```
000.001      QUERYF  EQU     1                Don't Assemble for Query
             ***     SYSGEN IS A CUT-DOWN FROM PIP.
             *
             ***     PIP - PERIHPERAL INTERCHANGE PROGRAM.
             *
             *          J. G. L.: 11/1977 FOR *HEATH* COMPANY
             *
             *       COPYRIGHT 1977, 1979 BY HEATH COMPANY
             *
             *          G. C.:     9/78       Maintenence release
             *                        79/04   Issue --.04.--
             *                        79/11   Issue --.05.--
             *                        80/07   Issue --.06.--
             *
             *       80.07.sc
             *               Linked list structure modified to remove page boundary
             *               requirements.
             *               H17 dependency removed.
             *               Multiple Unit
             *               Multiple Device.
             *               Command Line Switch Processing
             *


             ***     USE:
             *
             *               Command Line File specification replacing default.
             *               /Minimal Switch
             *               Destination Device specification
             *


             **      Assembly Constants                                      /80.07.sc/

000.010      FDNCNT  EQU     8                Number of File Descriptor Nodes


             **      SYSTEM EQUIVALENCES

000.000      CN.SOU  EQU     0                SOURCE CHANNEL NUMBER
000.001      CN.DES  EQU     1                DESTINATION CHANNEL NUMBER
000.002      CN.DIR  EQU     2                DIRECTORY CHANNEL NUMBER


             **      PROGRAM ERROR CODES      (Must not equal ENL)

000.200      PEC.DF  EQU     200Q             DEVICE FORMAT ERROR
000.201      PEC.DNC EQU     201Q             DEVICES NOT CONSISTANT
000.202      PEC.RSE EQU     202Q             RENAME SPECIFICATION ERROR
000.203      PEC.TFI EQU     203Q             TARGET FILE ILLEGAL
000.204      PEC.CS  EQU     204Q             CONTRADICTORY SWITCHES
000.205      PEC.IUW EQU     205Q             ILLEGAL USE OF WILDCARD
```

```
000.206            PEC.IDF EQU     206Q          ILLEGAL DESTINATION FILE FORMAT
000.207            PEC.CO  EQU     207Q          Command Overflow               /80.07.sc/

```

```
 000.000             56           XTEXT   DIRDEF


                     58X **       DIRECTORY ENTRY FORMAT.
                     59X
 000.000             60X          ORG     0
                     61X
                     62X
 000.377             63X DF.EMP   EQU     377Q               FLAGS ENTRY EMPTY
 000.376             64X DF.CLR   EQU     376Q               FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
                     65X
 000.000             66X DIR.NAM  DS      8                  NAME
 000.010             67X DIR.EXT  DS      3                  EXTENSION
 000.013             68X DIR.PRO  DS      1                  PROJECT
 000.014             69X DIR.VER  DS      1                  VERSION
 000.015             70X DIRIDL   EQU     *                  FILE IDENTIFICATION LENGTH
                     71X
 000.015             72X DIR.CLU  DS      1                  CLUSTER FACTOR
 000.016             73X DIR.FLG  DS      1                  FLAGS
 000.017             74X          DS      1                  RESERVED
 000.020             75X DIR.FGN  DS      1                  FIRST GROUP NUMBER
 000.021             76X DIR.LGN  DS      1                  LAST GROUP NUMBER
 000.022             77X DIR.LSI  DS      1                  LAST SECTOR INDEX (IN LAST GROUP)
 000.023             78X DIR.CRD  DS      2                  CREATION DATE
 000.025             79X DIR.ALD  DS      2                  LAST ALTERATION DATE
                     80X
 000.027             81X DIRELEN  EQU     *                  DIRECTORY ENTRY LENGTH
 000.027             82           XTEXT   DIFDEF


                     84X **       DIRECTORY FILE FLAGS.
                     85X
 000.200             86X DIF.SYS  EQU     10000000B                    SYSTEM FILE
 000.100             87X DIF.LOC  EQU     01000000B                    LOCKED FOR CHANGE
 000.040             88X DIF.WP   EQU     00100000B                    WRITE PROTECTED
 000.020             89X DIF.CNT  EQU     00010000B                    CONTIGUOUS FILE
                     90X
 000.027             91           XTEXT   DEVDEF


                     93X **       DEVICE TABLE ENTRYS.
                     94X
 000.000             95X          ORG     0
                     96X
 000.000             97X DEV.NAM  DS      2                  DEVICE NAME
 000.000             98X DV.EL    EQU     00000000B          END OF DEVICE LIST FLAG
 000.001             99X DV.NU    EQU     00000001B          DEVICE ENTRY NOT IN USE
                    100X
 000.002            101X DEV.RES  DS      1                  DRIVER RESIDENSE CODE
 000.001            102X DR.IM    EQU     00000001B          DRIVER IN MEMORY
 000.002            103X DR.PR    EQU     00000010B          DRIVER PERMINANTLY RESIDENT
```

```
                       104X
000.003                105X DEV.JMP DS      1               JMP TO PROCESSOR
000.004                106X DEV.DDA DS      2               DRIVER ADDRESS
000.006                107X DEV.FLG DS      1               FLAG BYTE
000.001                108X DT.DD   EQU     00000001B       DIRECTORY DEVICE
000.002                109X DT.CR   EQU     00000010B       CAPABLE OF READ OPERATION
000.004                110X DT.CW   EQU     00000100B       CAPABLE OF WRITE OPERATION
000.010                111X DT.RN   EQU     00001000B       Capable of random access        /80.02.sc/
000.020                112X DT.CH   EQU     00010000B       Capable of Character mode       /80.02.sc/
                       113X
000.007                114X DEV.MUM DS      1               MOUNTED UNIT MASK
000.010                115X DEV.MNU DS      1               MAXIMUM NUMBER OF UNITS
000.011                116X DEV.UNT DS      2               ADDRESS OF UNIT SPECIFIC DATA TABLE
                       117X
000.013                118X DEV.DVL DS      2               DRIVER BYTE LENGTH
000.015                119X DEV.DVG DS      1               DRIVER ROUTINE GROUP ADDRESS
                       120X
000.016                121X DEVELEN EQU     *               DEVICE TABLE ENTRY LENGTH


                       123X **      UNIT SPECIFIC DEVICE DATA TABLE ENTRIES
                       124X
000.000                125X         ORG     0
                       126X
000.000                127X UNT.FLG DS      1               UNIT SPECIFIC  *DEV.FLG*
000.001                128X UNT.SPG DS      1               Sectors Per Group               /80.04.GC/
000.002                129X UNT.GRT DS      2               ADDRESS OF GROUP RESERVATION TABLE (IF DT.DD)
000.004                130X UNT.GTS DS      2               GRT SECTOR NUMBER
000.006                131X UNT.DIS DS      2               DIRECTORY FIRST SECTOR NUMBER
                       132X
000.010                133X UNT.SIZ EQU     *               SIZE OF UNIT SPECIFIC DATA TABLE PER UNIT
000.010                134          XTEXT   IOCDEF


                       136X **      I/O CHANNEL DEFINITIONS.
