072 001 063           1577          LDA     FDNHEAD
044.106  247                   1578          ANA     A
044.107  312 137 044           1579          JZ      OCOPY6          NO FILES ARE READ, ERGO NONE ARE LEFT
044.112  072 112 063           1580          LDA     VOLFLAG
044.115  247                   1581          ANA     A
044.116  302 131 044           1582          JNZ     OCOPY3
044.121  006 177               1583          MVI     B,177Q          (B) = PERIODS MASK
044.123  021 250 044           1584          LXI     D,OCOPYG
044.126  315 224 046           1585          CALL    MAD             MOUNT ALTERNATE DISK
044.131  315 252 045           1586  OCOPY3  CALL    WPH             WRITE PHASE
044.134  303 053 044           1587          JMP     OCOPY1
                                1588
                                1589  *       ALL DONE, FINISH MESSAGE
                                1590
044.137  072 204 044           1591  OCOPY6  LDA     OCOPYC          (A) = FILE COUNT
044.142  006 000               1592          MVI     B,0             (BC) = COUNT OF FILES COPIED
044.144  117                   1593          MOV     C,A
                                1594
                                1595  *       TYPE FILE COUNT
                                1596
044.145  076 003               1597          MVI     A,3
044.147  041 161 044           1598          LXI     H,OCOPYE
044.152  315 371 060           1599          CALL    $UDDN           UNPACK COUNT INTO MESSAGE
044.155  315 136 031           1600          CALL    $TYPTX
044.160  012                   1601          DB      NL              for aesthetics                  /2.0a/
044.161  130 130 130           1602  OCOPYE  DB      'XXX'
044.164  040 106 151           1603          DB      ' Files Copied',ENL
044.202  311                   1604          RET
                                1605
044.203  000                   1606  OCOPYA  DB      0               DESTINATION FILE WILDCARD FLAG (=0 IF WC)
```

```
044.204  000          1607  OCOPYC   DB      0               FILES COPIED COUNT
044.205                1608  OCOPYD   DS      FB.NAML         HOLD AREA FOR WILDCARD DESTINATION
000.021                1609  OCOPYDL  EQU     *-OCOPYD
044.226  244 306 307   1610  OCOPYF   DB      244Q,306Q,307Q
044.231  012 111 156   1611           DB      NL,'Insert Source',':'+200Q
044.250  102 014 044   1612  OCOPYG   DB      102Q,014Q,44Q
044.253  012 111 156   1613           DB      NL,'Insert Destination',':'+200Q
```

```
                                1617  **      RPH - READ PHASE.
                                1618  *
                                1619  *       RPH  HANDLES THE READ PHASE OF THE COPY PROCESS.
                                1620  *
                                1621  *       IT IS ENTERED WITH THE NAMTAB AND FDN TABLE SETUP, AND
                                1622  *       WITH THE SOURCE DISK MOUNTED.
                                1623  *
                                1624  *       READ PHASE:
                                1625  *
                                1626  *       DURING THE READ PHASE, THE SOURCE DISK IS MOUNTED. SOURCE FILES ARE
                                1627  *       OPENED IN THE ORDER OF THEIR APPEARANCE. FOR EACH OPENED
                                1628  *       FILE, A 'FILE DESCRIPTOR NODE' *FDN* IS ADDED  TO  THE ACTIVE
                                1629  *       CHAIN. THEN, AS MUCH AS THE FILE AS POSSIBLE IS READ INTO MEMORY.
                                1630  *
                                1631  *       THE PROCESS CONTINUES UNTIL
                                1632  *               1) THERE IS NO MORE FREE RAM
                                1633  *               2) OR, THERE ARE NO MORE FILE DESCRIPTOR NODES IN THE FREE CHAIN
                                1634  *               3) OR, THERE ARE NO MORE FILES IN NAMTAB (INPUT FILE LIST)
                                1635  *
                                1636  *       ENTRY   NONE
                                1637  *       EXIT    NONE
                                1638  *       USES    ALL
                                1639
                                1640
044.277                         1641  RPH     EQU     *
                                1642
                                1643
                                1644  *       SEE IF ANY MEMORY TO HAVE
                                1645
044.277  315 165 046            1646          CALL    CBR             COMPUTE BUFFER ROOM
044.302  310                    1647          RZ                      NONE
                                1648
                                1649  *       SEE IF WE NEED TO READ SOME MORE INTO A PART-COPIED FILE
                                1650
044.303  041 001 063            1651          LXI     H,FDNHEAD
044.306  156                    1652          MOV     L,M             (HL) = ADDRESS IF FIRST NODE
044.307  175                    1653          MOV     A,L
044.310  247                    1654          ANA     A
044.311  312 326 044            1655          JZ      RPH1            IS NO FIRST NODE, ERGO NO FILE
044.314  043                    1656          INX     H
000.000                         1657          ERRNZ   FDN.STA-1
044.315  176                    1658          MOV     A,M             (A) = .STA
044.316  346 002                1659          ANI     ST.OPR
044.320  021 114 066            1660          LXI     D,NAMTAB
044.323  302 021 045            1661          JNZ     RPH2.5          FILE IS INCOMPLETELY READ
                                1662
                                1663  *       SEE IF ANY FREE FILE DESCRIPTOR NODES TO USE
                                1664
044.326  072 000 063            1665  RPH1    LDA     FDNFRE
044.331  247                    1666          ANA     A
044.332  310                    1667          RZ                      NO MORE
                                1668
                                1669  *       SEE IF THERE IS A FILE IN NAMTAB WITHOUT AN ENTRY IN FNDLIST.
                                1670  *       SINCE THE FIRST ENTRY IN FDNLIST CORRESPONDS TO THE FIRST IN
                                1671  *       NAMTAB, ETC., WE'LL JUST RUN DOWN FDNLIST UNTIL THE END, AND
                                1672  *       THE NEXT NAMTAB FILE WILL BE THE ONE WE WANT...
```

```
                          1673
044.333  001 021 000     1674          LXI     B,FB.NAML       (BC) = ENTRY SIZE IN NAMTAB
044.336  021 357 377     1675          LXI     D,-FB.NAML              (DE) = POINTER INTO NAMTAB
044.341  041 001 063     1676          LXI     H,FDNHEAD
044.344  175             1677          MOV     A,L             START WITH FDNHEAD
044.345  157             1678  RPH2    MOV     L,A             FOLLOW LINK
044.346  176             1679          MOV     A,M             (A) = NEXT NODE
044.347  353             1680          XCHG
044.350  011             1681          DAD     B               ADVANCE POINTER INTO NAMTAB
044.351  353             1682          XCHG
044.352  247             1683          ANA     A
044.353  302 345 044     1684          JNZ     RPH2            LINK SOME MORE
044.356  345             1685          PUSH    H               (HL) = ADDRESS OF LAST NODE
044.357  052 264 063     1686          LHLD    NAMTLEN
044.362  315 216 030     1687          CALL    $CDEHL          SEE IF HAVE ACCOUNTED FOR ALL NAMTAB ENTRYS
044.365  341             1688          POP     H
044.366  310             1689          RE                      FILES ALL USED UP
                         1690
                         1691  *       HAVE ROOM FOR DATA, HAVE A NODE FOR THE FILE COUNTS, AND
                         1692  *       HAVE A FILE NAME. ALL SET FOR BUSINESS..
                         1693  *
                         1694  *       (DE) = INDEX INTO NAMTAB FOR FILE
                         1695  *       (HL) = NODE ADDRESS OF LAST ENTRY IN LIST
                         1696  *
                         1697  *       CHAIN THE FIRST FREE NODE ONTO THE END OF THE LIST
                         1698
044.367  072 000 063     1699          LDA     FDNFRE
044.372  167             1700          MOV     M,A             CHAIN TO  NEW END  NODE
044.373  157             1701          MOV     L,A
044.374  176             1702          MOV     A,M             (A) =  NEXT NODE IN FREE CHAIN
044.375  062 000 063     1703          STA     FDNFRE
045.000  006 011         1704          MVI     B,FDNELEN
045.002  345             1705          PUSH    H               SAVE NODE ADDRESS
045.003  315 212 031     1706          CALL    $ZERO           ZERO ENTIRE NODE, ENCLUDING CHAIN (AT END,  NOW)
045.006  001 114 066     1707          LXI     B,NAMTAB
045.011  353             1708          XCHG
045.012  011             1709          DAD     B               (HL) = ADDRESS OF NAMTAB ENTRY
045.013  042 270 063     1710          SHLD    NAMTPTR         POINTER TO CURRENT NAMTAB ENTRY
045.016  353             1711          XCHG
045.017  341             1712          POP     H
000.000                  1713          ERRNZ   FDN.STA-1
045.020  043             1714          INX     H               (HL) = ADDR OF FDN.STA OF NODE
                         1715
                         1716  *       READY TO OPEN FILE
                         1717  *
                         1718  *       (DE) = NAMTAB ENTRY ADDRESS
                         1719  *       (HL) = $FDN.STA OF ENTRY
                         1720
045.021  345             1721  RPH2.5  PUSH    H               SAVE ADDRESS
045.022  353             1722          XCHG
045.023  257             1723          XRA     A
000.000                  1724          ERRNZ   CN.SOU          (A) = SOURCE CHANNEL NUMBER
045.024  377 042         1725          DB      SYSCALL,.OPENR  OPEN
045.026  332 175 052     1726          JC      NAMERR          ERROR
045.031  321             1727          POP     D
045.032  032             1728          LDAX    D               (A) = FDN.STA
```

```
045.033  346 002       1729          ANI     ST.OPR
045.035  325           1730          PUSH    D               SAVE ADDRESS
045.036  302 124 045   1731          JNZ     RPH3            ALREADY OPENED IN PREVIOUS PASSES
                       1732
                       1733  *       FIRST TIME THIS FILE HAS BEEN OPENED. SEE IF CONTIGUOUS
                       1734
045.041  345           1735          PUSH    H
045.042  041 204 044   1736          LXI     H,OCOPYC
045.045  064           1737          INR     M
045.046  341           1738          POP     H
045.047  032           1739          LDAX    D
045.050  366 002       1740          ORI     ST.OPR          SET OPEN FOR READ
045.052  022           1741          STAX    D
045.053  052 352 040   1742          LHLD    S.CFWA          (HL) = CHANNEL 0 FWA
000.000               1743          ERRNZ   IOCCTD-1        WE NEED TO CHAIN ONE TO GET TO USER #0
045.056  315 211 030   1744          CALL    $HLIHL
000.000               1745          ERRNZ   CN.SOU          ASSUME WE WANT CHANNEL 0
045.061  315 234 030   1746          CALL    $INDL
045.064  041 000       1747          DW      IOC.DIR+DIR.FLG
045.066  173           1748          MOV     A,E             (A) = DIR.FLG
045.067  346 000       1749          ANI     0 DIF.CNT       * * PATCH * *
045.071  312 124 045   1750          JZ      RPH3            NOT CONTIG
                       1751
                       1752  *       IS CONTIG, GET FILE SIZE
                       1753
045.074  315 234 030   1754          CALL    $INDL
045.077  005 000       1755          DW      IOC.GRT
045.101  325           1756          PUSH    D               SAVE GRT ADDRESS
045.102  315 234 030   1757          CALL    $INDL
045.105  043 000       1758          DW      IOC.DIR+DIR.FGN (E) = DIR.FGN
045.107  173           1759          MOV     A,E
045.110  341           1760          POP     H               (HL) = GRT TABLE ADDRESS
045.111  315 354 053   1761          CALL    CFS.            COMPUTE BLOCK SIZE
045.114  341           1762          POP     H               (HL) = ADDRESS OF FDN.STA
045.115  345           1763          PUSH    H
045.116  176           1764          MOV     A,M             (A) = FDN.STA
045.117  366 020       1765          ORI     ST.CNT          FLAG CONTIG
045.121  167           1766          MOV     M,A
045.122  043           1767          INX     H
000.000               1768          ERRNZ   FDN.SIZ-FDN.STA-1
045.123  163           1769          MOV     M,E             SET BLOCK COUNT
                       1770
                       1771  *       READY TO READ DATA. POSITION FILE (IN CASE SOME WAS READ IN
                       1772  *       PREVIOUS PASSES) AND COMPUTE THE MAX POSSIBLE READ COUNT
                       1773  *
                       1774  *       ((SP)) = ADDRESS OF FDN.STA FOR NODE
                       1775
045.124  341           1776  RPH3    POP     H               (HL) = ADDRESS OF FDN.STA
045.125  345           1777          PUSH    H
045.126  315 234 030   1778          CALL    $INDL
045.131  002 000       1779          DW      FDN.AMR-FDN.STA (DE) = AMOUNT READ (IN SECTORS)
045.133  102           1780          MOV     B,D
045.134  113           1781          MOV     C,E             (BC) = AMOUNT READ
045.135  076 000       1782          MVI     A,CN.SOU
045.137  377 047       1783          DB      SYSCALL,.POSIT          POSIT
045.141  332 227 052   1784          JC      IERR3           POSIT BLEW UP
```

```
045.144  315 165 046  1785          CALL    CBR             COMPUTE BUFFER ROOM
045.147  353          1786          XCHG                    (D) = POINTER/256, (E) = LIMIT/256
045.150  341          1787          POP     H               (HL) = #FDN.STA
045.151  001 006 000  1788          LXI     B,FDN.ADR-FDN.STA
045.154  011          1789          DAD     B               (HL) = #FDN.ADR
045.155  162          1790          MOV     M,D             SET ADDRESS/256
045.156  345          1791          PUSH    H               SAVE #FDN.ADR
045.157  036 000       1792          MVI     E,0             (DE) = ADDRESS
045.161  107          1793          MOV     B,A             (B) = SECTORS OF RAM AVAILABLE
045.162  113          1794          MOV     C,E             (C) = 0
045.163  305          1795          PUSH    B               SAVE TRY COUNT
045.164  076 000       1796          MVI     A,CN.SOU
045.166  377 004       1797          DB      SYSCALL,.READ   READ THE STUFF
                      1798
                      1799  *       COMPUTE THE AMOUNT READ (IN CASE OF EOF)
                      1800
045.170  321          1801          POP     D               (DE) = TRY COUNT
045.171  322 216 045  1802          JNC     RPH4            GOT  ALL WE TRYED
045.174  376 001       1803          CPI     EC.EOF
045.176  302 175 052  1804          JNE     NAMERR          NOT JUST EOF, GOT TROUBLES
045.201  172          1805          MOV     A,D
045.202  220          1806          SUB     B               REMOVE AMOUNT WE DIDNT GET
045.203  127          1807          MOV     D,A
045.204  341          1808          POP     H               (HL) = #FDN.ADR
045.205  345          1809          PUSH    H
045.206  001 372 377  1810          LXI     B,FDN.STA-FDN.ADR
045.211  011          1811          DAD     B
045.212  176          1812          MOV     A,M             (A) = FDN.STA
045.213  346 375       1813          ANI     377Q-ST.OPR     EOF, NOT OPEN FOR READ ANYMORE
045.215  167          1814          MOV     M,A             POST READ COMPLETE FOR THIS GUY
                      1815
                      1816  *       STORE RESULTS OF READ IN NODE
                      1817  *
                      1818  *       (D) = SECTORS READ
                      1819  *       ((SP)) = #FDN.ADR
                      1820
045.216  341          1821  RPH4    POP     H               (HL) = #FDN.ADR
045.217  043          1822          INX     H
000.000               1823          ERRNZ   FDN.AIM-FDN.ADR-1       (HL) = ADDRESS IF AMOUNT IN MEMORY BYTE
045.220  162          1824          MOV     M,D             STORE SECTORS IN MEMORY COUNT
045.221  001 373 377  1825          LXI     B,FDN.AMR-FDN.AIM
045.224  011          1826          DAD     B               (HL) = #FDN.AMR (AMOUNT READ)
045.225  176          1827          MOV     A,M             (A) = AMOUNT READ BEFORE
045.226  202          1828          ADD     D               ADD NEW AMOUNT
045.227  167          1829          MOV     M,A
045.230  043          1830          INX     H
045.231  176          1831          MOV     A,M
045.232  316 000       1832          ACI     0               PROPIGATE FOR VERY LARGE FILES
045.234  167          1833          MOV     M,A
045.235  041 115 063  1834          LXI     H,OBUFPTR
045.240  176          1835          MOV     A,M
045.241  202          1836          ADD     D               ADVANCE FREE RAM POINTER BY AMOUNT READ
045.242  167          1837          MOV     M,A
045.243  076 000       1838          MVI     A,CN.SOU
045.245  377 046       1839          DB      SYSCALL,.CLOSE  CLOSE FILE
045.247  303 277 044  1840          JMP     RPH             SEE IF MORE TO READ
```

```
                                1842  **      WPH - WRITE PHASE.
                                1843  *
                                1844  *       WPH HANDLES THE WRITE PHASE PROCESSING. IT IS ENTERED WITH
                                1845  *       THE FDN CHAIN SETUP, THE NAMTAB SETUP, AND
                                1846  *       THE DESTINATION DISK MOUNTED.
                                1847  *
                                1848  *
                                1849  *       WRITE PHASE
                                1850  *
                                1851  *       DURING THE WRITE PHASE, THE DESTINATION DISK IS MOUNTED. THE NODES
                                1852  *       ARE TAKEN FROM THE ACTIVE CHAIN, AND PROCESSED. IF THE FILE HAD
                                1853  *       BEEN PARTIALLY WRITTEN THE LAST PASS, IT IS  RE-OPENED AND POSITIONED.
                                1854  *       IF THERE IS NOT MORE DATA TO READ FOR A PROCESSED
                                1855  *       NODE, IT IS REMOVED, AND THE CORRESPONDING ENTRY IN NAMTAB IS DELETED.
                                1856  *
                                1857  *       WRITE PHASE CONTINUES UNTIL
                                1858  *
                                1859  *               1) THERE ARE NO MORE FILE NODES IN THE ACTIVE LIST
                                1860  *               2) OR, THE FIRST (AND ONLY) ENTRY IN THE LIST HAS NO
                                1861  *                  MORE DATA IN MEMORY, BUT HAS NOT BEEN COMPLETELY READ.
                                1862  *
                                1863  *       ENTRY   NONE
                                1864  *       EXIT    NONE
                                1865  *       USES    ALL
                                1866
                                1867
045.252                         1868  WPH     EQU     *
                                1869
                                1870  *       SEE IF MORE TO WRITE
                                1871
045.252  041 001 063           1872          LXI     H,FDNHEAD
045.255  156                   1873          MOV     L,M
045.256  175                   1874          MOV     A,L             (A) = FIRST NODE INDEX
045.257  247                   1875          ANA     A
045.260  310                   1876          RZ                      NO MORE
045.261  315 234 030           1877          CALL    $INDL
045.264  010 000               1878          DW      FDN.AIM         (E) = AMOUNT IN MEMORY FOR THIS GUY
045.266  173                   1879          MOV     A,E
045.267  247                   1880          ANA     A
045.270  302 305 045           1881          JNZ     WPH0            GOT DATA
                                1882
                                1883  *       NO DATA IN NODE. IF STILL READING, RETURN FOR MORE
                                1884
045.273  043                   1885          INX     H
045.274  176                   1886          MOV     A,M
045.275  053                   1887          DCX     H
045.276  346 002               1888          ANI     ST.OPR
045.300  300                   1889          RNZ                     STILL READING, GET MORE
045.301  353                   1890          XCHG                    (DE) = ADDRESS
045.302  303 126 046           1891          JMP     WPH4            REMOVE NODE, AM DONE WITH FILE
                                1892
                                1893  *       HAVE DATA TO WRITE. SEE IF WE HAVE OPENED THIS FILE BEFORE,,
                                1894  *       OR IF THIS IS THE FIRST TIME
                                1895
045.305  345                   1896  WPH0    PUSH    H               SAVE NODE POINTER
045.306  043                   1897          INX     H
```

```
000.000                 1898            ERRNZ   FDN.STA-1
045.307  176           1899            MOV     A,M             (A) = FDN.STA
045.310  346 001       1900            ANI     ST.OPW
045.312  302 021 046   1901            JNZ     WPH2            OPENED BEFORE
000.000                 1902            ERRNZ   ST.OPW-1
045.315  064           1903            INR     M               SET '1' BIT
                        1904
                        1905  *         BUILD NAME INTO DESTFB
                        1906
045.316  345           1907            PUSH    H               SAVE NODE ADDRESS
045.317  001 205 044   1908            LXI     B,OCOPYD
045.322  021 114 066   1909            LXI     D,NAMTAB
045.325  041 243 063   1910            LXI     H,DESTFB+FB.NAM
045.330  315 320 056   1911            CALL    MWN             MERGE WILDCARD NAME
045.333  341           1912            POP     H
                        1913
                        1914  *         IS 1ST TIME FOR THIS FILE. IF CONTIGUOUS FLAG, OPEN THE FILE
                        1915  *         FOR CONTIGUOUS
                        1916
045.334  176           1917            MOV     A,M             (A) = FLAG BYTE
045.335  346 020       1918            ANI     ST.CNT
045.337  302 357 045   1919            JNZ     WPH1            IS CONTIG
045.342  041 243 063   1920            LXI     H,DESTFB+FB.NAM
045.345  076 001       1921            MVI     A,CN.DES
045.347  377 043       1922            DB      SYSCALL,.OPENW  JUST OPEN FOR WRITE
045.351  332 207 052   1923            JC      DESTERR         ERROR
045.354  303 053 046   1924            JMP     WPH3            WRITE THE DATA
                        1925
                        1926  *         IS CONTIG FILE. OPEN IN CONTIG MODE
                        1927
045.357  043           1928  WPH1      INX     H
000.000                 1929            ERRNZ   FDN.SIZ-FDN.STA-1
045.360  116           1930            MOV     C,M             (C) = COUNT (IN BLOCKS)
045.361  006 000       1931            MVI     B,0
045.363  041 243 063   1932            LXI     H,DESTFB+FB.NAM
045.366  076 001       1933            MVI     A,CN.DES
045.370  305           1934            PUSH    B               SAVE COUNT
045.371  377 050       1935            DB      SYSCALL,.DELET  DELETE OLD ONE
045.373  322 003 046   1936            JNC     WPH1.5          DELETED
045.376  376 014       1937            CPI     EC.FNF
046.000  302 026 053   1938            JNE     ERROR           MUST BE WRITE PROTECTED, OR  SOMETHING...
046.003  301           1939  WPH1.5    POP     B               (BC) = COUNT
046.004  041 243 063   1940            LXI     H,DESTFB+FB.NAM
046.007  076 001       1941            MVI     A,CN.DES
046.011  377 045       1942            DB      SYSCALL,.OPENC  OPEN CONTIG
046.013  332 207 052   1943            JC      DESTERR
046.016  303 053 046   1944            JMP     WPH3
                        1945
                        1946  *         THIS FILE HAS ALREADY BEEN PARTIALLY WRITTEN. OPEN IN UPDATE MODE
                        1947  *         SO WE CAN EXTEND IT.
                        1948
046.021  041 243 063   1949  WPH2      LXI     H,DESTFB+FB.NAM
046.024  076 001       1950            MVI     A,CN.DES
046.026  377 044       1951            DB      SYSCALL,.OPENU  OPEN FOR UPDATE
046.030  332 207 052   1952            JC      DESTERR         PROBLEMS
046.033  341           1953            POP     H
```

```
046.034  345           1954          PUSH    H               (HL) = #FDN.STA
046.035  315 234 030   1955          CALL    $INDL
046.040  005 000       1956          DW      FDN.AMW         (DE) = AMOUNT WRITTEN
046.042  102           1957          MOV     B,D
046.043  113           1958          MOV     C,E             (BC) = SECTORS WRITTEN
046.044  076 001       1959          MVI     A,CN.DES
046.046  377 047       1960          DB      SYSCALL,.POSIT  POSITION FOR EXTEND
046.050  332 215 052   1961          JC      IERR1           COULDNT GET THERE!
                       1962
                       1963  *       FILE OPEN AND POSITIONED, WRITE DATA
                       1964
046.053  341           1965  WPH3    POP     H
046.054  345           1966          PUSH    H               (HL) = #FDN.LNK
046.055  315 234 030   1967          CALL    $INDL
046.060  007 000       1968          DW      FDN.ADR         (E) = ADDR/256, (D) = CNT/256
046.062  102           1969          MOV     B,D
046.063  123           1970          MOV     D,E
046.064  036 000       1971          MVI     E,0             (DE) = ADDRESS
046.066  113           1972          MOV     C,E             (BC) = COUNT
046.067  076 001       1973          MVI     A,CN.DES
046.071  305           1974          PUSH    B               SAVE WRITE COUNT
046.072  377 005       1975          DB      SYSCALL,.WRITE  WRITE IT
046.074  332 207 052   1976          JC      DESTERR         PROBABLY OUT OF ROOM
046.077  076 001       1977          MVI     A,CN.DES
046.101  377 046       1978          DB      SYSCALL,.CLOSE  CLOSE IT
046.103  332 207 052   1979          JC      DESTERR
046.106  301           1980          POP     B               (B) = SECTORS WRITTEN
046.107  341           1981          POP     H
046.110  345           1982          PUSH    H               (HL) = #FDN.LNK
046.111  021 005 000   1983          LXI     D,FDN.AMW-FDN.LNK
046.114  031           1984          DAD     D               (HL) = FDN.AMW
046.115  176           1985          MOV     A,M
046.116  200           1986          ADD     B
046.117  167           1987          MOV     M,A
046.120  043           1988          INX     H
046.121  176           1989          MOV     A,M
046.122  316 000       1990          ACI     0               INCREMENT AMOUNT WRITTEN
046.124  167           1991          MOV     M,A
                       1992
                       1993  *       CLEAR 'IN MEMORY' COUNT IN NODE. IF THE FILE HAS NO MORE TO
                       1994  *       READ, REMOVE IT FROM THE CHAIN AND NAMTAB
                       1995
046.125  321           1996          POP     D               (DE) = FDN.LNK
046.126  041 010 000   1997  WPH4    LXI     H,FDN.AIM
046.131  031           1998          DAD     D
046.132  066 000       1999          MVI     M,0             CLEAR AMOUNT IN MEMORY
046.134  353           2000          XCHG                    (HL) = FDN.LNK
046.135  043           2001          INX     H
000.000                2002          ERRNZ   FDN.STA-FDN.LNK-1
046.136  176           2003          MOV     A,M             (A) = FDN.STA
046.137  346 002       2004          ANI     ST.OPR
046.141  300           2005          RNZ                     STILL READING, AM DONE FOR THIS PHASE
                       2006
                       2007  *       UNLINK NODE FROM LIST
                       2008
046.142  053           2009          DCX     H
```

```
046.143  176            2010          MOV     A,M
046.144  062 001 063    2011          STA     FDNHEAD         UNLINK FROM ACTIVE LIST
046.147  072 000 063    2012          LDA     FDNFRE
046.152  167            2013          MOV     M,A             PUT THIS GUY ON HEAD OF FREE LIST
046.153  175            2014          MOV     A,L
046.154  062 000 063    2015          STA     FDNFRE
046.157  315 374 056    2016          CALL    REN             REMOVE ENTRY FROM NAMTAB
046.162  303 252 045    2017          JMP     WPH             TRY TO WRITE THE NEXT GUY


                        2019  **      CBR - COMPUTE BUFFER ROOM.
                        2020  *
                        2021  *       CBR COMPUTES THE NUMBER OF SECTORS WORTH OF RAM
                        2022  *       STILL FREE.
                        2023  *
                        2024  *       ENTRY   NONE
                        2025  *       EXIT    (A) = SECTORS OF RAM FREE
                        2026  *               'Z' SET IFF (A)  = 0
                        2027  *               (H) = BUFFPTR/256
                        2028  *               (L) = OBUFLIM/256
                        2029  *       USES    A,F
                        2030
                        2031
046.165  052 114 063    2032  CBR     LHLD    OBUFLIM
000.000                 2033          ERRNZ   OBUFFPTR-OBUFLIM-1
046.170  175            2034          MOV     A,L
046.171  224            2035          SUB     H
046.172  311            2036          RET


                        2038  **      IFL - INITIALIZE FDN LIST.
                        2039  *
                        2040  *       IFL CHAINS ALL THE FDN NODES TO THE FREE LIST. THIS
                        2041  *       CLEANUP IS NECESSARY IN CASE A CTL-C OR SOMETHING
                        2042  *       LEFT THE LIST GARBAGED.
                        2043  *
                        2044  *       ENTRY   NONE
                        2045  *       EXIT    NONE
                        2046  *       USES    ALL
                        2047
                        2048
046.173  041 002 063    2049  IFL     LXI     H,FDN.1
046.176  175            2050          MOV     A,L             (A) = FIRST LINK
046.177  062 000 063    2051          STA     FDNFRE
046.202  257            2052          XRA     A
046.203  062 001 063    2053          STA     FDNHEAD         NONE IN LIST
046.206  006 007        2054          MVI     B,FDNCNT-1      (B) = NUMBER OF NODES-1
046.210  076 011        2055  IFL1    MVI     A,FDNELEN
046.212  205            2056          ADD     L               (A) = #ADDR OF NEXT NODE
046.213  167            2057          MOV     M,A             SET LINK
046.214  157            2058          MOV     L,A             FORWARD TO NEXT LINK
046.215  005            2059          DCR     B
```

```
046.216  302 210 046  2060          JNZ     IFL1            MORE TO GO
046.221  066 000      2061          MVI     M,0             LAST ONE CHAINS NOWHERE
046.223  311          2062          RET


