```
000.000                     .PIP.   EQU     0               ASSEMBLE AS PIP
000.001                     ONECOPY EQU     1               DONT ASSEMBLE AS ONECOPY

000.000                             IF      .PIP.
                                    ELSE
                                    TITLE   'ONECOPY - ONE DRIVE COPY UTILITY'
                                    ENDIF


                            ***     PIP - PERIHPERAL INTERCHANGE PROGRAM.
                            *
                            *          J. G. L.,  11/1977 FOR *HEATH* COMPANY
                            *
                            *       COPYRIGHT 1977 BY HEATH COMPANY
                            *
                            *             G. C.,    78/09   Maintenence Release
                            *                       79/04
                            *
                            *                       79/11   50.05.00
                            *                       80      50.06.00
                            *                                /2.0a/ = /80.09.sc/
                            *                                /2.0b/ = /80.10.sc/
                            *

                            ***     USE:
                            *
                            *       DEST=SOURCE1 [,SOURCE2,...,SOURCEN] [/SWITCH1.../SWITCHN]
                            *
                            *       SWITCHES:
                            *
                            *       /ALL[OCATE]
                            *       /R[ENAME]               RENAME
                            *       /DEL[ETE]       DELETE
                            *       /L[IST]         LIST
                            *       /B[RIEF]        BRIEF LIST
                            *       /S[YSTEM]       ENCLUDE SYSTEM FILES
                            *       /V[ERSION]      PIP VERSION NUMBER
                            *       /MOU[NT]        MOUNT DEVICE
                            *       /DIS[MOUNT]     DISMOUNT DEVICE
                            *       /RES[ET]        RESET DEVICE
                            *
                            *       /SU[PRESS]      SUPRESS
                            *       /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
     000.001             61          IF      ONECOPY
                         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               UC.SB   EQU     01000000B       SET BREAK
000.200               UC.DLA  EQU     10000000B       DIVISOR LATCH ACCESS

000.004               UR.MCR  EQU     4              MODEM CONTROL REGISTER
000.001               UC.DTR  EQU     00000001B       DATA TERMINAL READY
000.002               UC.RTS  EQU     00000010B       REQUEST TO SEND
000.004               UC.OU1  EQU     00000100B       OUT 1
000.010               UC.OU2  EQU     00001000B       OUT 2
000.020               UC.LOO  EQU     00010000B       LOOP

000.005               UR.LSR  EQU     5              LINE STATUS REGISTER
000.001               UC.DR   EQU     00000001B       DATA READY
000.002               UC.OR   EQU     00000010B       OVERRUN
000.004               UC.PE   EQU     00000100B       PARITY ERROR
000.010               UC.FE   EQU     00001000B       FRAMING ERROR
000.020               UC.BI   EQU     00010000B       BREAK INTERRUPT
000.040               UC.THE  EQU     00100000B       TRANSMITTER HOLDING REGISTER EMPTY
000.100               UC.TSE  EQU     01000000B       TRANSMITTER SHIFT REGISTER EMPTY

000.006               UR.MSR  EQU     6              MODEM STATUS REGISTER
000.001               UC.DCS  EQU     00000001B       DELTA CLEAR TO SEND
000.002               UC.DDR  EQU     00000010B       DELTA DATA SET READY
000.004               UC.TER  EQU     00000100B       TRAILING EDGE OF RING
000.010               UC.DRL  EQU     00001000B       DELTA RECEIVE LINE SIGNAL DETECT
000.020               UC.CTS  EQU     00010000B       CLEAR TO SEND
000.040               UC.DSR  EQU     00100000B       DATA SET READY
000.100               UC.RI   EQU     01000000B       RING INDICATOR
000.200               UC.RLS  EQU     10000000B       RECEIVED LINE SIGNAL DETECT
000.000                       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     CB.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 **       Configuration Flags                                       /80.07.gc/
                         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.VPL EQU      *-LAB.VPR     Length 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     Length 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  342 021         897          DW      MEML-USERFWA    SIZE
      042.176  034 064         898          DW      ENTRY           ENTRY
                               899
      000.001                  900          IF      ONECOPY
                               901
                               902  *       Since this code overlays PRS, it is included here          /2.0a/
                               903
                               904  PRS3    CALL    GETLAB          Get Label
                               905          RC
                               906          LXI     B,256
                               907          LXI     D,LABEL
                               908          LXI     H,SLABEL
                               909          CALL    $MOVE           Save Current Label
                               910
                               911          CALL    MND             Mount New Disk
                               912          JC      ERROR
                               913          LDA     LABEL+LAB.SER
                               914          STA     VOLSER          Set Current Volume Number
                               915          JMP     START
                               916
                               917          ENDIF
                               918
      042.200                  919  PIP     EQU     *
                               920
                               921  *       COMMAND INTERPRETATION COMES HERE
                               922
      042.200                  923  RESTART EQU     *
                               924
      042.200  072 346 063     925          LDA     MODE
      042.203  247             926          ANA     A
      042.204  302 352 042     927          JNZ     EXIT            ENTERED WITH COMMAND, WILL NOW EXIT
      042.207  061 200 042     928  START   LXI     SP,STACK        CLEAN STACK
      042.212  315 220 042     929          CALL    PIP1            EXECUTE COMMAND
                               930
                               931  *       COMMANDS EXIT HERE IF NO ERRORS FOUND
                               932
      042.215  303 200 042     933          JMP     RESTART
                               934
                               935  *       GET READY TO PROCESS COMMAND
                               936
      042.220  315 343 056     937  PIP1    CALL    SDD             SET DEFAULT DEFAULT
                               938
                               939  *       CLEAR CHANNELS AND FILE BUFFER
                               940
      042.223  377 056         941          DB      SYSCALL,.CLEARA CLEAR CHANNELS
      042.225  257             942          XRA     A
      042.226  062 376 063     943          STA     DESTFB+FB.FLG   FLAG FILE NOT OPEN
                               944
                               945  *       CLEAR DYNAMIC BUFFERS
                               946
      042.231  041 000 000     947          LXI     H,0
      042.234  042 373 063     948          SHLD    BUFSIZ          EMPTY BUFFER
      042.237  042 030 064     949          SHLD    NAMTLEN         CLEAR NAMTAB
```

```
042.242  042 032 064   950          SHLD    NAMTMAX          CLEAR NAMTAB AREA
042.245  041 256 067   951          LXI     H,BUFF
042.250  042 371 063   952          SHLD    BUFPTR           SET BUFFER AGAINST END OF NAMTAB
                       953
                       954   *      INPUT COMMAND LINE
                       955
042.253  315 073 057   956          CALL    $CCO             CLEAR CONTROL-O
042.256  072 346 063   957          LDA     MODE
042.261  247           958          ANA     A
042.262  314 320 043   959          CZ      ACL              ACCEPT COMMAND LINE (UNLESS WAS PASSED ONE BY CALLER)
042.265  332 352 042   960          JC      EXIT             EOF
042.270  041 136 067   961          LXI     H,LINE           (HL) = COMMAND ADDRESS
042.273  021 367 042   962          LXI     D,PIPA           (DE) = SWITCH LIST
000.000                963          ERRNZ   I.COP
042.276  257           964          XRA     A                (A) = $I.COP
042.277  062 345 063   965          STA     COMAND           ASSUME COPY COMMAND
042.302  062 350 063   966          STA     SUPRES           CLEAR /SUP FLAG
042.305  062 344 063   967          STA     ALLOCA           Clear /ALL flag                  /80.06.sc/
042.310  074           968          INR     A                FLAG NO /S FLAG
042.311  062 351 063   969          STA     SYSTEM           CLEAR /S FLAG
042.314  315 012 061   970          CALL    $DRS             DETECT AND REMOVE SWITCHES
042.317  332 325 051   971          JC      ERROR            ERROR
042.322  072 345 063   972          LDA     COMAND
042.325  315 061 031   973          CALL    $TJMP            PROCESS COMMAND
```

```
                             975  **      COMMAND LIST
                             976
042.330                      977  PIPB    DS      0               COMMAND PROCESSOR TABLE
000.000                      978  I.COP   EQU     *-PIPB/2                COMMAND INDEX
042.330  343 043             979          DW      COPY
000.001                      980  I.LIS   EQU     *-PIPB/2                COMMAND INDEX
042.332  350 045             981          DW      LIST
000.002                      982  I.BRE   EQU     *-PIPB/2                COMMAND INDEX
042.334  356 045             983          DW      BRIEF           /BR
000.003                      984  I.VER   EQU     *-PIPB/2                COMMAND INDEX
042.336  033 051             985          DW      VERSN           /V
000.004                      986  I.MOU   EQU     *-PIPB/2        /MOU,/M
042.340  015 045             987          DW      MOUNT
000.000                      988          IF      .PIP.
000.005                      989  I.DEL   EQU     *-PIPB/2
042.342  124 045             990          DW      DELETE          /DEL
000.006                      991  I.REN   EQU     *-PIPB/2
042.344  203 045             992          DW      RENAME          /RE
000.007                      993  I.DIS   EQU     *-PIPB/2
042.346  023 045             994          DW      DISMOU          /DIS
000.010                      995  I.RES   EQU     *-PIPB/2
042.350  031 045             996          DW      RESET           /RES
                             997          ENDIF
                             998
                             999  *       CTL-D HIT
                            1000
042.352  257                1001  EXIT    XRA     A
042.353  377 000            1002          DB      SYSCALL,.EXIT   EXIT


                            1004  **      CCHIT - CTL-C HIT
                            1005  *
                            1006  *       ENTRY   FROM SYSTEM
                            1007
                            1008
042.355  315 136 031        1009  CCHIT   CALL    $TYPTX
042.360  136 303            1010          DB      '^','C'+200Q
042.362  377 007            1011          DB      SYSCALL,.CLRCO  CLEAR CONSOLE TYPEAHEAD
042.364  303 200 042        1012          JMP     RESTART         GET NEW COMMAND
```

```
                       ***   SWITCH PROCESSING TABLES AND ROUTINES.
                       *
                       *     COMMAND SWITCHES ARE PROCESSED VIA THE ROUTINE $DRS, 'DECODE AND
                       *     REMOVE SWITCHES'. $DRS IS SUPPLIED WITH A SWITCH DESCRIPTION
                       *     TABLE, WHICH CONTAINS THE ADDRESSES OF ROUTINES
                       *     WHICH ARE ENVOKED WHEN THE SWITCHES ARE ENCOUNTERED.


                       **    SWITCH TABLE

042.367                PIPA  DS      0                FWA SWITCH TABLE
000.000                      IF      .PIP.
042.367  104 105 114         DB      'DEL'            /DELETE
042.372  305 324 305         DB      'E'+200Q,'T'+200Q,'E'+200Q,200Q
042.376  142 043             DW      SW.DEL            PROCESSING ROUTINES

043.000  122                 DB      'R'              /RENAME
043.001  305 316 301         DB      'E'+200Q,'N'+200Q,'A'+200Q,'M'+200Q,'E'+200Q,200Q
043.007  147 043             DW      SW.REN            PROCESS RENAME

043.011  104 111 123         DB      'DIS'            /DISMOUNT
043.014  315 317 325         DB      'M'+200Q,'O'+200Q,'U'+200Q,'N'+200Q,'T'+200Q,200Q
043.022  154 043             DW      SW.DIS

043.024  122 105 123         DB      'RES'            /RESET
043.027  305 324 200         DB      'E'+200Q,'T'+200Q,200Q
043.032  161 043             DW      SW.RES
                             ENDIF

043.034  101 114 114         DB      'ALL'            /ALLOCATE                         /80.06.sc/
043.037  317 303 301         DB      'O'+200Q,'C'+200Q,'A'+200Q,'T'+200Q,'E'+200Q,200Q /.06.sc/
043.045  204 043             DW      SW.ALL                                             /80.06.sc/

043.047  114                 DB      'L'              /LIST
043.050  311 323 324         DB      'I'+200Q,'S'+200Q,'T'+200Q,200Q
043.054  265 043             DW      SW.LIS            PROCESS LIST

043.056  102                 DB      'B'              /BRIEF
043.057  322 311 305         DB      'R'+200Q,'I'+200Q,'E'+200Q,'F'+200Q,200Q
043.064  242 043             DW      SW.BRE            PROCESS BRIEF

043.066  126                 DB      'V'              /VERSION
043.067  305 322 323         DB      'E'+200Q,'R'+200Q,'S'+200Q,'I'+200Q,'O'+200Q,'N'+200Q,200Q
043.076  306 043             DW      SW.VER           PROCESS VERSION

043.100  115 117 125         DB      'MOU'            /MOUNT
043.103  316 324 200         DB      'N'+200Q,'T'+200Q,200Q
043.106  313 043             DW      SW.MOU

043.110  123                 DB      'S'              /SYSTEM
043.111  331 323 324         DB      'Y'+200Q,'S'+200Q,'T'+200Q,'E'+200Q,'M'+200Q,200Q
043.117  212 043             DW      SW.SYS            PROCESS SYSTEM

043.121  123 125             DB      'SU'             /SUPRESS
043.123  320 322 305         DB      'P'+200Q,'R'+200Q,'E'+200Q,'S'+200Q,'S'+200Q,200Q
043.131  217 043             DW      SW.SUP
```

```
                         1071
  043.133  112 107 114   1072          DB      'JGL'         /JGL INTERNAL SWITCH
  043.136  200           1073          DB      200Q
  043.137  225 043       1074          DW      SW.JGL
                         1075
  043.141  000           1076          DB      0             END OF TABLE
```

```
000.000                  1078          IF      .PIP.

                         1080  **      SW.DEL - /DELETE SWITCH DETECTED.
                         1081
043.142  076 005         1082  SW.DEL  MVI     A,I.DEL
043.144  303 166 043     1083          JMP     SWIT1            IS MAJOR FUNCTION

                         1085  **      SW.REN - /RENAME SWITCH DETECTED.
                         1086
043.147  076 006         1087  SW.REN  MVI     A,I.REN
043.151  303 166 043     1088          JMP     SWIT1            IS MAJOR FUNCTION

                         1090  **      SW.DIS  -  /DISMOUNT SWITCH DETECTED
                         1091
043.154  076 007         1092  SW.DIS  MVI     A,I.DIS
043.156  303 166 043     1093          JMP     SWIT1            IS MAJOR FUNCTION

                         1095  **      SW.RES  -  /RESET SWITCH DETECTED.
                         1096
043.161  076 010         1097  SW.RES  MVI     A,I.RES
043.163  303 166 043     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
043.166  001 345 063     1107  SWIT1   LXI     B,COMAND
043.171  365             1108          PUSH    PSW              SAVE COMMAND
043.172  012             1109          LDAX    B                (A) = PREVIOUS COMMAND
043.173  247             1110          ANA     A
043.174  076 204         1111          MVI     A,PEC.CS         CONTRADICTORY SWITCHES
043.176  302 325 051     1112          JNZ     ERROR            IF SO
043.201  361             1113          POP     PSW              (A) = NEW CODE
043.202  002             1114          STAX    B                STORE IT
043.203  311             1115          RET
```

```
                         1117  **      SW.ALL  - /ALLOCATE Switch Detected                       /80.06.sc/
                         1118
043.204  076 001         1119  SW.ALL  MVI     A,1
043.206  062 344 063     1120          STA     ALLOCA
043.211  311             1121          RET


                         1123  **      SW.SYS - /SYSTEM SWITCH DETECTED.
                         1124
043.212  257             1125  SW.SYS  XRA     A                SET /S FLAG
043.213  062 351 063     1126          STA     SYSTEM
043.216  311             1127          RET


                         1129  **      SW.SUP - /SUPPRESS SWITCH.
                         1130
                         1131
043.217  076 001         1132  SW.SUP  MVI     A,1
043.221  062 350 063     1133          STA     SUPRES
043.224  311             1134          RET


                         1136  **      SW.JGL - /JGL SYSTEM SWITCH.
                         1137
                         1138
043.225  076 001         1139  SW.JGL  MVI     A,1
043.227  062 347 063     1140          STA     JGL
043.232  076 103         1141          MVI     A,'C'
043.234  062 025 051     1142          STA     PFIB1            SET 'C' CHARACTER FOR FLAGS DISPLAY
043.237  303 212 043     1143          JMP     SW.SYS


                         1145  **      SW.BRE - /BRIEF SWITCH DETECTED.
                         1146
043.242  072 345 063     1147  SW.BRE  LDA     COMAND           ALLOW TO SUPERCEDE /LIST
043.245  247             1148          ANA     A
043.246  312 257 043     1149          JZ      SW.BRE1          NO OTHER COMMAND
000.000                  1150          ERRNZ   I.LIS-1
043.251  075             1151          DCR     A
043.252  076 204         1152          MVI     A,PEC.CS         ASSUME CONTRADICTORY SWITCHES
043.254  302 325 051     1153          JNZ     ERROR
043.257  076 002         1154  SW.BRE1 MVI     A,I.BRE          IS /BREIF
043.261  062 345 063     1155          STA     COMAND
043.264  311             1156          RET
```

```
                             1158  **       SW.LST - /LIST SWITCH DETECTED.
                             1159
     043.265  072 345 063  1160  SW.LIS  LDA     COMAND
     043.270  247          1161          ANA     A
     043.271  312 300 043  1162          JZ      SW.LIS1          NO FUNCTION
     000.000               1163          ERRNZ   I.BRE-2
     000.000               1164          ERRNZ   I.LIS-1
     043.274  326 003      1165          SUI     3
     043.276  077          1166          CMC
     043.277  320          1167          RNC                      ALREADY HAVE ONE SPECIFIED, I.BRE OVERRULES
     043.300  076 001      1168  SW.LIS1 MVI     A,I.LIS          /LIST
     043.302  062 345 063  1169          STA     COMAND
     043.305  311          1170          RET


                             1172  **       SW.VER  - /VERSION SWITCH DETECTED
                             1173
     043.306  076 003      1174  SW.VER  MVI     A,I.VER
     043.310  303 166 043  1175          JMP     SWIT1


                             1177  **       SW.MOU  - /MOUNT SWITCH DETECTED
                             1178
     043.313  076 004      1179  SW.MOU  MVI     A,I.MOU
     043.315  303 166 043  1180          JMP     SWIT1
```

```
                      ***     ACL - ACCEPT COMMAND LINE.
                      *
                      *       ACL PROMPTS FOR AND READS A COMMAND LINE FROM
                      *       THE CONSOLE.
                      *
                      *       ENTRY   NONE
                      *       EXIT    'C' CLEAR, GOT LINE
                      *                'LINE' = COMMAND LINE
                      *               'C' SET IF EOF
                      *       USES    ALL


043.320  315 110 057  ACL     CALL    $GNL            GUARANTEE NEW LINE
043.323  315 136 031          CALL    $TYPTX
000.000                       IF      .PIP.
043.326  072 120 272          DB      ':P',':'+200Q
                              ELSE    ONECOPY
                              DB      ':OC',':'+200Q
                              ENDIF
043.331  257                  XRA     A
043.332  062 326 040          STA     S.CSLMD         CLEAR SPECIAL MODES
043.335  041 136 067          LXI     H,LINE
043.340  303 155 057          JMP     $RTL.           READ UPPER CASE LINE AND EXIT
```

```
000.000                        IF      .PIP.           PIP USES 'COPY'
                        ***    COPY - PROCESS COPY COMMAND.
                        *
                        *      SYNTAX:
                        *
                        *      DEST=SOURCE1,...,SOURCEN
                        *
                        *      D'DEST' IS THE DESTINATION FILE DESIGNATOR. IF NULL
                        *      (IN WHICH CASE THE '=' MAY BE OMITTED) IT DEFAULTS  TO
                        *                KB:PIPDEST.JGL
                        *
                        *      THE 'SOURCE' FIELDS ARE THE SOURCE FILE DESIGNATORS. WILDCARDS
                        *      MAY BE USED FOR FILE NAME AND EXTENSION.
                        *      IF NO WILDCARDS ARE USED IN THE DESTINATION, MULTIPLE SOURCE FILES
                        *      ARE CONCATINATED TOGETHER.
                        *
                        *      IF WILDCARDS ARE PRESENT IN THE DESTINATION FILE DESCRIPTION,
                        *      THE SOURCE FILES ARE COPIED TO INDIVIDUAL OUTPUT FILES. THE
                        *      NAMES OF THE OUTPUT FILES ARE CREATED BY FILLING
                        *      THE 'WILD' SPOTS IN THE DESTINATION NAME WITH THE CORRESPONDING
                        *      CHARACTERS IN THE SOURCE NAME.


043.343                 COPY   EQU     *
043.343  257                   XRA     A
043.344  062 373 044           STA     COPYC           CLEAR FILE COUNT
043.347  315 324 053           CALL    DDF             DECODE DESTINATION FILE
043.352  332 325 051           JC      ERROR           ERROR
043.355  062 372 044           STA     COPYA           SAVE DESTIONATION TYPE
043.360  315 343 056           CALL    SDD             RESET DEFAULT DEFAULTS
043.363  257                   XRA     A               ALLOW *.*
043.364  315 042 053           CALL    BSL             BUILD SOURCE FILE LIST
043.367  332 325 051           JC      ERROR
043.372  315 345 060           CALL    $MOVEL
043.375  021 000               DW      COPYDL
043.377  007 064               DW      DESTFB+FB.NAM
044.001  374 044               DW      COPYD           SAVE WILDCARD DESTINATION

                        *      HAVE DESTINATION AND SOURCE FILE NAMES. DO THE COPYING.
                        *
                        *      IF NO DESTINATION WILD CARDS, THUS COPIING TO A SINGLE OUTPUT
                        *      FILE, OPEN THAT FILE NOW.

044.003  072 372 044           LDA     COPYA
044.006  247                   ANA     A
044.007  312 027 044           JZ      COPY1           IS WILDCARDED
044.012  041 007 064           LXI     H,DESTFB+FB.NAM
044.015  076 001               MVI     A,CN.DES        (A) = DESTINATION CHANNEL
044.017  377 043               DB      SYSCALL,.OPENW          OPEN IT
044.021  041 375 063           LXI     H,DESTFB
044.024  332 262 063           JC      $FERROR         IF ERROR

                        *      OPEN NEXT SOURCE FILE

044.027  052 030 064    COPY1  LHLD    NAMTLEN
044.032  174                   MOV     A,H
```

```
044.033    265                           ORA     L
044.034    312 241 044                   JZ      COPY5             NO MORE INPUT FILES
044.037    041 373 044                   LXI     H,COPYC
044.042    064                           INR     M                 COUNT FILE
044.043    041 256 067                   LXI     H,NAMTAB          (HL) = NAME ADDRESS
044.046    076 000                       MVI     A,CN.SOU          SOURCE CHANNEL
044.050    377 042                       DB      SYSCALL,.OPENR    OPEN FOR READ
044.052    332 106 051                   JC      NAMERR            IF ERROR

                                *        OPEN DESTINATION FILE IFF WILDCARDS

044.055    072 372 044                   LDA     COPYA
044.060    247                           ANA     A
044.061    302 114 044                   JNZ     COPY2             NOT WILDCARDS
044.064    001 374 044                   LXI     B,COPYD           (BC) = WILDCARD PATTERN ADDRESS
044.067    021 256 067                   LXI     D,NAMTAB          (DE) = SOURCE NAME
044.072    041 007 064                   LXI     H,DESTFB+FB.NAM   (HL) = RESULT AREA
044.075    345                           PUSH    H                 SAVE POINTER TO RESULT AREA
044.076    315 221 056                   CALL    MWN               MERGE WILDCARD NAME
044.101    341                           POP     H                 (HL) = #DESTFB+FB.NAM
044.102    076 001                       MVI     A,CN.DES
044.104    377 043                       DB      SYSCALL,.OPENW
044.106    041 375 063                   LXI     H,DESTFB
044.111    332 262 063                   JC      $FERROR           CANT GET FILE OPEN

                                *        INPUT AND OUTPUT FILES OPEN, COPY

044.114    315 026 055          COPY2    CALL    EBM               EXPAND BUFFER TO MAX SIZE
044.117    052 373 063          COPY3    LHLD    BUFSIZ
044.122    104                           MOV     B,H
044.123    115                           MOV     C,L               (BC) = LENGTH OF BUFFER
044.124    052 371 063                   LHLD    BUFPTR
044.127    353                           XCHG                      (DE) = BUFFER FWA
044.130    076 000                       MVI     A,CN.SOU
044.132    325                           PUSH    D
044.133    377 004                       DB      SYSCALL,.READ
044.135    321                           POP     D                 (DE) = BUFFER FWA
044.136    365                           PUSH    PSW
044.137    322 153 044                   JNC     COPY4             GOT IT ALL
044.142    376 001                       CPI     EC.EOF
044.144    312 153 044                   JE      COPY4             IS EOF
044.147    361                           POP     PSW               RESTORE ERROR CODE
044.150    303 106 051                   JMP     NAMERR

044.153    072 374 063          COPY4    LDA     BUFSIZ+1          (A) = # OF SECTORS IN BUFFER
044.156    220                           SUB     B
044.157    107                           MOV     B,A               (B) = SECTORS READ
044.160    016 000                       MVI     C,0
044.162    076 001                       MVI     A,CN.DES
044.164    377 005                       DB      SYSCALL,.WRITE    WRITE IT OUT
044.166    041 375 063                   LXI     H,DESTFB
044.171    332 262 063                   JC      $FERROR           ERROR ON WRITE
044.174    361                           POP     PSW               (PSW) = STATUS FROM READ
044.175    322 117 044                   JNC     COPY3             NOT EOF
044.200    315 322 056                   CALL    SBE               SHRINK BUFFER TO MINIMUM SIZE
044.203    076 000                       MVI     A,CN.SOU
```

```
044.205  377 046                DB      SYSCALL,.CLOSE  CLOSE SOURCE
044.207  332 106 051            JC      NAMERR          ERROR ON CLOSE
044.212  315 275 056            CALL    REN             REMOVE ENTRY FROM NAMTAB

                        *       IF DOING INDIVIDUAL FILE COPIES, CLOSE OUTPUT FILE.

044.215  072 372 044            LDA     COPYA
044.220  247                    ANA     A
044.221  302 027 044            JNZ     COPY1           CONCATINATING
044.224  076 001                MVI     A,CN.DES
044.226  377 046                DB      SYSCALL,.CLOSE  CLOSE DESTINATION
044.230  041 375 063            LXI     H,DESTFB
044.233  332 262 063            JC      $FERROR         ERROR ON CLOSE
044.236  303 027 044            JMP     COPY1           GET NEXT FILE

                        **      ALL COPIES COMPLETE. CLOSE FILES AND CLEAN UP

044.241  072 373 044    COPY5   LDA     COPYC
044.244  247                    ANA     A
044.245  302 301 044            JNZ     COPY6

                        *       NO FILES COPIED

044.250  315 136 031            CALL    $TYPTX
044.253  007 116 157            DB      BELL,'No Files Copied',ENL
044.274  076 001                MVI     A,CN.DES
044.276  377 055                DB      SYSCALL,.CLEAR  CLEAR CHANNEL
044.300  311                    RET

044.301  006 000        COPY6   MVI     B,0             (BC) = COUNT OF FILES COPIED
044.303  117                    MOV     C,A
044.304  072 372 044            LDA     COPYA
044.307  247                    ANA     A
044.310  312 327 044            JZ      COPY7           WILDCARDED
044.313  305                    PUSH    B               SAVE COUNT
044.314  076 001                MVI     A,CN.DES
044.316  377 046                DB      SYSCALL,.CLOSE  CLOSE DESTINATION
044.320  301                    POP     B               (BC) = FILES COPIED COUNT
044.321  041 375 063            LXI     H,DESTFB
044.324  332 262 063            JC      $FERROR         ERROR ON CLOSE

                        *       TYPE FILE COUNT

044.327  072 350 063    COPY7   LDA     SUPRES
044.332  247                    ANA     A
044.333  300                    RNZ                     SUPPRESS TRAIL MESSAGE
044.334  076 003                MVI     A,3
044.336  041 350 044            LXI     H,COPYE
044.341  315 272 060            CALL    $UDDN           UNPACK COUNT INTO MESSAGE
044.344  315 136 031            CALL    $TYPTX
044.347  012                    DB      NL
044.350  130 130 130    COPYE   DB      'XXX'
044.353  040 106 151            DB      ' Files Copied',ENL
044.371  311                    RET

044.372  000            COPYA   DB      0               DESTINATION FILE WILDCARD FLAG (=0 IF WC)
```

```
044.373  000          1377  COPYC   DB      0               FILES COPIED COUNT
044.374               1378  COPYD   DS      FB.NAML         HOLD AREA FOR WILDCARD DESTINATION
000.021               1379  COPYDL  EQU     *-COPYD
```

```
                              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:/MOU[NT]
                              1388  *
                              1389
      045.015                 1390  MOUNT   EQU     *
      045.015  076 200        1391          MVI     A,.MOUNT
      045.017  315 037 045    1392          CALL    MDR.                MOUNT/DISMOUNT/RESET
      045.022  311            1393          RET
```

```
                          ***     DISMOU  -  DISMOUNT CURRENT DISK
                          *
                          *       DISMOU DISMOUNTS THE CURRENT DISK ON THE SPECIFIED UNIT OF THE
                          *       SELECTED DEVICE.
                          *
                          *       DEV:/DIS[MOUNT]
                          *

045.023                   DISMOU  EQU     *
045.023  076 201                  MVI     A,.DMOUN
045.025  315 037 045              CALL    MDR.            MOUNT/DISMOUNT/RESET
045.030  311                      RET
```

```
                        ***     RESET   -  RESET THE CURRENT DISK
                        *
                        *       RESET RESETS THE SPECIFIED UNIT OF THE SELECTED DEVICE BY ISSUING
                        *       THE HDOS RESET CALL, WHICH IN TURN ISSUES A DISMOUNT AND MOUNT
                        *       ASKING THE USER TO OPEN THE DRIVE IN BETWEEN THE TWO.
                        *
                        *       DEV:/RES[ET]
                        *

045.031                 RESET   EQU     *
045.031  076 204                MVI     A,.RESET
045.033  315 037 045            CALL    MDR.             MOUNT/DISMOUNT/RESET
045.036  311                    RET


                        **      MDR.    -  MOUNT/DISMOUNT/RESET
                        *
                        *       MDR. PERFORMS THE SIMILAR FUNCTIONS OF MOUNT, DISMOUNT, AND RESET.
                        *
                        *
                        *       ENTRY   (A)     =  SYSCALL CODE FOR OPERATION TO BE PERFORMED
                        *
                        *       EXIT    IF  NO ERROR
                        *                   TO CALLER
                        *                 ELSE
                        *                   TO ERROR
                        *
                        *       USES    ALL
                        *

045.037  062 070 045    MDR.    STA     MDRA             STORE SYSCALL VALUE
045.042  315 271 053            CALL    CTS              CHECK FOR TARGET FILE SPECIFICATION
045.045  067                    STC
045.046  302 325 051            JNZ     ERROR            THERE WAS A TARGET FILE
045.051  041 136 067            LXI     H,LINE
045.054  315 211 061            CALL    $DTB             DELETE TRAILING BLANKS
045.057  376 001                CPI     1                (A) = LINE LENGTH INCLUDING <00> BYTE
045.061  076 200                MVI     A,PEC.DF         DEVICE FORMAT ERROR
045.063  312 325 051            JZ      ERROR            NULL DEVICE IS ILLEGAL, ONLY BYTE IS NULL
045.066  345            MDR1    PUSH    H                SAVE SPEC. ADDRESS FOR RETRY
045.067  377 000                DB      SYSCALL,0
045.070                 MDRA    EQU     *-1              SYSCALL VALUE
045.071  341                    POP     H
045.072  320                    RNC                      NO ERROR
045.073  345                    PUSH    H                SAVE SPEC. ADDRESS
045.074  376 044                CPI     EC.NPM           NO PROVISIONS MADE FOR REMOUNT
045.076  067                    STC
045.077  302 325 051            JNZ     ERROR            ALL ERRORS BUT 'EC.NPM' CONSIDERED FATAL
045.102  076 000                MVI     A,OVL0
045.104  377 010                DB      SYSCALL,.LOADO   LOAD *HDOSOVL0.SYS*
045.106  332 325 051            JC      ERROR
045.111  076 001                MVI     A,OVL1
045.113  377 010                DB      SYSCALL,.LOADO   LOAD *HDOSOVL1.SYS*
045.115  332 325 051            JC      ERROR            SYSCALL ERROR
```

```
045.120  341                    POP     H               RESTORE SPEC. ADDRESS
045.121  303 066 045            JMP     MDR1            TRY AGAIN
                                ELSE
                                STL     'MOUNT - MOUNT A DIFFERENT DISK'
                                EJECT
                        MOUNT   SPACE   4,10
                        ***     MOUNT - MOUNT A DIFFERENT DISK.
                        *
                        *       MOUNT CAUSES A NEW DISK TO BE MOUNTED.
                        *
                        *       INSERT THE DISK IN SY0, THEN TYPE
                        *
                        *       /MOUNT



                        MOUNT   LXI     H,SLABEL                                        /2.0a/
                                MVI     B,0             Count of 256                    /2.0a/
                                CALL    $ZERO           Zero the old label              /2.0a/

                                LXI     D,MOUNTA
                                MVI     B,377Q          OFF PERIODS
                                CALL    MAD             MOUNT ALTERNATE DISK
                                RET

                        MOUNTA  DB      244Q,306Q,307Q
                                DB      NL,'Insert New Disk',':'+200Q
                                STL     'ONECOPY - COPY FILES BETWEEN VOLUMES.'
                                EJECT
                        ONECOPY SPACE   4,10
                        ***     ONECOPY - COPY FILES BETWEEN TWO VOLUMES, WITH ONLY ONE
                        *       DRIVE.
                        *
                        *       (AND FOR MY NEXT TRICK...)
                        *
                        *       OPECOPY COPIES FILES BETWEEN TWO VOLUMES BY ALTERNATING BETWEEN
                        *       TWO PHASES, THE READ PHASE AND THE WRITE PHASE.
                        *
                        *       READ PHASE:
                        *
                        *       DURING THE READ PHASE, THE SOURCE DISK IS MOUNTED. SOURCE FILES ARE
                        *       OPENED IN THE ORDER OF THEIR APPEARANCE. FOR EACH OPENED
                        *       FILE, A 'FILE DESCRIPTOR NODE' *FDN* IS ADDED  TO  THE ACTIVE
                        *       CHAIN. THEN, AS MUCH AS THE FILE AS POSSIBLE IS READ INTO MEMORY.
                        *
                        *       THE PROCESS CONTINUES UNTIL
                        *               1) THERE IS NO MORE FREE RAM
                        *               2) OR, THERE ARE NO MORE FILE DESCRIPTOR NODES IN THE FREE CHAIN
                        *               3) OR, THERE ARE NO MORE FILES IN NAMTAB (INPUT FILE LIST)
                        *
                        *
                        *       WRITE PHASE
                        *
                        *       DURING THE WRITE PHASE, THE DESTINATION DISK IS MOUNTED. THE NODES
                        *       ARE TAKEN FROM THE ACTIVE CHAIN, AND PROCESSED. IF THE FILE HAD
                        *       BEEN PARTIALLY WRITTEN THE LAST PASS, IT IS  RE-OPENED AND POSITIONED.
*       IF THERE IS NOT MORE DATA TO READ FOR A PROCESSED
*       NODE, IT IS REMOVED, AND THE CORRESPONDING ENTRY IN NAMTAB IS DELETED.
*
*       WRITE PHASE CONTINUES UNTIL
*
*               1) THERE ARE NO MORE FILE NODES IN THE ACTIVE LIST
*               2) OR, THE FIRST (AND ONLY) ENTRY IN THE LIST HAS NO
*                  MORE DATA IN MEMORY, BUT HAS NOT BEEN COMPLETELY READ.


COPY    EQU     *                       CALLED 'COPY' BY MAINLINE CODE
OCOPY   EQU     *
        CALL    IFL                     INITIALIZE  FDN LISTS
        XRA     A
        STA     OCOPYC                  CLEAR FILE COUNT
        STA     VOLFLAG                 FLAG SOURCE  VOLUME MOUNTED
        LDA     LABEL+LAB.SER           A  = Volume Label                       /2.0a/
        STA     VOLSER                  SET VOLUME SERIAL NUMBER
        CALL    DDF                     DECODE DESTINATION FILE
        JC      ERROR                   ERROR
        STA     OCOPYA                  SAVE DESTIONATION TYPE
        CALL    SDD                     RESET DEFAULT DEFAULTS
        XRA     A                       ALLOW *.*
        CALL    BSL                     BUILD SOURCE FILE LIST
        JC      ERROR
        CALL    $MOVEL
        DW      OCOPYDL
        DW      DESTFB+FB.NAM
        DW      OCOPYD                  SAVE WILDCARD DESTINATION
        CALL    EBM                     EXPAND BUFFER TO MAX

*       MAKE SURE HE'S NOT TRYING TO CONCATINATE

        LDA     OCOPYA
        ANA     A
        JZ      OCOPY1                  HAVE WILDCARDS
        LHLD    NAMTLEN                 NO WILDCARDS, ONLY LET HIM SPEFICY ONE SOURCE
        LXI     D,-FB.NAML
        DAD     D
        MOV     A,H
        ORA     L
        MVI     A,PEC.FCI               FILE CONCATINATION IS ILLEGAL
        JNZ     ERROR

*       START READ PHASE

OCOPY1  LDA     BUFPTR+1                (A) = BUFFER FWA/256
        INR     A                       ROUND UP TO NEXT PAGE
        STA     OBUFPTR                 SET SECTOR BUFFER  FWA/256
        LDA     VOLFLAG
        ANA     A
        JZ      OCOPY2                  SOURCE IS MOUNTED
        LXI     D,OCOPYF
        MOV     B,A                     (B) = 377Q = PERIODS MASK
        CALL    MAD                     MOUNT ALTERNATE DISK
OCOPY2  CALL    RPH                     READ PHASE
```

```
          LDA     FDNHEAD
          ANA     A
          JZ      OCOPY6             NO FILES ARE READ, ERGO NONE ARE LEFT
          LDA     VOLFLAG
          ANA     A
          JNZ     OCOPY3
          MVI     B,177Q             (B) = PERIODS MASK
          LXI     D,OCOPYG
          CALL    MAD                MOUNT ALTERNATE DISK
OCOPY3    CALL    WPH                WRITE PHASE
          JMP     OCOPY1

*         ALL DONE, FINISH MESSAGE

OCOPY6    LDA     OCOPYC             (A) = FILE COUNT
          MVI     B,0                (BC) = COUNT OF FILES COPIED
          MOV     C,A

*         TYPE FILE COUNT

          MVI     A,3
          LXI     H,OCOPYE
          CALL    $UDDN              UNPACK COUNT INTO MESSAGE
          CALL    $TYPTX
          DB      NL                 for aesthetics                          /2.0a/
OCOPYE    DB      'XXX'
          DB      ' Files Copied',ENL
          RET

OCOPYA    DB      0                  DESTINATION FILE WILDCARD FLAG (=0 IF WC)
OCOPYC    DB      0                  FILES COPIED COUNT
OCOPYD    DS      FB.NAML            HOLD AREA FOR WILDCARD DESTINATION
OCOPYDL   EQU     *-OCOPYD
OCOPYF    DB      244Q,306Q,307Q
          DB      NL,'Insert Source',':'+200Q
OCOPYG    DB      102Q,014Q,44Q
          DB      NL,'Insert Destination',':'+200Q
          STL     'ONECOPY SUBROUTINES'
          EJECT
RPH       SPACE   4,10
**        RPH - READ PHASE.
*
*         RPH  HANDLES THE READ PHASE OF THE COPY PROCESS.
*
*         IT IS ENTERED WITH THE NAMTAB AND FDN TABLE SETUP, AND
*         WITH THE SOURCE DISK MOUNTED.
*
*         READ PHASE:
*
*         DURING THE READ PHASE, THE SOURCE DISK IS MOUNTED. SOURCE FILES ARE
*         OPENED IN THE ORDER OF THEIR APPEARANCE. FOR EACH OPENED
*         FILE, A 'FILE DESCRIPTOR NODE' *FDN* IS ADDED  TO  THE ACTIVE
*         CHAIN. THEN, AS MUCH AS THE FILE AS POSSIBLE IS READ INTO MEMORY.
*
*         THE PROCESS CONTINUES UNTIL
*                 1) THERE IS NO MORE FREE RAM
*                   2) OR, THERE ARE NO MORE FILE DESCRIPTOR NODES IN THE FREE CHAIN
*                   3) OR, THERE ARE NO MORE FILES IN NAMTAB (INPUT FILE LIST)
*
*       ENTRY       NONE
*       EXIT        NONE
*       USES        ALL


RPH     EQU         *


*       SEE IF ANY MEMORY TO HAVE

        CALL        CBR                 COMPUTE BUFFER ROOM
        RZ                              NONE

*       SEE IF WE NEED TO READ SOME MORE INTO A PART-COPIED FILE

        LXI         H,FDNHEAD
        MOV         L,M                 (HL) = ADDRESS IF FIRST NODE
        MOV         A,L
        ANA         A
        JZ          RPH1                IS NO FIRST NODE, ERGO NO FILE
        INX         H
        ERRNZ       FDN.STA-1
        MOV         A,M                 (A) = .STA
        ANI         ST.OPR
        LXI         D,NAMTAB
        JNZ         RPH2.5              FILE IS INCOMPLETELY READ

*       SEE IF ANY FREE FILE DESCRIPTOR NODES TO USE

RPH1    LDA         FDNFRE
        ANA         A
        RZ                              NO MORE

*       SEE IF THERE IS A FILE IN NAMTAB WITHOUT AN ENTRY IN FNDLIST.
*       SINCE THE FIRST ENTRY IN FDNLIST CORRESPONDS TO THE FIRST IN
*       NAMTAB, ETC., WE'LL JUST RUN DOWN FDNLIST UNTIL THE END, AND
*       THE NEXT NAMTAB FILE WILL BE THE ONE WE WANT...

        LXI         B,FB.NAML           (BC) = ENTRY SIZE IN NAMTAB
        LXI         D,-FB.NAML                  (DE) = POINTER INTO NAMTAB
        LXI         H,FDNHEAD
        MOV         A,L                 START WITH FDNHEAD
RPH2    MOV         L,A                 FOLLOW LINK
        MOV         A,M                 (A) = NEXT NODE
        XCHG
        DAD         B                   ADVANCE POINTER INTO NAMTAB
        XCHG
        ANA         A
        JNZ         RPH2                LINK SOME MORE
        PUSH        H                   (HL) = ADDRESS OF LAST NODE
        LHLD        NAMTLEN
        CALL        $CDEHL              SEE IF HAVE ACCOUNTED FOR ALL NAMTAB ENTRYS
        POP         H
```

```
                RE                              FILES ALL USED UP

        *       HAVE ROOM FOR DATA, HAVE A NODE FOR THE FILE COUNTS, AND
        *       HAVE A FILE NAME. ALL SET FOR BUSINESS..
        *
        *       (DE) = INDEX INTO NAMTAB FOR FILE
        *       (HL) = NODE ADDRESS OF LAST ENTRY IN LIST
        *
        *       CHAIN THE FIRST FREE NODE ONTO THE END OF THE LIST

                LDA     FDNFRE
                MOV     M,A                     CHAIN TO  NEW END  NODE
                MOV     L,A
                MOV     A,M                     (A) =  NEXT NODE IN FREE CHAIN
                STA     FDNFRE
                MVI     B,FDNELEN
                PUSH    H                       SAVE NODE ADDRESS
                CALL    $ZERO                   ZERO ENTIRE NODE, ENCLUDING CHAIN (AT END,  NOW)
                LXI     B,NAMTAB
                XCHG
                DAD     B                       (HL) = ADDRESS OF NAMTAB ENTRY
                SHLD    NAMTPTR                 POINTER TO CURRENT NAMTAB ENTRY
                XCHG
                POP     H
                ERRNZ   FDN.STA-1
                INX     H                       (HL) = ADDR OF FDN.STA OF NODE

        *       READY TO OPEN FILE
        *
        *       (DE) = NAMTAB ENTRY ADDRESS
        *       (HL) = #FDN.STA OF ENTRY

        RPH2.5  PUSH    H                       SAVE ADDRESS
                XCHG
                XRA     A
                ERRNZ   CN.SOU                  (A) = SOURCE CHANNEL NUMBER
                DB      SYSCALL,.OPENR          OPEN
                JC      NAMERR                  ERROR
                POP     D
                LDAX    D                       (A) = FDN.STA
                ANI     ST.OPR
                PUSH    D                       SAVE ADDRESS
                JNZ     RPH3                    ALREADY OPENED IN PREVIOUS PASSES

        *       FIRST TIME THIS FILE HAS BEEN OPENED. SEE IF CONTIGUOUS

                PUSH    H
                LXI     H,QCOPYC
                INR     M
                POP     H
                LDAX    D
                ORI     ST.OPR                  SET OPEN FOR READ
                STAX    D
                LHLD    S.CFWA                  (HL) = CHANNEL 0 FWA
                ERRNZ   IOCCTD-1                WE NEED TO CHAIN ONE TO GET TO USER #0
                CALL    $HLIHL
```

```
       ERRNZ  CN.SOU        ASSUME WE WANT CHANNEL 0
       CALL   $INDL
       DW     IOC.DIR+DIR.FLG
       MOV    A,E           (A) = DIR.FLG
       ANI    0 DIF.CNT     * * PATCH * *
       JZ     RPH3          NOT CONTIG

*      IS CONTIG, GET FILE SIZE

       CALL   $INDL
       DW     IOC.GRT
       PUSH   D             SAVE GRT ADDRESS
       CALL   $INDL
       DW     IOC.DIR+DIR.FGN (E) = DIR.FGN
       MOV    A,E
       POP    H             (HL) = GRT TABLE ADDRESS
       CALL   CFS.          COMPUTE BLOCK SIZE
       POP    H             (HL) = ADDRESS OF FDN.STA
       PUSH   H
       MOV    A,M           (A) = FDN.STA
       ORI    ST.CNT        FLAG CONTIG
       MOV    M,A
       INX    H
       ERRNZ  FDN.SIZ-FDN.STA-1
       MOV    M,E           SET BLOCK COUNT

*      READY TO READ DATA, POSITION FILE (IN CASE SOME WAS READ IN
*      PREVIOUS PASSES) AND COMPUTE THE MAX POSSIBLE READ COUNT
*
*      ((SP)) = ADDRESS OF FDN.STA FOR NODE

RPH3   POP    H             (HL) = ADDRESS OF FDN.STA
       PUSH   H
       CALL   $INDL
       DW     FDN.AMR-FDN.STA (DE) = AMOUNT READ (IN SECTORS)
       MOV    B,D
       MOV    C,E           (BC) = AMOUNT READ
       MVI    A,CN.SOU
       DB     SYSCALL,.POSIT       POSIT
       JC     IERR3         POSIT BLEW UP
       CALL   CBR           COMPUTE BUFFER ROOM
       XCHG                 (D) = POINTER/256, (E) = LIMIT/256
       POP    H             (HL) = #FDN.STA
       LXI    B,FDN.ADR-FDN.STA
       DAD    B             (HL) = #FDN.ADR
       MOV    M,D           SET ADDRESS/256
       PUSH   H             SAVE #FDN.ADR
       MVI    E,0           (DE) = ADDRESS
       MOV    B,A           (B) = SECTORS OF RAM AVAILABLE
       MOV    C,E           (C) = 0
       PUSH   B             SAVE TRY COUNT
       MVI    A,CN.SOU
       DB     SYSCALL,.READ READ THE STUFF

*      COMPUTE THE AMOUNT READ (IN CASE OF EOF)