                       137X
000.000                138X         ORG     0
                       139X
000.000                140X IOC.LNK DS      2               ADDRESS OF NEXT CHANNEL, =0 IF LAST
000.002                141X IOC.DDA DS      2               THREAD JUMP TO DEVICE DRIVER (VIA DEV TABLE)
                       142X
000.004                143X IOC.FLG DS      1               FILE TYPE FLAGS
000.001                144X FT.DD   EQU     00000001B       =1 IF DIRECTORY DEVICE
000.002                145X FT.OR   EQU     00000010B       =1 IF OPEN FOR READ
000.004                146X FT.OW   EQU     00000100B       =1 IF OPEN FOR WRITE
000.010                147X FT.OU   EQU     00001000B       =1 IF OPEN FOR UPDATE
000.020                148X FT.OC   EQU     00010000B       =1 IF OPEN FOR CHARACTER MODE   /80.02.GC/
000.003                149X IOC.SQL EQU     *-IOC.DDA       LENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
                       150X
000.005                151X IOC.GRT DS      2               ADDRESS OF GROUP RESERVATION TABLE
000.007                152X IOC.SPG DS      1               SECTORS PER GROUP, THIS DEVICE
000.010                153X IOC.CGN DS      1               CURRENT GROUP NUMBER
000.011                154X IOC.CSI DS      1               CURRENT SECTOR INDEX (IN CURRENT GROUP)
```

```
     000.012          155X IOC.LGN DS      1               LAST GROUP NUMBER
     000.013          156X IOC.LSI DS      1               LAST SECTOR INDEX (IN LAST GROUP)
     000.010          157X IOC.DRL EQU     *-IOC.FLG       LENGTH OF INFO NORMALLY COPIED BACK TO
                      158X *                               THE CHANNEL TABLE
     000.014          159X IOC.DTA DS      2               DEVICE TABLE ADDRESS FOR THIS DEVICE
     000.016          160X IOC.DES DS      2               SECTOR NUMBER OF DIRECTORY ENTRY
     000.020          161X IOC.DEV DS      2               DEVICE CODE
     000.022          162X IOC.UNI DS      1               UNIT NUMBER (0-9)
     000.021          163X IOC.DIL EQU     *-IOC.DDA       LENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)
                      164X
     000.023          165X IOC.DIR DS      DIRELEN         DIRECTORY ENTRY
                      166X
     000.052          167X IOCELEN EQU     *               IOC ENTRY LENGTH
                      168X
     000.001          169X IOCCTD  EQU     1               INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)
     000.052          170          XTEXT   DISDEF


                      172X **      DIRECTORY BLOCK FORMAT.
                      173X
     000.000          174X         ORG     0
                      175X
     000.000          176X DIS.ENT EQU     *               FIRST ENTRY ADDRESS
     000.000          177X         DS      22*DIRELEN      22 DIRECTORY ENTRYS PER BLOCK
     001.372          178X         DS      1               0 BYTE = END OF ENTRYS IN THIS BLOCK
                      179X
     001.373          180X         ORG     512-5           AT END OF BLOCK
     001.373          181X DIS.ENL DS      1               LENGTH OF EACH ENTRY (=DIRELEN)
     001.374          182X DIS.SEC DS      2               BLOCK # OF THIS BLOCK.
     001.376          183X DIS.LNK DS      2               BLOCK # OF NEXT BLOCK, =0 IF THIS IS LAST
     002.000          184          XTEXT   FBDEF


                      186X **      FILE BLOCK DEFINITIONS.
                      187X
     000.000          188X         ORG     0
     000.000          189X FB.CHA  DS      1               CHANNEL NUMBER
     000.001          190X FB.FLG  DS      1               FLAGS
     000.002          191X FB.FWA  DS      2               BUFFER FWA
     000.004          192X FB.PTR  DS      2               BUFFER POINTER
     000.006          193X FB.LIM  DS      2               LIMIT OF DATA IN BUFFER (READ OPERATIONS)
     000.010          194X FB.LWA  DS      2               LWA OF BUFFER
     000.012          195X FB.NAM  DS      4+8+4+1         NAME OF FILE
     000.021          196X FB.NAML EQU     *-FB.NAM
     000.033          197X FBENL   EQU     *               ENTRY LENGTH
     000.033          198          XTEXT   ECDEF
```

```
                    200X **       ERROR CODE DEFINITIONS.