                      2064  **      MAD - MOUNT ALTERNATE DISK.
                      2065  *
                      2066  *       MAD DISMOUNTES THE CURRENT DISK, HAS THE USER INSERT THE
                      2067  *       OTHER DISK, AND MOUNTS IT.
                      2068  *
                      2069  *       ENTRY   (B) = FRONT PANEL LED PATTERN
                      2070  *               (DE) = PROMPT PATTERNS FOR PANEL AND CONSOLE
                      2071  *       EXIT    (HL) = $VOLFLAG
                      2072  *       USES    ALL
                      2073
                      2074
046.224               2075  MAD     EQU     *
                      2076
                      2077  *       DISMOUNT CURRENT DISK
                      2078
046.224  325          2079          PUSH    D
046.225  305          2080          PUSH    B               SAVE ENTRY PARAMETERS IN CASE OF RETRY
046.226  325          2081          PUSH    D
046.227  305          2082          PUSH    B               SAVE ENTRY PARAMETERS OVER SYDD CALL
046.230  041 051 047  2083          LXI     H,MNDA          DEVICE SPECIFICATION
046.233  377 203      2084          DB      SYSCALL,.DMNMS  DISMOUNT WITHOUT MESSAGE
046.235  332 026 053  2085          JC      ERROR           IF ERROR
                      2086
                      2087  *       SETUP PROMPT ON FP LEDS AND CONSOLE FOR NEW DISK
                      2088
046.240  076 203      2089  MAD0    MVI     A,U0.DDU+U0.CLK+U0.HLT                          /2.0a/
046.242  062 010 040  2090          STA     .MFLAG          HALT DISPLAY UPDATE
                      2091
046.245  041 013 040  2092          LXI     H,.ALEDS
046.250  076 011      2093          MVI     A,9
046.252  301          2094          POP     B               (B) = PERIOD PATTERN
046.253  160          2095  MAD2    MOV     M,B             SET PATTERN
046.254  043          2096          INX     H
046.255  075          2097          DCR     A
046.256  302 253 046  2098          JNZ     MAD2            IF MORE TO BLANK
                      2099
046.261  041 016 040  2100          LXI     H,.ALEDS+3
046.264  001 003 000  2101          LXI     B,3
046.267  321          2102          POP     D               (DE) = PROMPT LIST
046.270  315 252 030  2103          CALL    $MOVE           MOVE IN PROMPT PATTERN
                      2104
046.273  353          2105          XCHG                    (HL) = PATTERN
046.274  377 003      2106          DB      SYSCALL,.PRINT  CONSOLE PROMPT
046.276  315 136 031  2107          CALL    $TYPTX
046.301  207          2108          DB      BELL+200Q       BEEP CONSOLE, TOO
046.302  076 144      2109          MVI     A,100
046.304  315 140 002  2110          CALL    .HORN           BEEP A WARNING
                      2111
                      2112  *       WAIT FOR SIGNAL THAT NEW DISK IS IN
```

```
                          2113
046.307  076 012          2114  MAD3    MVI     A,DC.RDY                                  /2.0a/
046.311  315 130 040      2115          CALL    SYDD                                      /2.0a/
046.314  322 307 046      2116          JNC     MAD3            Wait for device to go non-ready  /2.0a/
                          2117
046.317  076 012          2118  MAD4    MVI     A,DC.RDY                                  /2.0a/
046.321  315 130 040      2119          CALL    SYDD                                      /2.0a/
046.324  332 317 046      2120          JC      MAD4            Wait for device to go ready  /2.0a/
                          2121
                          2122  *       READ NEW DISK'S LABEL
                          2123
046.327  315 056 047      2124          CALL    GETLAB
046.332  332 026 053      2125          JC      ERROR
                          2126
                          2127  *       SEE IF LABEL CHANGED FROM BEFORE
                          2128
046.335  016 000          2129          MVI     C,0             Compare 256               /2.0a/
046.337  021 321 063      2130          LXI     D,SLABEL        DE = address of last label  /2.0a/
046.342  041 321 064      2131          LXI     H,LABEL         HL = Address of current label  /2.0a/
046.345  315 060 030      2132          CALL    $COMP           See if the label changed  /2.0a/
046.350  301              2133          POP     B
046.351  321              2134          POP     D               RESTORE ENTRY PARAMETERS
                          2135
046.352  041 113 063      2136          LXI     H,VOLSER
046.355  072 321 064      2137          LDA     LABEL+LAB.SER
046.360  302 372 046      2138          JNE     MAD4.5          IS THE RIGHT DISK         /2.0a/
046.363  325              2139          PUSH    D               SAVE PARAMS AS IN BEGINNING
046.364  305              2140          PUSH    B
046.365  325              2141          PUSH    D               SAVE FOR RETRY
046.366  305              2142          PUSH    B
046.367  303 240 046      2143          JMP     MADO            IT WAS NOT THE RIGHT DISK
                          2144
046.372  167              2145  MAD4.5  MOV     M,A             SET NEW SERIAL
046.373  041 112 063      2146          LXI     H,VOLFLAG
046.376  176              2147          MOV     A,M
046.377  057              2148          CMA
047.000  167              2149          MOV     M,A             COMPLEMENT VOLUME FLAG
                          2150
                          2151  *       ERASE FRONT PANEL DISLPLAY
                          2152
047.001  041 013 040      2153          LXI     H,.ALEDS
047.004  076 011          2154          MVI     A,9
047.006  160              2155  MAD5    MOV     M,B             SET TO PATTERN
047.007  043              2156          INX     H
047.010  075              2157          DCR     A
047.011  302 006 047      2158          JNZ     MAD5
                          2159
047.014  001 000 001      2160          LXI     B,256                                     /2.0a/
047.017  021 321 064      2161          LXI     D,LABEL                                   /2.0a/
047.022  041 321 063      2162          LXI     H,SLABEL                                  /2.0a/
047.025  315 252 030      2163          CALL    $MOVE           Save Current Label        /2.0a/
                          2164
047.030  315 040 047      2165          CALL    MND             MOUNT NEW DISK
047.033  315 136 031      2166          CALL    $TYPTX          Show user that disk is OK  /2.0a/
047.036  212              2167          DB      ENL                                       /2.0a/
047.037  311              2168          RET
```

```
                                2170  **      MND     - MOUNT NEW DISK
                                2171  *
                                2172  *       MOUNT NEW DISK ONTO DEVICE SSECIFIED IN MNDA
                                2173  *
                                2174  *
                                2175  *       ENTRY   NONE
                                2176  *
                                2177  *       EXIT    LABEL   = LABEL SECTOR
                                2178  *
                                2179  *       USES    ALL
                                2180  *
                                2181
047.040  041 051 047           2182  MND     LXI     H,MNDA
047.043  377 202               2183          DB      SYSCALL,.MONMS  MOUNT WITHOUT MESSAGE
047.045  332 026 053           2184          JC      ERROR           IF ERROR IN MOUNT
047.050  311                   2185          RET                                             /2.0a/
                                2186
047.051  123 131 060           2187  MNDA    DB      'SY0:',0



                                2189  **      GETLAB  - GET LABEL
                                2190  *
                                2191  *       GETLAB READS THE DISK LABEL
                                2192  *
                                2193  *       NOTE:   This routine leaves the volume mounted as       /2.0a/
                                2194  *               zero.
                                2195  *
                                2196  *       ENTRY   NONE
                                2197  *
                                2198  *       EXIT    LABEL IN LABEL
                                2199  *               (PSW)   = 'C' CLEAR IF NO ERROR
                                2200  *                       = 'C' SET   IF    ERROR
                                2201  *                             (A)  = ERROR CODE
                                2202  *
                                2203  *       USES    ALL
                                2204  *
                                2205
047.056  041 000 000           2206  GETLAB  LXI     H,0                                     /2.0a/
047.061  076 010               2207          MVI     A,DC.MOU                                /2.0a/
047.063  315 130 040           2208          CALL    SYDD            Mount the Disk as volume 0  /2.0a/
047.066  330                   2209          RC                      Some type of problem    /2.0a/
                                2210
047.067  041 011 000           2211          LXI     H,DDF.LAB
047.072  021 321 064           2212          LXI     D,LABEL
047.075  001 000 001           2213          LXI     B,256
047.100  076 002               2214          MVI     A,DC.RER                                /2.0a/
047.102  315 130 040           2215          CALL    SYDD
047.105  311                   2216          RET
                                2217          ENDIF
```

```
                    2220  ***     DELETE - PROCESS DELETE COMMAND.
                    2221  *
                    2222  *       SYNTAX:
                    2223  *
                    2224  *       SOURCE1,...,SOURCEN/DELETE
                    2225  *
                    2226  *       AT LEAST ONE SOURCE FILE MUST BE SPECIFIED.
                    2227  *       IF *.* IS SPECIFIED, DELETE ASKS,
                    2228  *              DELETE ALL ?!? ARE YOU SURE?
                    2229
                    2230
000.001             2231          IF      .PIP.
                    2232  DELETE  EQU     *
                    2233          LXI     H,LINE
                    2234
                    2235  *       SEE IF A DESTINATION FILE SPECIFIED
                    2236
                    2237  DEL1    MOV     A,M
                    2238          INX     H
                    2239          ANA     A
                    2240          JZ      DEL2            END OF LINE
                    2241          CPI     '='
                    2242          JNE     DEL1
                    2243
                    2244  *       HE SPECIFIED A DESTINATION FILE
                    2245
                    2246          MVI     A,PEC.TFI       TARGET FILE ILLEGAL
                    2247          JMP     ERROR           FORMAT ERROR
                    2248
                    2249  *       NO TARGET FILE SPECIFIED
                    2250
                    2251  DEL2    MVI     A,1             CHECK FOR *.*
                    2252          CALL    BSL             BUILD SOURCE FILE LIST
                    2253          JC      ERROR           NO GOOD
                    2254
                    2255  *       DELETE FILES ONE BY ONE
                    2256
                    2257  DEL5    LHLD    NAMTLEN
                    2258          MOV     A,H
                    2259          ORA     L
                    2260          RZ                      END OF LIST
                    2261          LXI     H,NAMTAB
                    2262          DB      SYSCALL,.DELET  REMOVE IT
                    2263          JC      NAMERR          ERROR ON DELETE
                    2264          CALL    REN             REMOVE ENTRY FROM NAMTAB
                    2265          JMP     DEL5            DELETE THE NEXT ONE
                    2266          STL     'RENAME - PROCESS RENAME COMMAND'
                    2267          EJECT
                    2268  ***     RENAME - RENAME FILES.
                    2269  *
                    2270  *       SYNTAX:
                    2271  *
                    2272  *       DEST = SOURCE1,...,SOURCEN
                    2273  *
                    2274  *       RENAME IS PROCESSED IN A MANNER SIMILAR TO COPY, EXCEPT THAT THE
                    2275  *       FILE IS RENAMED, RATHER THAN COPIED.
```

```
2276
2277
2278  RENAME  EQU     *
2279          CALL    DDF             DECODE DESTINATION FILE
2280          JC      ERROR
2281          XRA     A               ALLOW *.*
2282          CALL    BSL             BUILD SOURCEFILE LIST
2283          JC      ERROR
2284
2285  *       DO MULTIPLE RENAMES
2286
2287  REN1    LXI     B,DESTFB+FB.NAM (BC) = WILDCARDED TARGET NAME
2288          LXI     D,NAMTAB        (DE) = NORMAL SOURCE NAME
2289          LXI     H,RENA          (HL) = BUFFER FOR RESULT NAME
2290          PUSH    B               SAVE #DESTFB+FB.NAM
2291          PUSH    D               SAVE #NAMTAB
2292          CALL    MWN             MERGE WILDCARD NAME
2293          POP     D               (DE) = #NAMTAB
2294          POP     H               (HL) = #DESTFB+FB.NAM
2295
2296
2297  *       SEE IF SOURCE AND DEST FILE ON SAME DEVICE
2298
2299          PUSH    D               SAVE #NAMTAB (SOURCE NAME)
2300          MVI     C,3
2301          CALL    $COMP           COMPARE DEVICES
2302          MVI     A,PEC.DNC       DEVICES NOT CONSISTANT
2303          JNE     ERROR
2304
2305  *       SEE IF TARGET ALREADY EXISTS
2306
2307          LXI     H,RENA
2308          MVI     A,CN.SOU
2309          DB      SYSCALL,.OPENR
2310          LXI     H,RENA-FB.NAM
2311          JC      REN2            HAVE AN ERROR (AS WE SHOULD)
2312          MVI     A,EC.FAP        FILE AALREADY PRESENT
2313          JMP     $FERROR         ALREADY THERE
2314
2315  REN2    CPI     EC.FNF          MUST BE NOT FOUND
2316          JNE     $FERROR         OTHER ERROR
2317          POP     H               (HL) = SOURCE NAME
2318          LXI     B,RENA          (BC) = NEW (TARGET) NAME
2319          DB      SYSCALL,.RENAM  RENAME IT
2320          JC      NAMERR          ERROR ON RENAME
2321
2322  *       REMOVE NAME FROM NAMTAB
2323
2324          CALL    REN             REMOVE ENTRY FROM NAMTAB
2325          LHLD    NAMTLEN
2326          MOV     A,H
2327          ORA     L
2328          JNZ     REN1
2329          RET
2330
2331  RENA    DS      FB.NAML         FILE NAME WORK AREA
```

```
                        2332            ENDIF
```

```
                                2335  ***     LIST - INDEX DIRECTORY.
                                2336  *
                                2337  *       DEST=SOURCE/LIST
                                2338  *                  /BRIEF
                                2339  *
                                2340  *       THESE SWITCHES CAUSE THE DIRECTORY CONTENTS OF THE SPECIFIED FILE(S)
                                2341  *       TO BE LISTED
                                2342  *
                                2343  *       IN /LI FIRM, THE OUTPUT IS:
                                2344  *
                                2345  *       NAME    EXT     SIZE      DATE          FLAGS
                                2346  *       XXX     .XXX     NNN    DD-MMM-YY       CWS
                                2347  *        .       .       .         .             .
                                2348  *        .       .       .         .             .
                                2349  *        .       .       .         .             .
                                2350  *                NNN FILES USING MMM SECTORS, XXX FREE
                                2351  *
                                2352  *       IN /BR FORM, ONLY THE NAME AND EXTENSION ARE LISTED,
                                2353  *       4 ACROSS THE PAGE.
                                2354  *
                                2355  *       SPECIAL CONSIDERATIONS:
                                2356  *
                                2357  *       A NULL NAME OR EXTENSION IS TAKEN AS '*' (WILDCARD)
                                2358  *
                                2359  *       IMPLIMENTATION:
                                2360  *
                                2361  *       A FILE LIST OF SOURCE FILES IS BUILT. THE DEVICE DIRECTORY FILE
                                2362  *       IS THEN READ, AND EACH FILE IN IT IS CHECKED FOR A MATCH
                                2363  *       AGAINST ANY SOURCE SPECIFICATIONS. ELIGIBLE FILES ARE LISTED.
                                2364
                                2365
047.106  041 000 000           2366  LIST    LXI     H,0
047.111  303 117 047           2367          JMP     LIST1
                                2368
047.114  041 001 000           2369  BRIEF   LXI     H,1
                                2370  *       JMP     LIST1
                                2371
047.117  042 224 050           2372  LIST1   SHLD    LSTA            (LSTA) = 0 IF LIST, 1 IF /BRIEF
000.000                         2373          ERRNZ   LSTB-LSTA-1     LSTB = FILE COUNT
047.122  041 000 000           2374          LXI     H,0
047.125  042 226 050           2375          SHLD    LSTC            CLEAR SECTORS USED COUNT
047.130  315 044 061           2376          CALL    $MOVEL
047.133  011 000 277           2377          DW      9,S.DATE,LSTG1  SET DATE IN HEADING
                                2378
                                2379  *       CRACK DESTINATION FILE NAMES
                                2380
000.001                         2381          IF      .PIP.
                                2382          CALL    DDF             DECODE DEST FILE NAME
                                2383          JC      ERROR           FILE NAME ERROR
                                2384          ANA     A
                                2385          MVI     A,PEC.IUW       ILLEGAL USE OF WILDCARD IN DEST
                                2386          JZ      ERROR
                                2387          ENDIF
                                2388
                                2389  *       BUILD LIST OF SPECIFICATIONS
                                2390
```

```
047.141  315 012 051  2391           CALL    BLS             BUILD LIST OF SOURCE SPECS
047.144  332 026 053  2392           JC      ERROR           ERROR IN LIST
047.147  001 003 000  2393           LXI     B,3
047.152  041 206 063  2394           LXI     H,DIRNAM
047.155  315 252 030  2395           CALL    $MOVE           MOVE DEVICE CODE INTO DIRECT.SYS NAME
047.160  041 210 063  2396           LXI     H,DIRNAM+2
047.163  176          2397           MOV     A,M             SEE IF UNIT NUMBER OMITTED
047.164  247          2398           ANA     A
047.165  302 172 047  2399           JNZ     LIST1.5         SPECIFIED
047.170  066 060      2400           MVI     M,'0'           DONT ALLOW NULL NUMBER
                      2401
                      2402  *        GET ADDRESS OF DEVICE'S GRT
                      2403
047.172  041 206 063  2404  LIST1.5  LXI     H,DIRNAM        (HL) = # OF XXX:DIRECT.SYS (XXX = DEVICE)
047.175  001 230 050  2405           LXI     B,LSTD          (BC) = ADDRESS FOR RETURN INFO
047.200  377 053      2406           DB      SYSCALL,.DECODE DECODE NAME
047.202  332 026 053  2407           JC      ERROR           UNKNOWN DEVICE
047.205  072 230 050  2408           LDA     LSTD+0
047.210  346 001      2409           ANI     DT.DD
047.212  076 005      2410           MVI     A,EC.DNS
047.214  312 026 053  2411           JZ      ERROR           NOT DIRECTORY DEVICE
047.217  052 251 050  2412           LHLD    LSTD+17         (HL) = DEV TBL ADDR                 /80.04,sc/
                      2413
047.222  021 011 000  2414           LXI     D,DEV.UNT                                           /80.04,sc
047.225  031          2415           DAD     D
047.226  072 233 050  2416           LDA     LSTD+3
047.231  315 027 041  2417           CALL    S.GUP           HL = UNIT TABLE POINTER
                      2418
047.234  315 052 060  2419           CALL    $INDLB                                              /80.04,sc/
047.237  001 000      2420           DW      UNT.SPG                                             /80.04,sc/
047.241  062 262 050  2421           STA     LSTF            SAVE  SECTORS PER GROUP           /80.04,sc/
                      2422
047.244  315 234 030  2423           CALL    $INDL
047.247  002 000      2424           DW      UNT,GRT
047.251  353          2425           XCHG
047.252  042 260 050  2426           SHLD    LSTE            SAVE  GRT ADDRESS
047.255  353          2427           XCHG
                      2428
                      2429  *        OPEN DEVICE'S DIRECTORY
                      2430
047.256  041 206 063  2431           LXI     H,DIRNAM
047.261  076 002      2432           MVI     A,CN.DIR
047.263  377 042      2433           DB      SYSCALL,.OPENR
047.265  076 200      2434           MVI     A,EC.DF         DEVICE FORMAT ERROR
047.267  332 026 053  2435           JC      ERROR           CANT OPEN DIRECTORY
                      2436
                      2437
                      2438  *        OPEN OUTPUT FILE
                      2439
000,001               2440           IF      .PIP.
                      2441           LXI     H,DESTFB
                      2442           CALL    $FOPEW          OPEN FOR WRITE
                      2443           ENDIF
                      2444
                      2445  *        GENERATE HEADING
                      2446
```

```
047.272  001 001 000  2447           LXI     B,1             (BC) = TEXT COUNT
047.275  021 263 050  2448           LXI     D,LSTG          (DE) = TEXT ADDRESS
047.300  072 224 050  2449           LDA     LSTA
047.303  247          2450           ANA     A
047.304  302 311 047  2451           JNZ     LIST2           IS SHORT
047.307  016 051      2452           MVI     C,LSTGL         PRINT FULL HEADING
000.001               2453           IF      .PIP.
                      2454  LIST2    CALL    $FWRIB          WRITE HEADING
                      2455           ELSE
047.311  171          2456  LIST2    MOV     A,C
047.312  353          2457           XCHG                    (HL) = LINE ADDRESS
047.313  315 156 057  2458           CALL    $TYPCC          PRINT ON CONSOLE
                      2459           ENDIF
                      2460
                      2461  *        READ DIRECTORY BLOCKS, LOOKING FOR FILE MATCHES
                      2462
047.316  001 000 002  2463  LIST3    LXI     B,512
047.321  315 234 056  2464           CALL    GDWP            DE = DIRECTORY WORKSPACE POINTER  /79.11.GC/
047.324  076 002      2465           MVI     A,CN.DIR
047.326  325          2466           PUSH    D                                                 /79.11.GC/
047.327  377 004      2467           DB      SYSCALL,.READ
047.331  321          2468           POP     D               DE = DIRECOTRY WORKSPACE          /79.11.GC/
047.332  332 104 050  2469           JC      LIST9           ALL DONE
                      2470
                      2471  *        CHECK NEXT ENTRY IN NAMTAB AGAINST DIRECTORY ENTRY.
                      2472  *        (DE) = DIRECTORY BUFFER POINTER
                      2473
047.335  032          2474  LIST4    LDAX    D               (A) = FIRST CHARACTER OF NAME
047.336  247          2475           ANA     A
047.337  312 316 047  2476           JZ      LIST3           END OF THIS BUFFER
047.342  074          2477           INR     A
000.000               2478           ERRNZ   DF.EMP-3770
047.343  312 036 050  2479           JZ      LIST7           THIS ENTRY IS EMPTY
047.346  074          2480           INR     A
047.347  312 104 050  2481           JZ      LIST9           NO MORE ENTRYS IN DIRECTORY
047.352  353          2482           XCHG
047.353  315 331 053  2483           CALL    CFE             CHECK FILE ELIGIBILITY
047.356  353          2484           XCHG
047.357  302 036 050  2485           JNE     LIST7           NOT ELIGIBLE
047.362  041 114 066  2486           LXI     H,NAMTAB
                      2487
047.365  345          2488  LIST5    PUSH    H
047.366  325          2489           PUSH    D               SAVE ADDRESS OF FILE AND PATTERN
047.367  315 136 054  2490           CALL    CAD             CONVERT ASCII NAMTAB ENTRY TO DIRECTORY FORMAT
047.372  021 324 065  2491           LXI     D,PIO.DIR+DIR.NAM       (DE) = NAMTAB PATTERN
047.375  341          2492           POP     H
047.376  345          2493           PUSH    H               (HL) = DIRECTORY PATTERN
047.377  006 013      2494           MVI     B,8+3           CHECK FOR MATCH
050.001  315 004 054  2495           CALL    CWM             CHECK FOR WILDCARD MATCH
050.004  321          2496  LIST6    POP     D
050.005  341          2497           POP     H
050.006  312 065 050  2498           JE      LIST8           GOT FILE TO LIST
050.011  001 021 000  2499           LXI     B,FB.NAML
050.014  011          2500           DAD     B               ADVANCE PAST ENTRY IN NAMTAB
                      2501
                      2502  *        SEE IF AT END OF NAMTAB
```

```
                        2503
050.015 325             2504            PUSH    D
050.016 353             2505            XCHG                    (DE) = NEW ADDRESS
050.017 052 264 063     2506            LHLD    NAMTLEN
050.022 001 114 066     2507            LXI     B,NAMTAB
050.025 011             2508            DAD     B               (HL) = LWA+1 OF TABLE
050.026 353             2509            XCHG
050.027 315 216 030     2510            CALL    $CDEHL          COMPARE
050.032 321             2511            POP     D
050.033 302 365 047     2512            JNE     LIST5           MORE IN TABLE
                        2513
                        2514 *          FILE DOESNT MATCH ANY SELECTED FILE, PASS TO NEXT ONE
                        2515
050.036 353             2516 LIST7      XCHG                    (HL) = DIR BUFFER ADDRESS
                        2517
050.037 345             2518            PUSH    H                                       /79.11.GC/
050.040 315 242 056     2519            CALL    GDWP,           HL = DIRECTORY WORKSPACE PTR,   /79.11.GC/
050.043 315 052 060     2520            CALL    $INDLB          A  = DIR. ENTRY LENGTH          /79.11.GC/
050.046 373 001         2521            DW      DIS.ENL                                         /79.11.GC/
050.050 341             2522            POP     H                                               /79.11.GC/
                        2523
050.051 315 101 030     2524            CALL    $DADA,          ADVANCE
050.054 176             2525            MOV     A,M
050.055 247             2526            ANA     A
050.056 353             2527            XCHG
050.057 302 335 047     2528            JNZ     LIST4           TRY THIS ONE
050.062 303 316 047     2529            JMP     LIST3           READ ANOTHER BLOCK
                        2530
                        2531 *          HAVE FILE TO LIST
                        2532
050.065 325             2533 LIST8      PUSH    D               SAVE DIR POINTER
050.066 072 262 050     2534            LDA     LSTF            (A) = SECTORS PER GROUP THIS DEVICE
050.071 315 141 051     2535            CALL    PFI             PRINT FILE INFO
050.074 321             2536            POP     D
050.075 041 225 050     2537            LXI     H,LSTB
050.100 064             2538            INR     M               COUNT FILE
050.101 303 036 050     2539            JMP     LIST7           ADVANCE TO NEXT FILE
                        2540
                        2541 *          ALL DONE, CLOSE DIRECTORY FILE
                        2542
050.104 076 002         2543 LIST9      MVI     A,CN.DIR
050.106 377 046         2544            DB      SYSCALL,.CLOSE  CLOSE FILE
050.110 001 001 000     2545            LXI     B,1             ASSUME SHOFT FORM, JUST WRITE NL
050.113 072 224 050     2546            LDA     LSTA            (A) = FORM FLAG
050.116 247             2547            ANA     A
050.117 302 207 050     2548            JNZ     LIST10          IS SHORT, NO TRAILER
                        2549
                        2550 *          PRINT SUMMARY:
                        2551 *
                        2552 *          NNN FILES, USING XXX SECTORS, YYY FREE
                        2553
050.122 072 225 050     2554            LDA     LSTB
050.125 117             2555            MOV     C,A
050.126 006 000         2556            MVI     B,0             (BC) = FILE COUNT
050.130 076 003         2557            MVI     A,3
050.132 041 340 050     2558            LXI     H,LSTH1
```

```
050.135  315 371 060  2559           CALL    $UDDN           FILE COUNT
                      2560
050.140  052 226 050  2561           LHLD    LSTC
050.143  104         2562           MOV     B,H
050.144  115         2563           MOV     C,L             (BC) = SECTOR COUNT
050.145  041 361 050  2564           LXI     H,LSTH2
050.150  076 004      2565           MVI     A,4                                     /80.05.sc/
050.152  315 371 060  2566           CALL    $UDDN           USED COUNT
                      2567
050.155  052 260 050  2568           LHLD    LSTE
050.160  176         2569           MOV     A,M
050.161  315 351 053  2570           CALL    CFS             FOLLOW GRT CHAIN
050.164  072 262 050  2571           LDA     LSTF
050.167  315 007 031  2572           CALL    $MU86           (HL) = SECTORS FREE
050.172  104         2573           MOV     B,H
050.173  115         2574           MOV     C,L
050.174  041 377 050  2575           LXI     H,LSTH3
050.177  076 004      2576           MVI     A,4                                     /80.05.sc/
050.201  315 371 060  2577           CALL    $UDDN           UNPACK FREE
                      2578
050.204  001 056 000  2579           LXI     B,LSTHL
050.207  021 334 050  2580  LIST10   LXI     D,LSTH
050.212  072 204 063  2581           LDA     SUPRES
050.215  247         2582           ANA     A
000.001               2583           IF      .PIP.
                      2584           LXI     H,DESTFB
                      2585           JNZ     $FCLO           CLOSE AND EXIT, SUMMARY SUPPRESSED
                      2586           CALL    $FWRIB          WRITE TRAILER
                      2587
                      2588  *        ALL DONE, CLOSE OUTPUT FILE
                      2589
                      2590           JMP     $FCLO           CLOSE AND EXIT
                      2591           ELSE
050.216  300         2592           RNZ                     NOT TO SUMMARYIZE
050.217  171         2593           MOV     A,C             (A) = COUNT
050.220  353         2594           XCHG                    (HL) = ADDRESS
050.221  303 156 057  2595           JMP     $TYPCC          TYPE TEXT AND EXIT
                      2596           ENDIF
                      2597
050.224  000         2598  LSTA     DB      0               <>0 IFF SHORT FORM
                      2599
050.225  000         2600  LSTB     DB      0               FILE COUNT
050.226  000 000      2601  LSTC     DW      0               SECTORS USED
050.230               2602  LSTD     DS      24              FILE NAME DECODE AREA
050.260  000 000      2603  LSTE     DW      0               GRT ADDRESS
050.262  000         2604  LSTF     DB      0               SECTORS PER GROUP FOR THIS DEVICE
050.263  012 116 141  2605  LSTG     DB      NL,'Name',TAB,'.Ext',TAB,'Size',TAB,'  Date',TAB,TAB,'Flags',TAB
050.321               2606  LSTG1    DS      9               DATE
050.332  012 012      2607           DB      NL,NL
000.051               2608  LSTGL    EQU     *-LSTG
                      2609
050.334  012 040 040  2610  LSTH     DB      NL,'   '                FIRST CHARACTER MUST BE <NL>
050.340  116 116 116  2611  LSTH1    DB      'NNN Files, Usins '
050.361  115 115 115  2612  LSTH2    DB      'MMMM Sectors ('
050.377  130 130 130  2613  LSTH3    DB      'XXXX Free)',NL
000.056               2614  LSTHL    EQU     *-LSTH
```

```
                                2616  **      BLS - BUILD LIST OF SOURCE FILES.
                                2617  *
                                2618  *       BLS BUILDS A LIST OF SOURCE FILES INTO *NAMTAB*
                                2619  *       NULL FIELDS ARE SET TO WILDCARDS. BLS REQUIRES THAT ALL
                                2620  *       FILES SPECIFIED HAVE THE SAME DEVICE.
                                2621  *
                                2622  *       IF THE COMMAND LINE CONTAINS NO FILES, BUT CONTAINS AT LEAST
                                2623  *       ONE BLANK (AS WOULD BE THE CASE IN PROCESSING THE /LIST SWITCH, SINCE
                                2624  *       THE '/LIST' IS REPLACED WITH BLANKS) A FILE NAME OF ????????.???
                                2625  *       IS DECODED.
                                2626  *       ENTRY   NAMTAB EMPTY
                                2627  *       EXIT    'C' CLEAR IF OK
                                2628  *                 (DE) = $BLSA = 3 CHARACTER DEVICE NAME
                                2629  *                 'C' SET IF ERROR
                                2630  *                  (A) = ERROR MESSAGE
                                2631  *       USES    ALL
                                2632
                                2633
051.012  315 044 061            2634  BLS     CALL    $MOVEL
051.015  003 000 134            2635          DW      3,BLSC,BLSA     SET INITIAL DEFAULT DEVICE
051.023  041 000 000            2636          LXI     H,0
051.026  042 264 063            2637          SHLD    NAMTLEN         CLEAR NAMTAB
051.031  076 377                2638          MVI     A,377Q
051.033  062 133 051            2639          STA     BLSB            FLAG PROCESSING OF FIRST FILE NAME
051.036  315 300 056            2640          CALL    LSN             LOCATE SOURCE NAMES
                                2641
                                2642  *       CRACK THE NEXT NAME
                                2643
051.041  176                   2644  BLS1    MOV     A,M
051.042  021 125 051            2645          LXI     D,BLSA          (DE) = DEFAULT ADDRESS
051.045  247                   2646          ANA     A
051.046  310                   2647          RZ                      NO MORE NAMES
051.047  315 321 057            2648          CALL    $SOB            SEE IF ALL NULL
051.052  176                   2649          MOV     A,M
051.053  247                   2650          ANA     A
051.054  302 062 051            2651          JNZ     BLS2            NOT ALL NULL
051.057  041 134 051            2652          LXI     H,BLSC          USE DEFAULT DEVICE
051.062  315 142 054            2653  BLS2    CALL    CAD.            CONVERT ASCII NAME TO DIRECTORY FORMAT
051.065  330                   2654          RC                      ERROR
                                2655
                                2656  *       IF FIRST NAME, RECORD DEVICE
                                2657  *       IF NOT FIRST, COMPARE DEVICE AGAINST FIRST DEVICE
                                2658
051.066  345                   2659          PUSH    H
051.067  021 321 065            2660          LXI     D,PIO.DEV
051.072  041 125 051            2661          LXI     H,BLSA
051.075  001 003 000            2662          LXI     B,3             SETUP COUNT, FROM AND TO
000.001                         2663          IF      .PIP.
                                2664          LDA     BLSB
                                2665          ANA     A
                                2666          JP      BLS3            NOT 1ST FILE
                                2667          CALL    $MOVE           MOVE IN REQUIRED DEVICE FOR REMAINING FILES
                                2668          XRA     A
                                2669          STA     BLSB            FLAG 1ST NAME PROCESSED
                                2670          JMP     BLS4
                                2671          ENDIF
```