```

```
        POP     D               (DE) = TRY COUNT
        JNC     RPH4            GOT  ALL WE TRYED
        CPI     EC.EOF
        JNE     NAMERR          NOT JUST EOF, GOT TROUBLES
        MOV     A,D
        SUB     B               REMOVE AMOUNT WE DIDNT GET
        MOV     D,A
        POP     H               (HL) = #FDN.ADR
        PUSH    H
        LXI     B,FDN.STA-FDN.ADR
        DAD     B
        MOV     A,M             (A) = FDN.STA
        ANI     377Q-ST.OPR     EOF, NOT OPEN FOR READ ANYMORE
        MOV     M,A             POST READ COMPLETE FOR THIS GUY

*       STORE RESULTS OF READ IN NODE
*
*       (D) = SECTORS READ
*       ((SP)) = #FDN.ADR

RPH4    POP     H               (HL) = #FDN.ADR
        INX     H
        ERRNZ   FDN.AIM-FDN.ADR-1       (HL) = ADDRESS IF AMOUNT IN MEMORY BYTE
        MOV     M,D             STORE SECTORS IN MEMORY COUNT
        LXI     B,FDN.AMR-FDN.AIM
        DAD     B               (HL) = #FDN.AMR (AMOUNT READ)
        MOV     A,M             (A) = AMOUNT READ BEFORE
        ADD     D               ADD NEW AMOUNT
        MOV     M,A
        INX     H
        MOV     A,M
        ACI     0               PROPIGATE FOR VERY LARGE FILES
        MOV     M,A
        LXI     H,OBUFPTR
        MOV     A,M
        ADD     D               ADVANCE FREE RAM POINTER BY AMOUNT READ
        MOV     M,A
        MVI     A,CN.SOU
        DB      SYSCALL,.CLOSE  CLOSE FILE
        JMP     RPH             SEE IF MORE TO READ
WPH     SPACE   4,10
**      WPH - WRITE PHASE.
*
*       WPH HANDLES THE WRITE PHASE PROCESSING. IT IS ENTERED WITH
*       THE FDN CHAIN SETUP, THE NAMTAB SETUP, AND
*       THE DESTINATION DISK MOUNTED.
*
*
*       WRITE PHASE
*
*       DURING THE WRITE PHASE, THE DESTINATION DISK IS MOUNTED. THE NODES
*       ARE TAKEN FROM THE ACTIVE CHAIN, AND PROCESSED. IF THE FILE HAD
*       BEEN PARTIALLY WRITTEN THE LAST PASS, IT IS  RE-OPENED AND POSITIONED.
*       IF THERE IS NOT MORE DATA TO READ FOR A PROCESSED
*       NODE, IT IS REMOVED, AND THE CORRESPONDING ENTRY IN NAMTAB IS DELETED.
*
```

```
*       WRITE PHASE CONTINUES UNTIL
*
*               1) THERE ARE NO MORE FILE NODES IN THE ACTIVE LIST
*               2) OR, THE FIRST (AND ONLY) ENTRY IN THE LIST HAS NO
*                  MORE DATA IN MEMORY, BUT HAS NOT BEEN COMPLETELY READ.
*
*       ENTRY   NONE
*       EXIT    NONE
*       USES    ALL


WPH     EQU     *

*       SEE IF MORE TO WRITE

        LXI     H,FDNHEAD
        MOV     L,M
        MOV     A,L             (A) = FIRST NODE INDEX
        ANA     A
        RZ                      NO MORE
        CALL    $INDL
        DW      FDN.AIM         (E) = AMOUNT IN MEMORY FOR THIS GUY
        MOV     A,E
        ANA     A
        JNZ     WPH0            GOT DATA

*       NO DATA IN NODE. IF STILL READING, RETURN FOR MORE

        INX     H
        MOV     A,M
        DCX     H
        ANI     ST.OPR
        RNZ                     STILL READING, GET MORE
        XCHG                    (DE) = ADDRESS
        JMP     WPH4            REMOVE NODE, AM DONE WITH FILE

*       HAVE DATA TO WRITE. SEE IF WE HAVE OPENED THIS FILE BEFORE.,
*       OR IF THIS IS THE FIRST TIME

WPH0    PUSH    H               SAVE NODE POINTER
        INX     H
        ERRNZ   FDN.STA-1
        MOV     A,M             (A) = FDN.STA
        ANI     ST.OPW
        JNZ     WPH2            OPENED BEFORE
        ERRNZ   ST.OPW-1
        INR     M               SET '1' BIT

*       BUILD NAME INTO DESTFB

        PUSH    H               SAVE NODE ADDRESS
        LXI     B,OCOPYD
        LXI     D,NAMTAB
        LXI     H,DESTFB+FB.NAM
        CALL    MWN             MERGE WILDCARD NAME
        POP     H
```

```

*       IS 1ST TIME FOR THIS FILE. IF CONTIGUOUS FLAG, OPEN THE FILE
*       FOR CONTIGUOUS

        MOV     A,M             (A) = FLAG BYTE
        ANI     ST.CNT
        JNZ     WPH1            IS CONTIG
        LXI     H,DESTFB+FB.NAM
        MVI     A,CN.DES
        DB      SYSCALL,.OPENW  JUST OPEN FOR WRITE
        JC      DESTERR         ERROR
        JMP     WPH3            WRITE THE DATA

*       IS CONTIG FILE. OPEN IN CONTIG MODE

WPH1    INX     H
        ERRNZ   FDN.SIZ-FDN.STA-1
        MOV     C,M             (C) = COUNT (IN BLOCKS)
        MVI     B,0
        LXI     H,DESTFB+FB.NAM
        MVI     A,CN.DES
        PUSH    B               SAVE COUNT
        DB      SYSCALL,.DELET  DELETE OLD ONE
        JNC     WPH1.5          DELETED
        CPI     EC.FNF
        JNE     ERROR           MUST BE WRITE PROTECTED, OR  SOMETHING...
WPH1.5  POP     B               (BC) = COUNT
        LXI     H,DESTFB+FB.NAM
        MVI     A,CN.DES
        DB      SYSCALL,.OPENC  OPEN CONTIG
        JC      DESTERR
        JMP     WPH3

*       THIS FILE HAS ALREADY BEEN PARTIALLY WRITTEN. OPEN IN UPDATE MODE
*       SO WE CAN EXTEND IT.

WPH2    LXI     H,DESTFB+FB.NAM
        MVI     A,CN.DES
        DB      SYSCALL,.OPENU  OPEN FOR UPDATE
        JC      DESTERR         PROBLEMS
        POP     H
        PUSH    H               (HL) = #FDN.STA
        CALL    $INDL
        DW      FDN.AMW         (DE) = AMOUNT WRITTEN
        MOV     B,D
        MOV     C,E             (BC) = SECTORS WRITTEN
        MVI     A,CN.DES
        DB      SYSCALL,.POSIT  POSITION FOR EXTEND
        JC      IERR1           COULDNT GET THERE!

*       FILE OPEN AND POSITIONED. WRITE DATA

WPH3    POP     H
        PUSH    H               (HL) = #FDN.LNK
        CALL    $INDL
        DW      FDN.ADR         (E) = ADDR/256, (D) = CNT/256
        MOV     B,D
        MOV     D,E
        MVI     E,0             (DE) = ADDRESS
        MOV     C,E             (BC) = COUNT
        MVI     A,CN.DES
        PUSH    B               SAVE WRITE COUNT
        DB      SYSCALL,.WRITE  WRITE IT
        JC      DESTERR         PROBABLY OUT OF ROOM
        MVI     A,CN.DES
        DB      SYSCALL,.CLOSE  CLOSE IT
        JC      DESTERR
        POP     B               (B) = SECTORS WRITTEN
        POP     H
        PUSH    H               (HL) = #FDN.LNK
        LXI     D,FDN.AMW-FDN.LNK
        DAD     D               (HL) = FDN.AMW
        MOV     A,M
        ADD     B
        MOV     M,A
        INX     H
        MOV     A,M
        ACI     0               INCREMENT AMOUNT WRITTEN
        MOV     M,A

*       CLEAR 'IN MEMORY' COUNT IN NODE. IF THE FILE HAS NO MORE TO
*       READ, REMOVE IT FROM THE CHAIN AND NAMTAB

        POP     D               (DE) = FDN.LNK
WPH4    LXI     H,FDN.AIM
        DAD     D
        MVI     M,0             CLEAR AMOUNT IN MEMORY
        XCHG                    (HL) = FDN.LNK
        INX     H
        ERRNZ   FDN.STA-FDN.LNK-1
        MOV     A,M             (A) = FDN.STA
        ANI     ST.OPR
        RNZ                     STILL READING, AM DONE FOR THIS PHASE

*       UNLINK NODE FROM LIST

        DCX     H
        MOV     A,M
        STA     FDNHEAD         UNLINK FROM ACTIVE LIST
        LDA     FDNFRE
        MOV     M,A             PUT THIS GUY ON HEAD OF FREE LIST
        MOV     A,L
        STA     FDNFRE
        CALL    REN             REMOVE ENTRY FROM NAMTAB
        JMP     WPH             TRY TO WRITE THE NEXT GUY
CBR     SPACE   4,10
**      CBR - COMPUTE BUFFER ROOM.
*
*       CBR COMPUTES THE NUMBER OF SECTORS WORTH OF RAM
*       STILL FREE.
*
*       ENTRY   NONE
*       EXIT    (A) = SECTORS OF RAM FREE
*               'Z' SET IFF (A)  = 0
*               (H) = BUFFTR/256
*               (L) = OBUFLIM/256
*       USES    A,F


CBR     LHLD    OBUFLIM
        ERRNZ   OBUFPTR-OBUFLIM-1
        MOV     A,L
        SUB     H
        RET
IFL     SPACE   4,10
**      IFL - INITIALIZE FDN LIST.
*
*       IFL CHAINS ALL THE FDN NODES TO THE FREE LIST. THIS
*       CLEANUP IS NECESSARY IN CASE A CTL-C OR SOMETHING
*       LEFT THE LIST GARBAGED.
*
*       ENTRY   NONE
*       EXIT    NONE
*       USES    ALL


IFL     LXI     H,FDN.1
        MOV     A,L                     (A) = FIRST LINK
        STA     FDNFRE
        XRA     A
        STA     FDNHEAD                 NONE IN LIST
        MVI     B,FDNCNT-1              (B) = NUMBER OF NODES-1
IFL1    MVI     A,FDNELEN
        ADD     L                       (A) = #ADDR OF NEXT NODE
        MOV     M,A                     SET LINK
        MOV     L,A                     FORWARD TO NEXT LINK
        DCR     B
        JNZ     IFL1                    MORE TO GO
        MVI     M,0                     LAST ONE CHAINS NOWHERE
        RET
MAD     SPACE   4,10
**      MAD - MOUNT ALTERNATE DISK.
*
*       MAD DISMOUNTES THE CURRENT DISK, HAS THE USER INSERT THE
*       OTHER DISK, AND MOUNTS IT.
*
*       ENTRY   (B) = FRONT PANEL LED PATTERN
*               (DE) = PROMPT PATTERNS FOR PANEL AND CONSOLE
*       EXIT    (HL) = #VOLFLAG
*       USES    ALL


MAD     EQU     *

*       DISMOUNT CURRENT DISK

        PUSH    D
        PUSH    B                       SAVE ENTRY PARAMETERS IN CASE OF RETRY
```

```
2081              PUSH     D
2082              PUSH     B                     SAVE ENTRY PARAMETERS OVER SYDD CALL
2083              LXI      H,MNDA                DEVICE SPECIFICATION
2084              DB       SYSCALL,.DMNMS        DISMOUNT WITHOUT MESSAGE
2085              JC       ERROR                 IF ERROR
2086
2087     *        SETUP PROMPT ON FP LEDS AND CONSOLE FOR NEW DISK
2088
2089     MAD0     MVI      A,UO.DDU+UO.CLK+UO.HLT                                     /2.0a/
2090              STA      .MFLAG                HALT DISPLAY UPDATE
2091
2092              LXI      H,.ALEDS
2093              MVI      A,9
2094              POP      B                     (B) = PERIOD PATTERN
2095     MAD2     MOV      M,B                   SET PATTERN
2096              INX      H
2097              DCR      A
2098              JNZ      MAD2                  IF MORE TO BLANK
2099
2100              LXI      H,.ALEDS+3
2101              LXI      B,3
2102              POP      D                     (DE) = PROMPT LIST
2103              CALL     $MOVE                 MOVE IN PROMPT PATTERN
2104
2105              XCHG                           (HL) = PATTERN
2106              DB       SYSCALL,.PRINT        CONSOLE PROMPT
2107              CALL     $TYPTX
2108              DB       BELL+200Q             BEEP CONSOLE, TOO
2109              MVI      A,100
2110              CALL     .HORN                 BEEP A WARNING
2111
2112     *        WAIT FOR SIGNAL THAT NEW DISK IS IN
2113
2114     MAD3     MVI      A,DC.RDY                                                   /2.0a/
2115              CALL     SYDD                                                       /2.0a/
2116              JNC      MAD3                  Wait for device to go non-ready      /2.0a/
2117
2118     MAD4     MVI      A,DC.RDY                                                   /2.0a/
2119              CALL     SYDD                                                       /2.0a/
2120              JC       MAD4                  Wait for device to go ready          /2.0a/
2121
2122     *        READ NEW DISK'S LABEL
2123
2124              CALL     GETLAB
2125              JC       ERROR
2126
2127     *        SEE IF LABEL CHANGED FROM BEFORE
2128
2129              MVI      C,0                   Compare 256                          /2.0a/
2130              LXI      D,SLABEL              DE = address of last label           /2.0a/
2131              LXI      H,LABEL               HL = Address of current label        /2.0a/
2132              CALL     $COMP                 See if the label changed             /2.0a/
2133              POP      B
2134              POP      D                     RESTORE ENTRY PARAMETERS
2135
2136              LXI      H,VOLSER
```

```
                LDA     LABEL+LAB.SER
                JNE     MAD4.5                  IS THE RIGHT DISK                       /2.0a/
                PUSH    D                       SAVE PARAMS AS IN BEGINNING
                PUSH    B
                PUSH    D                       SAVE FOR RETRY
                PUSH    B
                JMP     MAD0                    IT WAS NOT THE RIGHT DISK

        MAD4.5  MOV     M,A                     SET NEW SERIAL
                LXI     H,VOLFLAG
                MOV     A,M
                CMA
                MOV     M,A                     COMPLEMENT VOLUME FLAG

        *       ERASE FRONT PANEL DISLPLAY

                LXI     H,.ALEDS
                MVI     A,9
        MAD5    MOV     M,B                     SET TO PATTERN
                INX     H
                DCR     A
                JNZ     MAD5

                LXI     B,256                                                           /2.0a/
                LXI     D,LABEL                                                         /2.0a/
                LXI     H,SLABEL                                                        /2.0a/
                CALL    $MOVE                   Save Current Label                      /2.0a/

                CALL    MND                     MOUNT NEW DISK
                CALL    $TYPTX                  Show user that disk is OK               /2.0a/
                DB      ENL                                                             /2.0a/
                RET
        MND     SPACE   4,10
        **      MND     - MOUNT NEW DISK
        *
        *       MOUNT NEW DISK ONTO DEVICE SSECIFIED IN MNDA
        *
        *
        *       ENTRY   NONE
        *
        *       EXIT    LABEL   = LABEL SECTOR
        *
        *       USES    ALL
        *

        MND     LXI     H,MNDA
                DB      SYSCALL,.MONMS  MOUNT WITHOUT MESSAGE
                JC      ERROR           IF ERROR IN MOUNT
                RET                                                                     /2.0a/

        MNDA    DB      'SY0:',0
        GETLAB  SPACE   4,10
        **      GETLAB  - GET LABEL
        *
        *       GETLAB READS THE DISK LABEL
        *
```

```
*       NOTE:   This routine leaves the volume mounted as               /2.0a/
*               zero.
*
*       ENTRY   NONE
*
*       EXIT    LABEL IN LABEL
*               (PSW)    = 'C' CLEAR IF NO ERROR
*                        = 'C' SET   IF    ERROR
*                           (A)  = ERROR CODE
*
*       USES    ALL
*

GETLAB  LXI     H,0                                                     /2.0a/
        MVI     A,DC.MOU                                                /2.0a/
        CALL    SYDD                    Mount the Disk as volume 0      /2.0a/
        RC                              Some type of problem            /2.0a/

        LXI     H,DDF.LAB
        LXI     D,LABEL
        LXI     B,256
        MVI     A,DC.RER                                                /2.0a/
        CALL    SYDD
        RET
        ENDIF
```

```
                                 ***       DELETE - PROCESS DELETE COMMAND.
                                 *
                                 *         SYNTAX:
                                 *
                                 *         SOURCE1,...,SOURCEN/DELETE
                                 *
                                 *         AT LEAST ONE SOURCE FILE MUST BE SPECIFIED.
                                 *         IF *.* IS SPECIFIED, DELETE ASKS,
                                 *                 DELETE ALL ?!? ARE YOU SURE?


000.000                                    IF      .PIP.
045.124                          DELETE    EQU     *
045.124  041 136 067                       LXI     H,LINE

                                 *         SEE IF A DESTINATION FILE SPECIFIED

045.127  176                     DEL1      MOV     A,M
045.130  043                               INX     H
045.131  247                               ANA     A
045.132  312 147 045                       JZ      DEL2                END OF LINE
045.135  376 075                           CPI     '='
045.137  302 127 045                       JNE     DEL1

                                 *         HE SPECIFIED A DESTINATION FILE

045.142  076 203                           MVI     A,PEC.TFI           TARGET FILE ILLEGAL
045.144  303 325 051                       JMP     ERROR               FORMAT ERROR

                                 *         NO TARGET FILE SPECIFIED

045.147  076 001                 DEL2      MVI     A,1                 CHECK FOR *.*
045.151  315 042 053                       CALL    BSL                 BUILD SOURCE FILE LIST
045.154  332 325 051                       JC      ERROR               NO GOOD

                                 *         DELETE FILES ONE BY ONE

045.157  052 030 064             DEL5      LHLD    NAMTLEN
045.162  174                               MOV     A,H
045.163  265                               ORA     L
045.164  310                               RZ                          END OF LIST
045.165  041 256 067                       LXI     H,NAMTAB
045.170  377 050                           DB      SYSCALL,.DELET      REMOVE IT
045.172  332 106 051                       JC      NAMERR              ERROR ON DELETE
045.175  315 275 056                       CALL    REN                 REMOVE ENTRY FROM NAMTAB
045.200  303 157 045                       JMP     DEL5                DELETE THE NEXT ONE
```

```
                            ***     RENAME - RENAME FILES.
                            *
                            *       SYNTAX:
                            *
                            *       DEST = SOURCE1,...,SOURCEN
                            *
                            *       RENAME IS PROCESSED IN A MANNER SIMILAR TO COPY, EXCEPT THAT THE
                            *       FILE IS RENAMED, RATHER THAN COPIED.


045.203                     RENAME  EQU     *
045.203  315 324 053                CALL    DDF               DECODE DESTINATION FILE
045.206  332 325 051                JC      ERROR
045.211  257                        XRA     A                 ALLOW *.*
045.212  315 042 053                CALL    BSL               BUILD SOURCEFILE LIST
045.215  332 325 051                JC      ERROR

                            *       DO MULTIPLE RENAMES

045.220  001 007 064        REN1    LXI     B,DESTFB+FB.NAM   (BC) = WILDCARDED TARGET NAME
045.223  021 256 067                LXI     D,NAMTAB          (DE) = NORMAL SOURCE NAME
045.226  041 327 045                LXI     H,RENA            (HL) = BUFFER FOR RESULT NAME
045.231  305                        PUSH    B                 SAVE #DESTFB+FB.NAM
045.232  325                        PUSH    D                 SAVE #NAMTAB
045.233  315 221 056                CALL    MWN               MERGE WILDCARD NAME
045.236  321                        POP     D                 (DE) = #NAMTAB
045.237  341                        POP     H                 (HL) = #DESTFB+FB.NAM


                            *       SEE IF SOURCE AND DEST FILE ON SAME DEVICE

045.240  325                        PUSH    D                 SAVE #NAMTAB (SOURCE NAME)
045.241  016 003                    MVI     C,3
045.243  315 060 030                CALL    $COMP             COMPARE DEVICES
045.246  076 201                    MVI     A,PEC.DNC         DEVICES NOT CONSISTANT
045.250  302 325 051                JNE     ERROR

                            *       SEE IF TARGET ALREADY EXISTS

045.253  041 327 045                LXI     H,RENA
045.256  076 000                    MVI     A,CN.SOU
045.260  377 042                    DB      SYSCALL,.OPENR
045.262  041 315 045                LXI     H,RENA-FB.NAM
045.265  332 275 045                JC      REN2              HAVE AN ERROR (AS WE SHOULD)
045.270  076 026                    MVI     A,EC.FAP          FILE AALREADY PRESENT
045.272  303 262 063                JMP     $FERROR           ALREADY THERE

045.275  376 014            REN2    CPI     EC.FNF            MUST BE NOT FOUND
045.277  302 262 063                JNE     $FERROR           OTHER ERROR
045.302  341                        POP     H                 (HL) = SOURCE NAME
045.303  001 327 045                LXI     B,RENA            (BC) = NEW (TARGET) NAME
045.306  377 051                    DB      SYSCALL,.RENAM    RENAME IT
045.310  332 106 051                JC      NAMERR            ERROR ON RENAME

                            *       REMOVE NAME FROM NAMTAB