                    201X
    000.000         202X          ORG      0
    000.000         203X          DS       1           NO ERROR #0
    000.001         204X EC.EOF   DS       1           END OF FILE
    000.002         205X EC.EOM   DS       1           END OF MEDIA
    000.003         206X EC.ILC   DS       1           ILLEGAL SYSCALL CODE
    000.004         207X EC.CNA   DS       1           CHANNEL NOT AVAILABLE
    000.005         208X EC.DNS   DS       1           DEVICE NOT SUITABLE
    000.006         209X EC.IDN   DS       1           ILLEGAL DEVICE NAME
    000.007         210X EC.IFN   DS       1           ILLEGAL FILE NAME
    000.010         211X EC.NRD   DS       1           NO ROOM FOR DEVICE DRIVER
    000.011         212X EC.FNO   DS       1           CHANNEL NOT OPEN
    000.012         213X EC.ILR   DS       1           ILLEGAL REQUEST
    000.013         214X EC.FUC   DS       1           FILE USAGE CONFLICT
    000.014         215X EC.FNF   DS       1           FILE NAME NOT FOUND
    000.015         216X EC.UND   DS       1           UNKNOWN DEVICE
    000.016         217X EC.ICN   DS       1           ILLEGAL CHANNEL NUMBER
    000.017         218X EC.DIF   DS       1           DIRECTORY FULL
    000.020         219X EC.IFC   DS       1           ILLEGAL FILE CONTENTS
    000.021         220X EC.NEM   DS       1           NOT ENOUGH MEMORY
    000.022         221X EC.RF    DS       1           READ FAILURE
    000.023         222X EC.WF    DS       1           WRITE FAILURE
    000.024         223X EC.WPV   DS       1           WRITE PROTECTION VIOLATION
    000.025         224X EC.WP    DS       1           DISK WRITE PROTECTED
    000.026         225X EC.FAP   DS       1           FILE ALREADY PRESENT
    000.027         226X EC.DDA   DS       1           DEVICE DRIVER ABORT
    000.030         227X EC.FL    DS       1           FILE LOCKED
    000.031         228X EC.FAO   DS       1           FILE ALREADY OPEN
    000.032         229X EC.IS    DS       1           ILLEGAL SWITCH
    000.033         230X EC.UUN   DS       1           UNKNOWN UNIT NUMBER
    000.034         231X EC.FNR   DS       1           FILE NAME REQUIRED
    000.035         232X EC.DIW   DS       1           DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
    000.036         233X EC.UNA   DS       1           UNIT NOT AVAILABLE
    000.037         234X EC.ILV   DS       1           ILLEGAL VALUE
    000.040         235X EC.ILO   DS       1           ILLEGAL OPTION
    000.041         236X EC.VPM   DS       1           VOLUME PRESENTLY MOUNTED ON DEVICE
    000.042         237X EC.NVM   DS       1           NO VOLUME PRESENTLY MOUNTED
    000.043         238X EC.FOD   DS       1           FILE OPEN ON DEVICE
    000.044         239X EC.NPM   DS       1           NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
    000.045         240X EC.DNI   DS       1           DISK NOT INITIALIZED
    000.046         241X EC.DNR   DS       1           DISK IS NOT READABLE
    000.047         242X EC.DSC   DS       1           DISK STRUCTURE IS CORRUPT
    000.050         243X EC.NCV   DS       1           NOT CORRECT VERSION OF HDOS
    000.051         244X EC.NOS   DS       1           NO OPERATING SYSTEM MOUNTED
    000.052         245X EC.IOI   DS       1           ILLEGAL OVERLAY INDEX
    000.053         246X EC.OTL   DS       1           OVERLAY TO LARGE
    000.054         247           XTEXT    OVLDEF
```

```
                     249X **      OVERLAY TABLE ENTRYS.
                     250X
000.000              251X         ORG     0
                     252X
000.000              253X OVL.COD DS      2               FIRST SECTOR OF OVERLAY CODE
000.002              254X OVL.SIZ DS      2               OVERLAY SIZE
000.004              255X OVL.ENT DS      2               OVERLAY ENTRY POINT
000.006              256X OVL.FLB DS      1               OVERLAY FLAG BYTE
000.007              257X         DS      1               DUMMY BYTE TO ROUND TABLE SIZE UP TO 8
000.010              258X OVL.ENS EQU     *               OVERLAY ENTRY SIZE
                     259X
                     260X *       OVERLAY INDICES
                     261X
000.000              262X         ORG     0
                     263X
000.000              264X OVL0    DS      1
000.001              265X OVL1    DS      1
000.002              266          XTEXT   HOSEQU


                     268X **      HDOS SYSTEM EQUIVALENCES.
                     269X *
                     270X
024.000              271X S.GRT0  EQU     24000A          SYSTEM AREA FOR  GRT0
025.000              272X S.GRT1  EQU     25000A          SYSTEM AREA FOR  GRT1
026.000              273X S.GRT2  EQU     26000A          SYSTEM AREA FOR  GRT2
                     274X
030.000              275X ROMBOOT EQU     30000A          ROM BOOT ENTRY
                     276X
040.100              277X         ORG     40100A          FREE SPACE FROM PAM-8
                     278X
040.100              279X         DS      8               JUMP TO SYSTEM EXIT
040.110              280X D.CON   DS      16              DISK CONSTANTS
040.130              281X SYDD    EQU     *               SYSTEM DISK ENTRY POINT
040.130              282X D.VEC   DS      24*3            SYSTEM ROM ENTRY VECTORS
040.240              283X D.RAM   DS      31              SYSTEM ROM WORK AREA
040.277              284X S.VAL   DS      36              SYSTEM VALUES
040.343              285X S.INT   DS      115             SYSTEM INTERNAL WORK AREAS
041.126              286X         DS      16
041.146              287X S.SOVR  DS      2               STACK OVERFLOW WARNING
041.150              288X         DS      42200A-*        SYSTEM STACK
001.032              289X STACKL  EQU     *-S.SOVR        STACK SIZE
                     290X
042.200              291X STACK   EQU     *               LWA+1 SYSTEM STACK
042.200              292X USERFWA EQU     *               USER FWA
042.200              293          XTEXT   HOSDEF
```

```
                      295X **      HOSDEF - DEFINE HOS PARAMETER.