```
                                2672
051.100  315 060 030  2673  BLS3    CALL    $COMP           SEE IF THIS DEVICE SAME AS PREVIOUS
051.103  312 113 051  2674          JE      BLS4            OK
051.106  076 201      2675          MVI     A,PEC.DNC       MULTIPLE DEVICES ARE ILLEGAL
051.110  067          2676          STC
051.111  341          2677          POP     H
051.112  311          2678          RET                     RETURN WITH ERROR
                                2679
                                2680  *       GOT NAME DECODED. ENTER IN NAMTAB
                                2681
051.113  315 160 053  2682  BLS4    CALL    AEN             ADD ENTRY TO NAMTAB
051.116  341          2683          POP     H
051.117  315 067 057  2684          CALL    SFS             SKIP FILE SEPERATOR (BLANKS AND/OR COMMA)
051.122  303 041 051  2685          JMP     BLS1            SEE IF MORE
                                2686
051.125  123 131 060  2687  BLSA    DB      'SY0',200Q,200Q,200Q
051.133  000          2688  BLSB    DB      0               FIRST FILE NAME FLAG
051.134  123 131 060  2689  BLSC    DB      'SY0:',0                DEFAULT DEVICE


                                2691  **      PFI - PRINT FILE INFO.
                                2692  *
                                2693  *       PFI DECODES A DIRECTORY ENTRY INTO A CODED LINE, THEN
                                2694  *       WRITES IT TO 'DESTFB'.
                                2695  *
                                2696  *       THE PRODUCED FORMAT DEPENDS UPON THE LISTING FORMAT FLAG,
                                2697  *       LSTA.
                                2698  *
                                2699  *       SHORT FORM:
                                2700  *
                                2701  *       NAME    .EXT    (TAB)
                                2702  *
                                2703  *       LONG FORM:
                                2704  *
                                2705  *       NAME    .EXT    SIZE    DATE    FLAGS   (NL)
                                2706  *
                                2707  *       ENTRY   (A) = SECTORS PER GROUP FOR THIS DEVICE
                                2708  *               (DE) = DIRECTORY ENTRY POINTER
                                2709  *       EXIT    IF LONG FORM, SECTOR COUNT IS ACCUMULATED IN LSTC
                                2710  *       USES    ALL
                                2711
                                2712
051.141  062 115 052  2713  PFI     STA     PFIC            SAVE SECTORS PER GROUP
051.144  041 032 052  2714          LXI     H,PFIA
051.147  016 010      2715          MVI     C,8
051.151  315 014 052  2716          CALL    PFI20           COPY NAME
051.154  312 162 051  2717          JZ      PFI1            ALL 8 CHARACTERS
051.157  066 011      2718          MVI     M,TAB
051.161  043          2719          INX     H
051.162  066 056      2720  PFI1    MVI     M,'.'
051.164  043          2721          INX     H
051.165  016 003      2722          MVI     C,3
051.167  315 014 052  2723          CALL    PFI20           COPY EXTENSION
051.172  066 011      2724          MVI     M,TAB
```

```
051.174 043          2725           INX     H
051.175 072 224 050  2726           LDA     LSTA
051.200 247          2727           ANA     A
051.201 312 226 051  2728           JZ      PFI3            IS LONG FORM
                     2729
                     2730 *         IS SHORT FORM. SEE IF NEED TO END LINE
                     2731
051.204 074          2732           INR     A
051.205 376 005      2733           CPI     5
051.207 302 220 051  2734           JNE     PFI2            NOT TIME YET
051.212 053          2735           DCX     H
051.213 066 012      2736           MVI     M,NL
051.215 043          2737           INX     H               TIME TO END LINE
051.216 076 001      2738           MVI     A,1
051.220 062 224 050  2739 PFI2      STA     LSTA            RESET COUNT
051.223 303 374 051  2740           JMP     PFI6            OUTPUT TO FILE
                     2741
                     2742 *         IS LONG FORM.
                     2743
051.226 001 005 000  2744 PFI3      LXI     B,DIR.FGN-DIR.EXT-3
051.231 353          2745           XCHG                    (DE) = LINE ADDR, (HL) = #PIO.DIR+DIR.EXT+3
051.232 011          2746           DAD     B               (HL) = #DIR.FGN
051.233 176          2747           MOV     A,M             (A) = (DIR.FGN)
051.234 043          2748           INX     H
051.235 043          2749           INX     H
051.236 116          2750           MOV     C,M             (C) = DIR.LSI = SECTORS USED IN LAST GROUP
000.000              2751           ERRNZ   DIR.LSI-DIR.FGN-2
051.237 353          2752           XCHG                    (DE) = ADDRESS OF LSI
051.240 325          2753           PUSH    D               SAVE #DIR.LSI
051.241 345          2754           PUSH    H               SAVE LINE ADDRESS
051.242 052 260 050  2755           LHLD    LSTE
051.245 157          2756           MOV     L,A
051.246 176          2757           MOV     A,M
051.247 315 351 053  2758           CALL    CFS             COMPUTE FILE ISZE
051.252 072 115 052  2759           LDA     PFIC            (A) = SECTORS PER GROUP
051.255 107          2760           MOV     B,A                                     /80.06.GC/
051.256 315 007 031  2761           CALL    $MU86           (HL) = SECTORS USED (EXCEPT FOR THOSE IN LAST GROUP)
                     2762
051.261 072 200 063  2763           LDA     ALLOCA                                  /80.06.sc/
051.264 247          2764           ANA     A                                       /80.06.sc/
051.265 312 271 051  2765           JZ      PFI3.5                                  /80.06.sc/
051.270 110          2766           MOV     C,B             Use Group Size instead if /ALL  /80.06.sc/
051.271              2767 PFI3.5    EQU     *                                       /80.06.sc/
                     2768
051.271 006 000      2769           MVI     B,0
051.273 011          2770           DAD     B               (HL) = SECTORS USED
051.274 104          2771           MOV     B,H
051.275 115          2772           MOV     C,L             (BC) = SECTORS USED COUNT
051.276 052 226 050  2773           LHLD    LSTC
051.301 011          2774           DAD     B
051.302 042 226 050  2775           SHLD    LSTC            ACCUMULATE COUNT OF SECTORS
051.305 341          2776           POP     H               (HL) = LINE ADDRESS
051.306 076 004      2777           MVI     A,4             3 DIGITS MAX                /80.05.sc/
051.310 315 371 060  2778           CALL    $UDDN           UNPACK COUNT
051.313 066 011      2779           MVI     M,TAB
051.315 043          2780           INX     H
```

```
051.316  321          2781          POP     D               (DE) = $DIR.LSI
                       2782
                       2783  *       TYPE DATE
                       2784
051.317  353          2785          XCHG
000.000               2786          ERRNZ   DIR.CRD-DIR.LSI-1
051.320  043          2787          INX     H               (HL) = $DIR.CRD
051.321  345          2788          PUSH    H
051.322  315 211 030  2789          CALL    $HLIHL
051.325  353          2790          XCHG
051.326  315 155 060  2791          CALL    $DAD            DECODE AUGUSTAN DATE
                       2792
                       2793  *       CODE FLAGS
                       2794
051.331  353          2795          XCHG                    (DE) = LINE ADDRESS
051.332  341          2796          POP     H               (HL) = $DIR.CRD
051.333  001 373 377  2797          LXI     B,DIR.FLG-DIR.CRD
051.336  011          2798          DAD     B               (HL) = ADDRESS OF DIRFLG
051.337  176          2799          MOV     A,M             (A) = FLAGS
051.340  353          2800          XCHG                    (HL) = LINE ADDRESS
051.341  247          2801          ANA     A
051.342  312 371 051  2802          JZ      PFI5.5          NO FLAGS
051.345  066 011      2803          MVI     M,TAB           TAB BEFORE FLAGS
051.347  043          2804          INX     H
051.350  021 105 052  2805          LXI     D,PFIB
051.353  207          2806  PFI4    ADD     A
051.354  322 364 051  2807          JNC     PFI5            NOT SET
051.357  365          2808          PUSH    PSW             SAVE FLAGS
051.360  032          2809          LDAX    D
051.361  167          2810          MOV     M,A
051.362  361          2811          POP     PSW             RESTORE FLAGS
051.363  043          2812          INX     H
051.364  023          2813  PFI5    INX     D               SET FLAG
051.365  247          2814          ANA     A
051.366  302 353 051  2815          JNZ     PFI4            MORE FLAGS SET
051.371  066 012      2816  PFI5.5  MVI     M,NL
051.373  043          2817          INX     H
                       2818
                       2819  *       LINE ALL BUILT, WRITE TO DESTFB
                       2820
051.374  021 346 325  2821  PFI6    LXI     D,-PFIA
051.377  031          2822          DAD     D
000.001               2823          IF      .PIP.
                       2824          MOV     B,H
                       2825          MOV     C,L             (BC) = LEN
                       2826          LXI     D,PFIA          (DE) = DATA FWA
                       2827          LXI     H,DESTFB
                       2828          JMP     $FWRIB          WRITE AND EXIT
                       2829          ELSE
052.000  175          2830          MOV     A,L             (A) = COUNT
052.001  041 032 052  2831          LXI     H,PFIA
052.004  303 156 057  2832          JMP     $TYPCC          TYPE LINE AND EXIT
                       2833          ENDIF
```

```
                                2835  **      PFI20 - COPY FILE NAME.
                                2836  *
                                2837  *       PFI20 COPIES A NAME FILED FROM THE DIRECTORY ENTRY TO A CODED
                                2838  *       LINE
                                2839  *
                                2840  *       EENTRY  (DE) = DIRECTORY ADDRESS
                                2841  *               (C) = NAME LENGTH
                                2842  *               (HL) = LINE ADDRESS
                                2843  *       EXIT    (DE) = (DE) + (C)
                                2844  *               'Z' SET IF MAX CHARACTERS COPIED
                                2845  *       USES    A,F,C,D,E,H,L
                                2846
                                2847
052.007  167                   2848  PFI19   MOV     M,A             COPY
052.010  043                   2849          INX     H
052.011  023                   2850          INX     D
052.012  015                   2851          DCR     C
052.013  310                   2852          RZ                      ALL COPIED
052.014  032                   2853  PFI20   LDAX    D
052.015  247                   2854          ANA     A
052.016  302 007 052           2855          JNZ     PFI19           GOT CHAR
                                2856
                                2857  *       NO NAME. (C) = COUNT LEFT
                                2858
052.021  173                   2859          MOV     A,E
052.022  201                   2860          ADD     C
052.023  137                   2861          MOV     E,A
052.024  172                   2862          MOV     A,D
052.025  316 000               2863          ACI     0
052.027  127                   2864          MOV     D,A
052.030  263                   2865          ORA     E               CLEAR 'Z'
052.031  311                   2866          RET
                                2867
052.032                         2868  PFIA    DS      0               BUFFER AREA FOR LINE BUILD
052.032  130 130 130 130 130    2869          DB      'XXXXXXXX.YYY   NNNN    DD-MMM-YY'
052.065  011 011 106           2870          DB      '               FLAGS
052.105  123 114 127           2871  PFIB    DB      'SLW'           CODES
052.110  040 061 062           2872  PFIB1   DB      ' 1234'          ('C' FOR CONTIGUOUS IS OPTIONAL)
000.000                         2873          ERRNZ   DIF.SYS-200Q
000.000                         2874          ERRNZ   DIF.LOC-100Q
000.000                         2875          ERRNZ   DIF.WP-40Q
000.000                         2876          ERRNZ   DIF.CNT-20Q
052.115  000                   2877  PFIC    DB      0               SECTORS PER GROUP FOR THIS DEVICE
```

```
                                2880  ***     VERSN   - PIP VERSION INFORMATION
                                2881  *
                                2882  *       DEST=/V[ERSION]
                                2883  *
                                2884  *       PRINT THE PIP VERSION INFORMATION TO THE 'DEST' FILE.
                                2885  *
                                2886
052.116                         2887  VERSN   EQU     *
                                2888
052.116  315 367 053           2889          CALL    CTS             CHECK FOR TARGET FILE SPECIFICATION
052.121  067                    2890          STC
052.122  302 026 053           2891          JNZ     ERROR           TARGET FILE SPECIFICATION ILLEGAL
052.125  041 374 065           2892          LXI     H,LINE
052.130  315 321 057           2893          CALL    $SOB            SKIP OVER ALL THE BLANKS ($DRS TURNS SWITCHES
052.133  176                    2894          MOV     A,M              TO BLANKS)
052.134  247                    2895          ANA     A
052.135  076 207                2896          MVI     A,PEC.SFI       SOURCE FILE ILLEGAL
052.137  067                    2897          STC
052.140  302 026 053           2898          JNZ     ERROR           ONLY ALLOW SWITCH ON LINE
052.143  315 136 031           2899          CALL    $TYPTX
                                2900
000.001                         2901          IF      .PIP.
                                2902          DB      'PIP'
                                2903          ELSE
052.146  117 116 105           2904          DB      'ONECOPY'
                                2905          ENDIF
                                2906
052.155  011 126 145           2907          DB      TAB,'Version:  '
052.170  062 056 060           2908          DB      VERS/16+'0','.',VERS&00001111B+'0'
052.173  212                    2909          DB      ENL
                                2910
052.174  311                    2911          RET
```

```
                                2914  **      ERROR PROCESSING ROUTINES
                                2915  *
                                2916
                                2917  ***     NAMERR - FILE TYPE ERROR, OCCURRED ON FILE WHOSE NAME
                                2918  *       IS NEXT UP IN NAMTAB.
                                2919  *
                                2920  *       PROCESS VIA $FERROR
                                2921
000.001                         2922          IF      .PIP.
                                2923  NAMERR  LXI     H,NAMTAB-FB.NAM
                                2924          JMP     $FERROR
                                2925          ELSE
052.175  052 270 063            2926  NAMERR  LHLD    NAMTPTR
052.200  001 346 377            2927          LXI     B,-FB.NAM
052.203  011                    2928          DAD     B
052.204  303 116 063            2929          JMP     $FERROR
                                2930
                                2931  **      ERROR ON FILE IN DESTFB
                                2932
052.207  041 231 063            2933  DESTERR LXI     H,DESTFB
052.212  303 116 063            2934          JMP     $FERROR
                                2935          ENDIF
                                2936
                                2937  **      INTERNAL ERRORS. SHOULD NOT OCCUR.
                                2938
052.215  076 061                2939  IERR1   MVI     A,'1'
052.217  303 234 052            2940          JMP     INTERR
                                2941
052.222  076 062                2942  IERR2   MVI     A,'2'
052.224  303 234 052            2943          JMP     INTERR
052.227  076 063                2944  IERR3   MVI     A,'3'
052.231  303 234 052            2945          JMP     INTERR
                                2946
                                2947
052.234  365                    2948  INTERR  PUSH    PSW             SAVE CODE
052.235  315 136 031            2949          CALL    $TYPTX
052.240  007 012 120            2950          DB      BELL,NL,'PIP INTERNAL ERROR ','#'+200Q
052.266  361                    2951          POP     PSW
052.267  315 075 061            2952          CALL    $WCHAR
052.272  315 136 031            2953          CALL    $TYPTX
052.275  012 124 110            2954          DB      NL,'THIS ERROR SHOULD NOT OCCUR. CONTACT HEATH TECHNICAL'
052.362  012 103 117            2955          DB      NL,'CORRESPONDENCE FOR ASSISTANCE.',NL
053.022  076 001                2956          MVI     A,1
053.024  377 000                2957          DB      SYSCALL,,EXIT           ABORT
```

```
                                2959  **      ERROR - GENERAL AND SYNTAX ERRORS NOT DIRECTLY ASSOCIATED
                                2960  *       WITH A VALID FILE NAME.
                                2961
                                2962
053.026  365                   2963  ERROR   PUSH    PSW             SAVE CODE
053.027  315 136 031           2964          CALL    $TYPTX
053.032  007 105 122           2965          DB      BELL,'ERROR -',' '+200Q
053.043  361                   2966          POP     PSW
053.044  247                   2967          ANA     A
053.045  372 057 053           2968          JM      ERROR1          IS PRODUCT ERROR
053.050  046 012               2969          MVI     H,NL            USE NL AS MESSAGE TRAIL CHAR
053.052  377 057               2970          DB      SYSCALL,,ERROR  LOOK UP SYSTEM ERROR
053.054  303 237 042           2971          JMP     RESTART
                                2972
                                2973  *       IS PRODUCT ERROR
                                2974
053.057  041 120 053           2975  ERROR1  LXI     H,ERRORA
053.062  276                   2976  ERROR2  CMP     M
053.063  043                   2977          INX     H
053.064  302 062 053           2978          JNE     ERROR2          FIND ERROR MESSAGE
000.000                        2979          IF      ONECOPY
053.067  315 136 031           2980          CALL    $TYPTX
053.072  007 117 116           2981          DB      BELL,'ONECOPY Error #',' '+200Q
                                2982          ENDIF
053.113  377 003               2983          DB      SYSCALL,,PRINT  PRINT MESSAGE
053.115  303 237 042           2984          JMP     RESTART
                                2985
053.120                        2986  ERRORA  DS      0               ERROR MESSAGES
000.001                        2987          IF      .PIP.
                                2988          DB      PEC.DF,'Device Format Error',ENL
                                2989          DB      PEC.DNC,'All Files Must Reside on the Same Device',ENL
                                2990          DB      PEC.TFI,'Destination File Specification is Illegal',ENL
                                2991          DB      PEC.CS,'Contradictory Switches Specified',ENL
                                2992          DB      PEC.IUW,'Illegal Use of Wildcard',ENL
                                2993          DB      PEC.IDF,'Illegal Destination File Format',ENL
                                2994          DB      PEC.SFI,'Source File Specification is Illegal',ENL
                                2995          ELSE
053.120  200 060 061           2996          DB      PEC.DF,'01',ENL
053.124  201 060 062           2997          DB      PEC.DNC,'02',ENL
053.130  203 060 063           2998          DB      PEC.TFI,'03',ENL
053.134  204 060 064           2999          DB      PEC.CS,'04',ENL
053.140  205 060 065           3000          DB      PEC.IUW,'05',ENL
053.144  206 060 066           3001          DB      PEC.IDF,'06',ENL
053.150  207 060 067           3002          DB      PEC.SFI,'07',ENL
053.154  210 060 070           3003          DB      PEC.FCI,'08',ENL
                                3004          ENDIF
```

```
                                3008  **      AEN - ADD ENTRY TO 'NAMTAB'
                                3009  *
                                3010  *       AEN EXPANDS THE FILE INFO IN PIO.XXX INTO A FILE DESCRIPTOR
                                3011  *       AND ENTERS IT IN THE NAMTAB TABLE.
                                3012  *
                                3013  *       ENTRY   NONE
                                3014  *       EXIT    'C' SET IF WILDCARD
                                3015  *       USES    ALL
                                3016
                                3017
053.160  041 232 053           3018  AEN     LXI     H,AENA
053.163  315 230 055           3019          CALL    CDA             CONVERT DIRECTORY FORMAT TO ASCII FORMAT
053.166  326 001               3020          SUI     1               'C' SET IF WILDCARD
053.170  365                   3021          PUSH    PSW             SAVE FLAG
053.171  052 264 063           3022          LHLD    NAMTLEN
053.174  001 021 000           3023          LXI     B,FB.NAML
053.177  011                   3024          DAD     B               INCREASE SIZE
053.200  042 264 063           3025          SHLD    NAMTLEN
053.203  353                   3026          XCHG                    (DE) = NEW LENGTH
053.204  052 266 063           3027          LHLD    NAMTMAX
053.207  175                   3028          MOV     A,L             SEE IF WILL OVERFLOW
053.210  223                   3029          SUB     E
053.211  174                   3030          MOV     A,H
053.212  232                   3031          SBB     D
053.213  334 246 056           3032          CC      INA             INCREASE NAMTAB ALLOCATION
053.216  041 073 066           3033          LXI     H,NAMTAB-FB.NAML
053.221  031                   3034          DAD     D               (HL) = *TO* ADDRESS
053.222  021 232 053           3035          LXI     D,AENA          (DE) = *FROM* ADDRESS
053.225  315 252 030           3036          CALL    $MOVE           MOVE ENTRY IN
053.230  361                   3037          POP     PSW             (PSW) = WILDCARD FLAG
053.231  311                   3038          RET
                                3039
053.232                         3040  AENA    DS      FB.NAML


                                3042  **      BSL - BUILD SOURCE FILE LIST.
                                3043  *
                                3044  *       BSL CRACKS THE LIST OF THE SOURCE FILES FROM THE COMMAND LINE AND
                                3045  *       BUILDS THEM INTO THE NAMTAB MANAGED TABLE.
                                3046  *       WILD CARDS ENCOUNTERED ARE EXPANDED.
                                3047  *
                                3048  *       ENTRY   (A) <> 0 IF TO ASK ABOUT '*.*' USE
                                3049  *       EXIT    'C' CLEAR IF OK
                                3050  *               'C' SET IF ERROR
                                3051  *                       (A) = CODE
                                3052  *       USES    ALL
                                3053
                                3054
053.253  062 324 053           3055  BSL     STA     BSLA            SAVE ASK FLAG
053.256  315 300 056           3056          CALL    LSN             LOCATE SOURCE NAME
                                3057
                                3058  *       GO THROUGH SOURCE LIST CRACKING NAMES
                                3059
053.261  176                   3060  BSL1    MOV     A,M
```

```
053.262  247            3061           ANA     A
053.263  310            3062           RZ                      ALL DONE
053.264  021 272 063     3063           LXI     D,DEFALT
053.267  315 136 054     3064           CALL    CAD             CONVERT ASCII NAME TO DIRECTORY FORMAT
053.272  330            3065           RC                      ERROR
053.273  315 104 057     3066           CALL    SND             SET NEW DEFAULTS
053.276  345            3067           PUSH    H               SAVE LINE ADDRESS
053.277  072 324 053     3068           LDA     BSLA
053.302  247            3069           ANA     A
053.303  304 325 053     3070           CNZ     CCW             CHECK FOR COMPLETE WILDCARD (*.*)
053.306  332 237 042     3071           JC      RESTART         USER CHICKENED OUT                /79.12.GC/
053.311  315 321 055     3072           CALL    EWS             EXPAND WILDCARD SPECIFICATION
053.314  341            3073  BSL2     POP     H               RESTORE LINE ADDRESS
053.315  330            3074           RC                      USER REFUSED *.*
053.316  315 067 057     3075           CALL    SFS             SKIP FILE SEPERATOR (BLANKS AND/OR COMMA)
053.321  303 261 053     3076           JMP     BSL1            DO MORE
                        3077
053.324  000            3078  BSLA     DB      0               <>0 IF TO CHECK FOR *.*


                        3080  **       CCW - CHECK FOR COMPLETE WILDCARD.
                        3081  *
                        3082  *        CCW IS CALLED WITH A NAME CRACKED INTO PIO.XXX, TO SEE IF
                        3083  *        IT IS A *.* SPECIFICATION.
                        3084  *
                        3085  *        IF SO, CCW ASKS,
                        3086  *
                        3087  *        DELETE ALL FILES ON DEV: ?!? (Y/N)
                        3088  *
                        3089  *        THE USER REPLY IS ACCEPTED AND DECODED.
                        3090  *
                        3091  *        ENTRY   NONE
                        3092  *        EXIT    'C' CLEAR IF NOT *.*, OR 'Y' REPLIED
                        3093  *                'C' SET IF *.* AND NOT 'Y'
                        3094  *        USES    A,F,B,H,L
                        3095
                        3096
053.325  041 324 065     3097  CCW      LXI     H,PIO.DIR+DIR.NAM
000.001                 3098           IF      .PIP.
                        3099           MVI     B,8+3
                        3100           MVI     A,200Q
                        3101  CCW1     ANA     M               SEE IF ALL HAVE 200Q BIT SET
                        3102           INX     H
                        3103           DCR     B
                        3104           JNZ     CCW1
                        3105           ANA     A
                        3106           RP                      NOT *.*
                        3107
                        3108  *        IS *.*
                        3109
                        3110           CALL    $TYPTX
                        3111           DB      BELL,'!?! DELETE ALL FILES ON',' '+200Q
                        3112           LXI     H,PIO.DEV
                        3113           MVI     A,3
```

```
                                3114            CALL    $TYPCC          TYPE DEVICE NAME
                                3115            CALL    $TYPTX
                                3116            DB      '; (Y/N)?',' '+200Q
                                3117            LXI     H,DESTBUF
                                3118            CALL    $RTL.           READ REPLY
                                3119            LDA     DESTBUF
                                3120            CPI     'Y'
                                3121            RE                      IS OK
                                3122            STC
                                3123            MVI     A,PEC.IUW       FLAG ILLEGAL USE OF WILDCARD
                                3124            ENDIF
          053.330  311          3125            RET                     FORGET IT


                                3127  **        CFE - CHECK FILE ELIGIBILITY.
                                3128  *
                                3129  *         CFE CHECKS TO SEE IF A WILDCARD-SELECTED FILE IS ELIGIBLE
                                3130  *         FOR PROCESSING. IF THE FILE IS FLAGGED SYSTEM, AND /S IS NOT
                                3131  *         SPECIFIED, THE FILE IS NOT ELIGIBLE.
                                3132  *
                                3133  *         ENTRY   (HL) = DIRECTORY ENTRY POINTER
                                3134  *         EXIT    'Z' SET IF ELIGIBLE
                                3135  *         USES    A,F
                                3136
                                3137
          053.331  345          3138  CFE       PUSH    H
          053.332  076 016      3139            MVI     A,DIR.FLG
          053.334  315 101 030  3140            CALL    $DADA.
          053.337  176          3141            MOV     A,M             (A) = FLAG
          053.340  346 200      3142            ANI     DIF.SYS
          053.342  341          3143            POP     H
          053.343  310          3144            RZ                      ELIGIBLE
          053.344  072 205 063  3145            LDA     SYSTEM          CHECK /S FLAG
          053.347  247          3146            ANA     A
          053.350  311          3147            RET


                                3149  **        CFS - COMPUTE FILE SIZE
                                3150  *
                                3151  *         CFS COMPUTES THE SIZE OF A FILE. THE DEVICE'S GRT MUST BE IN
                                3152  *         THE 'GRT' BUFFER.
                                3153  *
                                3154  *         ENTRY   (A) = FIRST GROUP NUMBER
                                3155  *         EXIT    (DE) = SIZE
                                3156  *         USES    ALL
                                3157
                                3158
          053.351  052 260 050  3159  CFS       LHLD    LSTE
          053.354  021 000 000  3160  CFS.      LXI     D,0
          053.357  247          3161  CFS1      ANA     A
          053.360  310          3162            RZ                      ALL DONE
          053.361  157          3163            MOV     L,A
```

```
053.362  176          3164          MOV     A,M             (A) = NEXT GRT
053.363  023          3165          INX     D
053.364  303 357 053  3166          JMP     CFS1            TRY AGAIN


                      3168  **      CTS     -  CHECK TARGET FILE SPECIFICATION
                      3169  *
                      3170  *       CTS CHECKS FOR A TARGET FILE SPECIFICATION
                      3171  *
                      3172  *
                      3173  *       ENTRY   NONE
                      3174  *
                      3175  *       EXIT    (PSW)   = 'Z' SET IF NO TARGET FILE
                      3176  *                       = 'Z' CLEAR IF    TARGET FILE
                      3177  *                         (A)  = PEC.TFI ERROR CODE
                      3178  *
                      3179  *       USES    (PSW),(HL)
                      3180  *
                      3181
053.367  315 300 056  3182  CTS     CALL    LSN             (HL) = ADDRESS OF FIRST SOURCE NAME
053.372  021 004 312  3183          LXI     D,-LINE
053.375  031          3184          DAD     D               (HL) == 0 IF NO '=' IN COMMAND LINE
053.376  175          3185          MOV     A,L
053.377  264          3186          ORA     H
054.000  310          3187          RZ                      NO TARGET FILE
054.001  076 203      3188          MVI     A,PEC.TFI       TARGET FILE ILLEGAL
054.003  311          3189          RET                     TARGET FILE SPECIFIED


                      3191  **      CWM - CHECK WILDCARD MATCH.
                      3192  *
                      3193  *       CWM CHECKS TO SEE IF A WILDCARDED FIELD MATCHES A NON-WILDCARDED
                      3194  *       FIELD.
                      3195  *
                      3196  *       ENTRY   (DE) = ADDRESS OF WC NAME
                      3197  *               (HL) = ADDRESS OF NON/WC NAME
                      3198  *               (B) = NUMBER OF CHARACTERS TO CHECK
                      3199  *       EXIT    'Z' SET IF MATCH
                      3200  *                (HL) = (HL)+(B)
                      3201  *                (DE) = (DE) = (B)
                      3202  *               'Z' CLEAR IF NO MATCH
                      3203  *       USES    A,F,B,D,E,H,L
                      3204
                      3205
054.004  032          3206  CWM     LDAX    D
054.005  247          3207          ANA     A
054.006  372 013 054  3208          JM      CWM1            IS MATCH
054.011  276          3209          CMP     M
054.012  300          3210          RNE                     NO MATCH
054.013  023          3211  CWM1    INX     D
054.014  043          3212          INX     H               ADVANCE ADDRESSES
054.015  005          3213          DCR     B
```