```

```
045.313  315 275 056  2324          CALL    REN             REMOVE ENTRY FROM NAMTAB
045.316  052 030 064  2325          LHLD    NAMTLEN
045.321  174          2326          MOV     A,H
045.322  265          2327          ORA     L
045.323  302 220 045  2328          JNZ     REN1
045.326  311          2329          RET
                      2330
045.327               2331  RENA    DS      FB.NAML         FILE NAME WORK AREA
                      2332          ENDIF
                      ***     LIST - INDEX DIRECTORY.
                      *
                      *       DEST=SOURCE/LIST
                      *                  /BRIEF
                      *
                      *       THESE SWITCHES CAUSE THE DIRECTORY CONTENTS OF THE SPECIFIED FILE(S)
                      *       TO BE LISTED
                      *
                      *       IN /LI FIRM, THE OUTPUT IS:
                      *
                      *       NAME    EXT     SIZE      DATE          FLAGS
                      *       XXX     .XXX    NNN       DD-MMM-YY     CWS
                      *        .       .       .         .             .
                      *        .       .       .         .             .
                      *        .       .       .         .             .
                      *               NNN FILES USING MMM SECTORS, XXX FREE
                      *
                      *       IN /BR FORM, ONLY THE NAME AND EXTENSION ARE LISTED,
                      *       4 ACROSS THE PAGE.
                      *
                      *       SPECIAL CONSIDERATIONS:
                      *
                      *       A NULL NAME OR EXTENSION IS TAKEN AS '*' (WILDCARD)
                      *
                      *       IMPLIMENTATION:
                      *
                      *       A FILE LIST OF SOURCE FILES IS BUILT. THE DEVICE DIRECTORY FILE
                      *       IS THEN READ, AND EACH FILE IN IT IS CHECKED FOR A MATCH
                      *       AGAINST ANY SOURCE SPECIFICATIONS. ELIGIBLE FILES ARE LISTED.


045.350  041 000 000  LIST    LXI     H,0
045.353  303 361 045          JMP     LIST1

045.356  041 001 000  BRIEF   LXI     H,1
                      *       JMP     LIST1

045.361  042 114 047  LIST1   SHLD    LSTA            (LSTA) = 0 IF LIST, 1 IF /BRIEF
000.000                       ERRNZ   LSTB-LSTA-1     LSTB = FILE COUNT
045.364  041 000 000          LXI     H,0
045.367  042 116 047          SHLD    LSTC            CLEAR SECTORS USED COUNT
045.372  315 345 060          CALL    $MOVEL
045.375  011 000 277          DW      9,S.DATE,LSTG1  SET DATE IN HEADING

                      *       CRACK DESTINATION FILE NAMES

000.000                       IF      .PIF.
046.003  315 324 053          CALL    DDF             DECODE DEST FILE NAME
046.006  332 325 051          JC      ERROR           FILE NAME ERROR
046.011  247                  ANA     A
046.012  076 205              MVI     A,PEC.IUW       ILLEGAL USE OF WILDCARD IN DEST
046.014  312 325 051          JZ      ERROR
                              ENDIF

                      *       BUILD LIST OF SPECIFICATIONS

```

```
046.017  315 302 047             CALL    BLS              BUILD LIST OF SOURCE SPECS
046.022  332 325 051             JC      ERROR            ERROR IN LIST
046.025  001 003 000             LXI     B,3
046.030  041 352 063             LXI     H,DIRNAM
046.033  315 252 030             CALL    $MOVE            MOVE DEVICE CODE INTO DIRECT.SYS NAME
046.036  041 354 063             LXI     H,DIRNAM+2
046.041  176                     MOV     A,M              SEE IF UNIT NUMBER OMITTED
046.042  247                     ANA     A
046.043  302 050 046             JNZ     LIST1.5          SPECIFIED
046.046  066 060                 MVI     M,'0'            DONT ALLOW NULL NUMBER

                       *         GET ADDRESS OF DEVICE'S GRT

046.050  041 352 063   LIST1.5   LXI     H,DIRNAM         (HL) = # OF XXX:DIRECT.SYS (XXX = DEVICE)
046.053  001 120 047             LXI     B,LSTD           (BC) = ADDRESS FOR RETURN INFO
046.056  377 053                 DB      SYSCALL,.DECODE            DECODE NAME
046.060  332 325 051             JC      ERROR            UNKNOWN DEVICE
046.063  072 120 047             LDA     LSTD+0
046.066  346 001                 ANI     DT.DD
046.070  076 005                 MVI     A,EC.DNS
046.072  312 325 051             JZ      ERROR            NOT DIRECTORY DEVICE
046.075  052 141 047             LHLD    LSTD+17          (HL) = DEV TBL ADDR                 /80.04.sc/

046.100  021 011 000             LXI     D,DEV.UNT                                            /80.04.sc
046.103  031                     DAD     D
046.104  072 123 047             LDA     LSTD+3
046.107  315 027 041             CALL    S.GUP            HL = UNIT TABLE POINTER

046.112  315 353 057             CALL    $INDLB                                               /80.04.sc/
046.115  001 000                 DW      UNT.SPG                                              /80.04.sc/
046.117  062 152 047             STA     LSTF             SAVE  SECTORS PER GROUP             /80.04.sc/

046.122  315 234 030             CALL    $INDL
046.125  002 000                 DW      UNT.GRT
046.127  353                     XCHG
046.130  042 150 047             SHLD    LSTE             SAVE  GRT ADDRESS
046.133  353                     XCHG

                       *         OPEN DEVICE'S DIRECTORY

046.134  041 352 063             LXI     H,DIRNAM
046.137  076 002                 MVI     A,CN.DIR
046.141  377 042                 DB      SYSCALL,.OPENR
046.143  076 200                 MVI     A,FEC.DF         DEVICE FORMAT ERROR
046.145  332 325 051             JC      ERROR            CANT OPEN DIRECTORY


                       *         OPEN OUTPUT FILE

000.000                          IF      .PIP.
046.150  041 375 063             LXI     H,DESTFB
046.153  315 261 061             CALL    $FOPEW           OPEN FOR WRITE
                                 ENDIF

                       *         GENERATE HEADING