                      296X *
                      297X
                      298X
    000.040           299X VERS    EQU     2*16+0          VERSION 2.0
                      300X
    000.377           301X SYSCALL EQU     377Q            SYSCALL INSTRUCTION
                      302X
                      303X
    000.000           304X         ORG     0
                      305X
                      306X *       RESIDENT FUNCTIONS
                      307X
    000.000           308X .EXIT   DS      1               EXIT (MUST BE FIRST)
    000.001           309X .SCIN   DS      1               SCIN
    000.002           310X .SCOUT  DS      1               SCOUT
    000.003           311X .PRINT  DS      1               PRINT
    000.004           312X .READ   DS      1               READ
    000.005           313X .WRITE  DS      1               WRITE
    000.006           314X .CONSL  DS      1               SET/CLEAR CONSOLE OPTIONS
    000.007           315X .CLRCO  DS      1               CLEAR CONSOLE BUFFER
    000.010           316X .LOADO  DS      1               LOAD AN OVERLAY
    000.011           317X .VERS   DS      1               RETURN HDOS VERSION NUMBER
    000.012           318X .SYSRES DS      1               PRECEDING FUNCTIONS ARE RESIDENT
                      319X
                      320X
                      321X *       *HDOSOVL0.SYS*  FUNCTIONS
                      322X
    000.040           323X         ORG     40A
                      324X
    000.040           325X .LINK   DS      1               LINK  (MUST BE FIRST)
    000.041           326X .CTLC   DS      1               CTL-C
    000.042           327X .OPENR  DS      1               OPENR
    000.043           328X .OPENW  DS      1               OPENW
    000.044           329X .OPENU  DS      1               OPENU
    000.045           330X .OPENC  DS      1               OPENC
    000.046           331X .CLOSE  DS      1               CLOSE
    000.047           332X .POSIT  DS      1               POSITION
    000.050           333X .DELET  DS      1               DELETE
    000.051           334X .RENAM  DS      1               RENAME
    000.052           335X .SETTP  DS      1               SETTOP
    000.053           336X .DECODE DS      1               NAME DECODE
    000.054           337X .NAME   DS      1               GET FILE NAME FROM CHANNEL
    000.055           338X .CLEAR  DS      1               CLEAR CHAN
    000.056           339X .CLEARA DS      1               CLEAR ALL CHANS
    000.057           340X .ERROR  DS      1               LOOKUP ERROR
    000.060           341X .CHFLG  DS      1               CHANGE FLAGS
    000.061           342X .DISMT  DS      1               FLAG SYSTEM DISK DISMOUNTED
    000.062           343X .LOADD  DS      1               LOAD DEVICE DRIVER
    000.063           344X .OPEN   DS      1               Parametrized Open
                      345X
                      346X
                      347X *       *HDOSOVL1.SYS*  FUNCTIONS
                      348X
    000.200           349X         ORG     200Q
                      350X
```

```
       000.200             351X .MOUNT  DS      1               MOUNT  (MUST BE FIRST)
       000.201             352X .DMOUN  DS      1               DISMOUNT
       000.202             353X .MONMS  DS      1               MOUNT/NO MESSAGE
       000.203             354X .DMNMS  DS      1               DISMOUNT/NO MESSAGE
       000.204             355X .RESET  DS      1               RESET = DISMOUNT/MOUNT OF UNIT
       000.205             356X .CLEAN  DS      1               Clean device
       000.206             357X .DAD    DS      1               Dismount All Disks              /80.08.sc/
       000.207             358          XTEXT   ASCII


                           360X **      ASCII CHARACTER EQUIVALENCES.
                           361X
       000.015             362X CR      EQU     13              CARRIAGE RETURN
       000.012             363X LF      EQU     10              LINE FEED
       000.200             364X NULL    EQU     200Q            PAD CHARACTER
       000.000             365X NUL2    EQU     0
       000.007             366X BELL    EQU     7               BELL CHARACTER
       000.177             367X RUBOUT  EQU     177Q
       000.010             368X BKSP    EQU     10Q             CTL-H
       000.026             369X C.SYN   EQU     26Q             SYNC
       000.002             370X C.STX   EQU     2               STX
       000.047             371X QUOTE   EQU     47Q
       000.011             372X TAB     EQU     11Q
       000.033             373X ESC     EQU     33Q
       000.012             374X NL      EQU     12Q             NEW LINE (HDOS SYSTEMS)
       000.212             375X ENL     EQU     NL+200Q         NL + END-OF-LINE-FLAG
       000.014             376X FF      EQU     14Q             FORM FEED
       000.001             377X CTLA    EQU     01Q             CTL-A
       000.002             378X CTLB    EQU     02Q             CTL-B
       000.003             379X CTLC    EQU     03Q             CTL-C
       000.004             380X CTLD    EQU     04Q             CTL-D
       000.017             381X CTLO    EQU     17Q             CTL-O
       000.020             382X CTLP    EQU     20Q             CTL-P
       000.021             383X CTLQ    EQU     21Q             CTL-Q
       000.023             384X CTLS    EQU     23Q             CTL-S
       000.032             385X CTLZ    EQU     32Q             CTL-Z
       000.207             386          XTEXT   ESINT



                           388X **      S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
                           389X *
                           390X *       THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
                           391X *       MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
                           392X
                           393X
       040.343             394X         ORG     S.INT
                           395X
                           396X **      CONSOLE STATUS FLAGS
                           397X
       040.343             398X S.CDB   DS      1               CONSOLE DESCRIPTOR BYTE
       000.000             399X CDB.H85 EQU     00000000B
       000.001             400X CDB.H84 EQU     00000001B       =0 IF H8-5, =1 IF H8-4
       040.344             401X S.BAUD  DS      2               [0-14]  H8-4 BAUD RATE, =0 IF H8-5
                           402X *                               [15]    =1 IF BAUD RATE => 2 STOP BITS
```

```
                       403X
                       404X **      TABLE ADDRESS WORDS
                       405X
040.346                406X S.DLINK DS      2               ADDRESS OF DATA IN HDOS CODE
040.350                407X S.OFWA  DS      2               FWA  OVERLAY  TABLE