```
054.016  302 004 054  3214          JNZ     CWM             GO FOR MORE
054.021  311         3215          RET                     GOT MATCH


                      3217  **      DDF - DECODE DESTINATION FILE.
                      3218  *
                      3219  *       DDF DECODES THE DESTINATION FILE NAME FROM THE COMMAND LINE.
                      3220  *
                      3221  *       IF NO DESTINATION NAME IS SPECIFIED, IT DEFAULTS TO
                      3222  *
                      3223  *       KB:PIPDEST.JGL
                      3224  *
                      3225  *       ENTRY   NONE
                      3226  *       EXIT    'C' CLEAR IF OK
                      3227  *                (A) = 0 IF NAME HAS WILDCARDS
                      3228  *                (A) = 1 IF NO WILDCARD USED
                      3229  *                DESTFB+FB.NAM CONTAINS A COMPLETE DESTINATION FILE NAME
                      3230  *                (HL) = COMMAND LINE POINTER UPDATED
                      3231  *               'C' SET IF ERROR
                      3232  *                (A) = CODE
                      3233  *       USES    ALL
                      3234
                      3235
054.022  021 374 065  3236  DDF     LXI     D,LINE
054.025  142         3237          MOV     H,D
054.026  153         3238          MOV     L,E             (HL) = COMMAND POINTER
054.027  032         3239  DDF1    LDAX    D
054.030  023         3240          INX     D
054.031  376 075     3241          CPI     '='
054.033  312 045 054  3242          JE      DDF2            HAVE A SOURCE FILE
054.036  247         3243          ANA     A
054.037  302 027 054  3244          JNZ     DDF1            MORE TO CHECK
054.042  041 122 054  3245  DDF1.0  LXI     H,DDFA          USE DEFAULT
                      3246
                      3247  *       (HL) = ADDRESS FOR NAME
                      3248
054.045  021 272 063  3249  DDF2    LXI     D,DEFALT
054.050  315 136 054  3250          CALL    CAD             CONVERT ASCII NAME TO DIRECTORY FORMAT
054.053  330         3251          RC                      ERROR
054.054  312 042 054  3252          JZ      DDF1.0          NO FILE NAME SPECIFIED, USE DEFAULT
054.057  176         3253          MOV     A,M
054.060  376 075     3254          CPI     '='
054.062  076 206     3255          MVI     A,PEC.IDF       ASSUME ILLEGAL DESTINATION FORMAT
054.064  067         3256          STC
054.065  300         3257          RNE                     MUST HAVE '='
                      3258
                      3259  *       HAVE NAME DECODED. EXPAND INTO DESTFB+FB.NAM
                      3260
054.066  041 243 063  3261          LXI     H,DESTFB+FB.NAM
000.001              3262          IF      .PIP.
                      3263          JMP     CDA             CONVERT DIRECTORY FORMAT TO ASCII FORMAT
                      3264          ELSE    ONECOPY
054.071  315 230 055  3265          CALL    CDA             CONVERT DIRECTORY FORMAT TO ASCII FORMAT
054.074  365         3266          PUSH    PSW             SAVE CODE
```

```
054.075  016 003      3267          MVI     C,3
054.077  021 133 054  3268          LXI     D,DDFB
054.102  041 243 063  3269          LXI     H,DESTFB+FB.NAM
054.105  315 060 030  3270          CALL    $COMP           SEE IF DEVICE IS SY0
054.110  302 115 054  3271          JNE     DDF3            IS ERROR
054.113  361          3272          POP     PSW
054.114  311          3273          RET                     RETURN WITH 'C' CLEAR
                      3274
054.115  361          3275  DDF3    POP     PSW             ERROR, ILLEGAL DEVICE CODE
054.116  076 005      3276          MVI     A,EC.DNS
054.120  067          3277          STC
054.121  311          3278          RET
                      3279
054.122  123 131 060  3280  DDFA    DB      'SY0:*.*=',0    DEFAULT TARGET FOR ONECOPY
054.133  123 131 060  3281  DDFB    DB      'SY0'           REQUIRED DEVICE SPECIFICATION FOR ONECOPY
                      3282          ELSE
                      3283
                      3284  DDFA    DB      'TT:PIPDEST.JGL=',0
                      3285          ENDIF


                      3287  **      CAD - CONVERT ASCII FILE NAME INTO DIRECTORY FORMAT.
                      3288  *
                      3289  *       CAD CRACKS AN ALPHANUMERIC FILE DESCRIPTION, OF THE FORM
                      3290  *
                      3291  *       DEV:NAME.EXT
                      3292  *
                      3293  *       INTO THE PIO.XXX FIELDS.
                      3294  *
                      3295  *       THE DEFAULT BLOCK DETERMINES THE VALUES FOR THE DEVICE AND EXTENSION
                      3296  *       FIELDS, IF THEY ARE UNSPECIFIED. IF *CAD* IS ENTERED
                      3297  *       AT *CAD*, AN UNSPECIFIED NAME FIELD IS RETURNED AS ZERO BYTES.
                      3298  *       IF ENTERED AT *CAD.*, AN UNSPECIFIED NAME FIELD IS
                      3299  *       RETURNED AS 200Q (MATCH-ONE) BYTES.
                      3300  *
                      3301  *       ENTRY   (DE) = POINT TO DEFAULT BLOCK
                      3302  *               (HL) = POINTER TO TEXT
                      3303  *       EXIT    'C' SET IF ERROR
                      3304  *                (A) = ERROR CODE
                      3305  *               'C' CLEAR IF OK
                      3306  *                (HL) = POINTS PAST FILE NAME
                      3307  *                'Z' SET IF NULL NAME
                      3308  *                'Z' CLEAR IF NON-NULL
                      3309  *                 PIO.DIR.NAM = NAME
                      3310  *                 PIO.DIR.EXT = EXTENSION
                      3311  *                 PIO.DEV = DEVICE CODE
                      3312  *                 PIO.UNI = UNIT NUMBER (ASCII DIGIT)
                      3313  *       USES    ALL
                      3314
                      3315
054.136  257          3316  CAD     XRA     A               SET TO NULLS
054.137  303 144 054  3317          JMP     CAD0
                      3318
054.142  076 200      3319  CAD.    MVI     A,200Q
```

```
054.144  345            3320  CAD0    PUSH    H
054.145  062 022 055     3321          STA     CADA            SAVE DEFAULT VALUE
                         3322
                         3323  *       SET DEFAULTS IN PIO.xxx
                         3324
054.150  041 321 065     3325          LXI     H,PIO.DEV
054.153  001 003 000     3326          LXI     B,3
054.156  315 252 030     3327          CALL    $MOVE           SET DEFALUT DEVICE
054.161  001 003 000     3328          LXI     B,3
054.164  041 334 065     3329          LXI     H,PIO.DIR+DIR.EXT
054.167  315 252 030     3330          CALL    $MOVE           SET DEFAULT EXTENSION
054.172  341            3331          POP     H
054.173  315 321 057     3332          CALL    $SOB            SKIP BLANKS
054.176  006 000         3333          MVI     B,0
054.200  376 077         3334          CPI     '?'
054.202  312 231 054     3335          JE      CAD1            IS '?'
054.205  376 052         3336          CPI     '*'
054.207  312 231 054     3337          JE      CAD1            IS '*'
054.212  376 056         3338          CPI     '.'
054.214  312 231 054     3339          JE      CAD1            IS '.'
054.217  376 101         3340          CPI     'A'
054.221  332 003 055     3341          JC      CAD4            NOT NAME
054.224  376 133         3342          CPI     'Z'+1
054.226  322 003 055     3343          JNC     CAD4            NOT NAME
                         3344
                         3345  *       HAVE ALPHA STRING, CRACK IT
                         3346
054.231  315 023 055     3347  CAD1    CALL    DNT             DECODE NEXT TOKEN
054.234  332 016 055     3348          JC      CAD5            ERROR
054.237  376 072         3349          CPI     ':'
054.241  302 306 054     3350          JNE     CAD2            NOT DEVICE
                         3351
                         3352  *       HAVE EXPLICIT DEVICE
                         3353
054.244  043            3354          INX     H               SKIP ':'
054.245  076 003         3355          MVI     A,3
054.247  271            3356          CMP     C
054.250  332 016 055     3357          JC      CAD5            TOO MANY CHARACTERS
054.253  076 001         3358          MVI     A,PIO.UNI-PIO.DEV-1                     /2.0b/
054.255  271            3359          CMP     C                                       /2.0b/
054.256  322 016 055     3360          JNC     CAD5            Too Few characters      /2.0b/
                         3361
054.261  076 060         3362          MVI     A,'0'                                   /2.0b/
054.263  062 323 065     3363          STA     PIO.UNI         Assume Unit 0           /2.0b/
054.266  006 000         3364          MVI     B,0             BC = Move Count         /2.0b/
054.270  345            3365          PUSH    H               SAVE (HL)
054.271  041 321 065     3366          LXI     H,PIO.DEV
054.274  315 252 030     3367          CALL    $MOVE           SET EXPLICIT DEVICE
054.277  341            3368          POP     H
054.300  315 023 055     3369          CALL    DNT             DECODE NEXT TOKEN
054.303  332 016 055     3370          JC      CAD5            ERROR
                         3371
                         3372  *       DECODE NAME
                         3373
054.306  001 010 000     3374  CAD2    LXI     B,8             (BC) = COUNT
054.311  345            3375          PUSH    H               SAVE TEXT ADDR
```

```
                        3376
                        3377  *       SEE IF NAME IS  UNSPECIFIED
                        3378
054.312  041 324 065    3379          LXI     H,PIO.DIR+DIR.NAM
054.315  345            3380          PUSH    H               SAVE ADDRESS OF DIR.NAM
054.316  315 252 030    3381          CALL    $MOVE           MOVE IN NAME
054.321  341            3382          POP     H               (HL) = $PIO.DIR+DIR.NAM
054.322  176            3383          MOV     A,M
054.323  247            3384          ANA     A
054.324  302 342 054    3385          JNZ     CAD2.6          IS SPECIFIED
054.327  072 022 055    3386          LDA     CADA            (A) = FILL CHARACTER
054.332  016 010        3387          MVI     C,8             (C) = COUNT
054.334  167            3388  CAD2.4  MOV     M,A
054.335  043            3389          INX     H
054.336  015            3390          DCR     C
054.337  302 334 054    3391          JNZ     CAD2.4
054.342  341            3392  CAD2.6  POP     H
054.343  176            3393          MOV     A,M             (A) = DELIMITER
054.344  376 056        3394          CPI     '.'
054.346  302 001 055    3395          JNE     CAD3            NOT EXTENSION
                        3396
                        3397  *       HAVE EXPLICIT EXTENSION
                        3398
054.351  043            3399          INX     H
054.352  315 023 055    3400          CALL    DNT
054.355  332 016 055    3401          JC      CAD5            ERROR
054.360  076 003        3402          MVI     A,3
054.362  271            3403          CMP     C
054.363  332 016 055    3404          JC      CAD5            TOO LONG
054.366  001 003 000    3405          LXI     B,3
054.371  345            3406          PUSH    H               SAVE TEXT POINTER
054.372  041 334 065    3407          LXI     H,PIO.DIR+DIR.EXT
054.375  315 252 030    3408          CALL    $MOVE           MOVE EXTENSION
055.000  341            3409          POP     H
                        3410
                        3411  *       DONE WITH NAME. MUST HAVE LEGIT DELIMITER
                        3412
055.001  006 001        3413  CAD3    MVI     B,1             (B) = NAME PRESENT FLAG
                        3414
                        3415  *       END OF NAME. EXIT
                        3416  *       (B) = 0 IF NULL; (B) <> 0 IF NON-NULL
                        3417
055.003  315 321 057    3418  CAD4    CALL    $SOB            SKIP BLANKS
055.006  176            3419          MOV     A,M             (A) = NEXT CHARACTER
055.007  315 134 057    3420          CALL    $CFD            CHECK FILE NAME DELIMITER
055.012  330            3421          RC                      ERROR
055.013  170            3422          MOV     A,B
055.014  247            3423          ANA     A               SET 'Z' IF NULL
055.015  311            3424          RET
                        3425
                        3426  *       ERROR
                        3427
055.016  076 007        3428  CAD5    MVI     A,EC.IFN        ILLEGAL FILE NAME
055.020  067            3429          STC
055.021  311            3430          RET
                        3431
```

```
055.022  000            3432  CADA    DB      0               FILL CHARACTER FOR OMITTED NAME FIELD


                         3434  **      DNT - DECODE NEXT TOKEN.
                         3435  *
                         3436  *       DNT COPIES THE NEXT ALPHANUMERIC FIELD INTO A ZERO-FILLED WORK AREA.
                         3437  *
                         3438  *       ENTRY   (HL) = TEXT POINTER
                         3439  *       EXIT    'C' SET IF ERROR
                         3440  *               'C' CLEAR IF OK
                         3441  *               (A) = DELIMTER CHARACTER
                         3442  *               (HL) UPDATED TO DELIMITER CHARACTER
                         3443  *               (DNTA) = STRING
                         3444  *               (C) = LENGTH
                         3445  *               (DE) = #DNTA
                         3446  *       USES    ALL
                         3447
                         3448
055.023  021 135 055    3449  DNT     LXI     D,DNTA
055.026  016 011        3450          MVI     C,9             (C) = SIZE OF DNTA
055.030  101            3451          MOV     B,C             (B) = MAX ALLOWED +1
055.031  257            3452          XRA     A
055.032  022            3453  DNT1    STAX    D               ZERO BUFFER
055.033  023            3454          INX     D
055.034  015            3455          DCR     C
055.035  302 032 055    3456          JNZ     DNT1
055.040  021 135 055    3457          LXI     D,DNTA
                         3458
                         3459  *       COPY CHARACTERS
                         3460
055.043  176            3461  DNT2    MOV     A,M
055.044  376 077        3462          CPI     '?'
055.046  076 200        3463          MVI     A,200Q
055.050  312 105 055    3464          JE      DNT3            IS MATCHONE
055.053  176            3465          MOV     A,M
055.054  376 052        3466          CPI     '*'
055.056  312 117 055    3467          JE      DNT5            IS WILDCARD
055.061  376 060        3468          CPI     '0'
055.063  332 130 055    3469          JC      DNT4            NOT ALPHANUMERIC
055.066  376 072        3470          CPI     '9'+1
055.070  332 105 055    3471          JC      DNT3            NUMERIC
055.073  376 101        3472          CPI     'A'
055.075  332 130 055    3473          JC      DNT4            DELIMITER
055.100  376 133        3474          CPI     'Z'+1
055.102  322 130 055    3475          JNC     DNT4            DELIMITER
                         3476
                         3477  *       HAVE GOOD CHARACTER
                         3478
055.105  022            3479  DNT3    STAX    D               STORE CHAR
055.106  023            3480          INX     D
055.107  043            3481          INX     H
055.110  014            3482          INR     C               COUNT
055.111  005            3483          DCR     B               LIMIT DECREMENT
055.112  302 043 055    3484          JNZ     DNT2            NOT OVERFLOW
```

```
                        3485
                        3486  *       OVERFLOW
                        3487
055.115  067            3488          STC       FLAG ERR
055.116  311            3489          RET
                        3490
                        3491  *       IS '*' WILDCARD
                        3492
055.117  076 200        3493  DNT5    MVI       A,200Q
055.121  022            3494          STAX      D
055.122  023            3495          INX       D
055.123  005            3496          DCR       B
055.124  302 117 055    3497          JNZ       DNT5        FILL WITH MATCH ONE
055.127  043            3498          INX       H           SKIP '*'
                        3499
                        3500  *       END OF STRING
                        3501
055.130  247            3502  DNT4    ANA       A           CLEAR 'C'
055.131  021 135 055    3503          LXI       D,DNTA      SET POINTER
055.134  311            3504          RET
                        3505
055.135                 3506  DNTA    DS        9           WORK AREA


                        3508  **      EBM - EXPAND BUFFER TO MAXIMUM.
                        3509  *
                        3510  *       EBM IS CALLED TO EXPAND THE BUFFER 'BUF' TO THE MAXIMUM SIZE,
                        3511  *       WHICH DOES NOT REQUIRE THE OVERLAYING OF THE SYSTEM.
                        3512  *
                        3513  *       ENTRY   NONE
                        3514  *       EXIT    (BUFSIZ) = BUFFER SIZE (MULTIPLE OF 256)
                        3515  *       USES    ALL
                        3516
                        3517
055.146  052 320 040    3518  EBM     LHLD      S.SYSM
055.151  345            3519          PUSH      H
055.152  052 350 040    3520          LHLD      S.OFWA
055.155  021 006 000    3521          LXI       D,OVL0*OVL.ENS+OVL.FLB
055.160  031            3522          DAD       D           (HL) = ADDR. OF OVL0 OVL.FLB ENTRY
055.161  076 002        3523          MVI       A,OVL.RES
055.163  246            3524          ANA       M
055.164  021 010 000    3525          LXI       D,OVL.ENS
055.167  031            3526          DAD       D           (HL) = ADDR. OF OVL1 OVL.FLB ENTRY
000.000                 3527          ERRNZ     OVL1-OVL0-1
055.170  246            3528          ANA       M
055.171  302 206 055    3529          JNZ       EBM1        OVL0 AND OVL1 ARE PERM. RESIDENT
055.174  052 324 040    3530          LHLD      S.OMAX
055.177  315 224 030    3531          CALL      $CHL
055.202  353            3532          XCHG
055.203  341            3533          POP       H
055.204  031            3534          DAD       D           (HL) = NEW ADDRESS SOUGHT
055.205  345            3535          PUSH      H
                        3536
055.206  341            3537  EBM1    POP       H
```

```
055.207  021 372 377  3538          LXI     D,-6
055.212  031          3539          DAD     D               (HL) = NEW ADDRESS SOUGHT
055.213  377 052      3540          DB      SYSCALL,.SETTP
055.215  332 215 052  3541          JC      IERR1           INTERNAL ERROR 1
055.220  052 322 040  3542          LHLD    S.USRM
000.001               3543          IF      .PIP.
                      3544          XCHG
                      3545          LHLD    BUFPTR
                      3546          CALL    $CHL            (HL) = - BUFFER FWA
                      3547          DAD     D
                      3548          MVI     L,0
                      3549          SHLD    BUFSIZ
                      3550          MVI     A,BUFMINL/256-1
                      3551          CMP     H
                      3552          RC                      IF OK
                      3553          MVI     A,EC.NEM
                      3554          JMP     ERROR           NOT ENOUGH MEMORY
                      3555
                      3556          ELSE
                      3557
055.223  174          3558          MOV     A,H             (A) = LIMIT/256
055.224  062 114 063  3559          STA     OBUFLIM         SET LIMIT
055.227  311          3560          RET
                      3561          ENDIF


                      3563  **      CDA - CONVERT DIRECTORY FORMAT TO ASCII.
                      3564  *
                      3565  *       CDA COPIES A DIRECTORY ENTRY FROM PIO.XXX TO A TARGET FIELD.
                      3566  *       THE DEVICE SPECIFICATION (IN PIO.DEV AND PIO.UNI) IS ALSO ENCODED.
                      3567  *       THE TARGET FIELD IS LEFT IN THE FORM:
                      3568  *
                      3569  *       DEV:NAME.XXX <00>
                      3570  *
                      3571  *       ENTRY   (HL) = FWA NAME FIELD
                      3572  *       EXIT    (A) = 0, HAVE WILDCARD
                      3573  *                   = 1, NO WILDCARDS USED
                      3574  *               'C' CLEAR
                      3575  *       USES    ALL
                      3576
                      3577
055.230  001 000 003  3578  CDA     LXI     B,3*256         (B) = CHARACTER COUNT, (C) = WILDCARD FLAG
055.233  021 321 065  3579          LXI     D,PIO.DEV
055.236  315 274 055  3580          CALL    CDA5            COPY IT
055.241  066 072      3581          MVI     M,':'
055.243  043          3582          INX     H
055.244  006 010      3583          MVI     B,8
055.246  021 324 065  3584          LXI     D,PIO.DIR+DIR.NAM
055.251  315 274 055  3585          CALL    CDA5            COPY IT
055.254  066 056      3586          MVI     M,'.'
055.256  043          3587          INX     H
055.257  006 003      3588          MVI     B,3
000.000               3589          ERRNZ   DIR.EXT-DIR.NAM-8
055.261  315 274 055  3590          CALL    CDA5            COPY IT
```

```
055.264  066 000      3591          MVI     M,0             FLAG END OF NAME
055.266  171          3592          MOV     A,C             (A) (BIT 7) = 1 IF WILDCARDS
055.267  007          3593          RLC
055.270  057          3594          CMA
055.271  346 001      3595          ANI     1               =0 IF WILDCARD
055.273  311          3596          RET

                      3598  **      CDA5 - CONVERT DIRECTORY FIELD TO ASCII.
                      3599  *
                      3600  *       ZEROS ARE IGNORED, 200Q WILDCARDS ARE MAPPED TO '?'
                      3601  *
                      3602  *       ENTRY   (DE) = FROM
                      3603  *               (HL) = TO
                      3604  *               (B) = COUNT
                      3605  *               (C) = ORA ACCUMULATOR
                      3606  *       EXIT    (DE) ADVANCED
                      3607  *               (HL) = (HL)+(B)
                      3608  *               (C) = (C) .OR. (FROM CHARACTERS PROCESSED)
                      3609  *       USES    ALL
                      3610
                      3611
055.274  032          3612  CDA5    LDAX    D               (A) = CHARACTER
055.275  261          3613          ORA     C
055.276  117          3614          MOV     C,A
055.277  032          3615          LDAX    D
055.300  023          3616          INX     D
055.301  247          3617          ANA     A
055.302  312 314 055  3618          JZ      CDA7            IS 00
055.305  362 312 055  3619          JP      CDA6            NOT 200Q
055.310  076 077      3620          MVI     A,'?'
055.312  167          3621  CDA6    MOV     M,A
055.313  043          3622          INX     H               INCREMENT TO
055.314  005          3623  CDA7    DCR     B
055.315  302 274 055  3624          JNZ     CDA5            IF MORE TO GO
055.320  311          3625          RET


                      3627  **      EWS - EXPAND WILDCARD SPECIFICATION.
                      3628  *
                      3629  *       DWS ENTERS THE FILE NAME IN FIO.XXX INTO THE MANAGED TABLE
                      3630  *       NAMTAB. IF THE FILE NAME CONTAINS WILDCARDS, THE DIRECTORY
                      3631  *       IS READ FOR ELIGIBLE FILES.
                      3632  *
                      3633  *       ENTRY   FIO.XXX = FILE NAME
                      3634  *       EXIT    'C' CLEAR IF OK
                      3635  *               'C' SET IF ERROR
                      3636  *       USES    ALL
                      3637
                      3638
055.321  315 160 053  3639  EWS     CALL    AEN             TRY TO ENTER IT
055.324  320          3640          RNC                     NO WILDCARDS, AM DONE
                      3641
                      3642  *       IS WILDCARD, LOOK UP DEVICE TYPE
```

```
                                3643
055.325  052 264 063  3644          LHLD    NAMTLEN
055.330  021 073 066  3645          LXI     D,NAMTAB-FB.NAML
055.333  031          3646          DAD     D               (HL) = ADDRESS OF LAST ENTRY
055.334  315 136 054  3647          CALL    CAD             CONVERT ASCII NAME TO DIRECTORY FORMAT
055.337  330          3648          RC                      ERROR
055.340  052 264 063  3649          LHLD    NAMTLEN
055.343  021 357 377  3650          LXI     D,-FB.NAML
055.346  031          3651          DAD     D
055.347  042 264 063  3652          SHLD    NAMTLEN         REMOVE WILDCARD FROM TABLE
055.352  315 044 061  3653          CALL    $MOVEL
055.355  003 000 321  3654          DW      3,PIO.DEV,DIRNAM        SET DIRECTORY NAME IN XXX:DIRECT.SYS
055.363  315 044 061  3655          CALL    $MOVEL
055.366  013 000 324  3656          DW      8+3,PIO.DIR+DIR.NAM,EWSC        SAVE WILDCARD PATTERN
055.374  001 163 056  3657          LXI     B,EWSB
055.377  041 206 063  3658          LXI     H,DIRNAM
056.002  377 053      3659          DB      SYSCALL,.DECODE GET INFORMATION ABOUT DEVICE
056.004  330          3660          RC                      ERROR
056.005  072 163 056  3661          LDA     EWSB            SEE IF A DIRECTORY DEVICE
056.010  346 001      3662          ANI     DT.DD
056.012  076 005      3663          MVI     A,EC.DNS                ASSUME DEVICE NOT SUITABLE
056.014  067          3664          STC
056.015  310          3665          RZ                      ERROR
                                3666
                                3667  *       IS DIRECTORY DEVICE. OPEN DIRECTORY
                                3668
056.016  041 206 063  3669          LXI     H,DIRNAM
056.021  076 002      3670          MVI     A,CN.DIR
056.023  377 042      3671          DB      SYSCALL,.OPENR
056.025  076 200      3672          MVI     A,PEC.DF
056.027  330          3673          RC                      DEVICE FORMAT FAILURE
                                3674
                                3675  *       READ DIRECTORY ENTRYS FOR MATCH
                                3676
056.030  315 234 056  3677  EWS1    CALL    GDWP            DE = DIRECTORY WORKSPACE PTR    /79.11.GC/
056.033  001 000 002  3678          LXI     B,512
056.036  076 002      3679          MVI     A,CN.DIR
056.040  325          3680          PUSH    D               SAVE ADDRESS
056.041  377 004      3681          DB      SYSCALL,.READ   READ BLOCK
056.043  341          3682          POP     H               (HL) = DIRECTORY ADDRESS
056.044  332 150 056  3683          JC      EWS7            ALL DONE
                                3684
                                3685  *       LOOK AT DIRECTORY BLOCK FOR MATCHES
                                3686
056.047  345          3687          PUSH    H                                       /79.11.GC/
056.050  315 242 056  3688          CALL    GDWP.                                   /79.11.GC/
056.053  315 052 060  3689          CALL    $INDLB                                  /79.11.GC/
056.056  373 001      3690          DW      DIS.ENL         A  = DIRECTORY ENTRY LENGTH   /79.11.GC/
056.060  341          3691          POP     H                                       /79.11.GC/
                                3692
056.061  117          3693          MOV     C,A             (C) = LENGTH
                                3694
                                3695  *       CHECK NEXT ENTRY
                                3696
056.062  176          3697  EWS3    MOV     A,M             (A) = 1ST CHAR THIS ENTRY
056.063  247          3698          ANA     A
```

```
056.064  312 030 056  3699          JZ      EWS1            END OF BLOCK
000.000               3700          ERRNZ   DF.EMP-377Q
056.067  074        3701          INR     A
056.070  312 142 056  3702          JZ      EWS6            ENTRY EMPTY
000.000               3703          ERRNZ   DF.CLR-376Q
056.073  074        3704          INR     A
056.074  312 150 056  3705          JZ      EWS7            END OF LIST
056.077  315 331 053  3706          CALL    CFE             CHECK FOR FILE ELIGIBILITY
056.102  302 142 056  3707          JNZ     EWS6            NOT TO PROCESS
056.105  345        3708          PUSH    H
056.106  021 221 056  3709          LXI     D,EWSC
056.111  006 013     3710          MVI     B,8+3
056.113  315 004 054  3711          CALL    CWM             CHECK WILDCARD MATCH
056.116  302 141 056  3712          JNZ     EWS4            NO MATCH
                      3713
                      3714  *       HAVE MATCH, ADD TO LSIT
                      3715
056.121  321        3716          POP     D               (DE) = FROM
056.122  325        3717          PUSH    D
056.123  305        3718          PUSH    B               SAVE (C)
056.124  001 013 000  3719          LXI     B,8+3
056.127  041 324 065  3720          LXI     H,PIO.DIR+DIR.NAM
056.132  315 252 030  3721          CALL    $MOVE
056.135  315 160 053  3722          CALL    AEN             ADD TO TABLE
056.140  301        3723          POP     B               RESTORE (C)
                      3724
                      3725  *       LOOKUP NEXT ENTRY
                      3726
056.141  341        3727  EWS4    POP     H
056.142  006 000     3728  EWS6    MVI     B,0
056.144  011        3729          DAD     B               POINT TO NEXT
056.145  303 062 056  3730          JMP     EWS3
                      3731
                      3732  *       ALL DONE, CLOSE DIRECTORY FILE
                      3733
056.150  076 002     3734  EWS7    MVI     A,CN.DIR
056.152  377 046     3735          DB      SYSCALL+.CLOSE
056.154  311        3736          RET
                      3737
056.155  123 131 060  3738  EWSA    DB      'SY0',200Q,200Q,200Q
                      3739
056.163               3740  EWSB    DS      30
                      3741
056.221               3742  EWSC    DS      8+3             WILDCARD PATTERN FOR DIRECTORY SEARCH


                      3744  **      GDWP    -  GET DIRECTORY WORKSPACE POINTER              /79.11.GC/
                      3745  *
                      3746  *       GDWP GETS THE DIRECTORY WORKSPACE POINTER
                      3747  *
                      3748  *       ENTRY:  NONE
                      3749  *
                      3750  *       EXIT:   DE      = DIRECTORY WORKSPACE POINTER
                      3751  *
```

```
                        3752  *       USES:   DE
                        3753  *
                        3754
056.234  353          3755  GDWP    XCHG
056.235  315 242 056   3756          CALL    GDWP.           HL = DIRECTORY WORKSPACE POINTER
056.240  353          3757          XCHG
056.241  311          3758          RET
                        3759
056.242  052 121 041   3760  GDWP.   LHLD    S.SCR           HL = SYSTEM SCRATCH
056.245  311          3761          RET