```

```
046.156  001 001 000  2447         LXI     B,1                (BC) = TEXT COUNT
046.161  021 153 047  2448         LXI     D,LSTG             (DE) = TEXT ADDRESS
046.164  072 114 047  2449         LDA     LSTA
046.167  247          2450         ANA     A
046.170  302 175 046  2451         JNZ     LIST2              IS SHORT
046.173  016 051      2452         MVI     C,LSTGL            PRINT FULL HEADING
000.000               2453         IF      .PIP.
046.175  315 012 062  2454  LIST2  CALL    $FWRIB             WRITE HEADING
                      2455         ELSE
                      2456  LIST2  MOV     A,C
                      2457         XCHG                       (HL) = LINE ADDRESS
                      2458         CALL    $TYPCC             PRINT ON CONSOLE
                      2459         ENDIF
                      2460
                      2461  *      READ DIRECTORY BLOCKS, LOOKING FOR FILE MATCHES
                      2462
046.200  001 000 002  2463  LIST3  LXI     B,512
046.203  315 135 056  2464         CALL    GDWP               DE = DIRECTORY WORKSPACE POINTER  /79.11.GC/
046.206  076 002      2465         MVI     A,CN.DIR
046.210  325          2466         PUSH    D                                                   /79.11.GC/
046.211  377 004      2467         DB      SYSCALL,.READ
046.213  321          2468         POP     D                  DE = DIRECOTRY WORKSPACE         /79.11.GC/
046.214  332 366 046  2469         JC      LIST9              ALL DONE
                      2470
                      2471  *      CHECK NEXT ENTRY IN NAMTAB AGAINST DIRECTORY ENTRY.
                      2472  *      (DE) = DIRECTORY BUFFER POINTER
                      2473
046.217  032          2474  LIST4  LDAX    D                  (A) = FIRST CHARACTER OF NAME
046.220  247          2475         ANA     A
046.221  312 200 046  2476         JZ      LIST3              END OF THIS BUFFER
046.224  074          2477         INR     A
000.000               2478         ERRNZ   DF.EMP-377Q
046.225  312 320 046  2479         JZ      LIST7              THIS ENTRY IS EMPTY
046.230  074          2480         INR     A
046.231  312 366 046  2481         JZ      LIST9              NO MORE ENTRYS IN DIRECTORY
046.234  353          2482         XCHG
046.235  315 233 053  2483         CALL    CFE                CHECK FILE ELIGIBILITY
046.240  353          2484         XCHG
046.241  302 320 046  2485         JNE     LIST7              NOT ELIGIBLE
046.244  041 256 067  2486         LXI     H,NAMTAB
                      2487
046.247  345          2488  LIST5  PUSH    H
046.250  325          2489         PUSH    D                  SAVE ADDRESS OF FILE AND PATTERN
046.251  315 016 054  2490         CALL    CAD                CONVERT ASCII NAMTAB ENTRY TO DIRECTORY FORMAT
046.254  021 066 067  2491         LXI     D,PIO.DIR+DIR.NAM          (DE) = NAMTAB PATTERN
046.257  341          2492         POP     H
046.260  345          2493         PUSH    H                  (HL) = DIRECTORY PATTERN
046.261  006 013      2494         MVI     B,8+3              CHECK FOR MATCH
046.263  315 306 053  2495         CALL    CWM                CHECK FOR WILDCARD MATCH
046.266  321          2496  LIST6  POP     D
046.267  341          2497         POP     H
046.270  312 347 046  2498         JE      LIST8              GOT FILE TO LIST
046.273  001 021 000  2499         LXI     B,FB.NAML
046.276  011          2500         DAD     B                  ADVANCE PAST ENTRY IN NAMTAB
                      2501
                      2502  *      SEE IF AT END OF NAMTAB
```

```

046.277  325                       PUSH    D
046.300  353                       XCHG                    (DE) = NEW ADDRESS
046.301  052 030 064               LHLD    NAMTLEN
046.304  001 256 087               LXI     B,NAMTAB
046.307  011                       DAD     B               (HL) = LWA+1 OF TABLE
046.310  353                       XCHG
046.311  315 216 030               CALL    $CDEHL          COMPARE
046.314  321                       POP     D
046.315  302 247 046               JNE     LIST5           MORE IN TABLE

                          *        FILE DOESNT MATCH ANY SELECTED FILE. PASS TO NEXT ONE

046.320  353              LIST7    XCHG                    (HL) = DIR BUFFER ADDRESS

046.321  345                       PUSH    H                                               /79.11.GC/
046.322  315 143 056               CALL    GDWP.           HL = DIRECTORY WORKSPACE PTR.   /79.11.GC/
046.325  315 353 057               CALL    $INDLB          A  = DIR. ENTRY LENGTH          /79.11.GC/
046.330  373 001                   DW      DIS.ENL                                         /79.11.GC/
046.332  341                       POP     H                                               /79.11.GC/

046.333  315 101 030               CALL    $DADA.          ADVANCE
046.336  176                       MOV     A,M
046.337  247                       ANA     A
046.340  353                       XCHG
046.341  302 217 046               JNZ     LIST4           TRY THIS ONE
046.344  303 200 046               JMP     LIST3           READ ANOTHER BLOCK

                          *        HAVE FILE TO LIST

046.347  325              LIST8    PUSH    D               SAVE DIR POINTER
046.350  072 152 047               LDA     LSTF            (A) = SECTORS PER GROUP THIS DEVICE
046.353  315 052 050               CALL    PFI             PRINT FILE INFO
046.356  321                       POP     D
046.357  041 115 047               LXI     H,LSTB
046.362  064                       INR     M               COUNT FILE
046.363  303 320 046               JMP     LIST7           ADVANCE TO NEXT FILE

                          *        ALL DONE. CLOSE DIRECTORY FILE

046.366  076 002          LIST9    MVI     A,CN.DIR
046.370  377 046                   DB      SYSCALL,.CLOSE  CLOSE FILE
046.372  001 001 000               LXI     B,1             ASSUME SHOFT FORM, JUST WRITE NL
046.375  072 114 047               LDA     LSTA            (A) = FORM FLAG
047.000  247                       ANA     A
047.001  302 071 047               JNZ     LIST10          IS SHORT, NO TRAILER

                          *        PRINT SUMMARY:
                          *
                          *        NNN FILES, USING XXX SECTORS, YYY FREE

047.004  072 115 047               LDA     LSTB
047.007  117                       MOV     C,A
047.010  006 000                   MVI     B,0             (BC) = FILE COUNT
047.012  076 003                   MVI     A,3
047.014  041 230 047               LXI     H,LSTH1
```

```
047.017  315 272 060  2559           CALL     $UDDN            FILE COUNT
                      2560
047.022  052 116 047  2561           LHLD     LSTC
047.025  104          2562           MOV      B,H
047.026  115          2563           MOV      C,L              (BC) = SECTOR COUNT
047.027  041 251 047  2564           LXI      H,LSTH2
047.032  076 004      2565           MVI      A,4                                                  /80.05.sc/
047.034  315 272 060  2566           CALL     $UDDN            USED COUNT
                      2567
047.037  052 150 047  2568           LHLD     LSTE
047.042  176          2569           MOV      A,M
047.043  315 253 053  2570           CALL     CFS              FOLLOW GRT CHAIN
047.046  072 152 047  2571           LDA      LSTF
047.051  315 007 031  2572           CALL     $MU86            (HL) = SECTORS FREE
047.054  104          2573           MOV      B,H
047.055  115          2574           MOV      C,L
047.056  041 267 047  2575           LXI      H,LSTH3
047.061  076 004      2576           MVI      A,4                                                  /80.05.sc/
047.063  315 272 060  2577           CALL     $UDDN            UNPACK FREE
                      2578
047.066  001 056 000  2579           LXI      B,LSTHL
047.071  021 224 047  2580  LIST10   LXI      D,LSTH
047.074  072 350 063  2581           LDA      SUPRES
047.077  247          2582           ANA      A
000.000               2583           IF       .PIP.
047.100  041 375 063  2584           LXI      H,DESTFB
047.103  302 300 062  2585           JNZ      $FCLO            CLOSE AND EXIT, SUMMARY SUPPRESSED
047.106  315 012 062  2586           CALL     $FWRIB           WRITE TRAILER
                      2587
                      2588  *        ALL DONE, CLOSE OUTPUT FILE
                      2589
047.111  303 300 062  2590           JMP      $FCLO            CLOSE AND EXIT
                      2591           ELSE
                      2592           RNZ                       NOT TO SUMMARYIZE
                      2593           MOV      A,C              (A) = COUNT
                      2594           XCHG                      (HL) = ADDRESS
                      2595           JMP      $TYPCC           TYPE TEXT AND EXIT
                      2596           ENDIF
                      2597
047.114  000          2598  LSTA     DB       0                <>0 IFF SHORT FORM
                      2599
047.115  000          2600  LSTB     DB       0                FILE COUNT
047.116  000 000      2601  LSTC     DW       0                SECTORS USED
047.120               2602  LSTD     DS       24               FILE NAME DECODE AREA
047.150  000 000      2603  LSTE     DW       0                GRT ADDRESS
047.152  000          2604  LSTF     DB       0                SECTORS PER GROUP FOR THIS DEVICE
047.153  012 116 141  2605  LSTG     DB       NL,'Name',TAB,'.Ext',TAB,'Size',TAB,'  Date',TAB,TAB,'Flags',TAB
047.211               2606  LSTG1    DS       9                DATE
047.222  012 012      2607           DB       NL,NL
000.051               2608  LSTGL    EQU      *-LSTG
                      2609
047.224  012 040 040  2610  LSTH     DB       NL,'   '                    FIRST CHARACTER MUST BE <NL>
047.230  116 116 116  2611  LSTH1    DB       'NNN Files, Using '
047.251  115 115 115  2612  LSTH2    DB       'MMMM Sectors ('
047.267  130 130 130  2613  LSTH3    DB       'XXXX Free)',NL
000.056               2614  LSTHL    EQU      *-LSTH
```

```
                           **     BLS - BUILD LIST OF SOURCE FILES.
                           *
                           *      BLS BUILDS A LIST OF SOURCE FILES INTO *NAMTAB*
                           *      NULL FIELDS ARE SET TO WILDCARDS. BLS REQUIRES THAT ALL
                           *      FILES SPECIFIED HAVE THE SAME DEVICE.
                           *
                           *      IF THE COMMAND LINE CONTAINS NO FILES, BUT CONTAINS AT LEAST
                           *      ONE BLANK (AS WOULD BE THE CASE IN PROCESSING THE /LIST SWITCH, SINCE
                           *      THE '/LIST' IS REPLACED WITH BLANKS) A FILE NAME OF ????????.???
                           *      IS DECODED.
                           *      ENTRY   NAMTAB EMPTY
                           *      EXIT    'C' CLEAR IF OK
                           *               (DE) = #BLSA = 3 CHARACTER DEVICE NAME
                           *              'C' SET IF ERROR
                           *               (A) = ERROR MESSAGE
                           *      USES    ALL


047.302  315 345 060       BLS    CALL    $MOVEL
047.305  003 000 045              DW      3,BLSC,BLSA         SET INITIAL DEFAULT DEVICE
047.313  041 000 000              LXI     H,0
047.316  042 030 064              SHLD    NAMTLEN             CLEAR NAMTAB
047.321  076 377                  MVI     A,377Q
047.323  062 044 050              STA     BLSB                FLAG PROCESSING OF FIRST FILE NAME
047.326  315 201 056              CALL    LSN                 LOCATE SOURCE NAMES

                           *      CRACK THE NEXT NAME

047.331  176               BLS1   MOV     A,M
047.332  021 036 050              LXI     D,BLSA              (DE) = DEFAULT ADDRESS
047.335  247                      ANA     A
047.336  310                      RZ                          NO MORE NAMES
047.337  315 222 057              CALL    $SOB                SEE IF ALL NULL
047.342  176                      MOV     A,M
047.343  247                      ANA     A
047.344  302 352 047              JNZ     BLS2                NOT ALL NULL
047.347  041 045 050              LXI     H,BLSC              USE DEFAULT DEVICE
047.352  315 022 054       BLS2   CALL    CAD.                CONVERT ASCII NAME TO DIRECTORY FORMAT
047.355  330                      RC                          ERROR

                           *      IF FIRST NAME, RECORD DEVICE
                           *      IF NOT FIRST, COMPARE DEVICE AGAINST FIRST DEVICE

047.356  345                      PUSH    H
047.357  021 063 067              LXI     D,PIO.DEV
047.362  041 036 050              LXI     H,BLSA
047.365  001 003 000              LXI     B,3                 SETUP COUNT, FROM AND TO
000.000                           IF      .PIP.
047.370  072 044 050              LDA     BLSB
047.373  247                      ANA     A
047.374  362 011 050              JP      BLS3                NOT 1ST FILE
047.377  315 252 030              CALL    $MOVE               MOVE IN REQUIRED DEVICE FOR REMAINING FILES
050.002  257                      XRA     A
050.003  062 044 050              STA     BLSB                FLAG 1ST NAME PROCESSED
050.006  303 024 050              JMP     BLS4
                                  ENDIF
```

```

050.011  315 060 030  BLS3    CALL    $COMF            SEE IF THIS DEVICE SAME AS PREVIOUS
050.014  312 024 050          JE      BLS4             OK
050.017  076 201              MVI     A,PEC.DNC        MULTIPLE DEVICES ARE ILLEGAL
050.021  067                  STC
050.022  341                  POP     H
050.023  311                  RET                      RETURN WITH ERROR

                      *       GOT NAME DECODED. ENTER IN NAMTAB

050.024  315 347 052  BLS4    CALL    AEN              ADD ENTRY TO NAMTAB
050.027  341                  POP     H
050.030  315 370 056          CALL    SFS              SKIP FILE SEPERATOR (BLANKS AND/OR COMMA)
050.033  303 331 047          JMP     BLS1             SEE IF MORE

050.036  123 131 060  BLSA    DB      'SY0',200Q,200Q,200Q
050.044  000          BLSB    DB      0                FIRST FILE NAME FLAG
050.045  123 131 060  BLSC    DB      'SY0:',0                  DEFAULT DEVICE



                      **      PFI - PRINT FILE INFO.
                      *
                      *       PFI DECODES A DIRECTORY ENTRY INTO A CODED LINE, THEN
                      *       WRITES IT TO 'DESTFB'.
                      *
                      *       THE PRODUCED FORMAT DEPENDS UPON THE LISTING FORMAT FLAG,
                      *       LSTA.
                      *
                      *       SHORT FORM:
                      *
                      *       NAME    .EXT    (TAB)
                      *
                      *       LONG FORM:
                      *
                      *       NAME    .EXT    SIZE    DATE    FLAGS   (NL)
                      *
                      *       ENTRY   (A) = SECTORS PER GROUP FOR THIS DEVICE
                      *               (DE) = DIRECTORY ENTRY POINTER
                      *       EXIT    IF LONG FORM, SECTOR COUNT IS ACCUMULATED IN LSTC
                      *       USES    ALL


050.052  062 032 051  PFI     STA     PFIC             SAVE SECTORS PER GROUP
050.055  041 347 050          LXI     H,PFIA
050.060  016 010              MVI     C,8
050.062  315 331 050          CALL    PFI20            COPY NAME
050.065  312 073 050          JZ      PFI1             ALL 8 CHARACTERS
050.070  066 011              MVI     M,TAB
050.072  043                  INX     H
050.073  066 056      PFI1    MVI     M,'.'
050.075  043                  INX     H
050.076  016 003              MVI     C,3
050.100  315 331 050          CALL    PFI20            COPY EXTENSION
050.103  066 011              MVI     M,TAB
```

```
050.105  043          2725          INX     H
050.106  072 114 047  2726          LDA     LSTA
050.111  247          2727          ANA     A
050.112  312 137 050  2728          JZ      PFI3            IS LONG FORM
                      2729
                      2730  *       IS SHORT FORM, SEE IF NEED TO END LINE
                      2731
050.115  074          2732          INR     A
050.116  376 005      2733          CPI     5
050.120  302 131 050  2734          JNE     PFI2            NOT TIME YET
050.123  053          2735          DCX     H
050.124  066 012      2736          MVI     M,NL
050.126  043          2737          INX     H               TIME TO END LINE
050.127  076 001      2738          MVI     A,1
050.131  062 114 047  2739  PFI2    STA     LSTA            RESET COUNT
050.134  303 305 050  2740          JMP     PFI6            OUTPUT TO FILE
                      2741
                      2742  *       IS LONG FORM.
                      2743
050.137  001 005 000  2744  PFI3    LXI     B,DIR.FGN-DIR.EXT-3
050.142  353          2745          XCHG                    (DE) = LINE ADDR, (HL) = #PIO.DIR+DIR.EXT+3
050.143  011          2746          DAD     B               (HL) = #DIR.FGN
050.144  176          2747          MOV     A,M             (A) = (DIR.FGN)
050.145  043          2748          INX     H
050.146  043          2749          INX     H
050.147  116          2750          MOV     C,M             (C) = DIR.LSI = SECTORS USED IN LAST GROUP
000.000               2751          ERRNZ   DIR.LSI-DIR.FGN-2
050.150  353          2752          XCHG                    (DE) = ADDRESS OF LSI
050.151  325          2753          PUSH    D               SAVE #DIR.LSI
050.152  345          2754          PUSH    H               SAVE LINE ADDRESS
050.153  052 150 047  2755          LHLD    LSTE
050.156  157          2756          MOV     L,A
050.157  176          2757          MOV     A,M
050.160  315 253 053  2758          CALL    CFS             COMPUTE FILE ISZE
050.163  072 032 051  2759          LDA     PFIC            (A) = SECTORS PER GROUP
050.166  107          2760          MOV     B,A                                             /80.06.GC/
050.167  315 007 031  2761          CALL    $MU86           (HL) = SECTORS USED (EXCEPT FOR THOSE IN LAST GROUP)
                      2762
050.172  072 344 063  2763          LDA     ALLOCA                                          /80.06.sc/
050.175  247          2764          ANA     A                                               /80.06.sc/
050.176  312 202 050  2765          JZ      PFI3.5                                          /80.06.sc/
050.201  110          2766          MOV     C,B             Use Group Size instead if /ALL  /80.06.sc/
050.202               2767  PFI3.5  EQU     *                                               /80.06.sc/
                      2768
050.202  006 000      2769          MVI     B,0
050.204  011          2770          DAD     B               (HL) = SECTORS USED
050.205  104          2771          MOV     B,H
050.206  115          2772          MOV     C,L             (BC) = SECTORS USED COUNT
050.207  052 116 047  2773          LHLD    LSTC
050.212  011          2774          DAD     B
050.213  042 116 047  2775          SHLD    LSTC            ACCUMULATE COUNT OF SECTORS
050.216  341          2776          POP     H               (HL) = LINE ADDRESS
050.217  076 004      2777          MVI     A,4             3 DIGITS MAX                    /80.05.sc/
050.221  315 272 060  2778          CALL    $UDDN           UNPACK COUNT
050.224  066 011      2779          MVI     M,TAB
050.226  043          2780          INX     H
```

```
050.227  321                       POP     D                   (DE) = #DIR.LSI

                          *        TYPE DATE

050.230  353                       XCHG
000.000                            ERRNZ   DIR.CRD-DIR.LSI-1
050.231  043                       INX     H                   (HL) = #DIR.CRD
050.232  345                       PUSH    H
050.233  315 211 030               CALL    $HLIHL
050.236  353                       XCHG
050.237  315 056 060               CALL    $DAD                DECODE AUGUSTAN DATE

                          *        CODE FLAGS

050.242  353                       XCHG                        (DE) = LINE ADDRESS
050.243  341                       POP     H                   (HL) = #DIR.CRD
050.244  001 373 377               LXI     B,DIR.FLG-DIR.CRD
050.247  011                       DAD     B                   (HL) = ADDRESS OF DIRFLG
050.250  176                       MOV     A,M                 (A) = FLAGS
050.251  353                       XCHG                        (HL) = LINE ADDRESS
050.252  247                       ANA     A
050.253  312 302 050               JZ      PFI5.5              NO FLAGS
050.256  066 011                   MVI     M,TAB               TAB BEFORE FLAGS
050.260  043                       INX     H
050.261  021 022 051               LXI     D,PFIB
050.264  207              PFI4     ADD     A
050.265  322 275 050               JNC     PFI5                NOT SET
050.270  365                       PUSH    PSW                 SAVE FLAGS
050.271  032                       LDAX    D
050.272  167                       MOV     M,A
050.273  361                       POP     PSW                 RESTORE FLAGS
050.274  043                       INX     H
050.275  023              PFI5     INX     D                   SET FLAG
050.276  247                       ANA     A
050.277  302 264 050               JNZ     PFI4                MORE FLAGS SET
050.302  066 012          PFI5.5   MVI     M,NL
050.304  043                       INX     H

                          *        LINE ALL BUILT. WRITE TO DESTFB

050.305  021 031 327      PFI6     LXI     D,-PFIA
050.310  031                       DAD     D
000.000                            IF      .PIP.
050.311  104                       MOV     B,H
050.312  115                       MOV     C,L                 (BC) = LEN
050.313  021 347 050               LXI     D,PFIA              (DE) = DATA FWA
050.316  041 375 063               LXI     H,DESTFB
050.321  303 012 062               JMP     $FWRIB              WRITE AND EXIT
                                   ELSE
                                   MOV     A,L                 (A) = COUNT
                                   LXI     H,PFIA
                                   JMP     $TYPCC              TYPE LINE AND EXIT
                                   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
050.324  167                     2848 PFI19  MOV     M,A                COPY
050.325  043                     2849        INX     H
050.326  023                     2850        INX     D
050.327  015                     2851        DCR     C
050.330  310                     2852        RZ                         ALL COPIED
050.331  032                     2853 PFI20  LDAX    D
050.332  247                     2854        ANA     A
050.333  302 324 050             2855        JNZ     PFI19              GOT CHAR
                                 2856
                                 2857  *       NO NAME. (C) = COUNT LEFT
                                 2858
050.336  173                     2859        MOV     A,E
050.337  201                     2860        ADD     C
050.340  137                     2861        MOV     E,A
050.341  172                     2862        MOV     A,D
050.342  316 000                 2863        ACI     0
050.344  127                     2864        MOV     D,A
050.345  263                     2865        ORA     E                  CLEAR 'Z'
050.346  311                     2866        RET
                                 2867
050.347                          2868 PFIA   DS      0                  BUFFER AREA FOR LINE BUILD
050.347  130 130 130             2869        DB      'XXXXXXXX.YYY      NNNN    DD-MMM-YY'
051.002  011 011 106             2870        DB      '                  FLAGS         '
051.022  123 114 127             2871 PFIB   DB      'SLW'              CODES
051.025  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
051.032  000                     2877 PFIC   DB      0                  SECTORS PER GROUP FOR THIS DEVICE
```

```
                          ***     VERSN   - PIP VERSION INFORMATION
                          *
                          *       DEST=/V[ERSION]
                          *
                          *       PRINT THE PIP VERSION INFORMATION TO THE 'DEST' FILE.
                          *

051.033                   VERSN   EQU     *

051.033  315 271 053              CALL    CTS             CHECK FOR TARGET FILE SPECIFICATION
051.036  067                      STC
051.037  302 325 051              JNZ     ERROR           TARGET FILE SPECIFICATION ILLEGAL
051.042  041 136 067              LXI     H,LINE
051.045  315 222 057              CALL    $SOB            SKIP OVER ALL THE BLANKS ($DRS TURNS SWITCHES
051.050  176                      MOV     A,M              TO BLANKS)
051.051  247                      ANA     A
051.052  076 207                  MVI     A,PEC.SFI       SOURCE FILE ILLEGAL
051.054  067                      STC
051.055  302 325 051              JNZ     ERROR           ONLY ALLOW SWITCH ON LINE
051.060  315 136 031              CALL    $TYPTX

000.000                           IF      .PIP.
051.063  120 111 120              DB      'PIP'
                                  ELSE
                                  DB      'ONECOPY'
                                  ENDIF

051.066  011 126 145              DB      TAB,'Version:  '
051.101  062 056 060              DB      VERS/16+'0','.',VERS&00001111B+'0'
051.104  212                      DB      ENL

051.105  311                      RET
```

```
                        2914  **      ERROR PROCESSING ROUTINES
                        2915  *


                        2917  ***     NAMERR - FILE TYPE ERROR, OCCURRED ON FILE WHOSE NAME
                        2918  *       IS NEXT UP IN NAMTAB.
                        2919  *
                        2920  *       PROCESS VIA $FERROR
                        2921
000.000                 2922          IF      .PIP.
051.106  041 244 067    2923  NAMERR  LXI     H,NAMTAB-FB.NAM
051.111  303 262 063    2924          JMP     $FERROR
                        2925          ELSE
                        2926  NAMERR  LHLD    NAMTPTR
                        2927          LXI     B,-FB.NAM
                        2928          DAD     B
                        2929          JMP     $FERROR
                        2930  DESTERR SPACE   4,10
                        2931  **      ERROR ON FILE IN DESTFB
                        2932
                        2933  DESTERR LXI     H,DESTFB
                        2934          JMP     $FERROR
                        2935          ENDIF


                        2937  **      INTERNAL ERRORS. SHOULD NOT OCCUR.
                        2938
051.114  076 061        2939  IERR1   MVI     A,'1'
051.116  303 133 051    2940          JMP     INTERR
                        2941
051.121  076 062        2942  IERR2   MVI     A,'2'
051.123  303 133 051    2943          JMP     INTERR
051.126  076 063        2944  IERR3   MVI     A,'3'
051.130  303 133 051    2945          JMP     INTERR
                        2946
                        2947
051.133  365            2948  INTERR  PUSH    PSW                SAVE CODE
051.134  315 136 031    2949          CALL    $TYPTX
051.137  007 012 120    2950          DB      BELL,NL,'PIP INTERNAL ERROR ','#'+200Q
051.165  361            2951          POP     PSW
051.166  315 376 060    2952          CALL    $WCHAR
051.171  315 136 031    2953          CALL    $TYPTX
051.174  012 124 110    2954          DB      NL,'THIS ERROR SHOULD NOT OCCUR. CONTACT HEATH TECHNICAL'
051.261  012 103 117    2955          DB      NL,'CORRESPONDENCE FOR ASSISTANCE.',NL
051.321  076 001        2956          MVI     A,1
051.323  377 000        2957          DB      SYSCALL,.EXIT            ABORT
```

```
                          2959  **      ERROR - GENERAL AND SYNTAX ERRORS NOT DIRECTLY ASSOCIATED
                          2960  *       WITH A VALID FILE NAME.
                          2961
                          2962
051.325  365              2963  ERROR   PUSH    PSW             SAVE CODE
051.326  315 136 031      2964          CALL    $TYPTX
051.331  007 105 122      2965          DB      BELL,'ERROR -',' '+200Q
051.342  361              2966          POP     PSW
051.343  247              2967          ANA     A
051.344  372 356 051      2968          JM      ERROR1          IS PRODUCT ERROR
051.347  046 012          2969          MVI     H,NL            USE NL AS MESSAGE TRAIL CHAR
051.351  377 057          2970          DB      SYSCALL,.ERROR  LOOK UP SYSTEM ERROR
051.353  303 200 042      2971          JMP     RESTART
                          2972
                          2973  *       IS PRODUCT ERROR
                          2974
051.356  041 373 051      2975  ERROR1  LXI     H,ERRORA
051.361  276              2976  ERROR2  CMP     M
051.362  043              2977          INX     H
051.363  302 361 051      2978          JNE     ERROR2          FIND ERROR MESSAGE
000.001                   2979          IF      ONECOPY
                          2980          CALL    $TYPTX
                          2981          DB      BELL,'ONECOPY Error #',' '+200Q
                          2982          ENDIF
051.366  377 003          2983          DB      SYSCALL,.PRINT  PRINT MESSAGE
051.370  303 200 042      2984          JMP     RESTART
                          2985
051.373                   2986  ERRORA  DS      0               ERROR MESSAGES
000.000                   2987          IF      .PIP.
051.373  200 104 145      2988          DB      PEC.DF,'Device Format Error',ENL
052.020  201 101 154      2989          DB      PEC.DNC,'All Files Must Reside on the Same Device',ENL
052.072  203 104 145      2990          DB      PEC.TFI,'Destination File Specification is Illegal',ENL
052.145  204 103 157      2991          DB      PEC.CS,'Contradictory Switches Specified',ENL
052.207  205 111 154      2992          DB      PEC.IUW,'Illegal Use of Wildcard',ENL
052.240  206 111 154      2993          DB      PEC.IDF,'Illegal Destination File Format',ENL
052.301  207 123 157      2994          DB      PEC.SFI,'Source File Specification is Illegal',ENL
                          2995          ELSE
                          2996          DB      PEC.DF,'01',ENL
                          2997          DB      PEC.DNC,'02',ENL
                          2998          DB      PEC.TFI,'03',ENL
                          2999          DB      PEC.CS,'04',ENL
                          3000          DB      PEC.IUW,'05',ENL
                          3001          DB      PEC.IDF,'06',ENL
                          3002          DB      PEC.SFI,'07',ENL
                          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
052.347  041 021 053      3018  AEN     LXI     H,AENA
052.352  315 131 055      3019          CALL    CDA                     CONVERT DIRECTORY FORMAT TO ASCII FORMAT
052.355  326 001          3020          SUI     1                       'C' SET IF WILDCARD
052.357  365              3021          PUSH    PSW                     SAVE FLAG
052.360  052 030 064      3022          LHLD    NAMTLEN
052.363  001 021 000      3023          LXI     B,FB.NAML
052.366  011              3024          DAD     B                       INCREASE SIZE
052.367  042 030 064      3025          SHLD    NAMTLEN
052.372  353              3026          XCHG                            (DE) = NEW LENGTH
052.373  052 032 064      3027          LHLD    NAMTMAX
052.376  175              3028          MOV     A,L                     SEE IF WILL OVERFLOW
052.377  223              3029          SUB     E
053.000  174              3030          MOV     A,H
053.001  232              3031          SBB     D
053.002  334 147 056      3032          CC      INA                     INCREASE NAMTAB ALLOCATION
053.005  041 235 067      3033          LXI     H,NAMTAB-FB.NAML
053.010  031              3034          DAD     D                       (HL) = *TO* ADDRESS
053.011  021 021 053      3035          LXI     D,AENA                  (DE) = *FROM* ADDRESS
053.014  315 252 030      3036          CALL    $MOVE                   MOVE ENTRY IN
053.017  361              3037          POP     PSW                     (PSW) = WILDCARD FLAG
053.020  311              3038          RET
                          3039
053.021                   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
053.042  062 113 053      3055  BSL     STA     BSLA                    SAVE ASK FLAG
053.045  315 201 056      3056          CALL    LSN                     LOCATE SOURCE NAME
                          3057
                          3058  *       GO THROUGH SOURCE LIST CRACKING NAMES
                          3059
053.050  176              3060  BSL1    MOV     A,M
```

```
053.051  247           3061        ANA     A
053.052  310           3062        RZ                      ALL DONE
053.053  021 034 064   3063        LXI     D,DEFALT
053.056  315 016 054   3064        CALL    CAD             CONVERT ASCII NAME TO DIRECTORY FORMAT
053.061  330           3065        RC                      ERROR
053.062  315 005 057   3066        CALL    SND             SET NEW DEFAULTS
053.065  345           3067        PUSH    H               SAVE LINE ADDRESS
053.066  072 113 053   3068        LDA     BSLA
053.071  247           3069        ANA     A
053.072  304 114 053   3070        CNZ     CCW             CHECK FOR COMPLETE WILDCARD (*.*)
053.075  332 200 042   3071        JC      RESTART         USER CHICKENED OUT                /79.12.GC/
053.100  315 222 055   3072        CALL    EWS             EXPAND WILDCARD SPECIFICATION
053.103  341           3073  BSL2  POP     H               RESTORE LINE ADDRESS
053.104  330           3074        RC                      USER REFUSED *.*
053.105  315 370 056   3075        CALL    SFS             SKIP FILE SEPERATOR (BLANKS AND/OR COMMA)
053.110  303 050 053   3076        JMP     BSL1            DO MORE
                       3077
053.113  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
053.114  041 066 067   3097  CCW   LXI     H,PIO.DIR+DIR.NAM
000.000                3098        IF      .PIP.
053.117  006 013       3099        MVI     B,8+3
053.121  076 200       3100        MVI     A,200Q
053.123  246           3101  CCW1  ANA     M               SEE IF ALL HAVE 200Q BIT SET
053.124  043           3102        INX     H
053.125  005           3103        DCR     B
053.126  302 123 053   3104        JNZ     CCW1
053.131  247           3105        ANA     A
053.132  360           3106        RP                      NOT *.*
                       3107
                       3108  *     IS *.*
                       3109
053.133  315 136 031   3110        CALL    $TYPTX
053.136  007 041 077   3111        DB      BELL,'!?! DELETE ALL FILES ON',' '+200Q
053.167  041 063 067   3112        LXI     H,PIO.DEV
053.172  076 003       3113        MVI     A,3
```

```
053.174  315 057 057  3114            CALL     $TYPCC          TYPE DEVICE NAME
053.177  315 136 031  3115            CALL     $TYPTX
053.202  072 040 050  3116            DB       ': (Y/N)?',' '+200Q
053.213  041 063 064  3117            LXI      H,DESTBUF
053.216  315 155 057  3118            CALL     $RTL.           READ REPLY
053.221  072 063 064  3119            LDA      DESTBUF
053.224  376 131      3120            CPI      'Y'
053.226  310          3121            RE                       IS OK
053.227  067          3122            STC
053.230  076 205      3123            MVI      A,PEC.IUW       FLAG ILLEGAL USE OF WILDCARD
                      3124            ENDIF
053.232  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
053.233  345          3138  CFE       PUSH     H
053.234  076 016      3139            MVI      A,DIR.FLG
053.236  315 101 030  3140            CALL     $DADA.
053.241  176          3141            MOV      A,M             (A) = FLAG
053.242  346 200      3142            ANI      DIF.SYS
053.244  341          3143            POP      H
053.245  310          3144            RZ                       ELIGIBLE
053.246  072 351 063  3145            LDA      SYSTEM          CHECK /S FLAG
053.251  247          3146            ANA      A
053.252  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
053.253  052 150 047  3159  CFS       LHLD     LSTE
053.256  021 000 000  3160  CFS.      LXI      D,0
053.261  247          3161  CFS1      ANA      A
053.262  310          3162            RZ                       ALL DONE
053.263  157          3163            MOV      L,A
```

```
053.264  176          3164            MOV     A,M             (A) = NEXT GRT
053.265  023          3165            INX     D
053.266  303 261 053  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
053.271  315 201 056  3182  CTS       CALL    LSN             (HL) = ADDRESS OF FIRST SOURCE NAME
053.274  021 242 310  3183            LXI     D,-LINE
053.277  031          3184            DAD     D               (HL) == 0 IF NO '=' IN COMMAND LINE
053.300  175          3185            MOV     A,L
053.301  264          3186            ORA     H
053.302  310          3187            RZ                      NO TARGET FILE
053.303  076 203      3188            MVI     A,PEC.TFI       TARGET FILE ILLEGAL
053.305  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
053.306  032          3206  CWM       LDAX    D
053.307  247          3207            ANA     A
053.310  372 315 053  3208            JM      CWM1            IS MATCH
053.313  276          3209            CMP     M
053.314  300          3210            RNE                     NO MATCH
053.315  023          3211  CWM1      INX     D
053.316  043          3212            INX     H               ADVANCE ADDRESSES
053.317  005          3213            DCR     B
```

```
053.320  302 306 053  3214            JNZ     CWM             GO FOR MORE
053.323  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 UDPATED
                      3231  *                 'C' SET IF ERROR
                      3232  *                  (A) = CODE
                      3233  *         USES    ALL
                      3234
                      3235
053.324  021 136 067  3236  DDF       LXI     D,LINE
053.327  142          3237            MOV     H,D
053.330  153          3238            MOV     L,E             (HL) = COMMAND POINTER
053.331  032          3239  DDF1      LDAX    D
053.332  023          3240            INX     D
053.333  376 075      3241            CPI     '='
053.335  312 347 053  3242            JE      DDF2            HAVE A SOURCE FILE
053.340  247          3243            ANA     A
053.341  302 331 053  3244            JNZ     DDF1            MORE TO CHECK
053.344  041 376 053  3245  DDF1.0    LXI     H,DDFA          USE DEFAULT
                      3246
                      3247  *         (HL) = ADDRESS FOR NAME
                      3248
053.347  021 034 064  3249  DDF2      LXI     D,DEFALT
053.352  315 016 054  3250            CALL    CAD             CONVERT ASCII NAME TO DIRECTORY FORMAT
053.355  330          3251            RC                      ERROR
053.356  312 344 053  3252            JZ      DDF1.0          NO FILE NAME SPECIFIED, USE DEFAULT
053.361  176          3253            MOV     A,M
053.362  376 075      3254            CPI     '='
053.364  076 206      3255            MVI     A,PEC.IDF       ASSUME ILLEGAL DESTINATION FORMAT
053.366  067          3256            STC
053.367  300          3257            RNE                     MUST HAVE '='
                      3258
                      3259  *         HAVE NAME DECODED. EXPAND INTO DESTFB+FB.NAM
                      3260
053.370  041 007 064  3261            LXI     H,DESTFB+FB.NAM
000.000               3262            IF      .PIP.
053.373  303 131 055  3263            JMP     CDA             CONVERT DIRECTORY FORMAT TO ASCII FORMAT
                      3264            ELSE    ONECOPY
                      3265            CALL    CDA             CONVERT DIRECTORY FORMAT TO ASCII FORMAT
                      3266            PUSH    PSW             SAVE CODE
```

```
                              MVI     C,3
                              LXI     D,DDFB
                              LXI     H,DESTFB+FB.NAM
                              CALL    $COMP             SEE IF DEVICE IS SYO
                              JNE     DDF3              IS ERROR
                              POP     PSW
                              RET                       RETURN WITH 'C' CLEAR

                      DDF3    POP     PSW               ERROR, ILLEGAL DEVICE CODE
                              MVI     A,EC.DNS
                              STC
                              RET

                      DDFA    DB      'SYO:*.*=',0      DEFAULT TARGET FOR ONECOPY
                      DDFB    DB      'SYO'             REQUIRED DEVICE SPECIFICATION FOR ONECOPY
                              ELSE

053.376  124 124 072  DDFA    DB      'TT:PIPDEST.JGL=',0
                              ENDIF


                      **      CAD - CONVERT ASCII FILE NAME INTO DIRECTORY FORMAT.
                      *
                      *       CAD CRACKS AN ALPHANUMERIC FILE DESCRIPTION, OF THE FORM
                      *
                      *       DEV:NAME.EXT
                      *
                      *       INTO THE PIO.XXX FIELDS.
                      *
                      *       THE DEFAULT BLOCK DETERMINES THE VALUES FOR THE DEVICE AND EXTENSION
                      *       FIELDS, IF THEY ARE UNSPECIFIED. IF *CAD* IS ENTERED
                      *       AT *CAD*, AN UNSPECIFIED NAME FIELD IS RETURNED AS ZERO BYTES.
                      *       IF ENTERED AT *CAD.*, AN UNSPECIFIED NAME FIELD IS
                      *       RETURNED AS 200Q (MATCH-ONE) BYTES.
                      *
                      *       ENTRY   (DE) = POINT TO DEFAULT BLOCK
                      *               (HL) = POINTER TO TEXT
                      *       EXIT    'C' SET IF ERROR
                      *                (A) = ERROR CODE
                      *               'C' CLEAR IF OK
                      *                (HL) = POINTS PAST FILE NAME
                      *                'Z' SET IF NULL NAME
                      *                'Z' CLEAR IF NON-NULL
                      *                 PIO.DIR.NAM = NAME
                      *                 PIO.DIR.EXT = EXTENSION
                      *                 PIO.DEV = DEVICE CODE
                      *                 PIO.UNI = UNIT NUMBER (ASCII DIGIT)
                      *       USES    ALL


054.016  257          CAD     XRA     A                 SET TO NULLS
054.017  303 024 054          JMP     CADO

054.022  076 200      CAD.    MVI     A,200Q
```

```
054.024  345          3320  CAD0   PUSH    H
054.025  062 302 054  3321         STA     CADA              SAVE DEFAULT VALUE
                      3322
                      3323  *      SET DEFAULTS IN PIO.xxx
                      3324
054.030  041 063 067  3325         LXI     H,PIO.DEV
054.033  001 003 000  3326         LXI     B,3
054.036  315 252 030  3327         CALL    $MOVE             SET DEFALUT DEVICE
054.041  001 003 000  3328         LXI     B,3
054.044  041 076 067  3329         LXI     H,PIO.DIR+DIR.EXT
054.047  315 252 030  3330         CALL    $MOVE             SET DEFAULT EXTENSION
054.052  341          3331         POP     H
054.053  315 222 057  3332         CALL    $SOB              SKIP BLANKS
054.056  006 000      3333         MVI     B,0
054.060  376 077      3334         CPI     '?'
054.062  312 111 054  3335         JE      CAD1              IS '?'
054.065  376 052      3336         CPI     '*'
054.067  312 111 054  3337         JE      CAD1              IS '*'
054.072  376 056      3338         CPI     '.'
054.074  312 111 054  3339         JE      CAD1              IS '.'
054.077  376 101      3340         CPI     'A'
054.101  332 263 054  3341         JC      CAD4              NOT NAME
054.104  376 133      3342         CPI     'Z'+1
054.106  322 263 054  3343         JNC     CAD4              NOT NAME
                      3344
                      3345  *      HAVE ALPHA STRING. CRACK IT
                      3346
054.111  315 303 054  3347  CAD1   CALL    DNT               DECODE NEXT TOKEN
054.114  332 276 054  3348         JC      CAD5              ERROR
054.117  376 072      3349         CPI     ':'
054.121  302 166 054  3350         JNE     CAD2              NOT DEVICE
                      3351
                      3352  *      HAVE EXPLICIT DEVICE
                      3353
054.124  043          3354         INX     H                 SKIP ':'
054.125  076 003      3355         MVI     A,3
054.127  271          3356         CMP     C
054.130  332 276 054  3357         JC      CAD5              TOO MANY CHARACTERS
054.133  076 001      3358         MVI     A,PIO.UNI-PIO.DEV-1                      /2.0b/
054.135  271          3359         CMP     C                                        /2.0b/
054.136  322 276 054  3360         JNC     CAD5              Too Few characters     /2.0b/
                      3361
054.141  076 060      3362         MVI     A,'0'                                    /2.0b/
054.143  062 065 067  3363         STA     PIO.UNI           Assume Unit 0          /2.0b/
054.146  006 000      3364         MVI     B,0               BC = Move Count        /2.0b/
054.150  345          3365         PUSH    H                 SAVE (HL)
054.151  041 063 067  3366         LXI     H,PIO.DEV
054.154  315 252 030  3367         CALL    $MOVE             SET EXPLICIT DEVICE
054.157  341          3368         POP     H
054.160  315 303 054  3369         CALL    DNT               DECODE NEXT TOKEN
054.163  332 276 054  3370         JC      CAD5              ERROR
                      3371
                      3372  *      DECODE NAME
                      3373
054.166  001 010 000  3374  CAD2   LXI     B,8               (BC) = COUNT
054.171  345          3375         PUSH    H                 SAVE TEXT ADDR
```

```
                        3376
                        3377  *       SEE IF NAME IS  UNSPECIFIED
                        3378
054.172  041 066 067    3379          LXI     H,PIO.DIR+DIR.NAM
054.175  345            3380          PUSH    H                 SAVE ADDRESS OF DIR.NAM
054.176  315 252 030    3381          CALL    $MOVE             MOVE IN NAME
054.201  341            3382          POP     H                 (HL) = #PIO.DIR+DIR.NAM
054.202  176            3383          MOV     A,M
054.203  247            3384          ANA     A
054.204  302 222 054    3385          JNZ     CAD2.6            IS SPECIFIED
054.207  072 302 054    3386          LDA     CADA              (A) = FILL CHARACTER
054.212  016 010        3387          MVI     C,8               (C) = COUNT
054.214  167            3388  CAD2.4  MOV     M,A
054.215  043            3389          INX     H
054.216  015            3390          DCR     C
054.217  302 214 054    3391          JNZ     CAD2.4
054.222  341            3392  CAD2.6  POP     H
054.223  176            3393          MOV     A,M               (A) = DELIMITER
054.224  376 056        3394          CPI     '.'
054.226  302 261 054    3395          JNE     CAD3              NOT EXTENSION
                        3396
                        3397  *       HAVE EXPLICIT EXTENSION
                        3398
054.231  043            3399          INX     H
054.232  315 303 054    3400          CALL    DNT
054.235  332 276 054    3401          JC      CAD5              ERROR
054.240  076 003        3402          MVI     A,3
054.242  271            3403          CMP     C
054.243  332 276 054    3404          JC      CAD5              TOO LONG
054.246  001 003 000    3405          LXI     B,3
054.251  345            3406          PUSH    H                 SAVE TEXT POINTER
054.252  041 076 067    3407          LXI     H,PIO.DIR+DIR.EXT
054.255  315 252 030    3408          CALL    $MOVE             MOVE EXTENSION
054.260  341            3409          POP     H
                        3410
                        3411  *       DONE WITH NAME, MUST HAVE LEGIT DELIMITER
                        3412
054.261  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
054.263  315 222 057    3418  CAD4    CALL    $SOB              SKIP BLANKS
054.266  176            3419          MOV     A,M               (A) = NEXT CHARACTER
054.267  315 035 057    3420          CALL    $CFD              CHECK FILE NAME DELIMITER
054.272  330            3421          RC                        ERROR
054.273  170            3422          MOV     A,B
054.274  247            3423          ANA     A                 SET 'Z' IF NULL
054.275  311            3424          RET
                        3425
                        3426  *       ERROR
                        3427
054.276  076 007        3428  CAD5    MVI     A,EC.IFN          ILLEGAL FILE NAME
054.300  067            3429          STC
054.301  311            3430          RET
                        3431
```

```
054.302  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
054.303  021 015 055   3449  DNT      LXI     D,DNTA
054.306  016 011       3450           MVI     C,9               (C) = SIZE OF DNTA
054.310  101           3451           MOV     B,C               (B) = MAX ALLOWED +1
054.311  257           3452           XRA     A
054.312  022           3453  DNT1     STAX    D                 ZERO BUFFER
054.313  023           3454           INX     D
054.314  015           3455           DCR     C
054.315  302 312 054   3456           JNZ     DNT1
054.320  021 015 055   3457           LXI     D,DNTA
                       3458
                       3459  *        COPY CHARACTERS
                       3460
054.323  176           3461  DNT2     MOV     A,M
054.324  376 077       3462           CPI     '?'
054.326  076 200       3463           MVI     A,200Q
054.330  312 365 054   3464           JE      DNT3              IS MATCHONE
054.333  176           3465           MOV     A,M
054.334  376 052       3466           CPI     '*'
054.336  312 377 054   3467           JE      DNT5              IS WILDCARD
054.341  376 060       3468           CPI     '0'
054.343  332 010 055   3469           JC      DNT4              NOT ALPHANUMERIC
054.346  376 072       3470           CPI     '9'+1
054.350  332 365 054   3471           JC      DNT3              NUMERIC
054.353  376 101       3472           CPI     'A'
054.355  332 010 055   3473           JC      DNT4              DELIMITER
054.360  376 133       3474           CPI     'Z'+1
054.362  322 010 055   3475           JNC     DNT4              DELIMITER
                       3476
                       3477  *        HAVE GOOD CHARACTER
                       3478
054.365  022           3479  DNT3     STAX    D                 STORE CHAR
054.366  023           3480           INX     D
054.367  043           3481           INX     H
054.370  014           3482           INR     C                 COUNT
054.371  005           3483           DCR     B                 LIMIT DECREMENT
054.372  302 323 054   3484           JNZ     DNT2              NOT OVERFLOW
```

```
                          3485
                          3486  *       OVERFLOW
                          3487
054.375  067              3488          STC                         FLAG ERR
054.376  311              3489          RET
                          3490
                          3491  *       IS '*' WILDCARD
                          3492
054.377  076 200          3493  DNT5    MVI     A,200Q
055.001  022              3494          STAX    D
055.002  023              3495          INX     D
055.003  005              3496          DCR     B
055.004  302 377 054      3497          JNZ     DNT5                FILL WITH MATCH ONE
055.007  043              3498          INX     H                   SKIP '*'
                          3499
                          3500  *       END OF STRING
                          3501
055.010  247              3502  DNT4    ANA     A                   CLEAR 'C'
055.011  021 015 055      3503          LXI     D,DNTA              SET POINTER
055.014  311              3504          RET
                          3505
055.015                   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
055.026  052 320 040      3518  EBM     LHLD    S.SYSM
055.031  345              3519          PUSH    H
055.032  052 350 040      3520          LHLD    S.OFWA
055.035  021 006 000      3521          LXI     D,OVLO*OVL.ENS+OVL.FLB
055.040  031              3522          DAD     D                   (HL) = ADDR. OF OVLO OVL.FLB ENTRY
055.041  076 002          3523          MVI     A,OVL.RES
055.043  246              3524          ANA     M
055.044  021 010 000      3525          LXI     D,OVL.ENS
055.047  031              3526          DAD     D                   (HL) = ADDR. OF OVL1 OVL.FLB ENTRY
000.000                   3527          ERRNZ   OVL1-OVLO-1
055.050  246              3528          ANA     M
055.051  302 066 055      3529          JNZ     EBM1                OVLO AND OVL1 ARE PERM. RESIDENT
055.054  052 324 040      3530          LHLD    S.UMAX
055.057  315 224 030      3531          CALL    $CHL
055.062  353              3532          XCHG
055.063  341              3533          POP     H
055.064  031              3534          DAD     D                   (HL) = NEW ADDRESS SOUGHT
055.065  345              3535          PUSH    H
                          3536
055.066  341              3537  EBM1    POP     H
```

```
055.067  021 372 377  3538           LXI     D,-6
055.072  031          3539           DAD     D                 (HL) = NEW ADDRESS SOUGHT
055.073  377 052      3540           DB      SYSCALL,.SETTP
055.075  332 114 051  3541           JC      IERR1             INTERNAL ERROR 1
055.100  052 322 040  3542           LHLD    S.USRM
000.000               3543           IF      .FIP.
055.103  353          3544           XCHG
055.104  052 371 063  3545           LHLD    BUFPTR
055.107  315 224 030  3546           CALL    $CHL              (HL) = - BUFFER FWA
055.112  031          3547           DAD     D
055.113  056 000      3548           MVI     L,0
055.115  042 373 063  3549           SHLD    BUFSIZ
055.120  076 001      3550           MVI     A,BUFMINL/256-1
055.122  274          3551           CMP     H
055.123  330          3552           RC                        IF OK
055.124  076 021      3553           MVI     A,EC.NEM
055.126  303 325 051  3554           JMP     ERROR             NOT ENOUGH MEMORY
                      3555
                      3556           ELSE
                      3557
                      3558           MOV     A,H               (A) = LIMIT/256
                      3559           STA     OBUFLIM           SET LIMIT
                      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
055.131  001 000 003  3578  CDA      LXI     B,3*256           (B) = CHARACTER COUNT, (C) = WILDCARD FLAG
055.134  021 063 067  3579           LXI     D,PIO.DEV
055.137  315 175 055  3580           CALL    CDA5              COPY IT
055.142  066 072      3581           MVI     M,':'
055.144  043          3582           INX     H
055.145  006 010      3583           MVI     B,8
055.147  021 066 067  3584           LXI     D,PIO.DIR+DIR.NAM
055.152  315 175 055  3585           CALL    CDA5              COPY IT
055.155  066 056      3586           MVI     M,'.'
055.157  043          3587           INX     H
055.160  006 003      3588           MVI     B,3
000.000               3589           ERRNZ   DIR.EXT-DIR.NAM-8
055.162  315 175 055  3590           CALL    CDA5              COPY IT
```

```
055.165  066 000       3591            MVI     M,0                FLAG END OF NAME
055.167  171           3592            MOV     A,C                (A) (BIT 7) = 1 IF WILDCARDS
055.170  007           3593            RLC
055.171  057           3594            CMA
055.172  346 001       3595            ANI     1                  =0 IF WILDCARD
055.174  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
055.175  032           3612  CDA5      LDAX    D                  (A) = CHARACTER
055.176  261           3613            ORA     C
055.177  117           3614            MOV     C,A
055.200  032           3615            LDAX    D
055.201  023           3616            INX     D
055.202  247           3617            ANA     A
055.203  312 215 055   3618            JZ      CDA7               IS 00
055.206  362 213 055   3619            JP      CDA6               NOT 200Q
055.211  076 077       3620            MVI     A,'?'
055.213  167           3621  CDA6      MOV     M,A
055.214  043           3622            INX     H                  INCREMENT TO
055.215  005           3623  CDA7      DCR     B
055.216  302 175 055   3624            JNZ     CDA5               IF MORE TO GO
055.221  311           3625            RET


                       3627  **        EWS - EXPAND WILDCARD SPECIFICATION.
                       3628  *
                       3629  *         DWS ENTERS THE FILE NAME IN PIO.XXX INTO THE MANAGED TABLE
                       3630  *         NAMTAB. IF THE FILE NAME CONTAINS WILDCARDS, THE DIRECTORY
                       3631  *         IS READ FOR ELIGIBLE FILES.
                       3632  *
                       3633  *         ENTRY   PIO.XXX = FILE NAME
                       3634  *         EXIT    'C' CLEAR IF OK
                       3635  *                 'C' SET IF ERROR
                       3636  *         USES    ALL
                       3637
                       3638
055.222  315 347 052   3639  EWS       CALL    AEN                TRY TO ENTER IT
055.225  320           3640            RNC                        NO WILDCARDS, AM DONE
                       3641
                       3642  *         IS WILDCARD, LOOK UP DEVICE TYPE
```

```

055.226  052 030 064          LHLD    NAMTLEN
055.231  021 235 067          LXI     D,NAMTAB-FB.NAML
055.234  031                  DAD     D               (HL) = ADDRESS OF LAST ENTRY
055.235  315 016 054          CALL    CAD             CONVERT ASCII NAME TO DIRECTORY FORMAT
055.240  330                  RC                      ERROR
055.241  052 030 064          LHLD    NAMTLEN
055.244  021 357 377          LXI     D,-FB.NAML
055.247  031                  DAD     D
055.250  042 030 064          SHLD    NAMTLEN         REMOVE WILDCARD FROM TABLE
055.253  315 345 060          CALL    $MOVEL
055.256  003 000 063          DW      3,PIO.DEV,DIRNAM          SET DIRECTORY NAME IN XXX:DIRECT.SYS
055.264  315 345 060          CALL    $MOVEL
055.267  013 000 066          DW      8+3,PIO.DIR+DIR.NAM,EWSC          SAVE WILDCARD PATTERN
055.275  001 064 056          LXI     B,EWSB
055.300  041 352 063          LXI     H,DIRNAM
055.303  377 053              DB      SYSCALL,.DECODE GET INFORMATION ABOUT DEVICE
055.305  330                  RC                      ERROR
055.306  072 064 056          LDA     EWSB            SEE IF A DIRECTORY DEVICE
055.311  346 001              ANI     DT.DD
055.313  076 005              MVI     A,EC.DNS                  ASSUME DEVICE NOT SUITABLE
055.315  067                  STC
055.316  310                  RZ                      ERROR

                       *      IS DIRECTORY DEVICE. OPEN DIRECTORY

055.317  041 352 063          LXI     H,DIRNAM
055.322  076 002              MVI     A,CN.DIR
055.324  377 042              DB      SYSCALL,.OPENR
055.326  076 200              MVI     A,PEC.DF
055.330  330                  RC                      DEVICE FORMAT FAILURE

                       *      READ DIRECTORY ENTRYS FOR MATCH

055.331  315 135 056   EWS1   CALL    GDWP            DE = DIRECTORY WORKSPACE PTR    /79.11.GC/
055.334  001 000 002          LXI     B,512
055.337  076 002              MVI     A,CN.DIR
055.341  325                  PUSH    D               SAVE ADDRESS
055.342  377 004              DB      SYSCALL,.READ   READ BLOCK
055.344  341                  POP     H               (HL) = DIRECTORY ADDRESS
055.345  332 051 056          JC      EWS7            ALL DONE

                       *      LOOK AT DIRECTORY BLOCK FOR MATCHES

055.350  345                  PUSH    H
055.351  315 143 056          CALL    GDWP.                                           /79.11.GC/
055.354  315 353 057          CALL    $INDLB                                          /79.11.GC/
055.357  373 001              DW      DIS,ENL         A  = DIRECTORY ENTRY LENGTH     /79.11.GC/
055.361  341                  POP     H                                               /79.11.GC/
                                                                                      /79.11.GC/

055.362  117                  MOV     C,A             (C) = LENGTH

                       *      CHECK NEXT ENTRY

055.363  176           EWS3   MOV     A,M             (A) = 1ST CHAR THIS ENTRY
055.364  247                  ANA     A
```

```
055.365  312 331 055  3699          JZ      EWS1            END OF BLOCK
000.000               3700          ERRNZ   DF.EMP-377Q
055.370  074          3701          INR     A
055.371  312 043 056  3702          JZ      EWS6            ENTRY EMPTY
000.000               3703          ERRNZ   DF.CLR-376Q
055.374  074          3704          INR     A
055.375  312 051 056  3705          JZ      EWS7            END OF LIST
056.000  315 233 053  3706          CALL    CFE             CHECK FOR FILE ELIGIBILITY
056.003  302 043 056  3707          JNZ     EWS6            NOT TO PROCESS
056.006  345          3708          PUSH    H
056.007  021 122 056  3709          LXI     D,EWSC
056.012  006 013      3710          MVI     B,8+3
056.014  315 306 053  3711          CALL    CWM             CHECK WILDCARD MATCH
056.017  302 042 056  3712          JNZ     EWS4            NO MATCH
                      3713
                      3714 *        HAVE MATCH, ADD TO LSIT
                      3715
056.022  321          3716          POP     D               (DE) = FROM
056.023  325          3717          PUSH    D
056.024  305          3718          PUSH    B               SAVE (C)
056.025  001 013 000  3719          LXI     B,8+3
056.030  041 066 067  3720          LXI     H,PIO.DIR+DIR.NAM
056.033  315 252 030  3721          CALL    $MOVE
056.036  315 347 052  3722          CALL    AEN             ADD TO TABLE
056.041  301          3723          POP     B               RESTORE (C)
                      3724
                      3725 *        LOOKUP NEXT ENTRY
                      3726
056.042  341          3727 EWS4     POP     H
056.043  006 000      3728 EWS6     MVI     B,0
056.045  011          3729          DAD     B               POINT TO NEXT
056.046  303 363 055  3730          JMP     EWS3
                      3731
                      3732 *        ALL DONE, CLOSE DIRECTORY FILE
                      3733
056.051  076 002      3734 EWS7     MVI     A,CN.DIR
056.053  377 046      3735          DB      SYSCALL,.CLOSE
056.055  311          3736          RET
                      3737
056.056  123 131 060  3738 EWSA     DB      'SY0',200Q,200Q,200Q
                      3739
056.064               3740 EWSB     DS      30
                      3741
056.122               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 *
```

```
                              *       USES:   DE
                              *

056.135  353                  GDWP    XCHG
056.136  315 143 056                  CALL    GDWP.           HL = DIRECTORY WORKSPACE POINTER
056.141  353                          XCHG
056.142  311                          RET

056.143  052 121 041          GDWP.   LHLD    S.SCR           HL = SYSTEM SCRATCH
056.146  311                          RET



                              **      INA - INCREASE NAMTAB ALLOCATION.
                              *
                              *       INA IS CALLED TO INCREASE THE NAMTAB ALLOCATION. THE
                              *       BUFFER AREA IS MOVED UP TO MAKE ROOM.
                              *
                              *       ENTRY   NONE
                              *       EXIT    NONE
                              *       USES    A,F,H,L

056.147  041 033 064          INA     LXI     H,NAMTMAX+1
056.152  064                          INR     M               INCREMENT LENGTH
056.153  041 372 063                  LXI     H,BUFPTR+1
056.156  064                          INR     M               MOVE BUFFER
056.157  052 373 063                  LHLD    BUFSIZ
056.162  174                          MOV     A,H
056.163  265                          ORA     L
056.164  076 021                      MVI     A,EC.NEM        FLAG OUT OF MEMORY IF BUFFER NOT EMPTY
056.166  302 325 051                  JNZ     ERROR
056.171  305                          PUSH    B
056.172  325                          PUSH    D
056.173  315 322 056                  CALL    SBE             NOTIFY SYSTEM
056.176  321                          POP     D
056.177  301                          POP     B
056.200  311                          RET



                              **      LSN - LOCATE SOURCE NAME
                              *
                              *       LSN SCANS THE COMMAND LINE FOR THE FIRST SOURCE FILE NAME.
                              *
                              *       ENTRY   NONE
                              *       EXIT    (HL) = 1ST FILE NAME FWA
                              *       USES    A,F,H,L

056.201  041 136 067          LSN     LXI     H,LINE
056.204  176                  LSN1    MOV     A,M
056.205  043                          INX     H
056.206  376 075                      CPI     '='
056.210  310                          RE                      GOT IT
056.211  247                          ANA     A
```

```
056.212  302 204 056  3802          JNZ     LSN1               MORE LINE
056.215  041 136 067  3803          LXI     H,LINE  IS NO =
056.220  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
056.221  345          3824  MWN     PUSH    H                  SAVE TARGET ADDRESS
056.222  305          3825          PUSH    B                  SAVE WC PATTERN
056.223  353          3826          XCHG                       (HL) = MASTER NAME
056.224  315 016 054  3827          CALL    CAD                CONVERT TO DIRECTORY FORMAT
056.227  315 345 060  3828          CALL    $MOVEL
056.232  013 000 066  3829          DW      8+3,PIO.DIR,MWNA          (MWNA) = DECODED MASTER
056.240  341          3830          POP     H                  (HL) = WC PATTERN
056.241  315 016 054  3831          CALL    CAD                (PIO.DIR) = WC PATTERN
056.244  021 042 064  3832          LXI     D,MWNA             (DE) = MASTER PATTERN
056.247  041 066 067  3833          LXI     H,PIO.DIR          (DE) = WC PATTERN ADDRESS
056.252  016 013      3834          MVI     C,8+3              MERGE NAME AND EXTENSION
                      3835
                      3836  *       MERGE NAMES
                      3837
056.254  176          3838  MWN1    MOV     A,M                (A) = WC PATTERN
056.255  247          3839          ANA     A
056.256  362 262 056  3840          JP      MWN2               USE THIS
056.261  032          3841          LDAX    D                  IS MATCH CHARACTER, USE MASTER INSTEAD
056.262  167          3842  MWN2    MOV     M,A                STORE CHARACTER
056.263  023          3843          INX     D
056.264  043          3844          INX     H
056.265  015          3845          DCR     C
056.266  302 254 056  3846          JNZ     MWN1               MERGE TILL DONE
056.271  341          3847          POP     H                  (HL) = TARGET ADDRESS
056.272  303 131 055  3848          JMP     CDA                CONVERT DIRECTORY FORMAT TO ASCII
```

```
                                **      REN - REMOVE ENTRY FROM *NAMTAB*
                                *
                                *       REN REMOVES THE FIRST 'FB.NAML' BYTES FROM NAMTAB.
                                *
                                *       THE AMOUNT (FB.NAML) IS REMOVED FROM THE SIZE OF THE TABLE. THE
                                *       TABLE IS NOT CHECKED FOR UNDERFLOW, THE CALLER MUST GUARANTEE THE
                                *       PRESENSE OF AT LEAST FB.NAML BYTES IN NAMTAB.
                                *
                                *       ENTRY   NONE
                                *       EXIT    NONE
                                *       USES    ALL


056.275  052 030 064            REN     LHLD    NAMTLEN
056.300  021 357 377                    LXI     D,-FB.NAML
056.303  031                            DAD     D                       REMOVE COUNT FROM LEN
056.304  042 030 064                    SHLD    NAMTLEN
056.307  104                            MOV     B,H
056.310  115                            MOV     C,L                     (BC) = REMAINING LENGTH
056.311  021 277 067                    LXI     D,NAMTAB+FB.NAML                (DE) = START OF 2ND ENTRY
056.314  041 256 067                    LXI     H,NAMTAB
056.317  303 252 030                    JMP     $MOVE                   MOVE DOWN AND RETURN



                                **      SBE - SET BUFFER EMPTY.
                                *
                                *       THE SYSTEM IS NOTIFIED.
                                *
                                *       ENTRY   NONE
                                *       EXIT    NONE
                                *       USES    ALL


056.322  041 000 000            SBE     LXI     H,0
056.325  042 373 063                    SHLD    BUFSIZ
056.330  052 371 063                    LHLD    BUFPTR                  (HL) = BUFFER FWA (AND LWA!)
056.333  043                            INX     H
056.334  043                            INX     H
056.335  377 052                        DB      SYSCALL,.SETTP
056.337  320                            RNC                             OK
056.340  303 325 051                    JMP     ERROR                   NOT ENOUGH ROOM



                                **      SDD - SET DEFAULT DEFAULT.
                                *
                                *       SDD IS CALLED TO SETUP THE CURRENT DEFAULT DEVICE
                                *       AND EXTENSION TO 'SY0' AND <NULL>, RESPECTIVELY.
                                *
                                *       ENTRY   NONE
                                *       EXIT    NONE
                                *       USES    NONE