040.352                408X S.CFWA  DS      2               FWA  CHANNEL  TABLE
040.354                409X S.DFWA  DS      2               FWA  DEVICE   TABLE
040.356                410X S.RFWA  DS      2               FWA  RESIDENT HDOS CODE
                       411X
                       412X **      DEVICE DRIVER DELAYED LOAD FLAGS
                       413X
040.360                414X S.DDLDA DS      2               DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
040.362                415X S.DDLEN DS      2               CODE LENGTH IN BYTES
040.364                416X S.DDGRP DS      1               GROUP NUMBER FOR DRIVER
040.365                417X         DS      1               HOLD PLACE
                       418X *S.DDSEC        DS      2               SECTOR NUMBER FOR DRIVER ( * OBSOLETE ! * )
040.366                419X S.DDDTA DS      2               DEVICE'S ADDRESS IN DEVLST +DEV.RES
040.370                420X S.DDOPC DS      1               OPEN OPCODE PENDEDING
                       421X
                       422X **      OVERLAY MANAGEMENT FLAGS
                       423X
000.001                424X OVL.IN  EQU     00000001B       IN MEMORY
000.002                425X OVL.RES EQU     00000010B       PERMINANTLY RESIDENT
000.014                426X OVL.NUM EQU     00001100B       OVERLAY NUMBER MASK
000.200                427X OVL.UCS EQU     10000000B       USER CODE SWAPPED FOR OVERLAY
                       428X
040.371                429X S.OVLFL DS      1               OVERLAY FLAG
040.372                430X S.UCSF  DS      2               FWA SWAPPED USER CODE
040.374                431X S.UCSL  DS      2               LENGTH SWAPPED USER CODE
040.376                432X S.OVLS  DS      2               SIZE OF OVERLAY CODE
041.000                433X S.OVLE  DS      2               ENTRY POINT OF OVERLAY CODE
                       434X
041.002                435X S.SSN   DS      2               SWAP AREA SECTOR NUMBER
041.004                436X S.OSN   DS      2               OVERLAY SECTOR NUMBER
                       437X
                       438X *       SYSCALL PROCESSING WORK AREAS
                       439X
041.006                440X S.CACC  DS      1               (ACC) UPON SYSCALL
041.007                441X S.CODE  DS      1               SYSCALL INDEX IN PROGRESS
                       442X
                       443X *       JUMPS TO ROUTINES IN RESIDENT HDOS CODE
                       444X
041.010                445X S.JUMPS DS      0               START OF DUMP VECTORS
041.010                446X S.SDD   DS      3               JUMP TO STAND-IN DEVICE DRIVER
041.013                447X S.FASER DS      3               JUMP TO FATSERR (FATAL SYSTEM ERROR)
041.016                448X S.DIREA DS      3               JUMP TO DIREAD (DISK FILE READ)
041.021                449X S.FCI   DS      3               JUMP TO FCI (FETCH CHANNEL INFO)
041.024                450X S.SCI   DS      3               JUMP TO SCI (STORE CHANNEL INFO)
041.027                451X S.GUP   DS      3               JUMP TO GUP (GET UNIT POINTER)
                       452X
041.032                453X S.MOUNT DS      1               <>0 IF THE SYSTEM DISK IS MOUNTED
041.033                454X S.DCS   DS      1               DEFAULT CLUSTER SIZE-1
                       455X
041.034                456X S.BOOTF DS      1               BOOT FLAGS
000.001                457X BOOT.P  EQU     00000001B       EXECUTE PROLOGUE UPON BOOTUP
                       458X
```

```
                        459X *        STACK VALUE SAVED FOR OVERLAY SYSCALLS
                        460X
      041.035           461X S.OVSTK DS       2              VALUE OF SP UPON SYSCALLS USING OVERLAY
                        462X
      041.037           463X         DS       1              RESERVED


                        465X **       ACTIVE I/O AREA.
                        466X *
                        467X *        THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION
                        468X *        CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM
                        469X *        THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.
                        470X *
                        471X *        NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY
                        472X *        FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE
                        473X *        8080 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY
                        474X *        COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND
                        475X *        BACKDATED AFTER PROCESSING.
                        476X
      041.040           477X AIO.VEC DS       3              JUMP INSTRUCTION
      041.041           478X AIO.DDA EQU      *-2            DEVICE DRIVER ADDRESS
      041.043           479X AIO.FLG DS       1              FLAG BYTE
      041.044           480X AIO.GRT DS       2              ADDRESS OF GROUP RESERV TABLE
      041.046           481X AIO.SPG DS       1              SECTORS PER GROUP
      041.047           482X AIO.CGN DS       1              CURRENT GROUP NUMBER
      041.050           483X AIO.CSI DS       1              CURRENT SECTOR INDEX
      041.051           484X AIO.LGN DS       1              LAST GROUP NUMBER
      041.052           485X AIO.LSI DS       1              LAST SECTOR INDEX
      041.053           486X AIO.DTA DS       2              DEVICE TABLE ADDRESS
      041.055           487X AIO.DES DS       2              DIRECTORY SECTOR
      041.057           488X AIO.DEV DS       2              DEVICE CODE
      041.061           489X AIO.UNI DS       1              UNIT NUMBER (0-9)
                        490X
      041.062           491X AIO.DIR DS       DIRELEN        DIRECTORY ENTRY
                        492X
      041.111           493X AIO.CNT DS       1              SECTOR COUNT
      041.112           494X AIO.EOM DS       1              END OF MEDIA FLAG
      041.113           495X AIO.EOF DS       1              END OF FILE FLAG
      041.114           496X AIO.TFP DS       2              TEMP FILE POINTERS
      041.116           497X AIO.CHA DS       2              ADDRESS OF CHANNEL BLOCK (IOC.DDA)


      041.120           499X S.BDA   DS       1              Boot Device Address (Setup by ROM) /80.09.sc/
      041.121           500X S.SCR   DS       2              SYSTEM SCRATCH AREA ADDRESS
      041.123           501          XTEXT    ESVAL
```

```
                         503X **        S.VAL - SYSTEM VALUE DEFINTIONS.
                         504X *
                         505X *         THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.
                         506X *
                         507X *         THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.