                        3763  **      INA - INCREASE NAMTAB ALLOCATION.
                        3764  *
                        3765  *       INA IS CALLED TO INCREASE THE NAMTAB ALLOCATION. THE
                        3766  *       BUFFER AREA IS MOVED UP TO MAKE ROOM.
                        3767  *
                        3768  *       ENTRY   NONE
                        3769  *       EXIT    NONE
                        3770  *       USES    A,F,H,L
                        3771
056.246  041 267 063   3772  INA     LXI     H,NAMTMAX+1
056.251  064           3773          INR     M               INCREMENT LENGTH
056.252  041 226 063   3774          LXI     H,BUFPTR+1
056.255  064           3775          INR     M               MOVE BUFFER
056.256  052 227 063   3776          LHLD    BUFSIZ
056.261  174           3777          MOV     A,H
056.262  265           3778          ORA     L
056.263  076 021       3779          MVI     A,EC.NEM        FLAG OUT OF MEMORY IF BUFFER NOT EMPTY
056.265  302 026 053   3780          JNZ     ERROR
056.270  305           3781          PUSH    B
056.271  325           3782          PUSH    D
056.272  315 021 057   3783          CALL    SBE             NOTIFY SYSTEM
056.275  321           3784          POP     D
056.276  301           3785          POP     B
056.277  311           3786          RET


                        3788  **      LSN - LOCATE SOURCE NAME
                        3789  *
                        3790  *       LSN SCANS THE COMMAND LINE FOR THE FIRST SOURCE FILE NAME.
                        3791  *
                        3792  *       ENTRY   NONE
                        3793  *       EXIT    (HL) = 1ST FILE NAME FWA
                        3794  *       USES    A,F,H,L
                        3795
056.300  041 374 065   3796  LSN     LXI     H,LINE
056.303  176           3797  LSN1    MOV     A,M
056.304  043           3798          INX     H
056.305  376 075       3799          CPI     '='
056.307  310           3800          RE                      GOT IT
056.310  247           3801          ANA     A
```

```
056.311  302 303 056  3802          JNZ     LSN1            MORE LINE
056.314  041 374 065  3803          LXI     H,LINE  IS NO =
056.317  311          3804          RET

                      3806  **      MWN - MERGE WILDCARD NAMES.
                      3807  *
                      3808  *       MWN MERGES A COMPLETELY SPECIFIED FILENAME WITH A WILDCARDED COMPLETELY
                      3809  *       SPECIFIED FILE NAME.
                      3810  *
                      3811  *       BOTH FILE NAMES SHOULD HAVE THE SAME DEVICE SPECIFICATION.
                      3812  *
                      3813  *       FILE NAME FORMAT:
                      3814  *
                      3815  *       DEV:NAMEXXXX.EXT 00
                      3816  *
                      3817  *       ENTRY   (BC) = ADDRESS OF WILDCARDED ASCII NAME
                      3818  *               (DE) = ADDRESS OF NON-WC ASCII NAME
                      3819  *               (HL) = ADDRESS FOR RESULTANT ASCII NAME
                      3820  *       EXIT    NONE
                      3821  *       USES    ALL
                      3822
                      3823
056.320  345          3824  MWN     PUSH    H               SAVE TARGET ADDRESS
056.321  305          3825          PUSH    B               SAVE WC PATTERN
056.322  353          3826          XCHG                    (HL) = MASTER NAME
056.323  315 136 054  3827          CALL    CAD             CONVERT TO DIRECTORY FORMAT
056.326  315 044 061  3828          CALL    $MOVEL
056.331  013 000 324  3829          DW      8+3,PIO.DIR,MWNA        (MWNA) = DECODED MASTER
056.337  341          3830          POP     H               (HL) = WC PATTERN
056.340  315 136 054  3831          CALL    CAD             (PIO.DIR) = WC PATTERN
056.343  021 300 063  3832          LXI     D,MWNA          (DE) = MASTER PATTERN
056.346  041 324 065  3833          LXI     H,PIO.DIR       (DE) = WC PATTERN ADDRESS
056.351  016 013      3834          MVI     C,8+3           MERGE NAME AND EXTENSION
                      3835
                      3836  *       MERGE NAMES
                      3837
056.353  176          3838  MWN1    MOV     A,M             (A) = WC PATTERN
056.354  247          3839          ANA     A
056.355  362 361 056  3840          JP      MWN2            USE THIS
056.360  032          3841          LDAX    D               IS MATCH CHARACTER, USE MASTER INSTEAD
056.361  167          3842  MWN2    MOV     M,A             STORE CHARACTER
056.362  023          3843          INX     D
056.363  043          3844          INX     H
056.364  015          3845          DCR     C
056.365  302 353 056  3846          JNZ     MWN1            MERGE TILL DONE
056.370  341          3847          POP     H               (HL) = TARGET ADDRESS
056.371  303 230 055  3848          JMP     CDA             CONVERT DIRECTORY FORMAT TO ASCII
```

```
                                3850  **      REN - REMOVE ENTRY FROM *NAMTAB*
                                3851  *
                                3852  *       REN REMOVES THE FIRST 'FB.NAML' BYTES FROM NAMTAB.
                                3853  *
                                3854  *       THE AMOUNT (FB.NAML) IS REMOVED FROM THE SIZE OF THE TABLE. THE
                                3855  *       TABLE IS NOT CHECKED FOR UNDERFLOW, THE CALLER MUST GUARANTEE THE
                                3856  *       PRESENSE OF AT LEAST FB.NAML BYTES IN NAMTAB.
                                3857  *
                                3858  *       ENTRY   NONE
                                3859  *       EXIT    NONE
                                3860  *       USES    ALL
                                3861
                                3862
056,374  052 264 063           3863  REN     LHLD    NAMTLEN
056,377  021 357 377           3864          LXI     D,-FB.NAML
057,002  031                   3865          DAD     D               REMOVE COUNT FROM LEN
057,003  042 264 063           3866          SHLD    NAMTLEN
057,006  104                   3867          MOV     B,H
057,007  115                   3868          MOV     C,L             (BC) = REMAINING LENGTH
057,010  021 135 066           3869          LXI     D,NAMTAB+FB.NAML        (DE) = START OF 2ND ENTRY
057,013  041 114 066           3870          LXI     H,NAMTAB
057,016  303 252 030           3871          JMP     $MOVE           MOVE DOWN AND RETURN


                                3873  **      SBE - SET BUFFER EMPTY.
                                3874  *
                                3875  *       THE SYSTEM IS NOTIFIED.
                                3876  *
                                3877  *       ENTRY   NONE
                                3878  *       EXIT    NONE
                                3879  *       USES    ALL
                                3880
                                3881
057,021  041 000 000           3882  SBE     LXI     H,0
057,024  042 227 063           3883          SHLD    BUFSIZ
057,027  052 225 063           3884          LHLD    BUFPTR          (HL) = BUFFER FWA (AND LWA!)
057,032  043                   3885          INX     H
057,033  043                   3886          INX     H
057,034  377 052               3887          DB      SYSCALL,.SETTP
057,036  320                   3888          RNC                     OK
057,037  303 026 053           3889          JMP     ERROR           NOT ENOUGH ROOM


                                3891  **      SDD - SET DEFAULT DEFAULT.
                                3892  *
                                3893  *       SDD IS CALLED TO SETUP THE CURRENT DEFAULT DEVICE
                                3894  *       AND EXTENSION TO 'SYO' AND <NULL>, RESPECTIVELY.
                                3895  *
                                3896  *       ENTRY   NONE
                                3897  *       EXIT    NONE
                                3898  *       USES    NONE
                                3899
```

```
                                3900
057.042  315 054 031            3901  SDD     CALL    $SAVALL
057.045  315 044 061            3902          CALL    $MOVEL
057.050  006 000 061            3903          DW      6,SDDA,DEFALT   SET DEFAULT DEFAULT
057.056  303 047 031            3904          JMP     $RSTALL         RESTORE AND RETURN
                                3905
057.061  123 131 060            3906  SDDA    DB      'SY0',0,0,0     DEFAULT DEFAULT VALUES


                                3908  **      SFS - SKIP FILE SEPERATOR.
                                3909  *
                                3910  *       SFS IS CALLED TO SKIP OVER THE CHARACTERS SEPERATING ONE
                                3911  *       FILE NAME FROM ANOTHER ON THE LINE. THE FILES MAY BE SEPERATED
                                3912  *       BY BLANKS OR A COMMA ALONE, OR BY BLANKS WITH A COMMA. THE
                                3913  *       SYNTAX IS
                                3914  *
                                3915  *       <BLANKS> <,> <BLANKS>
                                3916  *
                                3917  *       ONE, TWO OR ALL THREE FIELDS MAY BE PRESENT.
                                3918  *
                                3919  *       ENTRY   (HL) = POINT TO START OF SEP FIELD
                                3920  *       EXIT    (HL) ADVANCED PAST SEPERATOR FIELD
                                3921  *       USES    A,F,H,L
                                3922
                                3923
057.067  315 321 057            3924  SFS     CALL    $SOB            SKIP BLANKS
057.072  176                    3925          MOV     A,M
057.073  376 054                3926          CPI     ','
057.075  302 101 057            3927          JNE     SFS1            NOT ,
057.100  043                    3928          INX     H               SKIP ,
057.101  303 321 057            3929  SFS1    JMP     $SOB            GET ANY MORE BLANKS AND EXIT


                                3931  **      SND - SET NEW DEFAULTS.
                                3932  *
                                3933  *       SND IS CALLED TO SET A NEW DEFAULT DEVICE AND EXTENSION
                                3934  *       IN THE 'DEFALT' AREA.
                                3935  *
                                3936  *       ENTRY   PIO.DEV = DEVICE CODE
                                3937  *               PIO.UNI = UNIT #
                                3938  *               PIO.DIR+DIR.EXT = EXTENSION
                                3939  *       EXIT    NONE
                                3940  *       USES    NONE
                                3941
                                3942
057.104  315 054 031            3943  SND     CALL    $SAVALL         SAVE REGS
000.000                         3944          ERRNZ   PIO.UNI-PIO.DEV-2
057.107  315 044 061            3945          CALL    $MOVEL
057.112  003 000                3946          DW      3
057.114  321 065                3947          DW      PIO.DEV
057.116  272 063                3948          DW      DEFALT
057.120  315 044 061            3949          CALL    $MOVEL
```

```
057.123  003 000      3950          DW      3
057.125  334 065      3951          DW      PIO.DIR+DIR.EXT
057.127  275 063      3952          DW      DEFALT+3
057.131  303 047 031  3953          JMP     $RSTALL         RETURN
```

```
057.134                 3956          XTEXT   CFD


                        3958X **      $CFD - CHECK FILE DELIMITER.
                        3959X *
                        3960X *       $CFD CHECKS AN ASCII CHARACTER TO SEE IF IT IS A LEGAL FILE
                        3961X *       NAME DELIMITER. LEGAL DELIMITERS ARE
                        3962X *
                        3963X *       , = / <BLANK> <00>
                        3964X *
                        3965X *       ENTRY   (A) = CHARACTER
                        3966X *       EXIT    'C' CLEAR IF OK
                        3967X *               'C' SET IF ERROR
                        3968X *                (A) = ERROR CODE
                        3969X *       USES    A,F
                        3970X
                        3971X
057.134  247           3972X $CFD    ANA     A
057.135  310           3973X         RZ                      IS 00
057.136  376 054       3974X         CPI     ','
057.140  310           3975X         RE                      IS ,
057.141  376 075       3976X         CPI     '='
057.143  310           3977X         RE                      IS =
057.144  376 057       3978X         CPI     '/'
057.146  310           3979X         RE                      IS /
057.147  376 040       3980X         CPI     ' '
057.151  310           3981X         RE                      IS ' '
057.152  076 007       3982X         MVI     A,EC.IFN        ILLEGAL FILE NAME
057.154  067           3983X         STC
057.155  311           3984X         RET
057.156                 3985          XTEXT   TYPCC


                        3987X **      $TYPCC - TYPE A CHARACTER STRING BY COUNT.
                        3988X *
                        3989X *       $TYPCC TYPES A STRING OF CHARACTERS. THE CALLER SUPPLIES
                        3990X *       THE CHARACTER ADDRESS AND COUNT.
                        3991X *
                        3992X *       ENTRY   (HL) = ADDRESS
                        3993X *               (A) = COUNT
                        3994X *       EXIT    (HL) = LAST CHARACTER ADDRESS+1
                        3995X *       USES    A,F,H,L
                        3996X
                        3997X
057.156                 3998X $TYPCC  EQU     *
057.156  247           3999X         ANA     A
057.157  310           4000X         RZ                      NOTHING TO TYPE
057.160  365           4001X         PUSH    PSW             SAVE COUNT
057.161  176           4002X         MOV     A,M             (A) = CHARACTER
057.162  043           4003X         INX     H
057.163  377 002       4004X         DB      SYSCALL;.SCOUT
057.165  361           4005X         POP     PSW
```

```
057.166  075          4006X          DCR     A
057.167  303 156 057  4007X          JMP     $TYPCC
057.172               4008           XTEXT   WER


                      4010X **       $WER - WRITE ENABLE RAM.
                      4011X *
                      4012X *        $WER IS CALLED TO ENABLE WRITTING TO THE H17 CONTROLLER'S
                      4013X *        RAM AREA.
                      4014X *
                      4015X *        ENTRY   NONE
                      4016X *        EXIT    NONE
                      4017X *        USES    NONE
                      4018X
                      4019X
031.241               4020X $WER     EQU     31241A          IN H17 ROM


                      4022X **       $WDR - WRITE DISABLE RAM.
                      4023X *
                      4024X *        $WDR IS CALLED TO DISABLE WRITTING TO THE H17 CONTROLLER'S
                      4025X *        RAM AREA.
                      4026X *
                      4027X *        ENTRY   NONE
                      4028X *        EXIT    NONE
                      4029X *        USES    NONE
                      4030X
                      4031X
031.222               4032X $WDR     EQU     31222A          IN H17 ROM
057.172               4033           XTEXT   ZERO


                      4035X **       $ZERO - ZERO MEMORY
                      4036X *
                      4037X *        $ZERO ZEROS A BLOCK OF MEMORY.
                      4038X *
                      4039X *        ENTRY   (HL) = ADDRESS
                      4040X *                (B) = COUNT
                      4041X *        EXIT    (A) = 0
                      4042X *        USES    A,B,F,H,L
                      4043X
                      4044X
031.212               4045X $ZERO    EQU     31212A          IN H17 ROM
057.172               4046           XTEXT   MU86
```

```
                                4048X **      $MU86 - MULTIPLY 8X16 UNSIGNED.
                                4049X *
                                4050X *       $MU86 MULTIPLIES A 16 BIT VALUE BY A 8
                                4051X *       BIT VALUE.
                                4052X *
                                4053X *       ENTRY   (A) = MULTIPLIER
                                4054X *               (DE) = MULTIPLICAND
                                4055X *       EXIT    (HL) = RESULT
                                4056X *               'Z' SET IF NOT OVERFLOW
                                4057X *       USES    A,F,H,L
                                4058X
                                4059X
031.007                         4060X $MU86   EQU     31007A          IN H17 ROM
057.172                         4061          XTEXT   CCO



                                4063X **      $CCO - CLEAR CONTROL-O
                                4064X *
                                4065X *       $CCO IS CALLED TO CLEAR THE EFFECT OF THE CTL-O CHARACTER.
                                4066X *
                                4067X *       ENTRY   NONE
                                4068X *       EXIT    NONE
                                4069X *       USES    NONE
                                4070X
                                4071X
057.172  315 054 031           4072X $CCO    CALL    $SAVALL         SAVE REGISTERS
057.175  076 004               4073X         MVI     A,I.CONFL
057.177  001 001 000           4074X         LXI     B,CO.FLG        CLEAR CO.FLG
057.202  377 006               4075X         DB      SYSCALL,.CONSL
057.204  303 047 031           4076X         JMP     $RSTALL         RESTORE REGISTERS AND RETURN
057.207                         4077          XTEXT   GNL



                                4079X **      $GNL - GUARANTEE NEW LINE.
                                4080X *
                                4081X *       $GNL GUARANTEES THE START OF A NEW LINE BY ISSUING A CRLF
                                4082X *       IF THE CURSOR IS NOT AT COLUMN 1.
                                4083X *
                                4084X *       ENTRY   NONE
                                4085X *       EXIT    NONE
                                4086X *       USES    ALL
                                4087X
                                4088X
057.207  076 002               4089X $GNL    MVI     A,I.CUSOR
057.211  001 000 000           4090X         LXI     B,0
057.214  377 006               4091X         DB      SYSCALL,.CONSL          READ CURSOR
057.216  075                   4092X         DCR     A
057.217  310                   4093X         RZ                      AT COLUMN 1
057.220  303 370 057           4094X         JMP     $CRLF           NEW LINE
057.223                         4095          XTEXT   MLU
```

```
                        4097X **      MLU - MAP LOWER CASE LINE TO UPPER CASE.
                        4098X *
                        4099X *      MLU MAPS THE LOWER CASE ALPHABETICS IN A LINE TO UPPER CASE.
                        4100X *
                        4101X *      ENTRY   (HL) = LINE FWA
                        4102X *      EXIT    NONE
                        4103X *      USES    NONE
                        4104X
                        4105X
057.223  365           4106X $MLU   PUSH    PSW             SAVE (PSW)
057.224  345           4107X        PUSH    H               SAVE FWA
057.225  053           4108X        DCX     H               ANTICIPATE INX H
057.226  043           4109X $MLU1  INX     H
057.227  176           4110X        MOV     A,M             (A)= CHARACTER
057.230  315 243 057   4111X        CALL    $MCU            MAP CHAR TO UPPER
057.233  167           4112X        MOV     M,A
057.234  247           4113X        ANA     A
057.235  302 226 057   4114X        JNZ     $MLU1           MORE TO GO
057.240  341           4115X        POP     H               RESTORE (HL)
057.241  361           4116X        POP     PSW             RESTORE (PSW)
057.242  311           4117X        RET
057.243               4118         XTEXT   MCU


                        4120X **     MCU - MAP LOWER CASE TO UPPER CASE.
                        4121X *
                        4122X *      MCU MAPS A LOWER CASE ALPHABETIC TO UPPER
                        4123X *      CASE.
                        4124X *
                        4125X *      ENTRY   (A) = CHARACTER
                        4126X *      EXIT    (A) = CHARACTER RESULT
                        4127X *      USES    A,F
                        4128X
                        4129X
057.243  376 141       4130X $MCU   CPI     'a'
057.245  330           4131X        RC                      NOT LOWER CASE
057.246  376 173       4132X        CPI     'z'+1
057.250  320           4133X        RNC                     NOT LOWER CASE
057.251  326 040       4134X        SUI     'a'-'A'
057.253  311           4135X        RET
057.254               4136         XTEXT   RTL


                        4138X **     $RTL - READ TEXT LINE.
                        4139X *
                        4140X *      $RTL READS A LINE FROM THE TERMINAL.
                        4141X *
                        4142X *      CHARACTER ARE ACCEPTED FROM THE TERMINAL, RUBOUT AND BACKSPACE
                        4143X *      CHARACTERS ARE PROCESSED. WHEN A CARRIAGE RETURN IS ENTERED,
                        4144X *      $RTL RETURNS.
                        4145X *
                        4146X *      ENTRY   (HL) = BUFFER FWA
```

```
                                4147X *         EXIT    'C' CLEAR IF OK
                                4148X *                  DATA IN BUFFER
                                4149X *                  (A) = TEXT LENGTH
                                4150X *                  'C' SET IF CTL-D STRUCK
                                4151X *         USES    A,F
                                4152X
                                4153X
        057.254  315 263 057    4154X $RTL.     CALL    $RTL            $RTL IN UPPER CASE
        057.257  330            4155X           RC                      CTL-D
        057.260  303 223 057    4156X           JMP     $MLU            MAP LINE TO UPPER CASE
                                4157X
        057.263                 4158X $RTL      EQU     *
        057.263  345            4159X           PUSH    H               SAVE FWA
        057.264  315 067 061    4160X $RTL1     CALL    $RCHAR
        057.267  376 004        4161X           CPI     CTLD
        057.271  312 316 057    4162X           JE      $RTL2           CTL-D STRUCK
        057.274  167            4163X           MOV     M,A
        057.275  043            4164X           INX     H
        057.276  376 012        4165X           CPI     NL
        057.300  302 264 057    4166X           JNE     $RTL1
        057.303  053            4167X           DCX     H
        057.304  066 000        4168X           MVI     M,0
        057.306  043            4169X           INX     H
                                4170X
                                4171X *         ALL DONE. COMPUTE LENGTH
                                4172X
        057.307  353            4173X           XCHG                    (DE) = LWA+1
        057.310  343            4174X           XTHL                    (HL) = FWA
        057.311  173            4175X           MOV     A,E
        057.312  225            4176X           SUB     L               (A) = LENGTH
        057.313  247            4177X           ANA     A               CLEAR CARRY
        057.314  321            4178X           POP     D               RESTORE (DE)
        057.315  311            4179X           RET
                                4180X
                                4181X *         CTL-D STRUCK
                                4182X
        057.316  341            4183X $RTL2     POP     H               (HL) = FWA
        057.317  067            4184X           STC
        057.320  311            4185X           RET
        057.321                 4186            XTEXT   MOVE
```

```
                                4188X **        $MOVE - MOVE DATA
                                4189X *
                                4190X *         $MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                                4191X *         IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                                4192X *         FIRST TO LAST.
                                4193X *
                                4194X *         IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                                4195X *         LAST TO FIRST.
                                4196X *
                                4197X *         THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                                4198X *
                                4199X *         ENTRY   (BC) = COUNT
```

```
                        4200X *             (DE) = FROM
                        4201X *             (HL) = TO
                        4202X *       EXIT  MOVED
                        4203X *             (DE) = ADDRESS OF NEXT FROM BYTE
                        4204X *             (HL) = ADDRESS OF NEXT *TO* BYTE
                        4205X *             'C' CLEAR
                        4206X *       USES  ALL
                        4207X
                        4208X
030.252                 4209X $MOVE   EQU   30252A          IN H17 ROM
057.321                 4210          XTEXT CHL


                        4212X **      $CHL - COMPLEMENT (HL).
                        4213X *
                        4214X *       (HL) = -(HL)          TWO'S COMPLEMENT
                        4215X *
                        4216X *       ENTRY NONE
                        4217X *       EXIT  NONE
                        4218X *       USES  A,F,H,L
                        4219X
                        4220X
030.224                 4221X $CHL    EQU   30224A          IN H17 ROM
057.321                 4222          XTEXT SOB


                        4224X **      $SOB - SKIP OVER BLANKS.
                        4225X *
                        4226X *       $SOB IS CALLED TO SKIP AN ARBITRARILY LONG STRING OF BLANKS AND TABS.
                        4227X *
                        4228X *       ENTRY (HL) = FWA OF (POSSIBLE) BLANK STRING
                        4229X *       EXIT  (HL) = LWA+1 OF BLANK STRING (UNCHANGED IF NO BLANKS)
                        4230X *             (A) = FIRST NON-BLANK, NON-TAB CHARACTER EEN
                        4231X *       USES  A,F,H,L
                        4232X
                        4233X
057.321  053           4234X $SOB    DCX   H               PRE-DECREMENT
057.322  043           4235X $SOB1   INX   H
057.323  176           4236X         MOV   A,M
057.324  376 040       4237X         CPI   ' '
057.326  312 322 057   4238X         JE    $SOB1           GOT BLANK
057.331  376 011       4239X         CPI   TAB
057.333  312 322 057   4240X         JE    $SOB1           GOT TAB
057.336  311           4241X         RET
057.337               4242          XTEXT TBLS
```

```
                        4244X **      $TBLS - TABLE SEARCH
                        4245X *
                        4246X *       TABLE FORMAT
                        4247X *
                        4248X *       DB      KEY1,VAL1,
                        4249X *       .       .
                        4250X *       .       .
                        4251X *       DB      KEYN,VALN
                        4252X *       DB      0
                        4253X *
                        4254X *       ENTRY   (A) = PATTERN
                        4255X *               (H,L) = TABLE FWA
                        4256X *       EXIT    (A) = PATTERN IF FOUND
                        4257X *               'Z' SET IF FOUND
                        4258X *               'Z' CLEAR IF NOT FOUND OR PATTERN=0          /78.10.GC/
                        4259X *       USES    A,F,H,L
                        4260X
                        4261X
057.337  305           4262X $TBLS   PUSH    B
057.340  376 000       4263X         CPI     0                                            /78.10.GC/
057.342  312 364 057   4264X         JZ      TBL2                                         /78.10.GC/
057.345  107           4265X         MOV     B,A
057.346  176           4266X TBL1    MOV     A,M             (A) = CHARACTER
057.347  043           4267X         INX     H
057.350  270           4268X         CMP     B
057.351  312 366 057   4269X         JZ      TBL3            IF MATCH
057.354  247           4270X         ANA     A
057.355  043           4271X         INX     H               SKIP PAST
057.356  302 346 057   4272X         JNZ     TBL1            IF NOT END OF TABLE
057.361  053           4273X         DCX     H
057.362  053           4274X         DCX     H
057.363  257           4275X         XRA     A               SET TO ZERO FOR OLD USERS    /78.10.GC/
057.364  376 001       4276X TBL2    CPI     1               CLEAR ZERO                   /78.10.GC/
                        4277X
                        4278X *       DONE
                        4279X
057.366  301           4280X TBL3    POP     B
057.367  311           4281X         RET
057.370                4282          XTEXT   DADA


                        4284X **      $DADA - PERFORM (H,L) = (H,L) + (0,A)
                        4285X *
                        4286X *       ENTRY   (H,L) = BEFORE VALUE
                        4287X *               (A) = BEFORE VALUE
                        4288X *       EXIT    (H,L) = (H,L) + (0,A)
                        4289X *               'C' SET IF OVERFLOW
                        4290X *       USES    F,H,L
                        4291X
                        4292X
030.072                 4293X $DADA   EQU     30072A          IN H17 ROM
057.370                 4294          XTEXT   TJMP
```

```
                        4296X **    $TJMP - TABLE JUMP.
                        4297X *
                        4298X *     USAGE
                        4299X *
                        4300X *     CALL    $TJMP          (A) = INDEX
                        4301X *     DW      ADDR1
                        4302X *     .         .
                        4303X *     .         .
                        4304X *     .         .
                        4305X *     DW      ADDRN
                        4306X *
                        4307X *     ENTRY   (A) = INDEX
                        4308X *     EXIT    TO PROCESSOR
                        4309X *             (A) = INDEX*2
                        4310X *     USES    NONE.
                        4311X
                        4312X
031.061                 4313X $TJMP EQU     31061A          IN H17 ROM, (A) = INDEX*2
                        4314X
031.062                 4315X $TJMP. EQU    31062A          IN H17 ROM
057.370                 4316        XTEXT   CRLF


                        4318X **    $CRLF - TYPE CARRIAGE RETURN/ LINE FEED
                        4319X *
                        4320X *     $CRLF IS USED TO GENERATE PADDED CRLF'S.
                        4321X *
                        4322X *     ENTRY   NONE
                        4323X *     EXIT    (A) = 0
                        4324X *     USES    A,F
                        4325X
                        4326X
057.370  076 012       4327X $CRLF  MVI     A,NL
057.372  377 002       4328X        DB      SYSCALL,.SCOUT
057.374  257          4329X        XRA     A
057.375  311          4330X        RET
057.376                4331         XTEXT   TYPCH


                        4333X **    $TYPCH - TYPE SINGLE CHARACTER.
                        4334X *
                        4335X *     ENTRY   (RET) = CHARACTER
                        4336X *     EXIT    TO (RET)+1
                        4337X *             (A) = CHARACTER TYPED
                        4338X
                        4339X
057.376  343           4340X $TYPCH XTHL                    (HL) = RETURN ADDRESS
057.377  176           4341X        MOV     A,M             (A) = CHARACTER
060.000  043           4342X        INX     H
060.001  343           4343X        XTHL                    RESTORE ADVANCED EXIT ADDRESS
                        4344X
                        4345X **    $TYPC. - TYPE SINGLE CHARACTER.
```

```
                                4346X *
                                4347X *         ENTRY   (A) = CHARACTER
                                4348X *         EXIT    TO (RET)
                                4349X
060.002  377 002                4350X $TYPC.    DB      SYSCALL,.SCOUT
060.004  311                    4351X           RET
000.001                         4352  $CMP$     EQU     1
060.005                         4353            XTEXT   TYPLN


                                4355X **        $TYPLN - TYPE LINE.
                                4356X *
                                4357X *         $TYPLN IS CALLED TO TYPE A LINE OF TEXT. ZERO BYTES ARE
                                4358X *         TAKEN AS CRLF (WITH THE PROPER PADDING)
                                4359X *
                                4360X *         CALL    $TYPLN
                                4361X *         DB      N                       BYTE COUNT OF FOLLOWING MESSAGE
                                4362X *         DB      'N-CHARACTER MESSAGE'
                                4363X *
                                4364X *         ENTRY   (RET) = TEXT COUNT
                                4365X *                 (RET)+1 - (RET)+N = TEXT
                                4366X *         EXIT    TO (RET)+N+1
                                4367X *         USES    A,F
                                4368X *
                                4369X
                                4370X
060.005  343                    4371X $TYPLN.   XTHL                    (H,L) = COUNT ADDRESS
060.006  176                    4372X           MOV     A,M             (A) = COUNT
060.007  043                    4373X           INX     H               (H,L) = TEXT ADDRESS
060.010  345                    4374X           PUSH    H               SAVE TEXT FWA
060.011  315 072 030            4375X           CALL    $DADA           CALCULATE RETURN ADDRESS
060.014  343                    4376X           XTHL                    (HL) = TEXT ADDRE
060.015  315 023 060            4377X           CALL    $TYPL.          OUTPUT LINE
060.020  341                    4378X           POP     H               (HL) = RETURN ADDRESS
060.021  343                    4379X           XTHL                    RESTORE (HL); SET RETURN ADDRESS
060.022  311                    4380X           RET
                                4381X
                                4382X **        $TYPL. - TYPE LINE.
                                4383X *
                                4384X *         ENTRY   (HL) = ADDRESS
                                4385X *                 (A) = COUNT
                                4386X *         EXIT    NONE
                                4387X *         USES    A,F,H,L
                                4388X
060.023                         4389X $TYPL.    EQU     *
060.023  247                    4390X           ANA     A
060.024  310                    4391X           RZ                      NOTHING TO TYPE
060.025  365                    4392X           PUSH    PSW             SAVE COUNT
060.026  176                    4393X           MOV     A,M             (A) = CHARACTER
060.027  043                    4394X           INX     H
060.030  247                    4395X           ANA     A
000.001                         4396X           IF      $CMP$           IF HAVE COMPRESSED SPACES
                                4397X           JM      TPL2            IS COMPRESSED SPACE
                                4398X           ENDIF
```