```

```
                         3900
056.343  315 054 031     3901  SDD     CALL    $SAVALL
056.346  315 345 060     3902          CALL    $MOVEL
056.351  006 000 362     3903          DW      6,SDDA,DEFALT   SET DEFAULT DEFAULT
056.357  303 047 031     3904          JMP     $RSTALL         RESTORE AND RETURN
                         3905
056.362  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
056.370  315 222 057     3924  SFS     CALL    $SOB            SKIP BLANKS
056.373  176             3925          MOV     A,M
056.374  376 054         3926          CPI     ','
056.376  302 002 057     3927          JNE     SFS1            NOT ,
057.001  043             3928          INX     H               SKIP ,
057.002  303 222 057     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
057.005  315 054 031     3943  SND     CALL    $SAVALL         SAVE REGS
000.000                  3944          ERRNZ   PIO.UNI-PIO.DEV-2
057.010  315 345 060     3945          CALL    $MOVEL
057.013  003 000         3946          DW      3
057.015  063 067         3947          DW      PIO.DEV
057.017  034 064         3948          DW      DEFALT
057.021  315 345 060     3949          CALL    $MOVEL
```

```
057.024  003 000       3950           DW      3
057.026  076 067       3951           DW      PIO.DIR+DIR.EXT
057.030  037 064       3952           DW      DEFALT+3
057.032  303 047 031   3953           JMP     $RSTALL           RETURN
```

```
  057.035                 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
  057.035  247            3972X $CFD     ANA     A
  057.036  310            3973X          RZ                      IS 00
  057.037  376 054        3974X          CPI     ','
  057.041  310            3975X          RE                      IS ,
  057.042  376 075        3976X          CPI     '='
  057.044  310            3977X          RE                      IS =
  057.045  376 057        3978X          CPI     '/'
  057.047  310            3979X          RE                      IS /
  057.050  376 040        3980X          CPI     ' '
  057.052  310            3981X          RE                      IS ' '
  057.053  076 007        3982X          MVI     A,EC.IFN        ILLEGAL FILE NAME
  057.055  067            3983X          STC
  057.056  311            3984X          RET
  057.057                 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
  057.057                 3998X $TYPCC   EQU     *
  057.057  247            3999X          ANA     A
  057.060  310            4000X          RZ                      NOTHING TO TYPE
  057.061  365            4001X          PUSH    PSW             SAVE COUNT
  057.062  176            4002X          MOV     A,M             (A) = CHARACTER
  057.063  043            4003X          INX     H
  057.064  377 002        4004X          DB      SYSCALL,.SCOUT
  057.066  361            4005X          POP     PSW
```

```
     057.067  075           4006X        DCR      A
     057.070  303 057 057   4007X        JMP      $TYPCC
     057.073                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
     057.073                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
     057.073                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
057.073               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
057.073  315 054 031  4072X $CCO    CALL    $SAVALL         SAVE REGISTERS
057.076  076 004      4073X         MVI     A,I.CONFL
057.100  001 001 000  4074X         LXI     B,CO.FLG        CLEAR CO.FLG
057.103  377 006      4075X         DB      SYSCALL,.CONSL
057.105  303 047 031  4076X         JMP     $RSTALL         RESTORE REGISTERS AND RETURN
057.110               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
057.110  076 002      4089X $GNL    MVI     A,I.CUSOR
057.112  001 000 000  4090X         LXI     B,0
057.115  377 006      4091X         DB      SYSCALL,.CONSL          READ CURSOR
057.117  075          4092X         DCR     A
057.120  310          4093X         RZ                      AT COLUMN 1
057.121  303 271 057  4094X         JMP     $CRLF           NEW LINE
057.124               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
    057.124  365         4106X $MLU     PUSH     PSW                SAVE (PSW)
    057.125  345         4107X          PUSH     H                  SAVE FWA
    057.126  053         4108X          DCX      H                  ANTICIPATE INX H
    057.127  043         4109X $MLU1    INX      H
    057.130  176         4110X          MOV      A,M                (A)= CHARACTER
    057.131  315 144 057 4111X          CALL     $MCU               MAP CHAR TO UPPER
    057.134  167         4112X          MOV      M,A
    057.135  247         4113X          ANA      A
    057.136  302 127 057 4114X          JNZ      $MLU1              MORE TO GO
    057.141  341         4115X          POP      H                  RESTORE (HL)
    057.142  361         4116X          POP      PSW                RESTORE (PSW)
    057.143  311         4117X          RET
    057.144              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
    057.144  376 141     4130X $MCU     CPI      'a'
    057.146  330         4131X          RC                          NOT LOWER CASE
    057.147  376 173     4132X          CPI      'z'+1
    057.151  320         4133X          RNC                         NOT LOWER CASE
    057.152  326 040     4134X          SUI      'a'-'A'
    057.154  311         4135X          RET
    057.155              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
057.155  315 164 057     4154X $RTL.   CALL    $RTL               $RTL IN UPPER CASE
057.160  330             4155X         RC                         CTL-D
057.161  303 124 057     4156X         JMP     $MLU               MAP LINE TO UPPER CASE
                         4157X
057.164                  4158X $RTL    EQU     *
057.164  345             4159X         PUSH    H                  SAVE FWA
057.165  315 370 060     4160X $RTL1   CALL    $RCHAR
057.170  376 004         4161X         CPI     CTLD
057.172  312 217 057     4162X         JE      $RTL2              CTL-D STRUCK
057.175  167             4163X         MOV     M,A
057.176  043             4164X         INX     H
057.177  376 012         4165X         CPI     NL
057.201  302 165 057     4166X         JNE     $RTL1
057.204  053             4167X         DCX     H
057.205  066 000         4168X         MVI     M,0
057.207  043             4169X         INX     H
                         4170X
                         4171X *       ALL DONE. COMPUTE LENGTH
                         4172X
057.210  353             4173X         XCHG                       (DE) = LWA+1
057.211  343             4174X         XTHL                       (HL) = FWA
057.212  173             4175X         MOV     A,E
057.213  225             4176X         SUB     L                  (A) = LENGTH
057.214  247             4177X         ANA     A                  CLEAR CARRY
057.215  321             4178X         POP     D                  RESTORE (DE)
057.216  311             4179X         RET
                         4180X
                         4181X *       CTL-D STRUCK
                         4182X
057.217  341             4183X $RTL2   POP     H                  (HL) = FWA
057.220  067             4184X         STC
057.221  311             4185X         RET
057.222                  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
057.222                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
057.222                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
057.222  053           4234X $SOB    DCX     H               PRE-DECREMENT
057.223  043           4235X $SOB1   INX     H
057.224  176           4236X         MOV     A,M
057.225  376 040       4237X         CPI     ' '
057.227  312 223 057   4238X         JE      $SOB1           GOT BLANK
057.232  376 011       4239X         CPI     TAB
057.234  312 223 057   4240X         JE      $SOB1           GOT TAB
057.237  311           4241X         RET
057.240                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
057.240  305                    4262X $TBLS    PUSH     B
057.241  376 000                4263X          CPI      0                                              /78.10.GC/
057.243  312 265 057            4264X          JZ       TBL2                                           /78.10.GC/
057.246  107                    4265X          MOV      B,A
057.247  176                    4266X TBL1     MOV      A,M             (A) = CHARACTER
057.250  043                    4267X          INX      H
057.251  270                    4268X          CMP      B
057.252  312 267 057            4269X          JZ       TBL3            IF MATCH
057.255  247                    4270X          ANA      A
057.256  043                    4271X          INX      H               SKIP PAST
057.257  302 247 057            4272X          JNZ      TBL1            IF NOT END OF TABLE
057.262  053                    4273X          DCX      H
057.263  053                    4274X          DCX      H
057.264  257                    4275X          XRA      A               SET TO ZERO FOR OLD USERS      /78.10.GC/
057.265  376 001                4276X TBL2     CPI      1               CLEAR ZERO                     /78.10.GC/
                                4277X
                                4278X *        DONE
                                4279X
057.267  301                    4280X TBL3     POP      B
057.270  311                    4281X          RET
057.271                         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
057.271                         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
 057.271              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
 057.271  076 012     4327X $CRLF    MVI     A,NL
 057.273  377 002     4328X          DB      SYSCALL,.SCOUT
 057.275  257         4329X          XRA     A
 057.276  311         4330X          RET
 057.277              4331           XTEXT   TYPCH


                      4333X **       $TYPCH - TYPE SINGLE CHARACTER.
                      4334X *
                      4335X *        ENTRY   (RET) = CHARACTER
                      4336X *        EXIT    TO (RET)+1
                      4337X *                (A) = CHARACTER TYPED
                      4338X
                      4339X
 057.277  343         4340X $TYPCH   XTHL                    (HL) = RETURN ADDRESS
 057.300  176         4341X          MOV     A,M             (A) = CHARACTER
 057.301  043         4342X          INX     H
 057.302  343         4343X          XTHL                    RESTORE ADVANCED EXIT ADDRESS
                      4344X
                      4345X **       $TYPC. - TYPE SINGLE CHARACTER.
                      4346X *
                      4347X *       ENTRY   (A) = CHARACTER
                      4348X *       EXIT    TO (RET)
                      4349X
057.303  377 002      4350X $TYPC.  DB      SYSCALL,.SCOUT
057.305  311          4351X         RET
000.001               4352  $CMP$   EQU     1
057.306               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
057.306  343          4371X $TYPLN. XTHL                     (H,L) = COUNT ADDRESS
057.307  176          4372X         MOV     A,M              (A) = COUNT
057.310  043          4373X         INX     H                (H,L)  = TEXT ADDRESS
057.311  345          4374X         PUSH    H                SAVE TEXT FWA
057.312  315 072 030  4375X         CALL    $DADA            CALCULATE RETURN ADDRESS
057.315  343          4376X         XTHL                     (HL) = TEXT ADDRE
057.316  315 324 057  4377X         CALL    $TYPL.           OUTPUT LINE
057.321  341          4378X         POP     H                (HL) = RETURN ADDRESS
057.322  343          4379X         XTHL                     RESTORE (HL), SET RETURN ADDRESS
057.323  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
057.324               4389X $TYPL.  EQU     *
057.324  247          4390X         ANA     A
057.325  310          4391X         RZ                       NOTHING TO TYPE
057.326  365          4392X         PUSH    PSW              SAVE COUNT
057.327  176          4393X         MOV     A,M              (A) = CHARACTER
057.330  043          4394X         INX     H
057.331  247          4395X         ANA     A
000.001               4396X         IF      $CMP$            IF HAVE COMPRESSED SPACES
                      4397X         JM      TPL2             IS COMPRESSED SPACE
                      4398X         ENDIF
```

```
057.332  314 271 057  4399X           CZ       $CRLF
057.335  315 303 057  4400X           CALL     $TYPC.             TYPE CHARACTER
057.340  361          4401X TPL1      POP      PSW
057.341  075          4402X           DCR      A
057.342  302 324 057  4403X           JNZ      $TYPL.
057.345  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
057.346               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
057.346               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
        057.346              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
        057.346              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
        057.346           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
        057.346           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
        057.346           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
```

```
    030.101                 4539X $DADA.  EQU      30101A          IN H17 ROM
    057.346                 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
    057.346                 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
    057.346  136            4566X ILDEHL  MOV      E,M
    057.347  043            4567X         INX      H
    057.350  126            4568X         MOV      D,M
    057.351  043            4569X         INX      H
    057.352  311            4570X         RET
    057.353                 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
       057.353                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
       057.353  353           4602X $INDLB   XCHG                       DE = BASE
       057.354  343           4603X          XTHL                       SAVE  .DE.
       057.355  325           4604X          PUSH    D                  SAVE  BASE
       057.356  305           4605X          PUSH    B                  SAVE  .BC.
                              4606X
       057.357  116           4607X          MOV     C,M
       057.360  043           4608X          INX     H
       057.361  106           4609X          MOV     B,M                BC = OFFSET
       057.362  043           4610X          INX     H                  HL =  .RET.
                              4611X
       057.363  353           4612X          XCHG                       HL = BASE
       057.364  011           4613X          DAD     B                  HL = BASE + OFFSET
       057.365  176           4614X          MOV     A,M                A  = ( BASE + OFFSET )
       057.366  353           4615X          XCHG                       HL =  .RET.
                              4616X
       057.367  301           4617X          POP     B                  RESTORE  .BC.
       057.370  321           4618X          POP     D                  RESTORE  BASE
       057.371  343           4619X          XTHL                       HL = .DE. ; (SP) = .RET.
       057.372  353           4620X          XCHG                       DE = .DE. ; HL = BASE
       057.373  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
       057.374  315 001 061   4635X $INDS    CALL    XCHGBC
```

```
057.377  343           4636X           XTHL                  SAVE   .BC.
060.000  325           4637X           PUSH    D
060.001  315 346 057   4638X           CALL    ILDEHL        DE = OFFSET
060.004  315 001 061   4639X           CALL    XCHGBC        BC = .RET.
060.007  353           4640X           XCHG                  DE = BASE ; HL = OFFSET
060.010  031           4641X           DAD     D             HL = BASE + OFFSET
060.011  353           4642X           XCHG
060.012  343           4643X           XTHL                  SAVE  BASE
060.013  353           4644X           XCHG                  DE = VALUE
060.014  315 051 060   4645X           CALL    ISDEHL
060.017  341           4646X           POP     H             HL = BASE
060.020  315 001 061   4647X           CALL    XCHGBC
060.023  343           4648X           XTHL                  RESTORE   .BC.
060.024  315 001 061   4649X           CALL    XCHGBC
060.027  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
060.030  353           4665X $INDSB    XCHG                  DE = BASE
060.031  343           4666X           XTHL                  SAVE   .DE.
060.032  325           4667X           PUSH    D             SAVE  BASE
060.033  305           4668X           PUSH    B             SAVE   .BC.
                       4669X
060.034  116           4670X           MOV     C,M
060.035  043           4671X           INX     H
060.036  106           4672X           MOV     B,M           BC = OFFSET
060.037  043           4673X           INX     H             HL =   .RET.
                       4674X
060.040  353           4675X           XCHG                  HL = BASE
060.041  011           4676X           DAD     B             HL = BASE + OFFSET
060.042  167           4677X           MOV     M,A           ( BASE + OFFSET ) = A
060.043  353           4678X           XCHG
                       4679X
060.044  301           4680X           POP     B             RESTORE   .BC.
060.045  321           4681X           POP     D             RESTORE  BASE
060.046  343           4682X           XTHL                  HL = .DE. ; (SP) = .RET.
060.047  353           4683X           XCHG                  DE = .DE. ; HL = BASE
060.050  311           4684X           RET
060.051                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
 060.051  163             4701X ISDEHL  MOV     M,E
 060.052  043             4702X         INX     H
 060.053  162             4703X         MOV     M,D
 060.054  043             4704X         INX     H
 060.055  311             4705X         RET
 060.056                  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
 060.056  172             4730X $DAD    MOV     A,D                                            /80.08.sc/
 060.057  263             4731X         ORA     E                                              /80.08.sc/
 060.060  312 204 060     4732X         JZ      DAD2            No-Date                        /80.08.sc/
                          4733X
 060.063  102             4734X         MOV     B,D
 060.064  113             4735X         MOV     C,E
 060.065  021 040 000     4736X         LXI     D,32
 060.070  345             4737X         PUSH    H               SAVE ADDRESS
 060.071  315 106 030     4738X         CALL    $DU66           (DE) = DAY, (HL) = YEAR & MONTH
 060.074  343             4739X         XTHL                    (HL) = ADDRESS
```

```
060.075  102          4740X         MOV     B,D
060.076  113          4741X         MOV     C,E
060.077  173          4742X         MOV     A,E
060.100  247          4743X         ANA     A
060.101  312 201 060  4744X         JZ      DAD1            BAD VALUE
060.104  076 002      4745X         MVI     A,2
060.106  315 157 031  4746X         CALL    $UDD            UNPACK DAY
060.111  066 055      4747X         MVI     M,'-'
060.113  043          4748X         INX     H
060.114  301          4749X         POP     B               (BC) = YEAR & MONTH
060.115  021 020 000  4750X         LXI     D,16
060.120  345          4751X         PUSH    H               SAVE ADDRESS
060.121  315 106 030  4752X         CALL    $DU66
060.124  343          4753X         XTHL                    (HL) = ADDRESS, ((SP)) = YEAR
060.125  173          4754X         MOV     A,E
060.126  207          4755X         ADD     A
060.127  203          4756X         ADD     E               (A) = 3*MONTH
060.130  312 201 060  4757X         JZ      DAD1            BAD VALUE
060.133  376 047      4758X         CPI     13*3
060.135  322 201 060  4759X         JNC     DAD1            TOO LARGE
060.140  353          4760X         XCHG                    (DE) = ADDRESS
060.141  041 212 060  4761X         LXI     H,DADB-3
060.144  315 101 030  4762X         CALL    $DADA.          (HL) = ADDRESS OF MONTH
060.147  001 003 000  4763X         LXI     B,3
060.152  353          4764X         XCHG                    (HL) = BUFFER ADDR, (DE) = ADDR IN 'DADB'
060.153  315 252 030  4765X         CALL    $MOVE           MOVE MONTH IN
060.156  066 055      4766X         MVI     M,'-'
060.160  043          4767X         INX     H
060.161  301          4768X         POP     B               (BC) = YEAR
060.162  171          4769X         MOV     A,C
060.163  306 106      4770X         ADI     70
060.165  376 144      4771X         CPI     100
060.167  077          4772X         CMC
060.170  330          4773X         RC                      TOO LARGE
060.171  117          4774X         MOV     C,A             (BC) = YEAR
060.172  076 002      4775X         MVI     A,2
060.174  315 157 031  4776X         CALL    $UDD            UNPACK YEAR
060.177  247          4777X         ANA     A
060.200  311          4778X         RET
                      4779X
                      4780X *       ILLEGAL FORMAT. (NOT ALL ILLEGALS EXIT HERE!)
                      4781X
060.201  341          4782X DAD1    POP     H               RESTORE STACK
060.202  067          4783X         STC                     FLAG ERROR
060.203  311          4784X         RET
                      4785X
                      4786X *       No-Date                                                  /80.08.sc/
                      4787X
060.204  001 011 000  4788X DAD2    LXI     B,DADCL                                          /80.08.sc/
060.207  021 261 060  4789X         LXI     D,DADC                                           /80.08.sc/
060.212  303 252 030  4790X         JMP     $MOVE                                            /80.08.sc/
                      4791X
060.215  112 141 156  4792X DADB    DB      'JanFebMarAprMayJunJulAugSepOctNovDec'
                      4793X
060.261  040 116 157  4794X DADC    DB      ' No-Date '                                      /80.08.sc/
000.011               4795X DADCL   EQU     *-DADC                                           /80.08.sc/
```

```
     060.272                   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
     060.272                   4810X $UDDN   EQU      *
     060.272  315 072 030      4811X         CALL     $DADA
     060.275  345              4812X         PUSH     H                    SAVE FINAL (H,L) VALUE
                               4813X
     060.276  365              4814X UDDN1   PUSH     PSW
     060.277  345              4815X         PUSH     H
     060.300  021 012 000      4816X         LXI      D,10
     060.303  315 106 030      4817X         CALL     $DU66                (H,L) = VALUE/10
     060.306  104              4818X         MOV      B,H
     060.307  115              4819X         MOV      C,L                  (BC) = QUOTIENT
     060.310  341              4820X         POP      H
     060.311  076 060          4821X         MVI      A,'0'
     060.313  203              4822X         ADD      E                    ADD REMAINDER
     060.314  053              4823X         DCX      H
     060.315  167              4824X         MOV      M,A                  STORE DIGIT
     060.316  170              4825X         MOV      A,B
     060.317  261              4826X         ORA      C
     060.320  312 332 060      4827X         JZ       UDDN2                ALL ZEROS
     060.323  361              4828X         POP      PSW
     060.324  075              4829X         DCR      A
     060.325  302 276 060      4830X         JNZ      UDDN1                IF MORE TO GO
                               4831X
                               4832X *       ALL DONE, EXIT
                               4833X
     060.330  341              4834X UDDN1.5 POP      H                    RESTORE H
     060.331  311              4835X         RET                              RETURN
                               4836X
                               4837X *       DIGITS LEADING THIS ONE ARE ZERO. STORE NULLS INSTEAD.
                               4838X
     060.332  361              4839X UDDN2   POP      PSW
     060.333  075              4840X UDDN3   DCR      A
     060.334  312 330 060      4841X         JE       UDDN1.5              ALL DONE
     060.337  053              4842X         DCX      H
     060.340  066 000          4843X         MVI      M,0
     060.342  303 333 060      4844X         JMP      UDDN3
     060.345                   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
     060.345  341             4874X $MOVEL  POP     H               (HL) = RET
     060.346  116             4875X         MOV     C,M
     060.347  043             4876X         INX     H
     060.350  106             4877X         MOV     B,M             (BC) = COUNT
     060.351  043             4878X         INX     H
     060.352  136             4879X         MOV     E,M
     060.353  043             4880X         INX     H
     060.354  126             4881X         MOV     D,M             (DE) = FROM
     060.355  043             4882X         INX     H
     060.356  325             4883X         PUSH    D               ((SP)) = FROM
     060.357  136             4884X         MOV     E,M
     060.360  043             4885X         INX     H
     060.361  126             4886X         MOV     D,M             (DE) = TO
     060.362  043             4887X         INX     H
     060.363  343             4888X         XTHL                    ((SP)) = RET, (HL) = FROM
     060.364  353             4889X         XCHG                    (DE) = FROM , (HL) = TO
     060.365  303 252 030     4890X         JMP     $MOVE           MOVE IT
     060.370                  4891          XTEXT   RCHAR


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