                         508X
                         509X
     040.277             510X           ORG     S.VAL
                         511X
     040.277             512X S.DATE  DS      9               SYSTEM DATE (IN ASCII)
     040.310             513X S.DATC  DS      2               CODED DATE
     040.312             514X S.TIME  DS      4               TIME FROM MIDNIGHT (IN TICS)
     040.316             515X S.HIMEM DS      2               HARDWARE HIGH MEMORY ADRESS+1
                         516X
     040.320             517X S.SYSM  DS      2               FWA RESIDENT SYSTEM
                         518X
     040.322             519X S.USRM  DS      2               LWA USER MEMORY
                         520X
     040.324             521X S.OMAX  DS      2               MAX OVERLAY SIZE FOR SYSTEM
                         522X
                         523X
                         524X **        THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL
                         525X
     000.200             526X CSL.ECH EQU     10000000B       SUPPRESS ECHO
     000.004             527X CSL.RAW EQU     00000100B       Raw Mode I/O                  /80.09.sc/
     000.002             528X CSL.WRP EQU     00000010B       WRAP LINES AT WIDTH
     000.001             529X CSL.CHR EQU     00000001B       OPERATE IN CHARACTER MODE
                         530X
     000.000             531X I.CSLMD EQU     0               S.CSLMD IS FIRST BYTE
     040.326             532X S.CSLMD DS      1               CONSOLE MODE
                         533X
     000.200             534X CTP.BKS EQU     10000000B       TERMINAL PROCESSES BACKSPACES
     000.100             535X CTP.FF  EQU     01000000B       Terminal Processes Form-Feed    /80.09.sc/
     000.040             536X CTP.MLI EQU     00100000B       MAP LOWER CASE TO UPPER ON INPUT
     000.020             537X CTP.MLO EQU     00010000B       MAP LOWER CASE TO UPPER ON OUTPUT
     000.010             538X CTP.2SB EQU     00001000B       TERMINAL NEEDS TWO STOP BITS
     000.002             539X CTP.BKM EQU     00000010B       MAP BKSP (UPON INPUT) TO RUBOUT
     000.001             540X CTP.TAB EQU     00000001B       TERMINAL SUPPORTS TAB CHARACTERS
                         541X
     000.001             542X I.CONTY EQU     1               S.CONTY IS 2ND BYTE
     000.000             543X         ERRNZ   *-S.CSLMD-I.CONTY
     040.327             544X S.CONTY DS      1               CONSOLE TYPE FLAGS
     000.002             545X I.CUSOR EQU     2               S.CUSOR IS 3RD BYTE
     000.000             546X         ERRNZ   *-S.CSLMD-I.CUSOR
     040.330             547X S.CUSOR DS      1               CURRENT CURSOR POSITION
     000.003             548X I.CONWI EQU     3               S.CONWI IS 4TH BYTE
     000.000             549X         ERRNZ   *-S.CSLMD-I.CONWI
     040.331             550X S.CONWI DS      1               CONSOLE WIDTH
                         551X
     000.001             552X CO.FLG  EQU     00000001B       CTL-O FLAG
     000.200             553X CS.FLG  EQU     10000000B       CTL-S FLAG
                         554X
     000.004             555X I.CONFL EQU     4               S.CONFL IS 5TH BYTE
     000.000             556X         ERRNZ   *-S.CSLMD-I.CONFL
     040.332             557X S.CONFL DS      1               CONSOLE FLAGS
                         558X
```

```
040.333             559X S.CAADR DS      2                ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040.335             560X S.CCTAB DS      6                ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
040.343             561          XTEXT   DDDEF


                    563X **      DEVICE DRIVER COMMUNICATION FLAGS.
                    564X *
                    565X
000.000             566X         ORG     0
                    567X
000.000             568X DC.REA  DS      1                READ
000.001             569X DC.WRI  DS      1                WRITE
000.002             570X DC.RER  DS      1                READ REGARDLESS
000.003             571X DC.OPR  DS      1                OPEN FOR READ
000.004             572X DC.OPW  DS      1                OPEN FOR WRITE
000.005             573X DC.OPU  DS      1                OPEN FOR UPDATE
000.006             574X DC.CLO  DS      1                CLOSE
000.007             575X DC.ABT  DS      1                ABORT
000.010             576X DC.MOU  DS      1                MOUNT DEVICE
000.011             577X DC.LOD  DS      1                LOAD DEVICE DRIVER
000.012             578X DC.RDY  DS      1                Device Ready                     /80.04.GC/
000.013             579X DC.MAX  DS      1                MAXIMUM ENTRY INDEX
000.014             580          XTEXT   MTR
```

```
                         583X **       MTR - PAM/8 EQUIVALENCES.
                         584X *
                         585X *        THIS DECK CONTAINS SYMBOLIC DEFINITIONS USED TO
                         586X *        MAKE USE OF THE PAM/8 CODE AND CONTROL BYTES.

                         588X **       IO PORTS
                         589X
      000.360            590X IP.PAD   EQU     360Q            PAD INPUT PORT
      000.360            591X OP.CTL   EQU     360Q            CONTROL OUTPUT PORT
      000.360            592X OP.DIG   EQU     360Q            DIGIT SELECT OUTPUT PORT
      000.361            593X OP.SEG   EQU     361Q            SEGMENT SELECT OUTPUT PORT
      000.362            594X IP.CON   EQU     362Q            H-88/H-89/HA-8-8 Configuration  /80.07.gc/
      000.362            595X OP2.CTL  EQU     362Q            H-88/H-89/HA-8-8 Control Port   /80.07.gc/

                         597X **       FRONT PANEL CONTROL BITS.                                /80.07.gc/
                         598X *
                         599X *        CB.*  set in OP.CTL
                         600X *        CB2.* set in OP2.CTL
                         601X *
                         602X
      000.020            603X CB.SSI   EQU     00010000B       SINGLE STEP INTERRUPT
      000.040            604X CB.MTL   EQU     00100000B       MONITOR LIGHT
      000.100            605X CB.CLI   EQU     01000000B       CLOCK INTERRUPT ENABLE
      000.200            606X CB.SPK   EQU     10000000B       SPEAKER ENABLE
                         607X
      000.001            608X CB2.SSI  EQU     00000001B       Single Step Interrupt
      000.002            609X CB2.CLI  EQU     00000010B       Clock Interrupt Enable
      000.040            610X CB2.ORG  EQU     00100000B       ORG 0 Select
      000.100            611X CB2.SID  EQU     01000000B       Side 1 Select

                         613X **       Secondary Control Bits
                         614X

                         616X **       MONITOR MODE FLAGS.
                         617X
      000.000            618X DM.MR    EQU     0               MEMORY READ
      000.001            619X DM.MW    EQU     1               MEMORY WRITE
      000.002            620X DM.RR    EQU     2               REGISTER READ
      000.003            621X DM.RW    EQU     3               REGISTER WRITE
```

```
                           623X **       USER OPTION BITS.
                           624X *
                           625X *        THESE BITS ARE SET IN CELL .MFLAG.
                           626X
     000.200               627X UO.HLT  EQU      10000000B        DISABLE HALT PROCESSING
     000.100               628X UO.NFR  EQU      CB.CLI           NO REFRESH OF FRONT PANEL
     000.002               629X UO.DDU  EQU      00000010B        DISABLE DISPLAY UPDATE
     000.001               630X UO.CLK  EQU      00000001B        ALLOW PRIVATE INTERRUPT PROCESSING


                           632X **       MONITOR IDENTIFICATION FLAGS
                           633X *
                           634X *        THESE BYTES IDENTIFY THE ROM MONITOR.