```
060.031  314 370 057  4399X          CZ      $CRLF
060.034  315 002 060  4400X          CALL    $TYPC.          TYPE CHARACTER
060.037  361         4401X TPL1     POP     PSW
060.040  075         4402X          DCR     A
060.041  302 023 060  4403X          JNZ     $TYPL.
060.044  311         4404X          RET
000.001              4405X          IF      $CMP$           IF COMPRESSED TEXT
                     4406X
                     4407X *        HAVE COMPRESSED SPACE.
                     4408X
                     4409X TPL2     DCR     A
                     4410X          CP      $TYPCH          TYPE 00 IF CHARACTER WAS 200Q
                     4411X          DB      0
                     4412X          ANA     A               SET CODES
                     4413X TPL3     JP      TPL1            ALL EXPANDED
                     4414X          PUSH    PSW             SAVE COUNT
                     4415X          CALL    $TYPCH
                     4416X          DB      ' '
                     4417X          POP     PSW
                     4418X          DCR     A
                     4419X          JMP     TPL3
                     4420X          ENDIF
060.045              4421           XTEXT   TYPT2


                     4423X **       $TYPTX - TYPE TEXT.
                     4424X *
                     4425X *        $TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
                     4426X *
                     4427X *        IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED.
                     4428X *        A BYTE WITH THE 200Q BIT SET IS THE LAST BYTE IN THE MESSAGE.
                     4429X *
                     4430X *        ENTRY   (RET) = TEXT
                     4431X *        EXIT    TO (RET+LENGTH)
                     4432X *        USES    A,F
                     4433X
                     4434X
031.136              4435X $TYPTX   EQU     31136A          IN H17 ROM
                     4436X
031.144              4437X $TYPTX.  EQU     31144A          IN H17 ROM
060.045              4438           XTEXT   COMP


                     4440X **       $COMP - COMPARE TWO CHARACTER STRINGS.
                     4441X *
                     4442X *        $COMP COMPARES TWO BYTE STRINGS.
                     4443X *
                     4444X *        ENTRY   (C) = COMPARE COUNT
                     4445X *                (DE) = FWA OF STRING #1
                     4446X *                (HL) = FWA OF STRING #2
                     4447X *        EXIT    'Z' CLEAR, IS MIS-MATCH
                     4448X *                 (C) = LENGTH REMAINING
```

```
                        4449X *                 (DE) = ADDRESS OF MISMATCH IN STRING#1
                        4450X *                 (HL) = ADDRESS OF MISMATCH IN STRING #2
                        4451X *                'C' SET, HAVE MATCH
                        4452X *                 (C) = 0
                        4453X *                 (DE) = (DE) + (OC)
                        4454X *                 (HL) = (HL) + (OC)
                        4455X *         USES    A,F,C,D,E,H,L
                        4456X
                        4457X
030.060                 4458X $COMP     EQU     30060A          IN H17 ROM
060.045                 4459            XTEXT   SAVALL


                        4461X **        $RSTALL - RESTORE ALL REGISTERS.
                        4462X *
                        4463X *         $RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND
                        4464X *         RETURNS TO THE PREVIOUS CALLER.
                        4465X *
                        4466X *         ENTRY   (SP) = PSW
                        4467X *                 (SP+2) = BC
                        4468X *                 (SP+4) = DE
                        4469X *                 (SP+6) = HL
                        4470X *                 (SP+8) = RET
                        4471X *         EXIT    TO *RET*, REGISTERS RESTORED
                        4472X *         USES    ALL
                        4473X
                        4474X
031.047                 4475X $RSTALL   EQU     31047A          IN H17 ROM


                        4477X **        $SAVALL - SAVE ALL REGISTERS ON STACK.
                        4478X *
                        4479X *         $SAVALL SAVES ALL THE REGISTERS ON THE STACK.
                        4480X *
                        4481X *         ENTRY   NONE
                        4482X *         EXIT    (SP) = PSW
                        4483X *                 (SP+2) = BC
                        4484X *                 (SP+4) = DE
                        4485X *                 (SP+6) = HL
                        4486X *         USES    H,L
                        4487X
                        4488X
031.054                 4489X $SAVALL   EQU     31054A          IN H17 ROM
060.045                 4490            XTEXT   CDEHL
```

```
                        4492X **      $CDEHL - COMPARE (DE) TO (HL)
                        4493X *
                        4494X *       $CDEHL COMPARES (DE) TO (HL) FOR EQUALITY.
                        4495X *
                        4496X *       ENTRY   NONE
                        4497X *       EXIT    'Z' SET IF (DE) = (HL)
                        4498X *       USES    A,F
                        4499X
                        4500X
030.216                 4501X $CDEHL  EQU     30216A          IN H17 ROM
060.045                 4502          XTEXT   UDD


                        4504X **      $UDD - UNPACK DECIMAL DIGITS.
                        4505X *
                        4506X *       UDD CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
                        4507X *       DECIMAL DIGITS. THE RESULT IS ZERO FILLED.
                        4508X *
                        4509X *       ENTRY   (B,C) = ADDRESS VALUE
                        4510X *               (A) = DIGIT COUNT
                        4511X *               (H,L) = MEMORY ADDRESS
                        4512X *       EXIT    (HL) = (HL) + (A)
                        4513X *       USES    ALL
                        4514X
                        4515X
031.157                 4516X $UDD    EQU     31157A          IN H17 ROM
060.045                 4517          XTEXT   DU66


                        4519X **      $DU66 - UNSIGNED 16 / 16 DIVIDE.
                        4520X *
                        4521X *       (HL) = (BC)/(DE)
                        4522X *
                        4523X *       ENTRY   (BC), (DE) PRESET
                        4524X *       EXIT    (HL) = RESULT
                        4525X *               (DE) = REMAINDER
                        4526X *       USES    ALL
                        4527X
                        4528X
030.106                 4529X $DU66   EQU     30106A          IN H17 ROM
060.045                 4530          XTEXT   DADA2


                        4532X **      $DADA. - ADD (0,A) TO (H,L)
                        4533X *
                        4534X *       ENTRY   NONE
                        4535X *       EXIT    (HL) = (HL) + (0A)
                        4536X *       USES    A,F,H,L
                        4537X
                        4538X
```

```
                    4539X $DADA.  EQU     30101A          IN H17 ROM
060.045             4540          XTEXT   HLIHL


                    4542X **      $HLIHL - LOAD HL INDIRECT THROUGH HL.
                    4543X *
                    4544X *       (HL) = ((HL))
                    4545X *
                    4546X *       ENTRY   NONE
                    4547X *       EXIT    NONE
                    4548X *       USES    A,H,L
                    4549X
030.211             4550X $HLIHL  EQU     30211A          IN H17 ROM
060.045             4551          XTEXT   ILDEHL


                    4553X **      ILDEHL  -  INDEXED LOAD OF DE FROM HL
                    4554X *
                    4555X *       'DE' GET THE FULL WORD VALUE POINTED TO BY 'HL', AND 'HL' IS
                    4556X *       INCREMENTED BY TWO.
                    4557X *
                    4558X *       ENTRY:  HL      = ADDRESS OF FULL WORD VALUE
                    4559X *
                    4560X *       EXIT:   DE      = (HL)
                    4561X *               HL      = HL + 2
                    4562X *
                    4563X *       USES:   DE
                    4564X *
                    4565X
060.045  136       4566X ILDEHL  MOV     E,M
060.046  043       4567X         INX     H
060.047  126       4568X         MOV     D,M
060.050  043       4569X         INX     H
060.051  311       4570X         RET
060.052             4571          XTEXT   INDL


                    4573X **      $INDL - INDEXED LOAD.
                    4574X *
                    4575X *       $INDL LOADS DE WITH THE TWO BYTES AT (HL)+DISPLACMENT
                    4576X *
                    4577X *       THIS ACTS AS AN INDEXED FULL WORD LOAD.
                    4578X *
                    4579X *       (DE) = ( (HL) + DSPLACEMENT )
                    4580X *
                    4581X *       ENTRY   ((RET)) = DISPLACMENT (FULL WORD)
                    4582X *               (HL) = TABLE ADDRESS
                    4583X *       EXIT    TO (RET+2)
                    4584X *       USES    A,F,D,E
                    4585X
```

```
                        4586X
030.234                 4587X $INDL   EQU     30234A          IN H17 ROM
060.052                 4588          XTEXT   INDXX


                        4590X **      $INDLB  -  INDEXED LOAD BYTE
                        4591X *
                        4592X *       BYTE INDEXED LOAD PRIMITIVE
                        4593X *
                        4594X *       ENTRY:  HL      = BASE ADDRESS
                        4595X *               (RET)   = FULL WORD RELOCATION
                        4596X *
                        4597X *       EXIT:   A       = ( HL + (RET) )
                        4598X *
                        4599X *       USES:   A
                        4600X *
                        4601X
060.052  353           4602X $INDLB  XCHG                    DE = BASE
060.053  343           4603X         XTHL                    SAVE  .DE.
060.054  325           4604X         PUSH    D               SAVE  BASE
060.055  305           4605X         PUSH    B               SAVE  .BC.
                        4606X
060.056  116           4607X         MOV     C,M
060.057  043           4608X         INX     H
060.060  106           4609X         MOV     B,M             BC = OFFSET
060.061  043           4610X         INX     H               HL =  .RET.
                        4611X
060.062  353           4612X         XCHG                    HL = BASE
060.063  011           4613X         DAD     B               HL = BASE + OFFSET
060.064  176           4614X         MOV     A,M             A  = ( BASE + OFFSET )
060.065  353           4615X         XCHG                    HL =  .RET.
                        4616X
060.066  301           4617X         POP     B               RESTORE  .BC.
060.067  321           4618X         POP     D               RESTORE  BASE
060.070  343           4619X         XTHL                    HL = .DE. ; (SP) = .RET.
060.071  353           4620X         XCHG                    DE = .DE. ; HL = BASE
060.072  311           4621X         RET


                        4623X **      $INDS   -  INDEXED STORE
                        4624X *
                        4625X *       INDEXED STORE PRIMITIVE.
                        4626X *
                        4627X *       ENTRY:  HL = BASE ADDRESS
                        4628X *               DE = VALUE TO STORE
                        4629X *
                        4630X *       EXIT:   ( HL + (RET) ) = DE
                        4631X *
                        4632X *       USES:   NONE
                        4633X *
                        4634X
060.073  315 100 061   4635X $INDS   CALL    XCHGBC
```

```
060.076  343           4636X          XTHL                    SAVE  .BC.
060.077  325           4637X          PUSH    D
060.100  315 045 060   4638X          CALL    ILDEHL          DE = OFFSET
060.103  315 100 061   4639X          CALL    XCHGBC          BC = .RET.
060.106  353           4640X          XCHG                    DE = BASE ; HL = OFFSET
060.107  031           4641X          DAD     D               HL = BASE + OFFSET
060.110  353           4642X          XCHG
060.111  343           4643X          XTHL                    SAVE  BASE
060.112  353           4644X          XCHG                    DE = VALUE
060.113  315 150 060   4645X          CALL    ISDEHL
060.116  341           4646X          POP     H               HL = BASE
060.117  315 100 061   4647X          CALL    XCHGBC
060.122  343           4648X          XTHL                    RESTORE  .BC.
060.123  315 100 061   4649X          CALL    XCHGBC
060.126  311           4650X          RET


                       4652X **       $INDSB  -  INDEXED BYTE STORE
                       4653X *
                       4654X *        INDEXED BYTE STORE.
                       4655X *
                       4656X *        ENTRY:  A       = VALUE TO STORE
                       4657X *                HL      = BASE ADDRESS
                       4658X *                (RET)   = OFFSET
                       4659X *
                       4660X *        EXIT:   NONE
                       4661X *
                       4662X *        USES:   PSW
                       4663X *
                       4664X
060.127  353           4665X $INDSB   XCHG                    DE = BASE
060.130  343           4666X          XTHL                    SAVE  .DE.
060.131  325           4667X          PUSH    D               SAVE  BASE
060.132  305           4668X          PUSH    B               SAVE  .BC.
                       4669X
060.133  116           4670X          MOV     C,M
060.134  043           4671X          INX     H
060.135  106           4672X          MOV     B,M             BC = OFFSET
060.136  043           4673X          INX     H               HL =  .RET.
                       4674X
060.137  353           4675X          XCHG                    HL = BASE
060.140  011           4676X          DAD     B               HL = BASE + OFFSET
060.141  167           4677X          MOV     M,A             ( BASE + OFFSET ) = A
060.142  353           4678X          XCHG
                       4679X
060.143  301           4680X          POP     B               RESTORE  .BC.
060.144  321           4681X          POP     D               RESTORE  BASE
060.145  343           4682X          XTHL                    HL = .DE. ; (SP) = .RET.
060.146  353           4683X          XCHG                    DE = .DE. ; HL = BASE
060.147  311           4684X          RET
060.150               4685           XTEXT   ISDEHL
```

```
                        4687X **      ISDEHL  -  INDEXED STORE OF DE AT HL
                        4688X *
                        4689X *       STORE 'DE' AT THE ADDRESS POINTED TO BY 'HL', AND INCREMENT 'HL'
                        4690X *       BY 2.
                        4691X *
                        4692X *       ENTRY:  DE      = VALUE
                        4693X *               HL      = ADDRESS OF VALUE
                        4694X *
                        4695X *       EXIT:   (HL)    = DE
                        4696X *               HL      = HL + 2
                        4697X *
                        4698X *       USES:   HL
                        4699X *
                        4700X
060.150  163            4701X ISDEHL  MOV     M,E
060.151  043            4702X         INX     H
060.152  162            4703X         MOV     M,D
060.153  043            4704X         INX     H
060.154  311            4705X         RET
060.155                 4706          XTEXT   DAD


                        4708X **      $DAD - DECODE AUGUSTAN DATE.
                        4709X *
                        4710X *       $DAD DECODES A 15 BIT DATE CODE OF THE FORMAT:
                        4711X *
                        4712X *       ----------------------------------------
                        4713X *       I  0  I  6 BITS  I  4 BITS  I  5 BITS  I
                        4714X *       ----------------------------------------
                        4715X *               YEAR-70       MON       DAY
                        4716X *                1-63        1-12      1-31
                        4717X *
                        4718X *       TO THE FORM:
                        4719X *
                        4720X *       DD-MMM-YY
                        4721X *
                        4722X *       ENTRY   (DE) = 15 BIT VALUE
                        4723X *               (HL) = ADDRESS FOR DECODE
                        4724X *       EXIT    'C' CLEAR IF OK
                        4725X *                (DE) = (DE)+9
                        4726X *               'C' SET IF ERROR
                        4727X *       USES    ALL
                        4728X
                        4729X
060.155  172            4730X $DAD    MOV     A,D                                     /80.08.sc/
060.156  263            4731X         ORA     E                                       /80.08.sc/
060.157  312 303 060    4732X         JZ      DAD2            No-Date                 /80.08.sc/
                        4733X
060.162  102            4734X         MOV     B,D
060.163  113            4735X         MOV     C,E
060.164  021 040 000    4736X         LXI     D,32
060.167  345            4737X         PUSH    H               SAVE ADDRESS
060.170  315 106 030    4738X         CALL    $DU66           (DE) = DAY, (HL) = YEAR & MONTH
060.173  343            4739X         XTHL                    (HL) = ADDRESS
```

```
060.174  102          4740X          MOV     B,D
060.175  113          4741X          MOV     C,E
060.176  173          4742X          MOV     A,E
060.177  247          4743X          ANA     A
060.200  312 300 060  4744X          JZ      DAD1            BAD VALUE
060.203  076 002      4745X          MVI     A,2
060.205  315 157 031  4746X          CALL    $UDD            UNPACK DAY
060.210  066 055      4747X          MVI     M,'-'
060.212  043          4748X          INX     H
060.213  301          4749X          POP     B               (BC) = YEAR & MONTH
060.214  021 020 000  4750X          LXI     D,16
060.217  345          4751X          PUSH    H               SAVE ADDRESS
060.220  315 106 030  4752X          CALL    $DU66
060.223  343          4753X          XTHL                    (HL) = ADDRESS, ((SP)) = YEAR
060.224  173          4754X          MOV     A,E
060.225  207          4755X          ADD     A
060.226  203          4756X          ADD     E               (A) = 3*MONTH
060.227  312 300 060  4757X          JZ      DAD1            BAD VALUE
060.232  376 047      4758X          CPI     13*3
060.234  322 300 060  4759X          JNC     DAD1            TOO LARGE
060.237  353          4760X          XCHG                    (DE) = ADDRESS
060.240  041 311 060  4761X          LXI     H,DADB-3
060.243  315 101 030  4762X          CALL    $DADA,          (HL) = ADDRESS OF MONTH
060.246  001 003 000  4763X          LXI     B,3
060.251  353          4764X          XCHG                    (HL) = BUFFER ADDR, (DE) = ADDR IN 'DADB'
060.252  315 252 030  4765X          CALL    $MOVE           MOVE MONTH IN
060.255  066 055      4766X          MVI     M,'-'
060.257  043          4767X          INX     H
060.260  301          4768X          POP     B               (BC) = YEAR
060.261  171          4769X          MOV     A,C
060.262  306 106      4770X          ADI     70
060.264  376 144      4771X          CPI     100
060.266  077          4772X          CMC
060.267  330          4773X          RC                      TOO LARGE
060.270  117          4774X          MOV     C,A             (BC) = YEAR
060.271  076 002      4775X          MVI     A,2
060.273  315 157 031  4776X          CALL    $UDD            UNPACK YEAR
060.276  247          4777X          ANA     A
060.277  311          4778X          RET
                      4779X
                      4780X *        ILLEGAL FORMAT. (NOT ALL ILLEGALS EXIT HERE!)
                      4781X
060.300  341          4782X DAD1     POP     H               RESTORE STACK
060.301  067          4783X          STC                     FLAG ERROR
060.302  311          4784X          RET
                      4785X
                      4786X *        No-Date                                         /80.08.sc/
                      4787X
060.303  001 011 000  4788X DAD2     LXI     B,DADCL                                 /80.08.sc/
060.306  021 360 060  4789X          LXI     D,DADC                                  /80.08.sc/
060.311  303 252 030  4790X          JMP     $MOVE                                   /80.08.sc/
                      4791X
060.314  112 141 156  4792X DADB     DB      'JanFebMarAprMayJunJulAusSepOctNovDec'
                      4793X
060.360  040 116 157  4794X DADC     DB      ' No-Date '                             /80.08.sc/
000.011               4795X DADCL    EQU     *-DADC                                  /80.08.sc/
```

```
060.371                  4796          XTEXT   UDDN


                         4798X **      $UDDN - UNPACK DECIMAL DIGITS,
                         4799X *
                         4800X *       UDDN CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
                         4801X *       DECIMAL DIGITS. THE RESULT IS NULL FILLED TO THE LEFT.
                         4802X *
                         4803X *       ENTRY   (B,C) = ADDRESS VALUE
                         4804X *               (A) = DIGIT COUNT
                         4805X *               (H,L) = MEMORY ADDRESS
                         4806X *       EXIT    (HL) = (HL) + (A)
                         4807X *       USES    ALL
                         4808X
                         4809X
060.371                  4810X $UDDN   EQU     *
060.371  315 072 030     4811X         CALL    $DADA
060.374  345             4812X         PUSH    H               SAVE FINAL (H,L) VALUE
                         4813X
060.375  365             4814X UDDN1   PUSH    PSW
060.376  345             4815X         PUSH    H
060.377  021 012 000     4816X         LXI     D,10
061.002  315 106 030     4817X         CALL    $DU66           (H,L) = VALUE/10
061.005  104             4818X         MOV     B,H
061.006  115             4819X         MOV     C,L             (BC) = QUOTIENT
061.007  341             4820X         POP     H
061.010  076 060         4821X         MVI     A,'0'
061.012  203             4822X         ADD     E               ADD REMAINDER
061.013  053             4823X         DCX     H
061.014  167             4824X         MOV     M,A             STORE DIGIT
061.015  170             4825X         MOV     A,B
061.016  261             4826X         ORA     C
061.017  312 031 061     4827X         JZ      UDDN2           ALL ZEROS
061.022  361             4828X         POP     PSW
061.023  075             4829X         DCR     A
061.024  302 375 060     4830X         JNZ     UDDN1           IF MORE TO GO
                         4831X
                         4832X *       ALL DONE, EXIT
                         4833X
061.027  341             4834X UDDN1.5 POP     H               RESTORE H
061.030  311             4835X         RET                       RETURN
                         4836X
                         4837X *       DIGITS LEADING THIS ONE ARE ZERO, STORE NULLS INSTEAD.
                         4838X
061.031  361             4839X UDDN2   POP     PSW
061.032  075             4840X UDDN3   DCR     A
061.033  312 027 061     4841X         JE      UDDN1.5         ALL DONE
061.036  053             4842X         DCX     H
061.037  066 000         4843X         MVI     M,0
061.041  303 032 061     4844X         JMP     UDDN3
061.044                  4845          XTEXT   MOVEL
```

```
                                4847X **      $MOVEL - MOVE DATA
                                4848X *
                                4849X *       $MOVEL MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                                4850X *       IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                                4851X *       FIRST TO LAST.
                                4852X *
                                4853X *       IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                                4854X *       LAST TO FIRST.
                                4855X *
                                4856X *       THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                                4857X *
                                4858X *       CALL    $MOVEL
                                4859X *       DW      COUNT
                                4860X *       DW      FROM
                                4861X *       DW      TO
                                4862X *
                                4863X *       ENTRY   ((SP)) = RET
                                4864X *               (RET+0) = COUNT (WORD VALUE)
                                4865X *               (RET+2) = FROM
                                4866X *               (RET+4) = TO
                                4867X *       EXIT    TO (RET+6)
                                4868X *               (DE) = ADDRESS OF NEXT FROM BYTE
                                4869X *               (HL) = ADDRESS OF NEXT *TO* BYTE
                                4870X *               'C' CLEAR
                                4871X *       USES    ALL
                                4872X
                                4873X
061.044  341                    4874X $MOVEL  POP     H               (HL) = RET
061.045  116                    4875X         MOV     C,M
061.046  043                    4876X         INX     H
061.047  106                    4877X         MOV     B,M             (BC) = COUNT
061.050  043                    4878X         INX     H
061.051  136                    4879X         MOV     E,M
061.052  043                    4880X         INX     H
061.053  126                    4881X         MOV     D,M             (DE) = FROM
061.054  043                    4882X         INX     H
061.055  325                    4883X         PUSH    D               ((SP)) = FROM
061.056  136                    4884X         MOV     E,M
061.057  043                    4885X         INX     H
061.060  126                    4886X         MOV     D,M             (DE) = TO
061.061  043                    4887X         INX     H
061.062  343                    4888X         XTHL                    ((SP)) = RET, (HL) = FROM
061.063  353                    4889X         XCHG                    (DE) = FROM , (HL) = TO
061.064  303 252 030            4890X         JMP     $MOVE           MOVE IT
061.067                         4891          XTEXT   RCHAR


                                4893X **      $RCHAR - READ SINGLE CHARACTER FROM CONSOLE.
                                4894X *
                                4895X *       ENTRY   NONE
                                4896X *       EXIT    (A) = CHARACTER
                                4897X *       USES    A,F
                                4898X
                                4899X
```

```
061.067  377 001      4900X $RCHAR  DB      SYSCALL,.SCIN
061.071  332 067 061  4901X         JC      $RCHAR          NOT READY
061.074  311          4902X         RET
                      4903X
061.075  377 002      4904X $WCHAR  DB      SYSCALL,.SCOUT
061.077  311          4905X         RET
061.100               4906          XTEXT   XCHGBC



                      4908X **      XCHGBC  -  XCHG BC
                      4909X *
                      4910X *       EXCHANGE THE 'BC' REGISTER PAIR WITH THE 'HL' REGISTER PAIR.
                      4911X *
                      4912X *       ENTRY:  BC      = ORIGINAL BC
                      4913X *               HL      = ORIGINAL HL
                      4914X *
                      4915X *       EXIT:   BC      = ORIGINAL HL
                      4916X *               HL      = ORIGINAL BC
                      4917X *
                      4918X *       USES:   BC,HL
                      4919X *
                      4920X
061.100  365          4921X XCHGBC  PUSH    PSW
061.101  170          4922X         MOV     A,B
061.102  104          4923X         MOV     B,H
061.103  147          4924X         MOV     H,A
061.104  171          4925X         MOV     A,C
061.105  115          4926X         MOV     C,L
061.106  157          4927X         MOV     L,A
061.107  361          4928X         POP     PSW
061.110  311          4929X         RET
061.111               4930          XTEXT   DRS



                      4932X **      $DRS - DECODE AND REMOVE SWITCHES.
                      4933X *
                      4934X *       $DRS IS CALLED TO DECODE COMMAND SWITCHES FROM A LINE
                      4935X *       OF TEXT. SWITCHES TAKE THE FORM:
                      4936X *
                      4937X *       /XXXXX
                      4938X *
                      4939X *       AFTER A SWITCH HAS BEEN LOCATED, IT (AND THE PRECEDING '/')
                      4940X *       ARE REPLACED WITH BLANKS.
                      4941X *
                      4942X *       VALID SWITCH DESCRIPTIONS ARE ENCODED INTO A TABLE
                      4943X *       SUPPLIED BY THE CALLER, IN THE FORMAT:
                      4944X *
                      4945X *       DB      'X...X'         REQUIRED SWITCH CHARACTERS
                      4946X *       DB      'C'+200Q,...,'C'+200Q   OPTIONAL CHARACTERS
                      4947X *       DB      200Q            END OF CHARACTERS
                      4948X *       DW      ADDR            PROCESSOR ADDRESS (CALLED WHEN SWITCH DETECTED)
                      4949X *
```

```
                        4950X *       DB      'Y...Y'         NEXT SWITCH
                        4951X *       .       .
                        4952X *       .       .
                        4953X *       .       .
                        4954X *
                        4955X *       DB      0               FLAGS END OF TABLE
                        4956X *
                        4957X *       SWITCHES MUST BE FOLLOWED BY A ':', A '/' (ANOTHER SWITCH)
                        4958X *       A ',', OR A 00 BYTE.
                        4959X *
                        4960X *       UPON DETECTION OF A VALID SWITCH, $DRS CALLS THE USER PROCESS
                        4961X *       ROUTINE. UPON ENTRY,
                        4962X *       (HL) = ADDRESS OF THE FIRST BYTE FOLLOWING THE SWTICH
                        4963X *       'Z' CLEAR IF CHARACTER = '/', ',', OR 00
                        4964X *       'Z' SET IF CHARACTER = ':'
                        4965X *
                        4966X *       THE USER ROUTINE CAN DECODE SWITCH SUB-OPTIONS, IF DESIRED.
                        4967X *       THE USER ROUTINE MAY USE ALL REGISTERS.
                        4968X *
                        4969X *       ENTRY   (DE) = SWITCH TABLE FWA
                        4970X *               (HL) = LINE FWA
                        4971X *       EXIT    'C' CLEAR IF OK
                        4972X *               'C' SET IF ERROR
                        4973X *                (HL) = ADDRESS OF START OF BAD SWITCH
                        4974X *                (A) = ERROR CODE
                        4975X *       USES    ALL
                        4976X
                        4977X
061.111                 4978X $DRS    EQU     *
                        4979X
                        4980X *       LOOK FOR SWITCHES
                        4981X
061.111  176            4982X $DRS1   MOV     A,M
061.112  247            4983X         ANA     A
061.113  310            4984X         RZ                      END OF LINE
061.114  043            4985X         INX     H
061.115  376 057        4986X         CPI     '/'
061.117  302 111 061    4987X         JNE     $DRS1           NOT A SWITCH
061.122  042 306 061    4988X         SHLD    $DRSB           ($DRSB) = SWITCH FWA (AFTER '/')
                        4989X
                        4990X *       GOT A SWITCH. LOOK FOR A MATCH IN THE CALLER'S TABLE
                        4991X
061.125  325            4992X         PUSH    D               SAVE TABLE FWA
061.126  052 306 061    4993X $DRS2   LHLD    $DRSB           (HL) = SWITCH FWA
061.131  032            4994X $DRS3   LDAX    D               (A) = TABLE ENTRY
061.132  346 177        4995X         ANI     177Q
061.134  312 204 061    4996X         JZ      $DRS6           GOT A MATCH
061.137  276            4997X         CMP     M
061.140  302 150 061    4998X         JNE     $DRS4           NO MATCH
061.143  023            4999X         INX     D
061.144  043            5000X         INX     H
061.145  303 131 061    5001X         JMP     $DRS3           SEE IF MORE MATCH
                        5002X
                        5003X *       HAVE MIS-MATCH. SEE IF THE MISSING CHARACTER IS SIGNIFICANT
                        5004X
061.150  176            5005X $DRS4   MOV     A,M             (A) = LINE CHARACTER WE COULDNT MATCH
```

```
061.151  315 255 061  5006X          CALL    $DRS15          SEE IF OK TERMINATOR
061.154  302 164 061  5007X          JNE     $DRS4.5         NO MATCH ON THIS SWITCH
061.157  032          5008X          LDAX    D               (A) = NEXT CHARACTER IN SWITCH PATTERN
061.160  247          5009X          ANA     A
061.161  372 204 061  5010X          JM      $DRS6           HAVE SUFFICIENT MATCH
061.164  315 270 061  5011X $DRS4.5 CALL    $DRS20          SKIP TABLE ENTRY
061.167  032          5012X          LDAX    D
061.170  247          5013X          ANA     A
061.171  302 126 061  5014X          JNZ     $DRS2           MORE SWITCHES IN TABLE TO CHECK
                      5015X
                      5016X *        BAD SWITCH
                      5017X
061.174  321          5018X $DRS5    POP     D               RESTORE STACK
061.175  052 306 061  5019X          LHLD    $DRSB           POINT TO BAD SWITCH
061.200  067          5020X          STC
061.201  076 032      5021X          MVI     A,EC.IS         ILLEGAL SWITCH
061.203  311          5022X          RET
                      5023X
                      5024X *        HAVE SWITCH. CHECK IT'S FOLLOWING CHARACTER
                      5025X
061.204  315 321 057  5026X $DRS6    CALL    $SOB            SKIP OVER BLANKS
061.207  176          5027X          MOV     A,M
061.210  315 255 061  5028X          CALL    $DRS15          CHECK CHARACTER
061.213  302 174 061  5029X          JNE     $DRS5           IN ERROR
061.216  315 270 061  5030X          CALL    $DRS20          GET PROCESSOR ADDRESS
061.221  021 233 061  5031X          LXI     D,$DRS7
061.224  345          5032X          PUSH    H               SAVE (HL)
061.225  325          5033X          PUSH    D               SET RETURN ADDRESS FOR TABLE CODE
061.226  305          5034X          PUSH    B               SAVE PROCESSOR ADDRESS
061.227  176          5035X          MOV     A,M             (A) = NEXT CHARACTER
061.230  376 072      5036X          CPI     ':'             SET CONDITION CODES
061.232  311          5037X          RET                     CALL USER PROCESS
                      5038X
                      5039X *        USER PROCESS RETURNS HERE
                      5040X
061.233  321          5041X $DRS7    POP     D               (DE) = LAST CHARACTER OF SWITCH+1
061.234  052 306 061  5042X          LHLD    $DRSB           (HL) = FIRST CHARACTER OF SWITCH AFTER /
061.237  053          5043X          DCX     H               (HL) = ADDRESS OF '/'
                      5044X
                      5045X *        REPLACE SWITCH WITH BLANKS
                      5046X
061.240  066 040      5047X $DRS8    MVI     M,' '
061.242  043          5048X          INX     H
061.243  315 216 030  5049X          CALL    $CDEHL
061.246  302 240 061  5050X          JNE     $DRS8           NOT THERE YET
061.251  321          5051X          POP     D               (DE) = SWITCH TABLE FWA
061.252  303 111 061  5052X          JMP     $DRS1           LOOK FOR MORE SWITCHES
```