```
      060.370  377 001         4900X $RCHAR  DB      SYSCALL,.SCIN
      060.372  332 370 060     4901X         JC      $RCHAR            NOT READY
      060.375  311             4902X         RET
                               4903X
      060.376  377 002         4904X $WCHAR  DB      SYSCALL,.SCOUT
      061.000  311             4905X         RET
      061.001                  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
      061.001  365             4921X XCHGBC  PUSH    PSW
      061.002  170             4922X         MOV     A,B
      061.003  104             4923X         MOV     B,H
      061.004  147             4924X         MOV     H,A
      061.005  171             4925X         MOV     A,C
      061.006  115             4926X         MOV     C,L
      061.007  157             4927X         MOV     L,A
      061.010  361             4928X         POP     PSW
      061.011  311             4929X         RET
      061.012                  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 *
```

```
                          *        DB      'Y...Y'          NEXT SWITCH
                          *        .         .
                          *        .         .
                          *        .         .
                          *
                          *        DB      0                FLAGS END OF TABLE
                          *
                          *        SWITCHES MUST BE FOLLOWED BY A ':', A '/' (ANOTHER SWITCH)
                          *        A ',', OR A 00 BYTE.
                          *
                          *        UPON DETECTION OF A VALID SWITCH, $DRS CALLS THE USER PROCESS
                          *        ROUTINE. UPON ENTRY,
                          *        (HL) = ADDRESS OF THE FIRST BYTE FOLLOWING THE SWTICH
                          *        'Z' CLEAR IF CHARACTER = '/', ',', OR 00
                          *        'Z' SET IF CHARACTER = ':'
                          *
                          *        THE USER ROUTINE CAN DECODE SWITCH SUB-OPTIONS, IF DESIRED.
                          *        THE USER ROUTINE MAY USE ALL REGISTERS.
                          *
                          *        ENTRY   (DE) = SWITCH TABLE FWA
                          *                (HL) = LINE FWA
                          *        EXIT    'C' CLEAR IF OK
                          *                'C' SET IF ERROR
                          *                 (HL) = ADDRESS OF START OF BAD SWITCH
                          *                 (A) = ERROR CODE
                          *        USES    ALL


061.012                   $DRS     EQU     *

                          *        LOOK FOR SWITCHES

061.012  176              $DRS1    MOV     A,M
061.013  247                       ANA     A
061.014  310                       RZ                       END OF LINE
061.015  043                       INX     H
061.016  376 057                   CPI     '/'
061.020  302 012 061               JNE     $DRS1            NOT A SWITCH
061.023  042 207 061               SHLD    $DRSB            ($DRSB) = SWITCH FWA (AFTER '/')

                          *        GOT A SWITCH. LOOK FOR A MATCH IN THE CALLER'S TABLE

061.026  325                       PUSH    D                SAVE TABLE FWA
061.027  052 207 061      $DRS2    LHLD    $DRSB            (HL) = SWITCH FWA
061.032  032              $DRS3    LDAX    D                (A) = TABLE ENTRY
061.033  346 177                   ANI     177Q
061.035  312 105 061               JZ      $DRS6            GOT A MATCH
061.040  276                       CMP     M
061.041  302 051 061               JNE     $DRS4            NO MATCH
061.044  023                       INX     D
061.045  043                       INX     H
061.046  303 032 061               JMP     $DRS3            SEE IF MORE MATCH

                          *        HAVE MIS-MATCH. SEE IF THE MISSING CHARACTER IS SIGNIFICANT

061.051  176              $DRS4    MOV     A,M              (A) = LINE CHARACTER WE COULDNT MATCH
```

```
061.052  315 156 061  5006X          CALL     $DRS15           SEE IF OK TERMINATOR
061.055  302 065 061  5007X          JNE      $DRS4.5          NO MATCH ON THIS SWITCH
061.060  032          5008X          LDAX     D                (A) = NEXT CHARACTER IN SWITCH PATTERN
061.061  247          5009X          ANA      A
061.062  372 105 061  5010X          JM       $DRS6            HAVE SUFFICIENT MATCH
061.065  315 171 061  5011X $DRS4.5  CALL     $DRS20           SKIP TABLE ENTRY
061.070  032          5012X          LDAX     D
061.071  247          5013X          ANA      A
061.072  302 027 061  5014X          JNZ      $DRS2            MORE SWITCHES IN TABLE TO CHECK
                      5015X
                      5016X *        BAD SWITCH
                      5017X
061.075  321          5018X $DRS5    POP      D                RESTORE STACK
061.076  052 207 061  5019X          LHLD     $DRSB            POINT TO BAD SWITCH
061.101  067          5020X          STC
061.102  076 032      5021X          MVI      A,EC.IS          ILLEGAL SWITCH
061.104  311          5022X          RET
                      5023X
                      5024X *        HAVE SWITCH. CHECK IT'S FOLLOWING CHARACTER
                      5025X
061.105  315 222 057  5026X $DRS6    CALL     $SOB             SKIP OVER BLANKS
061.110  176          5027X          MOV      A,M
061.111  315 156 061  5028X          CALL     $DRS15           CHECK CHARACTER
061.114  302 075 061  5029X          JNE      $DRS5            IN ERROR
061.117  315 171 061  5030X          CALL     $DRS20           GET PROCESSOR ADDRESS
061.122  021 134 061  5031X          LXI      D,$DRS7
061.125  345          5032X          PUSH     H                SAVE (HL)
061.126  325          5033X          PUSH     D                SET RETURN ADDRESS FOR TABLE CODE
061.127  305          5034X          PUSH     B                SAVE PROCESSOR ADDRESS
061.130  176          5035X          MOV      A,M              (A) = NEXT CHARACTER
061.131  376 072      5036X          CPI      ':'              SET CONDITION CODES
061.133  311          5037X          RET                       CALL USER PROCESS
                      5038X
                      5039X *        USER PROCESS RETURNS HERE
                      5040X
061.134  321          5041X $DRS7    POP      D                (DE) = LAST CHARACTER OF SWITCH+1
061.135  052 207 061  5042X          LHLD     $DRSB            (HL) = FIRST CHARACTER OF SWITCH AFTER /
061.140  053          5043X          DCX      H                (HL) = ADDRESS OF '/'
                      5044X
                      5045X *        REPLACE SWITCH WITH BLANKS
                      5046X
061.141  066 040      5047X $DRS8    MVI      M,' '
061.143  043          5048X          INX      H
061.144  315 216 030  5049X          CALL     $CDEHL
061.147  302 141 061  5050X          JNE      $DRS8            NOT THERE YET
061.152  321          5051X          POP      D                (DE) = SWITCH TABLE FWA
061.153  303 012 061  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
     061.156  247                5064X $DRS15   ANA     A
     061.157  310                5065X          RZ                      IS 00
     061.160  376 057            5066X          CPI     '/'
     061.162  310                5067X          RE
     061.163  376 054            5068X          CPI     ','
     061.165  310                5069X          RE
     061.166  376 072            5070X          CPI     ':'
     061.170  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
     061.171  032                5086X $DRS20   LDAX    D
     061.172  023                5087X          INX     D
     061.173  376 200            5088X          CPI     200Q
     061.175  302 171 061        5089X          JNE     $DRS20
     061.200  032                5090X          LDAX    D               (A) = LOW BYTE OF PROCESSOR ADDRESS
     061.201  117                5091X          MOV     C,A
     061.202  023                5092X          INX     D
     061.203  032                5093X          LDAX    D
     061.204  107                5094X          MOV     B,A             (BC) = PROCESSOR ADDRESS
     061.205  023                5095X          INX     D
     061.206  311                5096X          RET
                                 5097X
     061.207  000 000            5098X $DRSB    DW      0               POINTER TO SWITCH BEING PROCESSED
     000.000                     5099           IF      .PIP.
     061.211                     5100           XTEXT   DTB
```

```
                              5102X **       $DTB - DELETE TRAILING BLANKS.
                              5103X *
                              5104X *        $DTB DELETES THE TRAILING BLANKS FROM A CODED LINE.
                              5105X *
                              5106X *        ENTRY    (HL) = LINE FWA
                              5107X *        EXIT     (A) = LENGTH OF RESULT (ENCLUDING 00 TERMINATOR BYTE)
                              5108X *        USES     A,F
                              5109X
                              5110X
    061.211  325              5111X $DTB     PUSH     D                   SAVE (DE)
    061.212  124              5112X          MOV      D,H
    061.213  135              5113X          MOV      E,L                 (DE) = FWA
    061.214  033              5114X          DCX      D                   (DE) = FWA-1
    061.215  176              5115X $DTB1    MOV      A,M
    061.216  043              5116X          INX      H
    061.217  247              5117X          ANA      A                   FIND END OF LINE
    061.220  302 215 061      5118X          JNZ      $DTB1
    061.223  053              5119X          DCX      H                   (HL) = ADDRESS OF TERMINATING ZERO BYTE
                              5120X
                              5121X *        GOT END OF LINE, DELETE TRAILING BLANKS
                              5122X
    061.224  053              5123X $DTB2    DCX      H                   BACKUP ONE CHARACTER
    061.225  315 216 030      5124X          CALL     $CDEHL
    061.230  312 241 061      5125X          JE       $DTB3               GONE PAST FRONT OF LINE, MUST BE ALL BLANKS
    061.233  176              5126X          MOV      A,M
    061.234  376 040          5127X          CPI      ' '
    061.236  312 224 061      5128X          JE       $DTB2               GOT BLANK
                              5129X
                              5130X *        HAVE TRIMED LINE. COMPUTE LENGTH
                              5131X
    061.241  043              5132X $DTB3    INX      H
    061.242  066 000          5133X          MVI      M,0                 TERMINATE LINE
    061.244  175              5134X          MOV      A,L
    061.245  223              5135X          SUB      E                   (A) = LENGTH +1 (FOR 00 BYTE)
    061.246  353              5136X          XCHG
    061.247  043              5137X          INX      H                   (HL) = LINE FWA
    061.250  321              5138X          POP      D                   RESTORE (DE)
    061.251  311              5139X          RET
    061.252                   5140           XTEXT    FOPE



                              5142X **       $FOPEx - OPEN FILE BLOCK FOR I/O
                              5143X *
                              5144X *        $FOPEx IS CALLED BEFORE ANY I/O IS DONE VIA A
                              5145X *        FILE BLOCK. $FOPEx SETS UP THE FILE BLOCK, AND OPENS
                              5146X *        THE FILE VIA *HDOS*.
                              5147X *
                              5148X *        ENTRY    (DE) = ADDRESS OF DEFAULT BLOCK
                              5149X *                 (HL) = ADDRESS OF FILE BLOCK
                              5150X *        EXIT     TO $FERROR IF ERROR
                              5151X *                 TO CALLER IF OK
                              5152X *        USES     A,F,B,C,D,E
                              5153X
                              5154X
```

```
061.252  315 277 061  5155X $FOPER  CALL    $FOPER.
061.255  320          5156X         RNC
061.256  303 262 063  5157X         JMP     $FERROR         IN ERROR
                      5158X
061.261  315 302 061  5159X $FOPEW  CALL    $FOPEW.
061.264  320          5160X         RNC
061.265  303 262 063  5161X         JMP     $FERROR         IN ERROR
                      5162X
061.270  315 305 061  5163X $FOPEU  CALL    $FOPEU.
061.273  320          5164X         RNC
061.274  303 262 063  5165X         JMP     $FERROR         IN ERROR
                      5166X
                      5167X
061.277  076 002      5168X $FOPER. MVI     A,FT.OR         FILE TYPE OF OPEN FOR READ
061.301  001          5169X         DB      001Q            LXI,B TO SKIP NEXT MVI
061.302  076 004      5170X $FOPEW. MVI     A,FT.OW         OPEN FOR WRITE
061.304  001          5171X         DB      001Q            LXI,B TO SKIP NEXT MIV
061.305  076 006      5172X $FOPEU. MVI     A,FT.OR+FT.OW
                      5173X
                      5174X *       (A) = FILE FLAGS
                      5175X
061.307  345          5176X         PUSH    H               SAVE FILE BLOCK ADDRESS
061.310  365          5177X         PUSH    PSW             SAVE NEW FLAGS
000.000               5178X         ERRNZ   FB.CHA
061.311  106          5179X         MOV     B,M             (B) = CHANNEL NUMBER
061.312  305          5180X         PUSH    B               SAVE HANNEL NUMBER
000.000               5181X         ERRNZ   FB.FLG-FB.CHA-1
061.313  043          5182X         INX     H
061.314  117          5183X         MOV     C,A             (C) = NEW FILE FLAGS
061.315  176          5184X         MOV     A,M             (A) = CURRENT TYPE
061.316  247          5185X         ANA     A
061.317  171          5186X         MOV     A,C             (A) = NEW FLAGS TO BE SET
061.320  312 332 061  5187X         JZ      $FOPE1          NOT ALREADY OPEN
                      5188X
                      5189X *       ALREADY OPEN. SQUACK
                      5190X
061.323  301          5191X         POP     B               RESTORE (BC)
061.324  361          5192X         POP     PSW             DISCARD NEW FLAGS
061.325  341          5193X         POP     H               (HL) = FB ADDRESS
061.326  076 031      5194X         MVI     A,EC.FAO        FILE ALREADY OPEN
061.330  067          5195X         STC
061.331  311          5196X         RET
                      5197X
000.000               5198X         ERRNZ   FB.FWA-FB.FLG-1
061.332  043          5199X $FOPE1  INX     H               (HL) = $FB.FWA
061.333  116          5200X         MOV     C,M
061.334  043          5201X         INX     H
061.335  106          5202X         MOV     B,M             (BC) = FB.FWA
061.336  043          5203X         INX     H
000.000               5204X         ERRNZ   FB.PTR-FB.FWA-2
061.337  161          5205X         MOV     M,C             SET FB.PTR = FB.FWA
061.340  043          5206X         INX     H
061.341  160          5207X         MOV     M,B
061.342  043          5208X         INX     H
000.000               5209X         ERRNZ   FB.LIM-FB.PTR-2
061.343  161          5210X         MOV     M,C             SET FB.LIM = FB.FWA
```

```
061.344  043              5211X              INX      H
061.345  160              5212X              MOV      M,B
061.346  043              5213X              INX      H
000.000                   5214X              ERRNZ    FB.NAM-FB.LIM-4
061.347  043              5215X              INX      H
061.350  043              5216X              INX      H                  (HL) = $FB.NAM
                          5217X
                          5218X *            FILE BLOCK POINTERS SETUP, OPEN FILE
                          5219X
061.351  345              5220X              PUSH     H                  SAVE NEW ADDRESS FOR NAME
061.352  041 003 062      5221X              LXI      H,$FOPEB
061.355  247              5222X              ANA      A                                                 /78.10.GC/
061.356  312 365 061      5223X              JZ       $FOPE2
000.000                   5224X              ERRNZ    .EXIT
061.361  315 240 057      5225X              CALL     $TBLS              FIND CODE
061.364  176              5226X              MOV      A,M
061.365  062 373 061      5227X $FOPE2       STA      $FOPEA             SET SYSCALL CODE
061.370  341              5228X              POP      H                  (HL) = $FB.NAM
061.371  361              5229X              POP      PSW                (A) = CHANNEL NUMBER
061.372  377 000          5230X              DB       SYSCALL,.EXIT
061.373                   5231X $FOPEA       EQU      *-1                SYSCALL CODE
061.374  321              5232X              POP      D                  (D) = NEW FLAG
061.375  341              5233X              POP      H                  (HL) = FILE BLOCK ADDRESS
061.376  330              5234X              RC                          EXIT IF ERROR
061.377  043              5235X              INX      H
000.000                   5236X              ERRNZ    FB.FLG-1
062.000  162              5237X              MOV      M,D                SET NEW FLAGS
062.001  053              5238X              DCX      H                  RESTORE (HL)
062.002  311              5239X              RET
                          5240X
062.003  002 042          5241X $FOPEB       DB       FT.OR,.OPENR       TABLE OF SYSCALL CODES
062.005  004 043          5242X              DB       FT.OW,.OPENW
062.007  006 044          5243X              DB       FT.OR+FT.OW,.OPENU
062.011  000              5244X              DB       0                  SHOULD NOT OCCUR
062.012                   5245               XTEXT    FWRIB



                          5247X **           $FWRIB - WRITE BYTES FROM FILE BUFFER.
                          5248X *
                          5249X *            $FWRIB IS CALLED TO WRITE A NUMBER OF BYTES FROM A FILE BUFFER.
                          5250X *
                          5251X *            ENTRY    (BC) = BYTE COUNT
                          5252X *                     (DE) = FWA FOR BYTES
                          5253X *                     (HL) = ADDRESS OF FILE BUFFER
                          5254X *            EXIT     TO *FERROR* IF ERROR
                          5255X *                     TO CALLER IF OK
                          5256X *                      (DE) = ADDRESS OF FIRST UNWRITTEN BYTE
                          5257X *            USES     A,F,B,C,D,E
                          5258X
                          5259X
062.012  315 021 062      5260X $FWRIB       CALL     $FWRIB.
062.015  320              5261X              RNC                         RETURN IF OK
062.016  303 262 063      5262X              JMP      $FERROR            ERROR
                          5263X
```

```
                      5264X
062.021               5265X $FWRIB. EQU     *
062.021  345          5266X         PUSH    H
062.022  315 005 063  5267X         CALL    CBT             COPY BUFFER POINTERS TO TEMP CELLS
                      5268X
                      5269X *       COPY DATA FROM USER AREA TO BUFFER
                      5270X
062.025  325          5271X $WRIB2  PUSH    D               SAVE AREA ADDRESS
062.026  072 150 063  5272X         LDA     T.FLG
062.031  346 004      5273X         ANI     FT.OW           SEE IF OPEN FOR WRITE
062.033  312 167 062  5274X         JZ      $WRIB8          FILE NOT OPEN FOR WRITE
062.036  170          5275X         MOV     A,B
062.037  261          5276X         ORA     C
062.040  312 167 062  5277X         JZ      $WRIB8          ALL DONE
                      5278X
                      5279X *       COMPUTE MIN( ROOM IN BUFFER, WRITE COUNT REQUESTED)
                      5280X
062.043  052 153 063  5281X $WRIB3  LHLD    T.PTR
062.046  353          5282X         XCHG                    (DE) = (FB.PTR) = ADDRESS OF ROOM
062.047  052 157 063  5283X         LHLD    T.LWA           (HL) = LIMIT ADDRESS
062.052  175          5284X         MOV     A,L
062.053  223          5285X         SUB     E
062.054  157          5286X         MOV     L,A
062.055  174          5287X         MOV     A,H
062.056  232          5288X         SBB     D
062.057  147          5289X         MOV     H,A             (HL) = BYTES OF ROOM IN BUFFER
062.060  171          5290X         MOV     A,C             COMPARE REQUESTED COUNT TO BUFFER ROOM
062.061  225          5291X         SUB     L
062.062  170          5292X         MOV     A,B
062.063  234          5293X         SBB     H
062.064  322 071 062  5294X         JNC     $WRIB4          MORE REQUESTED THEN ROOM
062.067  140          5295X         MOV     H,B
062.070  151          5296X         MOV     L,C             USE REQUESTED COUNT
062.071  174          5297X $WRIB4  MOV     A,H
062.072  265          5298X         ORA     L
062.073  302 133 062  5299X         JNZ     $WRIB6          SOME ROOM IN BUFFER
                      5300X
                      5301X *       BUFFER IS FULL, EMPTY IT
                      5302X
062.076  305          5303X         PUSH    B               SAVE COUNT
062.077  052 151 063  5304X         LHLD    T.FWA
062.102  042 153 063  5305X         SHLD    T.PTR           CLEAR REMOVAL POINTER
062.105  353          5306X         XCHG
062.106  052 157 063  5307X         LHLD    T.LWA
062.111  175          5308X         MOV     A,L
062.112  223          5309X         SUB     E
062.113  117          5310X         MOV     C,A
062.114  174          5311X         MOV     A,H
062.115  232          5312X         SBB     D
062.116  107          5313X         MOV     B,A             (BC) = DATA IN BUFFER
062.117  072 147 063  5314X         LDA     T.CHA
062.122  377 005      5315X         DB      SYSCALL,.WRITE  WRITE BUFFER
062.124  301          5316X         POP     B               (BC) = DESIRED COUNT
062.125  322 043 062  5317X         JNC     $WRIB3          GOT THE DATA
                      5318X
                      5319X *       ERROR ON WRITE.
```