                           635X *        THEY ARE THE VARIOUS VALUES OF LOCATION .IDENT
                           636X
     000.021               637X M.PAM8  EQU      021Q             'LXI' INSTRUCTION AT 000.000 IN PAM-8
     000.303               638X M.FOX   EQU      303Q             'JMP' INSTRUCTION AT 000.000 IN FOX ROM


                           640X **       Configuration Flags                                        /80.07.sc/
                           641X *
                           642X *        These bits are read in IP.CON.
                           643X *
                           644X
     000.003               645X CN.174M EQU      00000011B        Port 174Q Device-Type Mask
     000.014               646X CN.170M EQU      00001100B        Port 170Q Device-Type Mask
     000.020               647X CN.PRI  EQU      00010000B        Primary/Secondary:  1=>primary == 170Q
     000.040               648X CN.MEM  EQU      00100000B        Memory Test/Normal Switch:  0=>Test; 1=>Normal
     000.100               649X CN.BAU  EQU      01000000B        Baud Rate:  0=>9600; 1=>19,200
     000.200               650X CN.ABO  EQU      10000000B        Auto-Boot:  1=>Auto-Boot
                           651X
     000.000               652X CND.H17 EQU      00B              H-17 Disk,             Valid only in CN.174M
     000.000               653X CND.NDI EQU      00B              No Device Installed,  Valid only in CN.170M
     000.001               654X CND.H47 EQU      01B              H-47 Disk


                           656X **       ROUTINE ENTRY POINTS.
                           657X *
                           658X
     000.000               659X .IDENT  EQU      0000A            IDENTIFICATION LOCATION
     000.053               660X .DLY    EQU      0053A            DELAY
     001.267               661X .LOAD   EQU      1267A            TAPE LOAD
     001.374               662X .DUMP   EQU      1374A            TAPE DUMP
     002.136               663X .ALARM  EQU      2136A            ALARM ROUTINE
     002.140               664X .HORN   EQU      2140A            HORN
     002.172               665X .CTC    EQU      2172A            CHECK TAPE CHECKSUM
     002.205               666X .TPERR  EQU      2205A            TAPE ERROR ROUTINE
     002.264               667X .PCHL   EQU      2264A            PCHL INSTRUCTION
     002.265               668X .SRS    EQU      2265A            SCAN RECORD START
     002.325               669X .RNP    EQU      2325A            READ NEXT PAIR
     002.331               670X .RNB    EQU      2331A            READ NEXT BYTE
```

```
    002.347                  671X .CRC     EQU      2347A            CRC-16 CALCULATOR
    003.017                  672X .WNP     EQU      3017A            WRITE NEXT PAIR
    003.024                  673X .WNB     EQU      3024A            WRITE NEXT BYTE
    003.122                  674X .DOD     EQU      3122A            DECODE FOR OCTAL DISPLAY
    003.260                  675X .RCK     EQU      3260A            READ CONSOLE KEYSET
    003.356                  676X .DODA    EQU      3356A            SEGMENT CODE TABLE


                             678X **       RAM CELLS USED BY H8MTR.
                             679X *
                             680X
    040.000                  681X .START   EQU      40000A           START DUMP ADDRESS
    040.002                  682X .IOWRK   EQU      40002A           IN OR OUT INSTRUCTION
    040.005                  683X .REGI    EQU      40005A           DISPLAYED REGISTER INDEX
    040.006                  684X .DSPROT  EQU      40006A           PERIOD FLAG BYTE
    040.007                  685X .DSPMOD  EQU      40007A           DISPLAY MODE
    040.010                  686X .MFLAG   EQU      40010A           USER OPTION BYTE
    040.011                  687X .CTLFLG  EQU      40011A           PANEL CONTROL BYTE
    040.013                  688X .ALEDS   EQU      40013A           ABUSS LEDS
    040.021                  689X .DLEDS   EQU      40021A           DBUSS LEDS
    040.024                  690X .ABUSS   EQU      40024A           ABUSS REGISTER
    040.027                  691X .CRCSUM  EQU      40027A           CRCSUM WORD
    040.031                  692X .TPERRX  EQU      40031A           TAPE ERROR EXIT VECTOR
    040.033                  693X .TICCNT  EQU      40033A           CLOCK TICK COUNTER
    040.035                  694X .REGPTR  EQU      40035A           REGISTER POINTER
    040.037                  695X .UIVEC   EQU      40037A           USER INTERRUPT VECTORS
    040.064                  696X .NMIRET  EQU      40064A           H88/H89 NMI Return Address      /80.07.sc/
    040.066                  697X .CTL2FL  EQU      40066A           OP2.CTL Control Byte            /80.07.sc/
    000.014                  698           XTEXT    DDFDEF


                             700X **       DIRECTORY DEVICE FORMAT DEFINITION.                      /80.09.sc/
                             701X *
                             702X *        Modified:       Sep-80
                             703X *                         No longer require 2 sectors per group
                             704X *                         Reserved Group Table dynamically allocated
                             705X *
                             706X
    000.000                  707X          ORG      0
                             708X
    000.000                  709X DDF.BOO DS       9                 2K BOOT PROGRAM
    000.011                  710X DDF.BOL EQU      *                 LENGTH OF BOOT
    000.011                  711X DDF.LAB DS       1                 LABEL SECTOR
    000.012                  712X DDF.USR DS       0                 BEGINNING OF OPEN SPACE
    000.012                  713           XTEXT    LABDEF
```

```
                    715X **      DISK LABEL SECTOR FORMATS.