```
                                5054X **      $DRS15 - CHECK FOR VALID DELIMITER CHARACTER.
                                5055X *
                                5056X *       $DRS15 CHECKS THE NEXT TEXT CHARACTER TO SEE IF IT IS
                                5057X *
                                5058X *       00, '/', ',', ':'
                                5059X *
                                5060X *       ENTRY   (A) = CHARACTER
                                5061X *       EXIT    'Z' SET IFF CHARACTER IS ONE OF THE ABOVE
                                5062X *       USES    F
                                5063X
061.255  247                    5064X $DRS15  ANA     A
061.256  310                    5065X         RZ                      IS 00
061.257  376 057                5066X         CPI     '/'
061.261  310                    5067X         RE
061.262  376 054                5068X         CPI     ','
061.264  310                    5069X         RE
061.265  376 072                5070X         CPI     ':'
061.267  311                    5071X         RET

                                5073X **      $DRS20 - GET PROCESSOR ADDRESS.
                                5074X *
                                5075X *       $DRS20 IS CALLED TO GET THE PROCESSOR ADDRESS FIELD OUT OF
                                5076X *       AN ENTRY IN THE SWITCH TABLE. THE CALLER SUPPLIES A POINTER
                                5077X *       TO SOMEWHERE IN THE TEXT PART OF THE SWITCH DESCRIPTION;
                                5078X *       $DRS20 ADVANCES THE POINTER TO THE PROCESSOR ADDRESS.
                                5079X *
                                5080X *       ENTRY   (DE) = POINTER TO TEXT PART OF SWITCH ENTRY
                                5081X *       EXIT    (DE) = POINTER TO 1ST BYTE OF NEXT SWITCH TABLE ENTRY
                                5082X *               (BC) = PROCESSOR ADDRESS FROM TABLE
                                5083X *       USES    A,F,B,C,D,E
                                5084X
                                5085X
061.270  032                    5086X $DRS20  LDAX    D
061.271  023                    5087X         INX     D
061.272  376 200                5088X         CPI     200Q
061.274  302 270 061            5089X         JNE     $DRS20
061.277  032                    5090X         LDAX    D               (A) = LOW BYTE OF PROCESSOR ADDRESS
061.300  117                    5091X         MOV     C,A
061.301  023                    5092X         INX     D
061.302  032                    5093X         LDAX    D
061.303  107                    5094X         MOV     B,A             (BC) = PROCESSOR ADDRESS
061.304  023                    5095X         INX     D
061.305  311                    5096X         RET
                                5097X
061.306  000 000                5098X $DRSB   DW      0               POINTER TO SWITCH BEING PROCESSED
000.001                         5099          IF      .PIP.
                                5100          XTEXT   DTB
                                5101          XTEXT   FOPE
                                5102          XTEXT   FWRIB
                                5103          XTEXT   FCLO
                                5104          XTEXT   FUTIL
                                5105          ENDIF
```

```
061.310                5108  PATCH   DS      64              PATCH AREA
000.000                5109          IF      ONECOPY                                         /2.0a/
062.010                5110          DS      *+255/256*256-* Auxiliary Patch Area (Round up 1 page)  /2.0a/
                       5111          ENDIF                                                   /2.0a/
```

```
000.000                 5114            IF      ONECOPY
                        5115
                        5116
                        5117    **      FDN - FILE DESCRIPTOR NODES.
                        5118    *
                        5119    *       THESE NODES ARE USED TO KEEP TRACK OF FILES WHICH ARE BEING
                        5120    *       HELD IN MEMORY WHILE TRANSFERING.
                        5121
063.000                 5122    FDN     DS      0               START OF TYPICAL NODE
000.000                 5123    FDN.LNK EQU     *-FDN           LINK TO NEXT NODE IN CHAIN
063.000                 5124            DS      1               ALL IN SAME PAGE,  JUST KEEP PAGE INDEX
000.001                 5125    FDN.STA EQU     *-FDN           STATUS BYTE
000.020                 5126    ST.CNT  EQU     DIF.CNT         IS CONTIGUOUS
000.002                 5127    ST.OPR  EQU     00000010B       IS BEING READ
000.001                 5128    ST.OPW  EQU     00000001B       OPEN FOR WRITE
063.001                 5129            DS      1               STATUS BYTE
000.002                 5130    FDN.SIZ EQU     *-FDN           TOTAL SIZE OF FILE (IF ST.CNT SET)
063.002                 5131            DS      1               SIZE IN GROUPS
000.003                 5132    FDN.AMR EQU     *-FDN           AMOUNT ALREADY READ
063.003                 5133            DS      2                IN SECTORS
000.005                 5134    FDN.AMW EQU     *-FDN           AMOUNT ALREADY WRITTEN
063.005                 5135            DS      2                IN SECTORS
000.007                 5136    FDN.ADR EQU     *-FDN           ADDRESS IN BUFFER
063.007                 5137            DS      1                ADDRESS/256 (MUST BE EVEN PAGE)
000.010                 5138    FDN.AIM EQU     *-FDN           AMOUNT IN MEMORY
063.010                 5139            DS      1                 IN SECTORS
000.011                 5140    FDNELEN EQU     *-FDN           ENTRY  LENGTH
063.000                 5141            ORG     FDN             ORG BACK OVER DEFINITION AREA
                        5142
                        5143
                        5144
                        5145    **      TABLE. A LINK OF 0 IS A NULL LINK.
                        5146    *
                        5147    *       THE ENTIRE GROUP OF NODES MUST RESIDE
                        5148    *       IN THE SAME PAGE
                        5149
063.000                 5150    FDNFWA  EQU     *               START OF NODES
                        5151
063.000  002           5152    FDNFRE  DB      #FDN.1          START OF FREE CHAIN
063.001  000           5153    FDNHEAD DB      0                       ACTIVE LIST NOW EMPTY
                        5154
063.002                 5155    FDN.1   DS      0
063.002  013           5156            DB      #FDN.2          FDN.LNK
063.003  000           5157            DB      0               FDN.STA
063.004  000           5158            DB      0               FDN.SIZ
063.005  000 000       5159            DW      0               FDN.AMR
063.007  000 000       5160            DW      0               FDN.AMW
063.011  000           5161            DB      0               FDN.ADR
063.012  000           5162            DB      0               FDN.AIM
                        5163
063.013                 5164    FDN.2   DS      0
063.013  024           5165            DB      #FDN.3          FDN.LNK
063.014  000           5166            DB      0               FDN.STA
063.015  000           5167            DB      0               FDN.SIZ
063.016  000 000       5168            DW      0               FDN.AMR
063.020  000 000       5169            DW      0               FDN.AMW
```

```
063.022  000          5170          DB      0               FDN.ADR
063.023  000          5171          DB      0               FDN.AIM
                       5172
063.024               5173  FDN.3   DS      0
063.024  035          5174          DB      #FDN.4          FDN.LNK
063.025  000          5175          DB      0               FDN.STA
063.026  000          5176          DB      0               FDN.SIZ
063.027  000 000      5177          DW      0               FDN.AMR
063.031  000 000      5178          DW      0               FDN.AMW
063.033  000          5179          DB      0               FDN.ADR
063.034  000          5180          DB      0               FDN.AIM
                       5181
063.035               5182  FDN.4   DS      0
063.035  046          5183          DB      #FDN.5          FDN.LNK
063.036  000          5184          DB      0               FDN.STA
063.037  000          5185          DB      0               FDN.SIZ
063.040  000 000      5186          DW      0               FDN.AMR
063.042  000 000      5187          DW      0               FDN.AMW
063.044  000          5188          DB      0               FDN.ADR
063.045  000          5189          DB      0               FDN.AIM
                       5190
063.046               5191  FDN.5   DS      0
063.046  057          5192          DB      #FDN.6          FDN.LNK
063.047  000          5193          DB      0               FDN.STA
063.050  000          5194          DB      0               FDN.SIZ
063.051  000 000      5195          DW      0               FDN.AMR
063.053  000 000      5196          DW      0               FDN.AMW
063.055  000          5197          DB      0               FDN.ADR
063.056  000          5198          DB      0               FDN.AIM
                       5199
063.057               5200  FDN.6   DS      0
063.057  070          5201          DB      #FDN.7          FDN.LNK
063.060  000          5202          DB      0               FDN.STA
063.061  000          5203          DB      0               FDN.SIZ
063.062  000 000      5204          DW      0               FDN.AMR
063.064  000 000      5205          DW      0               FDN.AMW
063.066  000          5206          DB      0               FDN.ADR
063.067  000          5207          DB      0               FDN.AIM
                       5208
063.070               5209  FDN.7   DS      0
063.070  101          5210          DB      #FDN.8          FDN.LNK
063.071  000          5211          DB      0               FDN.STA
063.072  000          5212          DB      0               FDN.SIZ
063.073  000 000      5213          DW      0               FDN.AMR
063.075  000 000      5214          DW      0               FDN.AMW
063.077  000          5215          DB      0               FDN.ADR
063.100  000          5216          DB      0               FDN.AIM
                       5217
063.101               5218  FDN.8   DS      0
063.101  000          5219          DB      0               FDN.LNK
063.102  000          5220          DB      0               FDN.STA
063.103  000          5221          DB      0               FDN.SIZ
063.104  000 000      5222          DW      0               FDN.AMR
063.106  000 000      5223          DW      0               FDN.AMW
063.110  000          5224          DB      0               FDN.ADR
063.111  000          5225          DB      0               FDN.AIM
```

```
                                5226
000.010                         5227  FDNCNT  EQU     *-FDN.1/FDNELEN         NUMBER OF NODES
                                5228
000.063                         5229  .       SET     */256
000.000                         5230          ERRNZ   FDNFWA/256-.    MUST BE ALL IN SAME PAGE
                                5231
063.112  000                    5232  VOLFLAG DB      0               =0 IF READING FROM SOURCE, =377Q IF WRITTING TO DEST
063.113  000                    5233  VOLSER  DB      0               SERIAL NUMBER OF CURRENT DISK
                                5234
063.114  000                    5235  OBUFLIM DB      0               BUFFER LIMIT/256
063.115  000                    5236  OBUFPTR DB      0               NEXT FREE PAGE IN BUFFER/256
                                5237
                                5238
                                5239          ENDIF
                                5240
063.116                         5241          XTEXT   FERROR          APPEARS HERE TO ALLOW FDN. TO BE IN ONE PAGE


                                5243X **       $FERROR - PROCESS FILE ERRORS.
                                5244X *
                                5245X *       $FERROR IS CALLED TO COMPLAIN ABOUT AN ERROR ENCOUNTERED
                                5246X *       WHEN PROCESSING FILES.
                                5247X *
                                5248X *       ENTRY   (A) = ERROR CODE
                                5249X *                 (HL) = ADDRESS OF FILE NAME - FB.NAM
                                5250X *       EXIT    TO RESTART
                                5251X *       USES    ALL
                                5252X
                                5253X
063.116  365                    5254X $FERROR PUSH    PSW             SAVE CODE
063.117  315 136 031            5255X         CALL    $TYPTX
063.122  012 007 105            5256X         DB      NL,BELL,'ERROR ON FILE',' '+200Q
063.142  021 012 000            5257X         LXI     D,FB.NAM
063.145  031                    5258X         DAD     D
                                5259X
                                5260X *       PRINT FILE NAME
                                5261X
063.146  176                    5262X $FERR1  MOV     A,M
063.147  043                    5263X         INX     H               ADVANCE MESSAGE
063.150  247                    5264X         ANA     A
063.151  312 162 063            5265X         JZ      $FERR2
063.154  315 075 061            5266X         CALL    $WCHAR
063.157  303 146 063            5267X         JMP     $FERR1
                                5268X
                                5269X *       TYPE ERROR MESSAGE
                                5270X
063.162  315 136 031            5271X $FERR2  CALL    $TYPTX
063.165  040 055 240            5272X         DB      ' -',' '+200Q
063.170  046 012                5273X         MVI     H,NL
063.172  361                    5274X         POP     PSW             (A) = CODE
063.173  377 057                5275X         DB      SYSCALL,.ERROR
063.175  303 237 042            5276X         JMP     RESTART         EXIT
```

```
063.200  000          5279  ALLOCA  DB      0               /ALL flag (<>0 if /ALL specified) /80.06.sc/
063.201  000          5280  COMAND  DB      0               COMMAND IN PROGRESS
063.202  000          5281  MODE    DB      0               <>0 IF LINE PASSED ON STACK
063.203  000          5282  JGL     DB      0               /JGL FLAG (<>0 IF /JGL SPECIFIED)
063.204  000          5283  SUPRES  DB      0               /SUP FLAG (<>0 OF /SUP SPECIFIED)
063.205  001          5284  SYSTEM  DB      1               /S FLAG (=0 IF /S SPECIFIED)
                      5285
063.206  130 130 130  5286  DIRNAM  DB      'XXX:DIRECT.SYS',0      DIRECTORY FILE NAME
                      5287
063.225  114 066      5288  BUFPTR  DW      BUFF            POINTER TO START  OF BUFFER
063.227  000 000      5289  BUFSIZ  DW      0               BUFFER LENGTH


                      5291  **      FILE BLOCKS
                      5292
000.001               5293          IF      .PIP.
                      5294  DESTFB  DS      0               DESTINATION FILE BLOCK
                      5295          DB      CN.DES          CHANNEL NUMBER
                      5296          DB      0               FLAGS
                      5297          DW      DESTBUF
                      5298          DW      DESTBUF
                      5299          DW      DESTBUF
                      5300          DW      DESTBFE         END OF BLOCK
                      5301          DS      FB.NAML         NAME AREA
                      5302          ELSE
063.231               5303  DESTFB  DS      0               DUMY BUFFER
063.231  310          5304          DB      200             ILLEGAL CHANNEL NUMBER
063.232  000          5305          DB      0               FLAGS
063.233  000 000      5306          DW      0
063.235  000 000      5307          DW      0
063.237  000 000      5308          DW      0
063.241  000 000      5309          DW      0               END OF BLOCK
063.243               5310          DS      FB.NAML         NAME AREA
                      5311          ENDIF


063.264  000 000      5313  NAMTLEN DW      0               NAME TABLE POINTER
063.266  000 000      5314  NAMTMAX DW      0               MAXIMUM SIZE OF NAME TABLE
000.000               5315          IF      ONECOPY
063.270  000 000      5316  NAMTPTR DW      0               POINTER TO ACTIVE ELEMENT IN NAMTAB
                      5317          ENDIF
                      5318
```

```
                                5322  ***     PRS - PRESET PIP PROGRAM.
                                5323  *
                                5324  *       PRS IS CALLED TO PERFORM ONE-TIME-ONLY PRESETTING OF
                                5325  *       THE PROGRAM ENVIRONMENT.
                                5326  *
                                5327  *       THE CODE IS OVERLAID BY BUFFERS AND WORK AREAS WHEN PIP IS RUNNING.
        000.001                 5328          IF      .PIP.
                                5329  *       BE CAREFUL NOT TO USE ANY OF THE BUFFERS AND WORK AREAS BEFORE
                                5330  *       THE AREA *LINE*.
                                5331          ELSE
                                5332  *       DO NOT USE ANY OF THE BUFFERS AND WORK AREAS IN *PRS*
                                5333          ENDIF
                                5334  *
                                5335  *
                                5336  *       ENTRY   NONE
                                5337  *
                                5338  *       EXIT    IF  CORRECT VERSION OF HDOS
                                5339  *                   NONE
                                5340  *                 ELSE
                                5341  *                   EXIT TO HDOS
                                5342  *
                                5343  *       USES    ALL
                                5344  *
                                5345
        063.272                 5346  ENTRY   EQU     *               INITIAL ENTRY POINT
        063.272  377 011        5347  PRS     DB      SYSCALL,.VERS
        063.274  332 324 064    5348          JC      PRS1            ERROR IN GETTING VERSION
        063.277  376 040        5349          CPI     VERS
        063.301  302 324 064    5350          JNZ     PRS1            NOT CORRECT VERSION OF HDOS
        063.304  041 114 066    5351          LXI     H,RMEML         (HL) = RUN-TIME HIGH MEMORY
        063.307  377 052        5352          DB      SYSCALL,.SETTP  SET HI MEMORY
        063.311  332 327 064    5353          JC      PRS2            IF  ERROR
        063.314  041 004 043    5354          LXI     H,CCHIT
        063.317  076 003        5355          MVI     A,CTLC
        063.321  377 041        5356          DB      SYSCALL,.CTLC   SET CTL-C PROCESSING
        063.323  076 377        5357          MVI     A,377Q
        063.325  377 046        5358          DB      SYSCALL,.CLOSE  CLOSE OVERLAY CHANNEL
        000.001                 5359          IF      .PIP.
                                5360
                                5361  *       SEE IF COMMAND LINE PASSED ON STACK
                                5362
                                5363          LXI     H,0
                                5364          DAD     SP
                                5365          XCHG
                                5366          MVI     A,$STACK
                                5367          SUB     E
                                5368          MOV     C,A
                                5369          MVI     A,STACK/256
                                5370          SBB     D
                                5371          MOV     B,A             (BC) = BYTES ON STACK
                                5372          ORA     C
                                5373          STA     MODE            SET MODE <>0 IF LINE ON STACK
                                5374          JZ      START           NO LINE
                                5375
                                5376  *       HAVE LCOMMAND ON STACK. COPY INTO LINE BUFFER
                                5377  *       (BC) = COUNT
```

```
                                5378  *       (DE) = FWA
                                5379
                                5380          LXI     H,LINE
                                5381          CALL    $MOVE           COPY
                                5382          MVI     M,0             ENSURE END
                                5383          ELSE    ONECOPY
063.327  315 377 064           5384          CALL    $DOS            DISMOUNT OPERATING SYSTEM
063.332  332 327 064           5385          JC      PRS2            IF ERROR
063.335  315 136 031           5386          CALL    $TYPTX
063.340  012 011 011           5387          DB      NL,TAB,TAB,TAB,'   ','ONECOPY'
063.356  012 011 011           5388          DB      NL,TAB,TAB,TAB,'Version:  ',VERS/16+'0','.',VERS&0FH+'0'
063.377  012 011 011           5389          DB      NL,TAB,TAB,'      ','Issue:  $50.06,00 '
064.032  012 012 011           5390          DB      NL,NL,' ONECOPY is used to copy files for systems with only one'
064.124  012 146 154           5391          DB      NL,'floppy drive. Read the appropriate manual before using.'
064.214  212                   5392          DB      ENL
064.215  315 136 031           5393          CALL    $TYPTX
064.220  012 111 156           5394          DB      NL,'Insert the initial source disk. Hit RETURN when ready:',' '+200Q
064.310  315 242 056           5395          CALL    GDWP,                                           /79.11.GC/
064.313  315 263 057           5396          CALL    $RTL            GET CR
                                5397
064.316  303 200 042           5398          JMP     PRS3            Jump the the rest of the code       /2.0a/
                                5399          ENDIF
064.321  303 246 042           5400          JMP     START           START PROGRAM
                                5401
064.324  076 050               5402  PRS1    MVI     A,EC.NCV        NOT CORRECT VERSION
064.326  067                   5403          STC
064.327  046 012               5404  PRS2    MVI     H,NL
064.331  377 057               5405          DB      SYSCALL,.ERROR
064.333  303 001 043           5406          JMP     EXIT
                                5407
000.000                         5408          IF      ONECOPY
064.336                         5409          XTEXT   DTB


                                5411X **      $DTB - DELETE TRAILING BLANKS.
                                5412X *
                                5413X *       $DTB DELETES THE TRAILING BLANKS FROM A CODED LINE.
                                5414X *
                                5415X *       ENTRY   (HL) = LINE FWA
                                5416X *       EXIT    (A) = LENGTH OF RESULT (ENCLUDING 00 TERMINATOR BYTE)
                                5417X *       USES    A,F
                                5418X
                                5419X
064.336  325                   5420X $DTB    PUSH    D               SAVE (DE)
064.337  124                   5421X         MOV     D,H
064.340  135                   5422X         MOV     E,L             (DE) = FWA
064.341  033                   5423X         DCX     D               (DE) = FWA-1
064.342  176                   5424X $DTB1   MOV     A,M
064.343  043                   5425X         INX     H
064.344  247                   5426X         ANA     A               FIND END OF LINE
064.345  302 342 064           5427X         JNZ     $DTB1
064.350  053                   5428X         DCX     H               (HL) = ADDRESS OF TERMINATING ZERO BYTE
                                5429X
                                5430X *       GOT END OF LINE. DELETE TRAILING BLANKS
```

```
                                5431X
064.351  053                   5432X $DTB2   DCX     H               BACKUP ONE CHARACTER
064.352  315 216 030            5433X         CALL    $CDEHL
064.355  312 366 064            5434X         JE      $DTB3           GONE PAST FRONT OF LINE, MUST BE ALL BLANKS
064.360  176                   5435X         MOV     A,M
064.361  376 040               5436X         CPI     ' '
064.363  312 351 064            5437X         JE      $DTB2           GOT BLANK
                                5438X
                                5439X *       HAVE TRIMED LINE. COMPUTE LENGTH
                                5440X
064.366  043                   5441X $DTB3   INX     H
064.367  066 000               5442X         MVI     M,0             TERMINATE LINE
064.371  175                   5443X         MOV     A,L
064.372  223                   5444X         SUB     E               (A) = LENGTH +1 (FOR 00 BYTE)
064.373  353                   5445X         XCHG
064.374  043                   5446X         INX     H               (HL) = LINE FWA
064.375  321                   5447X         POP     D               RESTORE (DE)
064.376  311                   5448X         RET
064.377                         5449          XTEXT   DOS


                                5451X **      $DOS - DISMOUNT OPERATING SYSTEM.
                                5452X *
                                5453X *       $DOS disounts all units of all directory devices        /80.04.sc/
                                5454X *
                                5455X *       THE USER IS MESSAGED ABOUT THE DISKS, AND THE OPERATING
                                5456X *       SYSTEM IS NOTIFIED.
                                5457X *
                                5458X *
                                5459X *       ENTRY   NONE
                                5460X *
                                5461X *       EXIT    (PSW)   = 'C' CLEAR IF NO ERROR
                                5462X *                         'C' SET   IF    ERROR
                                5463X *                          (A)  = ERROR CODE
                                5464X *
                                5465X *       USES    ALL
                                5466X *
                                5467X
064.377  315 136 031            5468X $DOS    CALL    $TYPTX
065.002  012 007 104            5469X         DB      NL,BELL,'Dismounting All Disks!',NL,ENL
                                5470X
065.034  315 131 065            5471X         CALL    $DOS.
065.037  330                   5472X         RC
                                5473X
065.040  315 136 031            5474X         CALL    $TYPTX
065.043  012 122 145            5475X         DB      NL,'Remove the Disk(s). Hit RETURN when ready!',' '+200Q
                                5476X
065.117  315 067 061            5477X DOS1    CALL    $RCHAR          READ CHARACTER
065.122  376 012                5478X         CPI     NL
065.124  302 117 065            5479X         JNE     DOS1
                                5480X
065.127  247                   5481X         ANA     A               CLEAR CARRY
065.130  311                   5482X         RET
```

```
065.131  076 000      5484X $DOS.   MVI     A,OVL0
065.133  377 010      5485X         SCALL   .LOADO
065.135  330          5486X         RC
                      5487X
065.136  076 001      5488X         MVI     A,OVL1
065.140  377 010      5489X         SCALL   .LOADO
065.142  330          5490X         RC
                      5491X
065.143  377 206      5492X         SCALL   .DAD          Dismount all Disks          /80.09.sc/
065.145  311          5493X         RET
                      5494          ENDIF
                      5495
065.146               5496  MEML    EQU     *             MEMORY LENGTH
```

```
                    5499  **      THE FOLLOWING BUFFERS AND AREAS OVERLAY THE PRS CODE.
                    5500  *
                    5501  *       *PRS* MAY NOT USE ANY CELLS BELOW *LINE*, AT THE
                    5502  *       RISK OF SMASHING ITSELF
                    5503
063.272             5504          ORG     PRS
                    5505
063.272             5506  DEFALT  DS      6               DEFAULT BLOCK
                    5507
063.300             5508  MWNA    DS      FB.NAML         MWN WORK AREA
                    5509
000.001             5510          IF      .PIP.
                    5511  DESTBUF DS      256             DESTINATION FILE BUFFER (ALSO USED BY *CCW*)
                    5512  DESTBFE EQU     *               END OF BUFFER
                    5513          ENDIF
                    5514
                    5515  **      * * NOTE * *
                    5516  *       DIRWORK  USES THE SYSTEM SCRATCH AREA, LABEL. DIRWORK WILL NOT
                    5517  *       BE PRESERVED DURING A SYSCALL !!
                    5518
063.321             5519  SLABEL  DS      256             Saved Label Sector                      /2.0a/
064.321             5520  LABEL   DS      256             Label Sector                            /2.0a/
                    5521
                    5522  *DIRWORK        EQU     SECSCR          USE SECTOR SCRATCH AREA /79.11.GC/


                    5524  **      PIO.XXX - IMAGE OF SYSTEM AIO.XXX AREA
                    5525  *
                    5526  *       THESE CELLS MIRROR THE SYSTEM AIO.XXX AREA
                    5527
                    5528
065.321             5529  PIO.DEV DS      2               DEVICE CODE
065.323             5530  PIO.UNI DS      1               UNIT NUMBER (0-9)
                    5531
065.324             5532  PIO.DIR DS      DIRELEN         DIRECTORY ENTRY
                    5533
065.353             5534  $FOPWRK DS      FB.NAML         WORK AREA FOR $FOPE
                    5535
                    5536
000.001             5537          IF      .PIP.
                    5538          ERRMI   *-MEML          FOLLOWING MUST NOT OVERLAY *PRS*
                    5539          ENDIF
065.374             5540  LINE    DS      80              COMMAND BUFFER
                    5541
                    5542
066.114             5543  NAMTAB  DS      0               NAME TABLE
                    5544
                    5545
002.000             5546  BUFMINL EQU     512             MINIMUM SIZE FOR BUFFER (WHEN IN USE)
066.114             5547  BUFF    EQU     *               BUFFER AREA STARTS AFTER NAMTAB
                    5548
066.114             5549  RMEML   EQU     *               INITIAL RUNNING MEMORY LENGTH
                    5550
                    5551
                    5552
```

```
    066.114                5553        END
ASSEMBLY COMPLETE
  5553 STATEMENTS
     0 ERRORS DETECTED
  8580 BYTES FREE
```

```
$CCO    057172   956    4072L
$CDEHL  030216  1687    2510   4501E  5049   5433
$CFD    057134  3420    3972L
$CHL    030224  3531    4221E
$CMP$   000001  4352E   4396   4405
$COMP   030060  2132    2673   3270   4458E
$CRLF   057370  4094    4327L  4399
$DAD    060155  2791    4730L
$DADA   030072  4293E   4375   4811
$DADA.  030101  2524    3140   4539E  4762
$DOS    064377  5384    5468L
$DOS.   065131  5471    5484L
$DRS    061111   970    4978E
$DRS1   061111  4982L   4987   5052
$DRS15  061255  5006    5028   5064L
$DRS2   061126  4993L   5014
$DRS20  061270  5011    5030   5086L  5089
$DRS3   061131  4994L   5001
$DRS4   061150  4998    5005L
$DRS4.5 061164  5007    5011L
$DRS5   061174  5018L   5029
$DRS6   061204  4996    5010   5026L
$DRS7   061233  5031    5041L
$DRS8   061240  5047L   5050
$DRSB   061306  4988    4993   5019   5042   5098L
$DTB    064336  5420L
$DTB1   064342  5424L   5427
$DTB2   064351  5432L   5437
$DTB3   064366  5434    5441L
$DU66   030106  4529E   4738   4752   4817
$FERR1  063146  5262L   5267
$FERR2  063162  5265    5271L
$FERROR 063116  2929    2934   5254L
$FOPWRK 065353  5534L
$GNL    057207  1196    4089L
$HLIHL  030211  1744    2789   4550E
$INDL   030234  1746    1754   1757   1778   1877   1955   1967   2423   4587E
$INDLB  060052  2419    2520   3689   4602L
$INDS   060073  4635L
$INDSB  060127  4665L
$MCU    057243  4111    4130L
$MLU    057223  4106L   4156
$MLU1   057226  4109L   4114
$MOVE   030252   909    2103   2163   2395   3036   3327   3330   3367   3381   3408   3721   3871
                4209E   4765   4790   4890
$MOVEL  061044  1546    2376   2634   3653   3655   3828   3902   3945   3949   4874L
$MU86   031007  2572    2761   4060E
$RCHAR  061067  4160    4900L  4901   5477
$RSTALL 031047  3904    3953   4076   4475E
$RTL    057263  4154    4158E  5396
$RTL.   057254  1206    4154L
$RTL1   057264  4160L   4166
$RTL2   057316  4162    4183L
$SAVALL 031054  3901    3943   4072   4489E
$SOB    057321  2648    2893   3332   3418   3924   3929   4234L  5026
$SOB1   057322  4235L   4238   4240
$TBLS   057337  4262L
$TJMP   031061   973    4313E
```