```
                            5320X
      062.130  303 167 062  5321X         JMP     $WRIB8          HAVE ERROR
                            5322X
                            5323X *       GOT THE DATA, MOVE IT FROM BUFFER TO TARGET
                            5324X *
                            5325X *       (BC) = REQUEST COUNT
                            5326X *       (DE) = TO
                            5327X *       (HL) = COUNT
                            5328X *       ((SP)) = FROM
                            5329X
      062.133  171          5330X $WRIB6  MOV     A,C
      062.134  225          5331X         SUB     L
      062.135  117          5332X         MOV     C,A
      062.136  170          5333X         MOV     A,B
      062.137  234          5334X         SBB     H
      062.140  107          5335X         MOV     B,A             REMOVE BYTES ABOUT TO BE MOVED FROM REQUEST COUNT
      062.141  305          5336X         PUSH    B
      062.142  343          5337X         XTHL                    (HL) = REMAINING REQUEST COUNT
      062.143  301          5338X         POP     B               (BC) = COUNT FOR THIS COPY
      062.144  343          5339X         XTHL                    (HL) = TARGET ADDR, ((SP)) = REMAINING REQ. COUNT
      062.145  176          5340X $WRIB7  MOV     A,M
      062.146  022          5341X         STAX    D
      062.147  023          5342X         INX     D
      062.150  043          5343X         INX     H
      062.151  013          5344X         DCX     B
      062.152  170          5345X         MOV     A,B
      062.153  261          5346X         ORA     C
      062.154  302 145 062  5347X         JNZ     $WRIB7          MORE TO GO
      062.157  353          5348X         XCHG
      062.160  042 153 063  5349X         SHLD    T.PTR           UPDATE POINTER
      062.163  301          5350X         POP     B               (BC) = REMAINING COUNT
      062.164  303 025 062  5351X         JMP     $WRIB2          SEE IF MORE IN BUFFER
                            5352X
                            5353X *       WRITE COMPLETE.
                            5354X *
                            5355X *       (PSW) = COMPLETION FLAGS
                            5356X
      062.167  321          5357X $WRIB8  POP     D               RESTORE TARGET ADDRESS
      062.170  341          5358X         POP     H
      062.171  303 033 063  5359X         JMP     CTB             COPY TEMP POINTERS BACK TO BLOCK, EXIT



                            5361X **      $FWBRK  -  BREAKOUTPUT                                   /80.02.GC/
                            5362X *
                            5363X *       $FWBRK empties the specified buffer by filling it with NULLs
                            5364X *       and then writing it.  Note this is used to insure that block
                            5365X *       mode I/O is output if it is not really a serial device (eg.
                            5366X *       writing to AT: from *EDIT*.
                            5367X *
                            5368X *
                            5369X *       ENTRY:  HL      =  FILE BLOCK POINTER
                            5370X *
                            5371X *       EXIT:   HL      =  FILE BLOCK POINTER
                            5372X *               TO  $FERROR IF ERROR
```

```
                        5373X *
                        5374X *        USES:    PSW,BC,DE
                        5375X *
                        5376X
 062.174  315 203 062   5377X $FWBRK   CALL     $FWBRK.
 062.177  320           5378X          RNC                      NO ERROR
                        5379X
 062.200  303 262 063   5380X          JMP      $FERROR
                        5381X
 062.203  345           5382X $FWBRK.  PUSH     H
 062.204  315 005 063   5383X          CALL     CBT             COPY BUFFER TO TEMPORARY
 062.207  315 217 062   5384X          CALL     $FWBRK1
 062.212  341           5385X          POP      H
 062.213  315 033 063   5386X          CALL     CTB             COPY TEMPORARY TO BUFFER
 062.216  311           5387X          RET
                        5388X
 062.217  052 157 063   5389X $FWBRK1  LHLD     T.LWA
 062.222  353           5390X          XCHG                     DE = BUFFER LWA
 062.223  052 153 063   5391X          LHLD     T.PTR           HL = BUFFER PTR
 062.226  173           5392X          MOV      A,E
 062.227  225           5393X          SUB      L
 062.230  117           5394X          MOV      C,A
 062.231  172           5395X          MOV      A,D
 062.232  234           5396X          SBB      H
 062.233  107           5397X          MOV      B,A             BC = DE - HL
 062.234  261           5398X          ORA      C
 062.235  310           5399X          RZ                       THE BUFFER IS ALREADY FLUSHED
                        5400X
                        5401X *        FILL THE BUFFER WITH NULLS
                        5402X
 062.236  170           5403X FWBRK2   MOV      A,B
 062.237  261           5404X          ORA      C
 062.240  312 252 062   5405X          JZ       FWBRK3          NO MORE LEFT TO FILL
                        5406X
 062.243  066 000       5407X          MVI      M,0
 062.245  043           5408X          INX      H
 062.246  013           5409X          DCX      B
 062.247  303 236 062   5410X          JMP      FWBRK2
                        5411X
 062.252  052 151 063   5412X FWBRK3   LHLD     T.FWA
 062.255  042 153 063   5413X          SHLD     T.PTR
 062.260  353           5414X          XCHG                     DE = BUFFER FWA
 062.261  052 157 063   5415X          LHLD     T.LWA           HL = BUFFER LWA
 062.264  175           5416X          MOV      A,L
 062.265  223           5417X          SUB      E
 062.266  117           5418X          MOV      C,A
 062.267  174           5419X          MOV      A,H
 062.270  232           5420X          SBB      D
 062.271  107           5421X          MOV      B,A             BC = HL - DE  ( BC = COUNT )
 062.272  072 147 063   5422X          LDA      T.CHA
 062.275  377 005       5423X          DB       SYSCALL,.WRITE
 062.277  311           5424X          RET
 062.300                5425           XTEXT    FCLO
```

```
                          5427X **       $FCLO - CLOSE FILE BLOCK.
                          5428X *
                          5429X *        $FCLO IS CALLED TO TERMINATE PROCESSING THROUGH A FILE
                          5430X *        BLOCK.
                          5431X *
                          5432X *        ENTRY   (HL) = FILE BLOCK ADDRESS
                          5433X *        EXIT    TO $FERROR IF ERROR
                          5434X *                 TO CALLER IF OK
                          5435X *        USES    A,F,B,C,D,E
                          5436X
                          5437X
062.300  315 307 062      5438X $FCLO    CALL    $FCLO.
062.303  320              5439X          RNC                     NO ERROR
062.304  303 262 063      5440X          JMP     $FERROR
                          5441X
062.307  345              5442X $FCLO.   PUSH    H               SAVE FILE BLOCK ADDRESS
000.000                   5443X          ERRNZ   FB.FLG-1
062.310  043              5444X          INX     H               (HL) = $FB.FLG
062.311  176              5445X          MOV     A,M
062.312  066 000          5446X          MVI     M,0             CLEAR FLAG
062.314  247              5447X          ANA     A
062.315  312 003 063      5448X          JZ      $FCLO4          FILE NOT OPEN
062.320  346 004          5449X          ANI     FT.OW
062.322  312 375 062      5450X          JZ      $FCLO3          NO WRITING, NO FLUSHING NEEDED
                          5451X
                          5452X *        WAS OPEN FOR WRITE. SEE IF NEED FLUSH THE LAST SECTOR
                          5453X
062.325  315 234 030      5454X          CALL    $INDL
062.330  003 000          5455X          DW      FB.PTR-FB.FLG
062.332  325              5456X          PUSH    D               SAVE (FB.PTR)
062.333  315 234 030      5457X          CALL    $INDL           (DE) = (FB.FWA)
062.336  001 000          5458X          DW      FB.FWA-FB.FLG
062.340  341              5459X          POP     H               (HL) = (FB.PTR)
062.341  175              5460X          MOV     A,L
062.342  223              5461X          SUB     E
062.343  117              5462X          MOV     C,A
062.344  174              5463X          MOV     A,H
062.345  232              5464X          SBB     D
062.346  107              5465X          MOV     B,A             (BC) = AMOUNT IN BLOCK
062.347  261              5466X          ORA     C
062.350  312 375 062      5467X          JZ      $FCLO3          NONE TO FLUSH
                          5468X
                          5469X *        NEED TO FLUSH BUFFER
                          5470X *
                          5471X *        (BC) = DATA AMOUNT
                          5472X *        (DE) = FWA
                          5473X *        (HL) = LWA+1
                          5474X
062.353  171              5475X          MOV     A,C
062.354  247              5476X          ANA     A
062.355  312 370 062      5477X          JZ      $FCLO2          DONT HAVE PARTIAL SECTOR
                          5478X
                          5479X *        ZERO FILL PARTIAL SECTOR
                          5480X
062.360  066 000          5481X $FCLO1   MVI     M,0
062.362  043              5482X          INX     H
```

```
062.363  014            5483X          INR     C
062.364  302 360 062    5484X          JNZ     $FCLO1
062.367  004            5485X          INR     B                  COUNT ANOTHER FULL SECTOR
062.370  341            5486X $FCLO2   POP     H                  (HL) = FB FWA
062.371  176            5487X          MOV     A,M                (A) = CHANNEL NUMBER
000.000                 5488X          ERRNZ   FB.CHA
062.372  345            5489X          PUSH    H
062.373  377 005        5490X          DB      SYSCALL,.WRITE           FLUSH
                        5491X
                        5492X *        READY TO CLOSE FILE.
                        5493X *
                        5494X *        'C' SET IF ERROR
                        5495X *         (A) = ERROR CODE
                        5496X
062.375  341            5497X $FCLO3   POP     H                  (HL) = FILE BLOCK ADDRESS
062.376  330            5498X          RC                         ERROR
000.000                 5499X          ERRNZ   FB.CHA
062.377  176            5500X          MOV     A,M                (A) = CHANNEL NUMBER
063.000  345            5501X          PUSH    H
063.001  377 046        5502X          DB      SYSCALL,.CLOSE     CLOSE CHANNEL
063.003  341            5503X $FCLO4   POP     H                  (HL) = FILE BLOCK ADDRESS
063.004  311            5504X          RET
063.005                 5505           XTEXT   FUTIL



                        5507X **       $FUTIL - UTILITY ROUTINES FOR FILE BLOCK ROUTINES.
                        5508X
                        5509X **       CBT - COPY BLOCK POINTERS TO TEMP CELLS.
                        5510X *
                        5511X *        ENTRY   (HL) = FILE BLOK FWA
                        5512X *        EXIT    NONE
                        5513X *        USES    A,F,H,L
                        5514X
063.005  325            5515X CBT      PUSH    D
063.006  305            5516X          PUSH    B                  SAVE REGISTERS
000.000                 5517X          ERRNZ   TLEN-10            ASSUME 10 BYTES TO MOVE
063.007  021 147 063    5518X          LXI     D,T.CHA            (DE) = TARGET FOR MOVE
063.012  006 005        5519X          MVI     B,10/2
063.014  176            5520X CBT1     MOV     A,M                COPY FILE BUFFER INTO WORK AREA
063.015  022            5521X          STAX    D
063.016  043            5522X          INX     H
063.017  023            5523X          INX     D
063.020  176            5524X          MOV     A,M
063.021  022            5525X          STAX    D
063.022  043            5526X          INX     H
063.023  023            5527X          INX     D
063.024  005            5528X          DCR     B
063.025  302 014 063    5529X          JNZ     CBT1               MORE TO GO
063.030  301            5530X          POP     B
063.031  321            5531X          POP     D                  (DE) = DATA TARGET ADDRESS
063.032  311            5532X          RET
                        5533X
                        5534X
                        5535X **       CTB - COPY TEMP CELLS BACK TO FILE BLOCK.
```

```
                            5536X *
                            5537X *         ENTRY   (HL) = FILE BLOCK ADDRESS
                            5538X *         EXIT    NONE
                            5539X *         USES    NONE
                            5540X
063.033  365                5541X CTB       PUSH    PSW
063.034  325                5542X           PUSH    D
063.035  305                5543X           PUSH    B
063.036  345                5544X           PUSH    H                       SAVE REGISTERS
063.037  006 004            5545X           MVI     B,8/2
063.041  021 147 063        5546X           LXI     D,T.CHA
063.044  032                5547X CTB1      LDAX    D
063.045  167                5548X           MOV     M,A
063.046  023                5549X           INX     D
063.047  043                5550X           INX     H
063.050  032                5551X           LDAX    D
063.051  167                5552X           MOV     M,A
063.052  023                5553X           INX     D
063.053  043                5554X           INX     H
063.054  005                5555X           DCR     B
063.055  302 044 063        5556X           JNZ     CTB1                    RESTORE FILE BUFFER VALUES
063.060  341                5557X           POP     H
063.061  301                5558X           POP     B
063.062  321                5559X           POP     D
063.063  361                5560X           POP     PSW
063.064  311                5561X           RET



                            5563X **        $FFB - FILE FILE BUFFER.
                            5564X *
                            5565X *         $FFB FILLS THE FILE BUFFER BY READING FROM THE FILE.
                            5566X *
                            5567X *         ENTRY   NONE
                            5568X *         EXIT    'C' SET IF READ INCOMPLETE
                            5569X *                  (A) = ERROR CODE
                            5570X *                 'C' CLEAR IF READ COMPLETEE
                            5571X *                  DATA IN BUFFER
                            5572X *         USES    A,F,D,E,H,L
                            5573X
                            5574X
063.065  072 161 063        5575X $FFB      LDA     EOFFLG
063.070  037                5576X           RAR
063.071  330                5577X           RC                              EOF
                            5578X
                            5579X *         CAN READ MORE, DO SO
                            5580X
063.072  305                5581X           PUSH    B                       SAVE COUNT
063.073  052 151 063        5582X           LHLD    T.FWA
063.076  042 153 063        5583X           SHLD    T.PTR                   CLEAR REMOVAL POINTER
063.101  353                5584X           XCHG
063.102  052 157 063        5585X           LHLD    T.LWA
063.105  042 155 063        5586X           SHLD    T.LIM                   SET DATA LIMIT
063.110  175                5587X           MOV     A,L
063.111  223                5588X           SUB     E
```

```
063.112  117          5589X          MOV     C,A
063.113  174          5590X          MOV     A,H
063.114  232          5591X          SBB     D
063.115  107          5592X          MOV     B,A              (BC) = ROOM IN BUFFER
063.116  072 147 063  5593X          LDA     T.CHA
063.121  377 004      5594X          DB      SYSCALL,.READ    READ BUFFER
063.123  120          5595X          MOV     D,B              (D) = SECTORS UNREAD
063.124  301          5596X          POP     B                (BC) = DESIRED COUNT
063.125  320          5597X          RNC                      GOT THE DATA
                      5598X
                      5599X *        ERROR ON READ. SEE IF EOF
                      5600X
063.126  027          5601X          RAL
063.127  062 161 063  5602X          STA     EOFFLG           SET EOF, WE HOPE
063.132  376 003      5603X          CPI     EC.EOF*2+1
063.134  037          5604X          RAR
063.135  300          5605X          RNE                      IS NOT EOF, RETURN NOW!
063.136  072 156 063  5606X          LDA     T.LIM+1
063.141  222          5607X          SUB     D
063.142  062 156 063  5608X          STA     T.LIM+1          SET AMOUNT OF DATA WE DID GET
063.145  247          5609X          ANA     A
063.146  311          5610X          RET                      EXIT WITH DATA
                      5611X
                      5612X
                      5613X **       TEMP CELLS TO HOLD FILE BLOCK POINTERS DURING I/O
                      5614X
000.000               5615X          ERRNZ   FB.CHA
063.147  000          5616X T.CHA    DB      0                CHANNEL NUMBER
000.000               5617X          ERRNZ   *-T.CHA-FB.FLG
063.150  000          5618X T.FLG    DB      0                FLAG BYTE
000.000               5619X          ERRNZ   *-T.CHA-FB.FWA
063.151  000 000      5620X T.FWA    DW      0
000.000               5621X          ERRNZ   *-T.CHA-FB.PTR
063.153  000 000      5622X T.PTR    DW      0
000.000               5623X          ERRNZ   *-T.CHA-FB.LIM
063.155  000 000      5624X T.LIM    DW      0
000.000               5625X          ERRNZ   *-T.CHA-FB.LWA
063.157  000 000      5626X T.LWA    DW      0
000.012               5627X TLEN     EQU     *-T.CHA          LENGTH OF TEMP CELLS
                      5628X
063.161  000          5629X EOFFLG   DB      0
                      5630           ENDIF
```

```
 063.162            5633  PATCH   DS     64              PATCH AREA
 000.001            5634          IF     ONECOPY                                                   /2.0a/
                    5635          DS     *+255/256*256-* Auxiliary Patch Area (Round up 1 page)    /2.0a/
                    5636          ENDIF                                                            /2.0a/
```

```
000.001          IF      ONECOPY


         **      FDN - FILE DESCRIPTOR NODES.
         *
         *       THESE NODES ARE USED TO KEEP TRACK OF FILES WHICH ARE BEING
         *       HELD IN MEMORY WHILE TRANSFERING.

         FDN     DS      0               START OF TYPICAL NODE
         FDN.LNK EQU     *-FDN           LINK TO NEXT NODE IN CHAIN
                 DS      1               ALL IN SAME PAGE,  JUST KEEP PAGE INDEX
         FDN.STA EQU     *-FDN           STATUS BYTE
         ST.CNT  EQU     DIF.CNT         IS CONTIGUOUS
         ST.OPR  EQU     00000010B       IS BEING READ
         ST.OPW  EQU     00000001B       OPEN FOR WRITE
                 DS      1               STATUS BYTE
         FDN.SIZ EQU     *-FDN           TOTAL SIZE OF FILE (IF ST.CNT SET)
                 DS      1               SIZE IN GROUPS
         FDN.AMR EQU     *-FDN           AMOUNT ALREADY READ
                 DS      2                IN SECTORS
         FDN.AMW EQU     *-FDN           AMOUNT ALREADY WRITTEN
                 DS      2                IN SECTORS
         FDN.ADR EQU     *-FDN           ADDRESS IN BUFFER
                 DS      1                ADDRESS/256 (MUST BE EVEN PAGE)
         FDN.AIM EQU     *-FDN           AMOUNT IN MEMORY
                 DS      1                 IN SECTORS
         FDNELEN EQU     *-FDN           ENTRY  LENGTH
                 ORG     FDN             ORG BACK OVER DEFINITION AREA



         **      TABLE. A LINK OF 0 IS A NULL LINK.
         *
         *       THE ENTIRE GROUP OF NODES MUST RESIDE
         *       IN THE SAME PAGE

         FDNFWA  EQU     *               START OF NODES

         FDNFRE  DB      #FDN.1          START OF FREE CHAIN
         FDNHEAD DB      0                       ACTIVE LIST NOW EMPTY

         FDN.1   DS      0
                 DB      #FDN.2          FDN.LNK
                 DB      0               FDN.STA
                 DB      0               FDN.SIZ
                 DW      0               FDN.AMR
                 DW      0               FDN.AMW
                 DB      0               FDN.ADR
                 DB      0               FDN.AIM

         FDN.2   DS      0
                 DB      #FDN.3          FDN.LNK
                 DB      0               FDN.STA
                 DB      0               FDN.SIZ
                 DW      0               FDN.AMR
                 DW      0               FDN.AMW
```

```
5695            DB      0               FDN.ADR
5696            DB      0               FDN.AIM
5697
5698  FDN.3     DS      0
5699            DB      #FDN.4          FDN.LNK
5700            DB      0               FDN.STA
5701            DB      0               FDN.SIZ
5702            DW      0               FDN.AMR
5703            DW      0               FDN.AMW
5704            DB      0               FDN.ADR
5705            DB      0               FDN.AIM
5706
5707  FDN.4     DS      0
5708            DB      #FDN.5          FDN.LNK
5709            DB      0               FDN.STA
5710            DB      0               FDN.SIZ
5711            DW      0               FDN.AMR
5712            DW      0               FDN.AMW
5713            DB      0               FDN.ADR
5714            DB      0               FDN.AIM
5715
5716  FDN.5     DS      0
5717            DB      #FDN.6          FDN.LNK
5718            DB      0               FDN.STA
5719            DB      0               FDN.SIZ
5720            DW      0               FDN.AMR
5721            DW      0               FDN.AMW
5722            DB      0               FDN.ADR
5723            DB      0               FDN.AIM
5724
5725  FDN.6     DS      0
5726            DB      #FDN.7          FDN.LNK
5727            DB      0               FDN.STA
5728            DB      0               FDN.SIZ
5729            DW      0               FDN.AMR
5730            DW      0               FDN.AMW
5731            DB      0               FDN.ADR
5732            DB      0               FDN.AIM
5733
5734  FDN.7     DS      0
5735            DB      #FDN.8          FDN.LNK
5736            DB      0               FDN.STA
5737            DB      0               FDN.SIZ
5738            DW      0               FDN.AMR
5739            DW      0               FDN.AMW
5740            DB      0               FDN.ADR
5741            DB      0               FDN.AIM
5742
5743  FDN.8     DS      0
5744            DB      0               FDN.LNK
5745            DB      0               FDN.STA
5746            DB      0               FDN.SIZ
5747            DW      0               FDN.AMR
5748            DW      0               FDN.AMW
5749            DB      0               FDN.ADR
5750            DB      0               FDN.AIM
```

```
                              5751
                              5752  FDNCNT  EQU     *-FDN.1/FDNELEN         NUMBER OF NODES
                              5753
                              5754  .       SET     */256
                              5755          ERRNZ   FDNFWA/256-.      MUST BE ALL IN SAME PAGE
                              5756
                              5757  VOLFLAG DB      0                 =0 IF READING FROM SOURCE, =377Q IF WRITTING TO DEST
                              5758  VOLSER  DB      0                 SERIAL NUMBER OF CURRENT DISK
                              5759
                              5760  OBUFLIM DB      0                 BUFFER LIMIT/256
                              5761  OBUFPTR DB      0                 NEXT FREE PAGE IN BUFFER/256
                              5762
                              5763
                              5764          ENDIF
                              5765
    063.262                   5766          XTEXT   FERROR            APPEARS HERE TO ALLOW FDN. TO BE IN ONE PAGE


                              5768X **      $FERROR - PROCESS FILE ERRORS.
                              5769X *
                              5770X *       $FERROR IS CALLED TO COMPLAIN ABOUT AN ERROR ENCOUNTERED
                              5771X *       WHEN PROCESSING FILES.
                              5772X *
                              5773X *       ENTRY   (A) = ERROR CODE
                              5774X *               (HL) = ADDRESS OF FILE NAME - FB.NAM
                              5775X *       EXIT    TO RESTART
                              5776X *       USES    ALL
                              5777X
                              5778X
    063.262  365              5779X $FERROR PUSH    PSW               SAVE CODE
    063.263  315 136 031      5780X         CALL    $TYPTX
    063.266  012 007 105      5781X         DB      NL,BELL,'ERROR ON FILE',' '+200Q
    063.306  021 012 000      5782X         LXI     D,FB.NAM
    063.311  031              5783X         DAD     D
                              5784X
                              5785X *       PRINT FILE NAME
                              5786X
    063.312  176              5787X $FERR1  MOV     A,M
    063.313  043              5788X         INX     H                 ADVANCE MESSAGE
    063.314  247              5789X         ANA     A
    063.315  312 326 063      5790X         JZ      $FERR2
    063.320  315 376 060      5791X         CALL    $WCHAR
    063.323  303 312 063      5792X         JMP     $FERR1
                              5793X
                              5794X *       TYPE ERROR MESSAGE
                              5795X
    063.326  315 136 031      5796X $FERR2  CALL    $TYPTX
    063.331  040 055 240      5797X         DB      ' -',' '+200Q
    063.334  046 012          5798X         MVI     H,NL
    063.336  361              5799X         POP     PSW               (A) = CODE
    063.337  377 057          5800X         DB      SYSCALL,.ERROR
    063.341  303 200 042      5801X         JMP     RESTART           EXIT
```

```
       063.344  000           5804  ALLOCA  DB      0                    /ALL flag (<>0 if /ALL specified) /80.06.sc/
       063.345  000           5805  COMAND  DB      0                    COMMAND IN PROGRESS
       063.346  000           5806  MODE    DB      0                    <>0 IF LINE PASSED ON STACK
       063.347  000           5807  JGL     DB      0                    /JGL FLAG (<>0 IF /JGL SPECIFIED)
       063.350  000           5808  SUPRES  DB      0                    /SUP FLAG (<>0 OF /SUP SPECIFIED)
       063.351  001           5809  SYSTEM  DB      1                    /S FLAG (=0 IF /S SPECIFIED)
                              5810
       063.352  130 130 130   5811  DIRNAM  DB      'XXX:DIRECT.SYS',0       DIRECTORY FILE NAME
                              5812
       063.371  256 067       5813  BUFPTR  DW      BUFF                 POINTER TO START  OF BUFFER
       063.373  000 000       5814  BUFSIZ  DW      0                    BUFFER LENGTH


                              5816  **      FILE BLOCKS
                              5817
       000.000                5818          IF      .PIP.
       063.375                5819  DESTFB  DS      0                    DESTINATION FILE BLOCK
       063.375  001           5820          DB      CN.DES               CHANNEL NUMBER
       063.376  000           5821          DB      0                    FLAGS
       063.377  063 064       5822          DW      DESTBUF
       064.001  063 064       5823          DW      DESTBUF
       064.003  063 064       5824          DW      DESTBUF
       064.005  063 065       5825          DW      DESTBFE              END OF BLOCK
       064.007                5826          DS      FB.NAML              NAME AREA
                              5827          ELSE
                              5828  DESTFB  DS      0                    DUMY BUFFER
                              5829          DB      200                  ILLEGAL CHANNEL NUMBER
                              5830          DB      0                    FLAGS
                              5831          DW      0
                              5832          DW      0
                              5833          DW      0
                              5834          DW      0                    END OF BLOCK
                              5835          DS      FB.NAML              NAME AREA
                              5836          ENDIF


       064.030  000 000       5838  NAMTLEN DW      0                    NAME TABLE POINTER
       064.032  000 000       5839  NAMTMAX DW      0                    MAXIMUM SIZE OF NAME TABLE
       000.001                5840          IF      ONECOPY
                              5841  NAMTPTR DW      0                    POINTER TO ACTIVE ELEMENT IN NAMTAB
                              5842          ENDIF
                              5843
```

```
                    ***     PRS - PRESET PIP PROGRAM.
                    *
                    *       PRS IS CALLED TO PERFORM ONE-TIME-ONLY PRESETTING OF
                    *       THE PROGRAM ENVIRONMENT.
                    *
                    *       THE CODE IS OVERLAID BY BUFFERS AND WORK AREAS WHEN PIP IS RUNNING.
000.000                     IF      .PIP.
                    *       BE CAREFUL NOT TO USE ANY OF THE BUFFERS AND WORK AREAS BEFORE
                    *       THE AREA *LINE*.
                            ELSE
                    *       DO NOT USE ANY OF THE BUFFERS AND WORK AREAS IN *PRS*
                            ENDIF
                    *
                    *
                    *       ENTRY   NONE
                    *
                    *       EXIT    IF  CORRECT VERSION OF HDOS
                    *                   NONE
                    *                 ELSE
                    *                   EXIT TO HDOS
                    *
                    *       USES    ALL
                    *

064.034             ENTRY   EQU     *               INITIAL ENTRY POINT
064.034 377 011     PRS     DB      SYSCALL,.VERS
064.036 332 130 064         JC      PRS1            ERROR IN GETTING VERSION
064.041 376 040             CPI     VERS
064.043 302 130 064         JNZ     PRS1            NOT CORRECT VERSION OF HDOS
064.046 041 256 067         LXI     H,RMEML         (HL) = RUN-TIME HIGH MEMORY
064.051 377 052             DB      SYSCALL,.SETTP  SET HI MEMORY
064.053 332 133 064         JC      PRS2            IF  ERROR
064.056 041 355 042         LXI     H,CCHIT
064.061 076 003             MVI     A,CTLC
064.063 377 041             DB      SYSCALL,.CTLC   SET CTL-C PROCESSING
064.065 076 377             MVI     A,377Q
064.067 377 046             DB      SYSCALL,.CLOSE  CLOSE OVERLAY CHANNEL
000.000                     IF      .PIP.