```
$TJMP.  031062  4315E
$TYPC.  060002  4350L   4400
$TYPCC  057156  2458    2595    2832    3998E   4007
$TYPCH  057376  4340L
$TYPL.  060023  4377    4389E   4403
$TYPLN. 060005  4371L
$TYPTX  031136  1009    1197    1600    2107    2166    2899    2949    2953    2964    2980    4435E   5255
                5271    5386    5393    5468    5474
$TYPTX. 031144  4437E
$UDD    031157  4516E   4746    4776
$UDDN   060371  1599    2559    2566    2577    2778    4810E
$WCHAR  061075  2952    4904L   5266
$WDR    031222  4032E
$WER    031241  4020E
$ZERO   031212  1483    1706    4045E
.       000063  5229S   5230
.ABUSS  040024  808E
.ALARM  002136  781E
.ALEDS  040013  806E    2092    2100    2153
.CHFLG  000060  459L
.CLEAN  000205  474L
.CLEAR  000055  456L
.CLEARA 000056  457L     941
.CLOSE  000046  449L    1839    1978    2544    3735    5358
.CLRCO  000007  433L    1011
.CONSL  000006  432L    4075    4091
.CRC    002347  789E
.CRCSUM 040027  809E
.CTC    002172  783E
.CTL2FL 040066  815E
.CTLC   000041  444L    5356
.CTLFLG 040011  805E
.DAD    000206  475L    5492
.DECODE 000053  454L    2406    3659
.DELET  000050  451L    1935
.DISMT  000061  460L
.DLEDS  040021  807E
.DLY    000053  778E
.DMNMS  000203  472L    2084
.DMOUN  000201  470L
.DOD    003122  792E
.DODA   003356  794E
.DSPMOD 040007  803E
.DSPROT 040006  802E
.DUMP   001374  780E
.ERROR  000057  458L    2970    5275    5405
.EXIT   000000  426L    1002    2957
.HORN   002140  782E    2110
.IDENT  000000  777E
.IOWRK  040002  800E
.LINK   000040  443L
.LOAD   001267  779E
.LOADD  000062  461L
.LOADO  000010  434L    5485    5489
.MFLAG  040010  804E    2090
.MONMS  000202  471L    2183
.MOUNT  000200  469L
.NAME   000054  455L
```

```
.NMIRET 040064   814E
.OPEN   000063   462L
.OPENC  000045   448L  1942
.OPENR  000042   445L  1725  2433  3671
.OPENU  000044   447L  1951
.OPENW  000043   446L  1922
.PCHL   002264   785E
.PIP.   000001     1E     4   988  1026  1078  1198  1209  2231  2381  2440  2453  2583
                2663  2823  2901  2922  2987  3098  3262  3543  5099  5293  5328  5359  5510
                5537
.POSIT  000047   450L  1783  1960
.PRINT  000003   429L  2106  2983
.RCK    003260   793E
.READ   000004   430L  1797  2467  3681
.REGI   040005   801E
.REGPTR 040035   812E
.RENAM  000051   452L
.RESET  000204   473L
.RNB    002331   788E
.RNP    002325   787E
.SCIN   000001   427L  4900
.SCOUT  000002   428L  4004  4328  4350  4904
.SETTP  000052   453L  3540  3887  5352
.SRS    002265   786E
.START  040000   799E
.SYSRES 000012   436L
.TICCNT 040033   811E
.TPERR  002205   784E
.TPERRX 040031   810E
.UIVEC  040037   813E
.VERS   000011   435L  5347
.WNB    003024   791E
.WNP    003017   790E
.WRITE  000005   431L  1975
ABS.COD 000010   891L   894
ABS.ENT 000006   889L
ABS.ID  000000   885L
ABS.LDA 000002   887L
ABS.LEN 000004   888L
AC.DLY  000156    70E
ACL     043256   959   1196L
AEN     053160  2682  3018L  3639  3722
AENA    053232  3018  3035   3040L
AIO.CGN 041047   600L
AIO.CHA 041116   615L
AIO.CNT 041111   611L
AIO.CSI 041050   601L
AIO.DDA 041041   596E
AIO.DES 041055   605L
AIO.DEV 041057   606L
AIO.DIR 041062   609L
AIO.DTA 041053   604L
AIO.EOF 041113   613L
AIO.EOM 041112   612L
AIO.FLG 041043   597L
AIO.GRT 041044   598L
AIO.LGN 041051   602L
AIO.LSI 041052   603L
```

```
AIO.SPG 041046    599L
AIO.TFP 041114    614L
AIO.UNI 041061    607L
AIO.VEC 041040    595L
ALLOCA  063200    967     1120    2763    5279L
BELL    000007    484E    2108    2950    2965    2981    5256    5469
BKSP    000010    486E
BLS     051012    2391    2634L
BLS1    051041    2644L   2685
BLS2    051062    2651    2653L
BLS3    051100    2673L
BLS4    051113    2674    2682L
BLSA    051125    2635    2645    2661    2687L
BLSB    051133    2639    2688L
BLSC    051134    2635    2652    2689L
BOOT.P  000001    575E
BRIEF   047114    983     2369L
BSL     053253    1544    3055L
BSL1    053261    3060L   3076
BSL2    053314    3073L
BSLA    053324    3055    3068    3078L
BUFF    066114    951     5288    5547E
BUFMINL 002000    5546E
BUFPTR  063225    952     1567    3774    3884    5288L
BUFSIZ  063227    948     3776    3883    5289L
C.STX   000002    488E
C.SYN   000026    487E
CAD     054136    2490    3064    3250    3316L   3647    3827    3831
CAD.    054142    2653    3319L
CAD0    054144    3317    3320L
CAD1    054231    3335    3337    3339    3347L
CAD2    054306    3350    3374L
CAD2.4  054334    3388L   3391
CAD2.6  054342    3385    3392L
CAD3    055001    3395    3413L
CAD4    055003    3341    3343    3418L
CAD5    055016    3348    3357    3360    3370    3401    3404    3428L
CADA    055022    3321    3386    3432L
CB.CLI  000100    723E     746
CB.MTL  000040    722E
CB.SPK  000200    724E
CB.SSI  000020    721E
CB2.CLI 000002    727E
CB2.ORG 000040    728E
CB2.SID 000100    729E
CB2.SSI 000001    726E
CBR     046165    1646    1785    2032L
CCHIT   043004    1009L   5354
CCW     053325    3070    3097L
CDA     055230    3019    3265    3578L   3848
CDA5    055274    3580    3585    3590    3612L   3624
CDA6    055312    3619    3621L
CDA7    055314    3618    3623L
CDB.H84 000001    518E
CDB.H85 000000    517E
CFE     053331    2483    3138L   3706
CFS     053351    2570    2758    3159L
CFS.    053354    1761    3160L
```

```
CFS1     053357     3161L     3166
CN.170M  000014      764E
CN.174M  000003      763E
CN.ABO   000200      768E
CN.BAU   000100      767E
CN.DES   000001       49E     1921    1933    1941    1950    1959    1973    1977
CN.DIR   000002       50E     2432    2465    2543    3670    3679    3734
CN.MEM   000040      766E
CN.PRI   000020      765E
CN.SOU   000000       48E     1724    1745    1782    1796    1838
CND.H17  000000      770E
CND.H47  000001      772E
CND.NDI  000000      771E
CO.FLG   000001      670E     4074
COMAND   063201       965      972    1107    1147    1155    1160    1169    5280L
COPY     043347       979     1531E
CR       000015      480E
CS.FLG   000200      671E
CSL.CHR  000001      647E
CSL.ECH  000200      644E
CSL.RAW  000004      645E
CSL.WRP  000002      646E
CTLA     000001      495E
CTLB     000002      496E
CTLC     000003      497E     5355
CTLD     000004      498E     4161
CTLO     000017      499E
CTLP     000020      500E
CTLQ     000021      501E
CTLS     000023      502E
CTLZ     000032      503E
CTP.2SB  000010      656E
CTP.BKM  000002      657E
CTP.BKS  000200      652E
CTP.FF   000100      653E
CTP.MLI  000040      654E
CTP.MLO  000020      655E
CTP.TAB  000001      658E
CTS      053367     2889      3182L
CWM      054004     2495      3206L   3214    3711
CWM1     054013     3208      3211L
D.CON    040110      398L
D.RAM    040240      401L
D.VEC    040130      400L
DAD1     060300     4744      4757    4759    4782L
DAD2     060303     4732      4788L
DADB     060314     4761      4792L
DADC     060360     4789      4794L   4795
DADCL    000011     4788      4795E
DC.ABT   000007      693L
DC.CLO   000006      692L
DC.LOD   000011      695L
DC.MAX   000013      697L
DC.MOU   000010      694L     2207
DC.OPR   000003      689L
DC.OPU   000005      691L
DC.OPW   000004      690L
DC.RDY   000012      696L     2114    2118
```

```
DC.REA  000000   686L
DC.RER  000002   688L   2214
DC.WRI  000001   687L
DDF     054022  1539   3236L
DDF.BOL 000011   828E
DDF.BOO 000000   827L
DDF.LAB 000011   829L   2211
DDF.USR 000012   830L
DDF1    054027  3239L  3244
DDF1.0  054042  3245L  3252
DDF2    054045  3242   3249L
DDF3    054115  3271   3275L
DDFA    054122  3245   3280L
DDFB    054133  3268   3281L
DEFALT  063272  3063   3249   3903   3948   3952   5506L
DESTERR 052207  1923   1943   1952   1976   1979   2933L
DESTFB  063231   943   1548   1910   1920   1932   1940   1949   2933   3261   3269   5303L
DEV.DDA 000004   243L
DEV.DVG 000015   256L
DEV.DVL 000013   255L
DEV.FLG 000006   244L
DEV.JMP 000003   242L
DEV.MNU 000010   252L
DEV.MUM 000007   251L
DEV.NAM 000000   234L
DEV.RES 000002   238L
DEV.UNT 000011   253L   2414
DEVELEN 000016   258E
DF.CLR  000376   182E   3703
DF.EMP  000377   181E   2478   3700
DIF.CNT 000020   207E   2876   5126
DIF.LOC 000100   205E   2874
DIF.SYS 000200   204E   2873   3142
DIF.WP  000040   206E   2875
DIR.ALD 000025   197L
DIR.CLU 000015   190L
DIR.CRD 000023   196L   2786   2797
DIR.EXT 000010   185L   2744   3329   3407   3589   3951
DIR.FGN 000020   193L   1758   2744   2751
DIR.FLG 000016   191L   1747   2797   3139
DIR.LGN 000021   194L
DIR.LSI 000022   195L   2751   2786
DIR.NAM 000000   184L   2491   3097   3379   3584   3589   3656   3720
DIR.PRO 000013   186L
DIR.VER 000014   187L
DIRELEN 000027   199E    302    314    609   5532
DIRIDL  000015   188E
DIRNAM  063206  2394   2396   2404   2431   3654   3658   3669   5286L
DIS.ENL 001373   318L   2521   3690
DIS.ENT 000000   313E
DIS.LNK 001376   320L
DIS.SEC 001374   319L
DM.MR   000000   736E
DM.MW   000001   737E
DM.RR   000002   738E
DM.RW   000003   739E
DNT     055023  3347   3369   3400   3449L
DNT1    055032  3453L  3456
```

| | | | | | |
|---|---|---|---|---|---|
| DNT2 | 055043 | 3461L | 3484 | | |
| DNT3 | 055105 | 3464 | 3471 | 3479L | |
| DNT4 | 055130 | 3469 | 3473 | 3475 | 3502L |
| DNT5 | 055117 | 3467 | 3493L | 3497 | |
| DNTA | 055135 | 3449 | 3457 | 3503 | 3506L |
| DOS1 | 065117 | 5477L | 5479 | | |
| DR.IM | 000001 | 239E | | | |
| DR.PR | 000002 | 240E | | | |
| DT.CH | 000020 | 249E | | | |
| DT.CR | 000002 | 246E | | | |
| DT.CW | 000004 | 247E | | | |
| DT.DD | 000001 | 245E | 2409 | 3662 | |
| DT.RN | 000010 | 248E | | | |
| DV.EL | 000000 | 235E | | | |
| DV.NU | 000001 | 236E | | | |
| EBM | 055146 | 1550 | 3518L | | |
| EBM1 | 055206 | 3529 | 3537L | | |
| EC.CNA | 000004 | 344L | | | |
| EC.DDA | 000027 | 363L | | | |
| EC.DIF | 000017 | 355L | | | |
| EC.DIW | 000035 | 369L | | | |
| EC.DNI | 000045 | 377L | | | |
| EC.DNR | 000046 | 378L | | | |
| EC.DNS | 000005 | 345L | 2410 | 3276 | 3663 |
| EC.DSC | 000047 | 379L | | | |
| EC.EOF | 000001 | 341L | 1803 | | |
| EC.EOM | 000002 | 342L | | | |
| EC.FAO | 000031 | 365L | | | |
| EC.FAP | 000026 | 362L | | | |
| EC.FL | 000030 | 364L | | | |
| EC.FNF | 000014 | 352L | 1937 | | |
| EC.FNO | 000011 | 349L | | | |
| EC.FNR | 000034 | 368L | | | |
| EC.FOD | 000043 | 375L | | | |
| EC.FUC | 000013 | 351L | | | |
| EC.ICN | 000016 | 354L | | | |
| EC.IDN | 000006 | 346L | | | |
| EC.IFC | 000020 | 356L | | | |
| EC.IFN | 000007 | 347L | 3428 | 3982 | |
| EC.ILC | 000003 | 343L | | | |
| EC.ILO | 000040 | 372L | | | |
| EC.ILR | 000012 | 350L | | | |
| EC.ILV | 000037 | 371L | | | |
| EC.IOI | 000052 | 382L | | | |
| EC.IS | 000032 | 366L | 5021 | | |
| EC.NCV | 000050 | 380L | 5402 | | |
| EC.NEM | 000021 | 357L | 3779 | | |
| EC.NOS | 000051 | 381L | | | |
| EC.NPM | 000044 | 376L | | | |
| EC.NRD | 000010 | 348L | | | |
| EC.NVM | 000042 | 374L | | | |
| EC.OTL | 000053 | 383L | | | |
| EC.RF | 000022 | 358L | | | |
| EC.UNA | 000036 | 370L | | | |
| EC.UND | 000015 | 353L | | | |
| EC.UUN | 000033 | 367L | | | |
| EC.VPM | 000041 | 373L | | | |
| EC.WF | 000023 | 359L | | | |

```
EC.WP   000025      361L
EC.WPV  000024      360L
ENL     000212      493E    1603    2167    2909    2996    2997    2998    2999    3000    3001    3002    3003
                    5392    5469
ENTRY   063272      898     5346E
ERROR   053026      912     971     1112    1153    1540    1545    1563    1938    2085    2125    2184    2392
                    2407    2411    2435    2891    2898    2963L   3780    3889
ERROR1  053057      2968    2975L
ERROR2  053062      2976L   2978
ERRORA  053120      2975    2986L
ESC     000033      491E
EWS     055321      3072    3639L
EWS1    056030      3677L   3699
EWS3    056062      3697L   3730
EWS4    056141      3712    3727L
EWS6    056142      3702    3707    3728L
EWS7    056150      3683    3705    3734L
EWSA    056155      3738L
EWSB    056163      3657    3661    3740L
EWSC    056221      3656    3709    3742L
EXIT    043001      927     960     1001L   5406
FB.CHA  000000      326L
FB.FLG  000001      327L    943
FB.FWA  000002      328L
FB.LIM  000006      330L
FB.LWA  000010      331L
FB.NAM  000012      332L    333     1548    1910    1920    1932    1940    1949    2927    3261    3269    5257
FB.NAML 000021      333E    1558    1608    1674    1675    2499    3023    3033    3040    3645    3650    3864
                    3869    5310    5508    5534
FB.PTR  000004      329L
FBENL   000033      334E
FDN     063000      5122L   5123    5125    5130    5132    5134    5136    5138    5140    5141
FDN.1   063002      2049    5152    5155L   5227
FDN.2   063013      5156    5164L
FDN.3   063024      5165    5173L
FDN.4   063035      5174    5182L
FDN.5   063046      5183    5191L
FDN.6   063057      5192    5200L
FDN.7   063070      5201    5209L
FDN.8   063101      5210    5218L
FDN.ADR 000007      1788    1810    1823    1968    5136E
FDN.AIM 000010      1823    1825    1878    1997    5138E
FDN.AMR 000003      1779    1825    5132E
FDN.AMW 000005      1956    1983    5134E
FDN.LNK 000000      1983    2002    5123E
FDN.SIZ 000002      1768    1929    5130E
FDN.STA 000001      1657    1713    1768    1779    1788    1810    1898    1929    2002    5125E
FDNCNT  000010      2054    5227E
FDNELEN 000011      1704    2055    5140E   5227
FDNFRE  063000      1665    1699    1703    2012    2015    2051    5152L
FDNFWA  063000      5150E   5230
FDNHEAD 063001      1577    1651    1676    1872    2011    2053    5153L
FF      000014      494E
FT.ABS  000000      875E    895
FT.BAC  000003      878E
FT.DD   000001      281E
FT.OC   000020      285E
FT.OR   000002      282E
```

```
FT.OU    000010    284E
FT.OW    000004    283E
FT.PIC   000001    876E
FT.REL   000002    877E
GDWP     056234   2464     3677    3755L
GDWP.    056242   2519     3688    3756    3760L   5395
GETLAB   047056    904     2124    2206L
I.BRE    000002    982E    1154    1163
I.CONFL  000004    673E     674    4073
I.CONTY  000001    660E     661
I.CONWI  000003    666E     667
I.COP    000000    963      978E
I.CSLMD  000000    649E
I.CUSOR  000002    663E     664    4089
I.LIS    000001    980E    1150    1164    1168
I.MOU    000004    986E    1179
I.VER    000003    984E    1174
IERR1    052215   1961     2939L   3541
IERR2    052222   2942L
IERR3    052227   1784     2944L
IFL      046173   1533     2049L
IFL1     046210   2055L    2060
ILDEHL   060045   4566L    4638
INA      056246   3032     3772L
INTERR   052234   2940     2943    2945    2948L
IOC.CGN  000010    290L
IOC.CSI  000011    291L
IOC.DDA  000002    278L     286     300
IOC.DES  000016    297L
IOC.DEV  000020    298L
IOC.DIL  000021    300E
IOC.DIR  000023    302L    1747    1758
IOC.DRL  000010    294E
IOC.DTA  000014    296L
IOC.FLG  000004    280L     294
IOC.GRT  000005    288L    1755
IOC.LGN  000012    292L
IOC.LNK  000000    277L
IOC.LSI  000013    293L
IOC.SPG  000007    289L
IOC.SQL  000003    286E
IOC.UNI  000022    299L
IOCCTD   000001    306E    1743
IOCELEN  000052    304E
IP.CON   000362    712E
IP.PAD   000360    708E
ISDEHL   060150   4645     4701L
JGL      063203   1140     5282L
LAB.AUX  000117    865E     867
LAB.AXL  000001    867E
LAB.DAT  000000    842E
LAB.DIS  000003    838L
LAB.GRT  000005    839L
LAB.IND  000001    837L
LAB.LAB  000021    861L     862
LAB.LBL  000074    862E
LAB.NOD  000002    844E
LAB.PSS  000016    853L
```

```
LAB.RGT 000012  849L
LAB.SER 000000  836L  913   1537  2137
LAB.SIZ 000014  852L
LAB.SPG 000007  840L
LAB.SPT 000117  866L
LAB.SYS 000001  843E
LAB.VER 000011  847L
LAB.VFL 000020  854L
LAB.VLT 000010  846L
LAB.VPL 000005  858E  858   859
LAB.VPR 000014  851E  856
LABEL   064321  907   913   1537  2131  2137  2161  2212  5520L
LF      000012  481E
LINE    065374  961   1205  2892  3183  3236  3796  3803  5540L
LIST    047106  981   2366L
LIST1   047117  2367  2372L
LIST1.5 047172  2399  2404L
LIST10  050207  2548  2580L
LIST2   047311  2451  2456L
LIST3   047316  2463L 2476  2529
LIST4   047335  2474L 2528
LIST5   047365  2488L 2512
LIST6   050004  2496L
LIST7   050036  2479  2485  2516L 2539
LIST8   050065  2498  2533L
LIST9   050104  2469  2481  2543L
LSN     056300  2640  3056  3182  3796L
LSN1    056303  3797L 3802
LSTA    050224  2372  2373  2449  2546  2598L 2726  2739
LSTB    050225  2373  2537  2554  2600L
LSTC    050226  2375  2561  2601L 2773  2775
LSTD    050230  2405  2408  2412  2416  2602L
LSTE    050260  2426  2568  2603L 2755  3159
LSTF    050262  2421  2534  2571  2604L
LSTG    050263  2448  2605L 2608
LSTG1   050321  2377  2606L
LSTGL   000051  2452  2608E
LSTH    050334  2580  2610L 2614
LSTH1   050340  2558  2611L
LSTH2   050361  2564  2612L
LSTH3   050377  2575  2613L
LSTHL   000056  2579  2614E
M.FOX   000303   756E
M.PAMB  000021   755E
MAD     046224  1487  1575  1585  2075E
MAD0    046240  2089L 2143
MAD2    046253  2095L 2098
MAD3    046307  2114L 2116
MAD4    046317  2118L 2120
MAD4.5  046372  2138  2145L
MAD5    047006  2155L 2158
MEML    065146   897  5496E
MND     047040   911  2165  2182L
MNDA    047051  2083  2182  2187L
MODE    063202   925   957  5281L
MOUNT   043302   987  1481L
MOUNTA  043323  1485  1490L
MWN     056320  1911  3824L
```

```
MWN1     056353   3838L   3846
MWN2     056361   3840    3842L
MWNA     063300   3829    3832    5508L
NAMERR   052175   1726    1804    2926L
NAMTAB   066114   1660    1707    1909    2486    2507    3033    3645    3869    3870    5543L
NAMTLEN  063264    949    1557    1686    2506    2637    3022    3025    3644    3649    3652    3863    3866
                  5313L
NAMTMAX  063266    950    3027    3772    5314L
NAMTPTR  063270   1710    2926    5316L
NL       000012    492E     493    1491    1601    1611    1613    2605    2607    2607    2610    2613    2736
                  2816    2950    2954    2955    2955    2969    4165    4327    5256    5273    5387    5388    5389
                  5390    5390    5391    5394    5404    5469    5469    5475    5478
NUL2     000000    483E
NULL     000200    482E
OBUFLIM  063114   2032    2033    3559    5235L
OBUFPTR  063115   1569    1834    2033    5236L
OCOPY    043347   1532E
OCOPY1   044053   1556    1567L   1587
OCOPY2   044100   1572    1576L
OCOPY3   044131   1582    1586L
OCOPY6   044137   1579    1591L
OCOPYA   044203   1541    1554    1606L
OCOPYC   044204   1535    1591    1607L   1736
OCOPYD   044205   1549    1608L   1609    1908
OCOPYDL  000021   1547    1609E
OCOPYE   044161   1598    1602L
OCOPYF   044226   1573    1610L
OCOPYG   044250   1584    1612L
ONECOPY  000000      2E      61     900    2979    5109    5114    5315    5408
OP.CTL   000360    709E
OP.DIG   000360    710E
OP.SEG   000361    711E
OP2.CTL  000362    713E
OVL.COD  000000    215L
OVL.ENS  000010    220E    3521    3525
OVL.ENT  000004    217L
OVL.FLB  000006    218L    3521
OVL.IN   000001    542E
OVL.NUM  000014    544E
OVL.RES  000002    543E    3523
OVL.SIZ  000002    216L
OVL.UCS  000200    545E
OVL0     000000    226L    3521    3527    5484
OVL1     000001    227L    3527    5488
PATCH    061310   5108L
PEC.CS   000204     57E    1111    1152    2999
PEC.DF   000200     54E    2434    2996    3672
PEC.DNC  000201     55E    2675    2997
PEC.FCI  000210     62E    1562    3003
PEC.IDF  000206     59E    3001    3255
PEC.IUW  000205     58E    3000
PEC.SFI  000207     60E    2896    3002
PEC.TFI  000203     56E    2998    3188
PFI      051141   2535    2713L
PFI1     051162   2717    2720L
PFI19    052007   2848L   2855
PFI2     051220   2734    2739L
PFI20    052014   2716    2723    2853L
```

```
PFI3      051226    2728     2744L
PFI3.5    051271    2765     2767E
PFI4      051353    2806L    2815
PFI5      051364    2807     2813L
PFI5.5    051371    2802     2816L
PFI6      051374    2740     2821L
PFIA      052032    2714     2821    2831    2868L
PFIB      052105    2805     2871L
PFIB1     052110    1142     2872L
PFIC      052115    2713     2759    2877L
PIO.DEV   065321    2660     3325    3358    3366    3579    3654    3944    3947    5529L
PIO.DIR   065324    2491     3097    3329    3379    3407    3584    3656    3720    3829    3833    3951    5532L
PIO.UNI   065323    3358     3363    3944    5530L
PIP       042237     919E
PIP1      042257     929      937L
PIPA      043016     962     1025L
PIPB      042367     977L     978     980     982     984     986
PRS       063272    5347L    5504
PRS1      064324    5348     5350    5402L
PRS2      064327    5353     5385    5404L
PRS3      042200     904L    5398
QUOTE     000047     489E
REN       056374    2016     3863L
RESTART   042237     923E     933    1012    2971    2984    3071    5276
RMEML     066114    5351     5549E
ROMBOOT   030000     393E
RPH       044277    1576     1641E   1840
RPH1      044326    1655     1665L
RPH2      044345    1678L    1684
RPH2.5    045021    1661     1721L
RPH3      045124    1731     1750    1776L
RPH4      045216    1802     1821L
RUBOUT    000177     485E
S.BAUD    040344     519L
S.BDA     041120     617L
S.BOOTF   041034     574L
S.CAADR   040333     677L
S.CACC    041006     558L
S.CCTAB   040335     678L
S.CDB     040343     516L
S.CFWA    040352     526L    1742
S.CODE    041007     559L
S.CONFL   040332     675L
S.CONTY   040327     662L
S.CONWI   040331     668L
S.CSLMD   040326     650L     661     664     667     674    1204
S.CUSOR   040330     665L
S.DATC    040310     631L
S.DATE    040277     630L    2377
S.DCS     041033     572L
S.DDDTA   040366     537L
S.DDGRP   040364     534L
S.DDLDA   040360     532L
S.DDLEN   040362     533L
S.DDOPC   040370     538L
S.DFWA    040354     527L
S.DIREA   041016     566L
S.DLINK   040346     524L
```

```
S.FASER 041013  565L
S.FCI   041021  567L
S.GRT0  024000  389E
S.GRT1  025000  390E
S.GRT2  026000  391E
S.GUP   041027  569L  2417
S.HIMEM 040316  633L
S.INT   040343  403L   512
S.JUMPS 041010  563L
S.MOUNT 041032  571L
S.OFWA  040350  525L  3520
S.OMAX  040324  639L  3530
S.OSN   041004  554L
S.OVLE  041000  551L
S.OVLFL 040371  547L
S.OVLS  040376  550L
S.OVSTK 041035  579L
S.RFWA  040356  528L
S.SCI   041024  568L
S.SCR   041121  618L  3760
S.SDD   041010  564L
S.SOVR  041146  405L   407
S.SSN   041002  553L
S.SYSM  040320  635L  3518
S.TIME  040312  632L
S.UCSF  040372  548L
S.UCSL  040374  549L
S.USRM  040322  637L  3542
S.VAL   040277  402L   628
SBE     057021  3783  3882L
SC.ACE  000350   69E
SC.UART 000372  138E
SDD     057042   937  1542  3901L
SDDA    057061  3903  3906L
SFS     057067  2684  3075  3924L
SFS1    057101  3927  3929L
SLABEL  063321   908  1481  2130  2162  5519L
SND     057104  3066  3943L
ST.CNT  000020  1765  1918  5126E
ST.OPR  000002  1659  1729  1740  1813  1888  2004  5127E
ST.OPW  000001  1900  1902  5128E
STACK   042200  409E   928
STACKL  001032  407E
START   042246   915   928L  5400
SUPRES  063204   966  1133  2581  5283L
SW.ALL  043142  1046  1119L
SW.BRE  043200  1054  1147L
SW.BRE1 043215  1149  1154L
SW.JGL  043163  1074  1139L
SW.LIS  043223  1050  1160L
SW.LIS1 043236  1162  1168L
SW.MOU  043251  1062  1179L
SW.SUP  043155  1070  1132L
SW.SYS  043150  1066  1125L  1143
SW.VER  043244  1058  1174L
SWIT1   043124  1107L  1175  1180
SYDD    040130   399E  2115  2119  2208  2215
SYSCALL 000377   419E   941  1002  1011  1725  1783  1797  1839  1922  1935  1942  1951
```

```
UMI.1B  000100  143E
UMI.1X  000001  152E
UMI.2B  000300  145E
UMI.64X 000003  154E
UMI.HB  000200  144E
UMI.L5  000000  148E
UMI.L6  000004  149E
UMI.L7  000010  150E
UMI.L8  000014  151E
UMI.PA  000020  147E
UMI.PE  000040  146E
UNT.DIS 000006  268L
UNT.FLG 000000  264L
UNT.GRT 000002  266L    2424
UNT.GTS 000004  267L
UNT.SIZ 000010  270E
UNT.SPG 000001  265L    2420
UO.CLK  000001  748E    2089
UO.DDU  000002  747E    2089
UO.HLT  000200  745E    2089
UO.NFR  000100  746E
UR.DLL  000000   76E
UR.DLM  000001   78E
UR.IER  000001   80E
UR.IIR  000002   86E
UR.LCR  000003   90E
UR.LSR  000005  109E
UR.MCR  000004  102E
UR.MSR  000006  118E
UR.RBR  000000   72E
UR.THR  000000   74E
USERFWA 042200  410E    894     896     897
USR     000001  136E
USR.BD  000100  167E
USR.FE  000040  168E
USR.OE  000020  169E
USR.PE  000010  170E
USR.RXR 000002  172E
USR.TXE 000004  171E
USR.TXR 000001  173E
VERS    000040  417E    2908    2908    5349    5388    5388
VERSN   052116  985     2887E
VFL.NSD 000001  855E
VOLFLAG 063112  1536    1570    1580    2146    5232L
VOLSER  063113   914    1538    2136    5233L
WPH     045252  1586    1868E   2017
WPH0    045305  1881    1896L
WPH1    045357  1919    1928L
WPH1.5  046003  1936    1939L
WPH2    046021  1901    1949L
WPH3    046053  1924    1944    1965L
WPH4    046126  1891    1997L
XCHGBC  061100  4635    4639    4647    4649    4921L

13654 BYTES FREE
```