                    *       SEE IF COMMAND LINE PASSED ON STACK

064.071 041 000 000         LXI     H,0
064.074 071                 DAD     SP
064.075 353                 XCHG
064.076 076 200             MVI     A,#STACK
064.100 223                 SUB     E
064.101 117                 MOV     C,A
064.102 076 042             MVI     A,STACK/256
064.104 232                 SBB     D
064.105 107                 MOV     B,A             (BC) = BYTES ON STACK
064.106 261                 ORA     C
064.107 062 346 063         STA     MODE            SET MODE <>0 IF LINE ON STACK
064.112 312 207 042         JZ      START           NO LINE

                    *       HAVE LCOMMAND ON STACK. COPY INTO LINE BUFFER
                    *       (BC) = COUNT
```

```
                                5903  *        (DE) = FWA
                                5904
 064.115  041 136 067  5905           LXI     H,LINE
 064.120  315 252 030  5906           CALL    $MOVE            COPY
 064.123  066 000      5907           MVI     M,0              ENSURE END
                       5908           ELSE    ONECOPY
                       5909           CALL    $DOS             DISMOUNT OPERATING SYSTEM
                       5910           JC      PRS2             IF ERROR
                       5911           CALL    $TYPTX
                       5912           DB      NL,TAB,TAB,TAB,'    ','ONECOPY'
                       5913           DB      NL,TAB,TAB,TAB,'Version:   ',VERS/16+'0','.',VERS&0FH+'0'
                       5914           DB      NL,TAB,TAB,'        ','Issue:  #50.06.00 '
                       5915           DB      NL,NL,' ONECOPY is used to copy files for systems with only one'
                       5916           DB      NL,'floppy drive. Read the appropriate manual before using.'
                       5917           DB      ENL
                       5918           CALL    $TYPTX
                       5919           DB      NL,'Insert the initial source disk. Hit RETURN when ready:',' '+200Q
                       5920           CALL    GDWP.                                       /79.11.GC/
                       5921           CALL    $RTL             GET CR
                       5922
                       5923           JMP     PRS3             Jump the the rest of the code          /2.0a/
                       5924           ENDIF
 064.125  303 207 042  5925           JMP     START            START PROGRAM
                       5926
 064.130  076 050      5927  PRS1     MVI     A,EC.NCV         NOT CORRECT VERSION
 064.132  067          5928           STC
 064.133  046 012      5929  PRS2     MVI     H,NL
 064.135  377 057      5930           DB      SYSCALL,.ERROR
 064.137  303 352 042  5931           JMP     EXIT
                       5932
 000.001               5933           IF      ONECOPY
                       5934           XTEXT   DTB
                       5935           XTEXT   DOS
                       5936           ENDIF
                       5937
 064.142               5938  MEML     EQU     *                MEMORY LENGTH
```

```
                **      THE FOLLOWING BUFFERS AND AREAS OVERLAY THE PRS CODE.
                *
                *       *PRS* MAY NOT USE ANY CELLS BELOW *LINE*, AT THE
                *       RISK OF SMASHING ITSELF

064.034                 ORG     PRS

064.034         DEFALT  DS      6               DEFAULT BLOCK

064.042         MWNA    DS      FB.NAML         MWN WORK AREA

000.000                 IF      .PIP.
064.063         DESTBUF DS      256             DESTINATION FILE BUFFER (ALSO USED BY *CCW*)
065.063         DESTBFE EQU     *               END OF BUFFER
                        ENDIF

                **      * * NOTE * *
                *       DIRWORK  USES THE SYSTEM SCRATCH AREA, LABEL. DIRWORK WILL NOT
                *       BE PRESERVED DURING A SYSCALL !!

065.063         SLABEL  DS      256             Saved Label Sector                          /2.0a/
066.063         LABEL   DS      256             Label Sector                                /2.0a/

                *DIRWORK        EQU     SECSCR          USE SECTOR SCRATCH AREA /79.11.GC/


                **      PIO.XXX - IMAGE OF SYSTEM AIO.XXX AREA
                *
                *       THESE CELLS MIRROR THE SYSTEM AIO.XXX AREA


067.063         PIO.DEV DS      2               DEVICE CODE
067.065         PIO.UNI DS      1               UNIT NUMBER (0-9)

067.066         PIO.DIR DS      DIRELEN         DIRECTORY ENTRY

067.115         $FOPWRK DS      FB.NAML         WORK AREA FOR $FOPE


000.000                 IF      .PIP.
002.374                 ERRMI   *-MEML          FOLLOWING MUST NOT OVERLAY *PRS*
                        ENDIF
067.136         LINE    DS      80              COMMAND BUFFER


067.256         NAMTAB  DS      0               NAME TABLE


002.000         BUFMINL EQU     512             MINIMUM SIZE FOR BUFFER (WHEN IN USE)
067.256         BUFF    EQU     *               BUFFER AREA STARTS AFTER NAMTAB

067.256         RMEML   EQU     *               INITIAL RUNNING MEMORY LENGTH
```

```
       067.256              5995            END
ASSEMBLY COMPLETE
 5995 STATEMENTS
    0 ERRORS DETECTED
 8558 BYTES FREE
```

```
$CCD    057073       956       4072L
$CDEHL  030216      2510       4501E   5049    5124
$CFD    057035      3420       3972L
$CHL    030224      3531       3546    4221E
$CMP$   000001      4352E      4396    4405
$COMF   030060      2301       2673    4458E
$CRLF   057271      4094       4327L   4399
$DAD    060056      2791       4730L
$DADA   030072      4293E      4375    4811
$DADA.  030101      2524       3140    4539E   4762
$DRS    061012       970       4978E
$DRS1   061012      4982L      4987    5052
$DRS15  061156      5006       5028    5064L
$DRS2   061027      4993L      5014
$DRS20  061171      5011       5030    5086L   5089
$DRS3   061032      4994L      5001
$DRS4   061051      4998       5005L
$DRS4.5 061065      5007       5011L
$DRS5   061075      5018L      5029
$DRS6   061105      4996       5010    5026L
$DRS7   061134      5031       5041L
$DRS8   061141      5047L      5050
$DRSB   061207      4988       4993    5019    5042    5098L
$DTB    061211      1446       5111L
$DTB1   061215      5115L      5118
$DTB2   061224      5123L      5128
$DTB3   061241      5125       5132L
$DU66   030106      4529E      4738    4752    4817
$FCLO   062300      2585       2590    5438L
$FCLO.  062307      5438       5442L
$FCLO1  062360      5481L      5484
$FCLO2  062370      5477       5486L
$FCLO3  062375      5450       5467    5497L
$FCLO4  063003      5448       5503L
$FERR1  063312      5787L      5792
$FERR2  063326      5790       5796L
$FERROR 063262      1259       1288    1316    1333    1360    2313    2316    2924    5157    5161    5165    5262
                    5380       5440    5779L
$FFB    063065      5575L
$FOPE1  061332      5187       5199L
$FOPE2  061365      5223       5227L
$FOPEA  061373      5227       5231E
$FOPEB  062003      5221       5241L
$FOPER  061252      5155L
$FOPER. 061277      5155       5168L
$FOPEU  061270      5163L
$FOPEU. 061305      5163       5172L
$FOPEW  061261      2442       5159L
$FOPEW. 061302      5159       5170L
$FOPWRK 067115      5976L
$FWBRK  062174      5377L
$FWBRK. 062203      5377       5382L
$FWBRK1 062217      5384       5389L
$FWRIB  062012      2454       2586    2828    5260L
$FWRIB. 062021      5260       5265E
$GNL    057110      1196       4089L
$HLIHL  030211      2789       4550E
$INDL   030234      2423       4587E   5454    5457
```

```
$INDLB   057353      2419       2520    3689    4602L
$INDS    057374      4635L
$INDSB   060030      4665L
$MCU     057144      4111       4130L
$MLU     057124      4106L      4156
$MLU1    057127      4109L      4114
$MOVE    030252      2395       2667    3036    3327    3330    3367    3381    3408    3721    3871    4209E   4765
                     4790       4890    5906
$MOVEL   060345      1242       2376    2634    3653    3655    3828    3902    3945    3949    4874L
$MU86    031007      2572       2761    4060E
$RCHAR   060370      4160       4900L   4901
$RSTALL  031047      3904       3953    4076    4475E
$RTL     057164      4154       4158E
$RTL.    057155      1206       3118    4154L
$RTL1    057165      4160L      4166
$RTL2    057217      4162       4183L
$SAVALL  031054      3901       3943    4072    4489E
$SOB     057222      2648       2893    3332    3418    3924    3929    4234L   5026
$SOB1    057223      4235L      4238    4240
$TBLS    057240      4262L      5225
$TJMP    031061       973       4313E
$TJMP.   031062      4315E
$TYPC.   057303      4350L      4400
$TYPCC   057057      3114       3998E   4007
$TYPCH   057277      4340L
$TYPL.   057324      4377       4389E   4403
$TYPLN.  057306      4371L
$TYPTX   031136      1009       1197    1344    1370    2899    2949    2953    2964    3110    3115    4435E   5780
                     5796
$TYPTX.  031144      4437E
$UDD     031157      4516E      4746    4776
$UDDN    060272      1369       2559    2566    2577    2778    4810E
$WCHAR   060376      2952       4904L   5791
$WDR     031222      4032E
$WER     031241      4020E
$WRIB2   062025      5271L      5351
$WRIB3   062043      5281L      5317
$WRIB4   062071      5294       5297L
$WRIB6   062133      5299       5330L
$WRIB7   062145      5340L      5347
$WRIB8   062167      5274       5277    5321    5357L
$ZERO    031212      4045E
.ABUSS   040024       808E
.ALARM   002136       781E
.ALEDS   040013       806E
.CHFLG   000060       459L
.CLEAN   000205       474L
.CLEAR   000055       456L      1347
.CLEARA  000056       457L       941
.CLOSE   000046       449L      1321    1331    1357    2544    3735    5502    5883
.CLRCO   000007       433L      1011
.CONSL   000006       432L      4075    4091
.CRC     002347       789E
.CRCSUM  040027       809E
.CTC     002172       783E
.CTL2FL  040066       815E
.CTLC    000041       444L      5881
.CTLFLG  040011       805E
```

```
.DAD    000206       475L
.DECODE 000053       454L    2406    3659
.DELET  000050       451L    2262
.DISMT  000061       460L
.DLEDS  040021       807E
.DLY    000053       778E
.DMNMS  000203       472L
.DMOUN  000201       470L    1406
.DOD    003122       792E
.DODA   003356       794E
.DSPMOD 040007       803E
.DSPROT 040006       802E
.DUMP   001374       780E
.ERROR  000057       458L    2970    5800    5930
.EXIT   000000       426L    1002    2957    5224    5230
.HORN   002140       782E
.IDENT  000000       777E
.IOWRK  040002       800E
.LINK   000040       443L
.LOAD   001267       779E
.LOADD  000062       461L
.LOADO  000010       434L    1460    1463
.MFLAG  040010       804E
.MONMS  000202       471L
.MOUNT  000200       469L    1391
.NAME   000054       455L
.NMIRET 040064       814E
.OPEN   000063       462L
.OPENC  000045       448L
.OPENR  000042       445L    1271    2309    2433    3671    5241
.OPENU  000044       447L    5243
.OPENW  000043       446L    1257    1286    5242
.PCHL   002264       785E
.PIP.   000000         1E       4     988    1026    1078    1198    1209    2231    2381    2440    2453    2583
                     2663    2823    2901    2922    2987    3098    3262    3543    5099    5818    5853    5884    5952
                     5979
.POSIT  000047       450L
.PRINT  000003       429L    2983
.RCK    003260       793E
.READ   000004       430L    1300    2467    3681    5594
.REGI   040005       801E
.REGPTR 040035       812E
.RENAM  000051       452L    2319
.RESET  000204       473L    1422
.RNB    002331       788E
.RNP    002325       787E
.SCIN   000001       427L    4900
.SCOUT  000002       428L    4004    4328    4350    4904
.SETTP  000052       453L    3540    3887    5877
.SRS    002265       786E
.START  040000       799E
.SYSRES 000012       436L
.TICCNT 040033       811E
.TPERR  002205       784E
.TPERRX 040031       810E
.UIVEC  040037       813E
.VERS   000011       435L    5872
.WNB    003024       791E
```

```
    .WNP     003017      790E
    .WRITE   000005      431L     1314    5315    5423    5490
    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      043320      959      1196L
    AEN      052347     2682      3018L   3639    3722
    AENA     053021     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.TFF  041114      614L
    AIO.UNI  041061      607L
    AIO.VEC  041040      595L
    ALLOCA   063344      967      1120    2763    5804L
    BELL     000007      484E     1345    2950    2965    3111    5781
    BKSP     000010      486E
    BLS      047302     2391      2634L
    BLS1     047331     2644L     2685
    BLS2     047352     2651      2653L
    BLS3     050011     2666      2673L
    BLS4     050024     2670      2674    2682L
    BLSA     050036     2635      2645    2661    2687L
    BLSB     050044     2639      2664    2669    2688L
    BLSC     050045     2635      2652    2689L
    BOOT.F   000001      575E
    BRIEF    045356      983      2369L
    BSL      053042     1240      2252    2282    3055L
    BSL1     053050     3060L     3076
    BSL2     053103     3073L
    BSLA     053113     3055      3068    3078L
    BUFF     067256      951      5813    5989E
    BUFMINL  002000     3550      5988E
    BUFPTR   063371      952      1296    3545    3774    3884    5813L
    BUFSIZ   063373      948      1293    1309    3549    3776    3883    5814L
    C.STX    000002      488E
    C.SYN    000026      487E
    CAD      054016     2490      3064    3250    3316L   3647    3827    3831
    CAD.     054022     2653      3319L
    CAD0     054024     3317      3320L
    CAD1     054111     3335      3337    3339    3347L
    CAD2     054166     3350      3374L
```

```
CAD2.4  054214      3388L     3391
CAD2.6  054222      3385      3392L
CAD3    054261      3395      3413L
CAD4    054263      3341      3343      3418L
CAD5    054276      3348      3357      3360      3370      3401      3404      3428L
CADA    054302      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
CBT     063005      5267      5383      5515L
CBT1    063014      5520L     5529
CCHIT   042355      1009L     5879
CCW     053114      3070      3097L
CCW1    053123      3101L     3104
CDA     055131      3019      3263      3578L     3848
CDA5    055175      3580      3585      3590      3612L     3624
CDA6    055213      3619      3621L
CDA7    055215      3618      3623L
CDB.H84 000001       518E
CDB.H85 000000       517E
CFE     053233      2483      3138L     3706
CFS     053253      2570      2758      3159L
CFS.    053256      3160L
CFS1    053261      3161L     3166
CN.170M 000014       764E
CN.174M 000003       763E
CN.ABO  000200       768E
CN.BAU  000100       767E
CN.DES  000001        49E     1256      1285      1313      1330      1346      1356      5820
CN.DIR  000002        50E     2432      2465      2543      3670      3679      3734
CN.MEM  000040       766E
CN.PRI  000020       765E
CN.SOU  000000        48E     1270      1298      1320      2308
CND.H17 000000       770E
CND.H47 000001       772E
CND.NDI 000000       771E
CO.FLG  000001       670E     4074
COMAND  063345       965       972      1107      1147      1155      1160      1169      5805L
COPY    043343       979      1232E
COPY1   044027      1254      1263L     1329      1334
COPY2   044114      1278      1292L
COPY3   044117      1293L     1318
COPY4   044153      1303      1305      1309L
COPY5   044241      1266      1338L
COPY6   044301      1340      1350L
COPY7   044327      1354      1364L
COPYA   044372      1237      1252      1276      1327      1352      1376L
COPYC   044373      1234      1267      1338      1377L
COPYD   044374      1245      1279      1378L     1379
COPYDL  000021      1243      1379E
COPYE   044350      1368      1372L
CR      000015       480E
CS.FLG  000200       671E
```

```
CSL.CHR 000001        647E
CSL.ECH 000200        644E
CSL.RAW 000004        645E
CSL.WRP 000002        646E
CTB     063033       5359        5386     5541L
CTB1    063044       5547L       5556
CTLA    000001        495E
CTLB    000002        496E
CTLC    000003        497E       5880
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     053271       1442        2889     3182L
CWM     053306       2495        3206L    3214     3711
CWM1    053315       3208        3211L
D.CON   040110        398L
D.RAM   040240        401L
D.VEC   040130        400L
DAD1    060201       4744        4757     4759     4782L
DAD2    060204       4732        4788L
DADB    060215       4761        4792L
DADC    060261       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
DC.OPR  000003        689L
DC.OPU  000005        691L
DC.OPW  000004        690L
DC.RDY  000012        696L
DC.REA  000000        686L
DC.RER  000002        688L
DC.WRI  000001        687L
DDF     053324       1235        2279     2382     3236L
DDF.BOL 000011        828E
DDF.BOO 000000        827L
DDF.LAB 000011        829L
DDF.USR 000012        830L
DDF1    053331       3239L       3244
DDF1.0  053344       3245L       3252
DDF2    053347       3242        3249L
DDFA    053376       3245        3284L
DEFALT  064034       3063        3249     3903     3948     3952     5948L
DEL1    045127       2237L       2242
DEL2    045147       2240        2251L
DEL5    045157       2257L       2265
```

```
  DELETE  045124       990        2232E
  DESTBFE 065063      5825        5954E
  DESTBUF 064063      3117        3119      5822      5823      5824      5953L
  DESTFB  063375       943        1244      1255      1258      1281      1287      1315      1332      1359      2287      2441      2584
                      2827        3261      5819L
  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
  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       2744      2751
  DIR.FLG 000016       191L       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      5974
  DIRIDL  000015       188E
  DIRNAM  063352      2394        2396      2404      2431      3654      3658      3669      5811L
  DIS.ENL 001373       318L       2521      3690
  DIS.ENT 000000       313E
  DIS.LNK 001376       320L
  DIS.SEC 001374       319L
  DISMOU  045023       994        1405E
  DM.MR   000000       736E
  DM.MW   000001       737E
  DM.RR   000002       738E
  DM.RW   000003       739E
  DNT     054303      3347        3369      3400      3449L
  DNT1    054312      3453L       3456
  DNT2    054323      3461L       3484
  DNT3    054365      3464        3471      3479L
  DNT4    055010      3469        3473      3475      3502L
  DNT5    054377      3467        3493L     3497
  DNTA    055015      3449        3457      3503      3506L
  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      055026     1292       3518L
  EBM1     055066     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      3663
  EC.DSC   000047      379L
  EC.EOF   000001      341L      1304      5603
  EC.EOM   000002      342L
  EC.FAO   000031      365L      5194
  EC.FAP   000026      362L      2312
  EC.FL    000030      364L
  EC.FNF   000014      352L      2315
  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      5927
  EC.NEM   000021      357L      3553      3779
  EC.NOS   000051      381L
  EC.NPM   000044      376L      1456
  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      1345      1373      2909      2988      2989      2990      2991      2992      2993      2994
  ENTRY    064034      898       5871E
  EOFFLG   063161     5575       5602      5629L
  ERROR    051325      971       1112      1153      1236      1241      1444      1449      1458      1461      1464      2247      2253
                      2280       2283      2303      2383      2386      2392      2407      2411      2435      2891      2898      2963L     3554
                      3780       3889
  ERROR1   051356     2968       2975L
  ERROR2   051361     2976L      2978
  ERRORA   051373     2975       2986L
  ESC      000033      491E
```

```
EWS     055222      3072    3639L
EWS1    055331      3677L   3699
EWS3    055363      3697L   3730
EWS4    056042      3712    3727L
EWS6    056043      3702    3707    3728L
EWS7    056051      3683    3705    3734L
EWSA    056056      3738L
EWSB    056064      3657    3661    3740L
EWSC    056122      3656    3709    3742L
EXIT    042352       927     960    1001L   5931
FB.CHA  000000       326L   5178    5181    5488    5499    5615
FB.FLG  000001       327L    943    5181    5198    5236    5443    5455    5458    5617
FB.FWA  000002       328L   5198    5204    5458    5619
FB.LIM  000006       330L   5209    5214    5623
FB.LWA  000010       331L   5625
FB.NAM  000012       332L    333    1244    1255    1281    2287    2310    2923    3261    5214    5782
FB.NAML 000021       333E   1378    2331    2499    3023    3033    3040    3645    3650    3864    3869    5826
                    5950    5976
FB.PTR  000004       329L   5204    5209    5455    5621
FBENL   000033       334E
FF      000014       494E
FT.ABS  000000       875E    895
FT.BAC  000003       878E
FT.DD   000001       281E
FT.OC   000020       285E
FT.OR   000002       282E   5168    5172    5241    5243
FT.OU   000010       284E
FT.OW   000004       283E   5170    5172    5242    5243    5273    5449
FT.PIC  000001       876E
FT.REL  000002       877E
FWBRK2  062236      5403L   5410
FWBRK3  062252      5405    5412L
GDWP    056135      2464    3677    3755L
GDWP.   056143      2519    3688    3756    3760L
I.BRE   000002       982E   1154    1163
I.CONFL 000004       673E    674    4073
I.CONTY 000001       660E    661
I.CONWI 000003       666E    667
I.COF   000000       963     978E
I.CSLMD 000000       649E
I.CUSOR 000002       663E    664    4089
I.DEL   000005       989E   1082
I.DIS   000007       993E   1092
I.LIS   000001       980E   1150    1164    1168
I.MOU   000004       986E   1179
I.REN   000006       991E   1087
I.RES   000010       995E   1097
I.VER   000003       984E   1174
IERR1   051114      2939L   3541
IERR2   051121      2942L
IERR3   051126      2944L
ILDEHL  057346      4566L   4638
INA     056147      3032    3772L
INTERR  051133      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
IOC.DRL 000010      294E
IOC.DTA 000014      296L
IOC.FLG 000004      280L      294
IOC.GRT 000005      288L
IOC.LGN 000012      292L
IOC.LNK 000000      277L
IOC.LSI 000013      293L
IOC.SPG 000007      289L
IOC.SUL 000003      286E
IOC.UNI 000022      299L
IOCCTD  000001      306E
IOCELEN 000052      304E
IF.CON  000362      712E
IF.PAD  000360      708E
ISDEHL  060051     4645      4701L
JGL     063347     1140      5807L
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
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      856E      858       859
LAB.VPR 000014      851E      856
LABEL   066063     5962L
LF      000012      481E
LINE    067136      961      1205      1445      2233      2892      3183      3236      3796      3803      5905      5982L
LIST    045350      981      2366L
LIST1   045361     2367      2372L
LIST1.5 046050     2399      2404L
LIST10  047071     2548      2580L
LIST2   046175     2451      2454L
LIST3   046200     2463L     2476      2529
LIST4   046217     2474L     2528
LIST5   046247     2488L     2512
LIST6   046266     2496L
LIST7   046320     2479      2485      2516L     2539
LIST8   046347     2498      2533L
LIST9   046366     2469      2481      2543L
LSN     056201     2640      3056      3182      3796L
LSN1    056204     3797L     3802
LSTA    047114     2372      2373      2449      2546      2598L     2726      2739
```

```
LSTB     047115       2373       2537       2554       2600L
LSTC     047116       2375       2561       2601L      2773       2775
LSTD     047120       2405       2408       2412       2416       2602L
LSTE     047150       2426       2568       2603L      2755       3159
LSTF     047152       2421       2534       2571       2604L
LSTG     047153       2448       2605L      2608
LSTG1    047211       2377       2606L
LSTGL    000051       2452       2608E
LSTH     047224       2580       2610L      2614
LSTH1    047230       2558       2611L
LSTH2    047251       2564       2612L
LSTH3    047267       2575       2613L
LSTHL    000056       2579       2614E
M.FOX    000303        756E
M.PAM8   000021        755E
MDR.     045037       1392       1407       1423       1441L
MDR1     045066       1450L      1466
MDRA     045070       1441       1452E
MEML     064142        897       5938E      5980
MODE     063346        925        957       5806L      5898
MOUNT    045015        987       1390E
MWN      056221       1283       2292       3824L
MWN1     056254       3838L      3846
MWN2     056262       3840       3842L
MWNA     064042       3829       3832       5950L
NAMERR   051106       1272       1307       1322       2263       2320       2923L
NAMTAB   067256       1269       1280       2261       2288       2486       2507       2923       3033       3645       3869       3870       5985L
NAMTLEN  064030        949       1263       2257       2325       2506       2637       3022       3025       3644       3649       3652       3863
                      3866       5838L
NAMTMAX  064032        950       3027       3772       5839L
NL       000012        492E       493       1371       2605       2607       2607       2610       2613       2736       2816       2950       2954
                      2955       2955       2969       4165       4327       5781       5798       5929
NUL2     000000        483E
NULL     000200        482E
ONECOPY  000001          2E        61        900       2979       5634       5639       5840       5933
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      1459       3521       3527
OVL1     000001        227L      1462       3527
PATCH    063162       5633L
PEC.CS   000204         57E      1111       1152       2991
PEC.DF   000200         54E      1448       2434       2988       3672
PEC.DNC  000201         55E      2302       2675       2989
PEC.IDF  000206         59E      2993       3255
PEC.IUW  000205         58E      2385       2992       3123
PEC.SFI  000207         60E      2896       2994
PEC.TFI  000203         56E      2246       2990       3188
```

```
PFI      050052     2535      2713L
PFI1     050073     2717      2720L
PFI19    050324     2848L     2855
PFI2     050131     2734      2739L
PFI20    050331     2716      2723      2853L
PFI3     050137     2728      2744L
PFI3.5   050202     2765      2767E
PFI4     050264     2806L     2815
PFI5     050275     2807      2813L
PFI5.5   050302     2802      2816L
PFI6     050305     2740      2821L
PFIA     050347     2714      2821      2826      2868L
PFIB     051022     2805      2871L
PFIB1    051025     1142      2872L
PFIC     051032     2713      2759      2877L
PIO.DEV  067063     2660      3112      3325      3358      3366      3579      3654      3944      3947      5971L
PIO.DIR  067066     2491      3097      3329      3379      3407      3584      3656      3720      3829      3833      3951      5974L
PIO.UNI  067065     3358      3363      3944      5972L
PIP      042200      919E
PIP1     042220      929       937L
PIPA     042367      962      1025L
PIPB     042330      977L      978       980       982       984       986       989       991       993       995
PRS      064034     5872L     5946
PRS1     064130     5873      5875      5927L
PRS2     064133     5878      5929L
QUOTE    000047      489E
REN      056275     1323      2264      2324      3863L
REN1     045220     2287L     2328
REN2     045275     2311      2315L
RENA     045327     2289      2307      2310      2318      2331L
RENAME   045203      992      2278E
RESET    045031      996      1421E
RESTART  042200      923E      933      1012      2971      2984      3071      5801
RMEML    067256     5876      5991E
ROMBOOT  030000      393E
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
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.USN   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     056322       1319       3783      3882L
   SC.ACE  000350         69E
   SC.UART 000372        138E
   SDD     056343        937       1238      3901L
   SDDA    056362       3903       3906L
   SFS     056370       2684       3075      3924L
   SFS1    057002       3927       3929L
   SLABEL  065063       5961L
   SND     057005       3066       3943L
   STACK   042200        409E       928      5891      5894
   STACKL  001032        407E
   START   042207        928L      5899      5925
   SUPRES  063350        966       1133      1364      2581      5808L
   SW.ALL  043204       1046       1119L
   SW.BRE  043242       1054       1147L
   SW.BRE1 043257       1149       1154L
   SW.DEL  043142       1029       1082L
   SW.DIS  043154       1037       1092L
   SW.JGL  043225       1074       1139L
   SW.LIS  043265       1050       1160L
   SW.LIS1 043300       1162       1168L
   SW.MOU  043313       1062       1179L
   SW.REN  043147       1033       1087L
   SW.RES  043161       1041       1097L
   SW.SUP  043217       1070       1132L
   SW.SYS  043212       1066       1125L     1143
```

```
SW.VER   043306    1058     1174L
SWIT1    043166    1083     1088     1093     1098     1107L    1175     1180
SYDD     040130     399E
SYSCALL  000377     419E     941     1002     1011     1257     1271     1286     1300     1314     1321     1331     1347
                   1357     1451     1460     1463     2262     2309     2319     2406     2433     2467     2544     2957     2970
                   2983     3540     3659     3671     3681     3735     3887     4004     4075     4091     4328     4350     4900
                   4904     5230     5315     5423     5490     5502     5594     5800     5872     5877     5881     5883     5930
SYSTEM   063351     969     1126     3145     5809L
T.CHA    063147    5314     5422     5518     5546     5593     5616L    5617     5619     5621     5623     5625     5627
T.FLG    063150    5272     5618L
T.FWA    063151    5304     5412     5582     5620L
T.LIM    063155    5586     5606     5608     5624L
T.LWA    063157    5283     5307     5389     5415     5585     5626L
T.PTR    063153    5281     5305     5349     5391     5413     5583     5622L
TAB      000011     490E    2605     2605     2605     2605     2605     2605     2718     2724     2779     2803     2907
                   4239
TBL1     057247    4266L    4272
TBL2     057265    4264     4276L
TBL3     057267    4269     4280L
TLEN     000012    5517     5627E
TPL1     057340    4401L
UC.2SB   000004      95E
UC.5BW   000000      91E
UC.6BW   000001      92E
UC.7BW   000002      93E
UC.8BW   000003      94E
UC.BI    000020     114E
UC.CTS   000020     123E
UC.DCS   000001     119E
UC.DDR   000002     120E
UC.DLA   000200     100E
UC.DR    000001     110E
UC.DRL   000010     122E
UC.DSR   000040     124E
UC.DTR   000001     103E
UC.EDA   000001      81E
UC.EPS   000020      97E
UC.FE    000010     113E
UC.IID   000006      88E
UC.IIP   000001      87E
UC.LOO   000020     107E
UC.MSI   000010      84E
UC.OR    000002     111E
UC.OU1   000004     105E
UC.OU2   000010     106E
UC.PE    000004     112E
UC.PEN   000010      96E
UC.RI    000100     125E
UC.RLS   000200     126E
UC.RSI   000004      83E
UC.RTS   000002     104E
UC.SB    000100      99E
UC.SKP   000040      98E
UC.TER   000004     121E
UC.THE   000040     115E
UC.TRE   000002      82E
UC.TSE   000100     116E
UCI.ER   000020     160E
UCI.IE   000002     162E
UCI.IR   000100     158E
UCI.RE   000004     161E
UCI.RO   000040     159E
UCI.TE   000001     163E
UDDN1    060276    4814L      4830
UDDN1.5  060330    4834L      4841
UDDN2    060332    4827       4839L
UDDN3    060333    4840L      4844
UDR      000000     135E
UMI.16X  000002     153E
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
UO.DDU   000002     747E
UO.HLT   000200     745E
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      5874
VERSN    051033     985       2887E
VFL.NSD  000001     855E
XCHGBC   061001    4635       4639      4647      4649      4921L

13594 BYTES FREE
```