                    716X
000.000             717X         ORG    0
000.000             718X LAB.SER DS     1               SERIAL NUMBER OF VOLUME
000.001             719X LAB.IND DS     2               INITTIALIZATION DATE
000.003             720X LAB.DIS DS     2               SECTOR NUMBER OF 1ST DIRECTORY SECTOR
000.005             721X LAB.GRT DS     2               INDEX OF GRT SECTOR
000.007             722X LAB.SPG DS     1               SECTORS PER GROUP
                    723X
000.000             724X LAB.DAT EQU    0               DATA VOLUME ONLY
000.001             725X LAB.SYS EQU    1               SYSTEM VOLUME
000.002             726X LAB.NOD EQU    2               => LAB.NOD MEANS VOLUME HAS NO DIRECTORY
                    727X
000.010             728X LAB.VLT DS     1               VOLUME TYPE
000.011             729X LAB.VER DS     1               VERSION OF INIT17 THAT INITED DISK
                    730X
000.012             731X LAB.RGT DS     2               RGT sector number                /80.06.sc/
                    732X
000.014             733X LAB.VPR EQU    *               Volume dependant data            /80.05.sc/
000.014             734X LAB.SIZ DS     2                Volume Size (Bytes/256)         /80.05.sc/
000.016             735X LAB.PSS DS     2                Physical Sector Size            /80.05.sc/
000.020             736X LAB.VFL DS     1                Volume dependant Flags          /80.09.sc/
000.001             737X VFL.NSD EQU    00000001B         Number of Sides:  1 => 2       /80.09.sc/
000.005             738X LAB.VPL EQU    *-LAB.VPR       Length of volume dependant data /80.05.sc/
                    739X
000.000             740X         ERRMI  5-LAB.VPL                                        /80.05.sc/
000.021             741X         DS     5-LAB.VPL       Reserved                         /80.05.sc/
                    742X
000.021             743X LAB.LAB DS     60              LABEL
000.074             744X LAB.LBL EQU    *-LAB.LAB       LABEL LENGTH
000.115             745X         DS     2               Reserved for 0 bytes             /80.09.sc/
                    746X
000.117             747X LAB.AUX EQU    *               Auxiliary Data                   /80.09.sc/
000.117             748X LAB.SPT DS     1                Sectors per Track               /80.09.sc/
000.001             749X LAB.AXL EQU    *-LAB.AUX       Length of Aux. Data              /80.09.sc/
000.120             750          XTEXT  FILDEF



                    752X **      FILDEF - FILE TYPE DEFINITIONS.
                    753X *
                    754X *       DB     377Q,FT.XXX
                    755X
                    756X
000.000             757X FT.ABS  EQU    0               ABSOLUTE BINARY
000.001             758X FT.PIC  EQU    1               POSITION INDEPENDANT CODE
000.002             759X FT.REL  EQU    2               RELOCATABLE CODE
000.003             760X FT.BAC  EQU    3               COMPILED BASIC CODE
000.120             761          XTEXT  ABSDEF
```

```
                       763X **      ABS FORMAT EQUIVALENCES.
                       764X
000.000                765X         ORG     0
                       766X
000.000                767X ABS.ID  DS      1               377Q = BINARY FILE FLAG
000.001                768X         DS      1               FILE TYPE (FT.ABS)
000.002                769X ABS.LDA DS      2               LOAD ADDRESS
000.004                770X ABS.LEN DS      2               LENGTH OF ENTIRE RECORD
000.006                771X ABS.ENT DS      2               ENTRY POINT
                       772X
000.010                773X ABS.COD DS      0               CODE STARTS HERE
```

```
                        776  **      Structure Definitions                              /80.07.sc/
                        777
     000.000            778          ORG     0
                        779
     000.000            780  DEFAULT DS      6                  Default device descriptor
                        781
     000.006            782  DEVTAB  DS      2                  Device Table Address Pointer
                        783
     000.010            784  DRIVER  DS      3                  Driver Entry Point
                        785
     000.013            786  UNIT    DS      1                  Unit Number
                        787
     000.014            788  DEVICE  DS      IOC.DIR-IOC.DEV+2         Device Specifier
                        789
     000.021            790  DVCLEN  EQU     *
```

```
042.170                               ORG     USERFWA-ABS.COD
042.170  377 000                      DB      377Q,FT.ABS
042.172  200 042                      DW      USERFWA          LOAD ADDRESS
042.174  154 023                      DW      MEML-USERFWA     SIZE
042.176  167 062                      DW      ENTRY            ENTRY

                        *             COMMAND INTERPRETATION COMES HERE

042.200  061 200 042    START         LXI     SP,STACK         CLEAN STACK

042.203  257                          XRA     A                                               /80.07.GC/
042.204  062 247 060                  STA     VOLFLAG          Initialize Source Mounted      /80.07.sc/
042.207  315 336 047                  CALL    MSD.             Mount Source Diskette          /80.07.sc/
042.212  052 167 063                  LHLD    SRCLAB+LAB.SER                                  /80.10.sc/
042.215  046 000                      MVI     H,0              HL = Volume number             /80.10.sc/
042.217  076 010                      MVI     A,DC.MOU                                        /80.10.sc/
042.221  315 327 056                  CALL    SRCDRVR          Call source driver             /80.10.sc/
042.224  332 123 052                  JC      ERROR                                           /80.10.sc/

042.227  072 241 060                  LDA     DRIVES2                                         /80.07.sc/
042.232  247                          ANA     A                'NZ' => 2-Drive Sysgen         /80.07.sc/
042.233  304 041 047                  CNZ     MDD              Mount Destination Diskette     /80.07.sc/

                        *             CLEAR CHANNELS AND FILE BUFFER

042.236  377 056                      SCALL   .CLEARA          CLEAR CHANNELS

042.240  041 000 000                  LXI     H,0
042.243  042 236 060                  SHLD    BUFSIZ           EMPTY BUFFER
042.246  042 364 060                  SHLD    NAMTLEN          CLEAR NAMTAB
042.251  042 366 060                  SHLD    NAMTMAX          CLEAR NAMTAB AREA
042.254  041 242 065                  LXI     H,BUFF
042.257  042 234 060                  SHLD    BUFPTR           SET BUFFER AGAINST END OF NAMTAB

                        *             Copy the files                                          /80.07.sc/

042.262  072 245 060                  LDA     QUERY
042.265  365                          PUSH    PSW              Save Query Flag
042.266  257                          XRA     A
042.267  062 245 060                  STA     QUERY            Force NO Query on required files
042.272  315 012 043                  CALL    CRF              Copy Required Files
042.275  315 147 043                  CALL    CSD              Copy System Device Drivers
042.300  361                          POP     PSW
042.301  062 245 060                  STA     QUERY            Restore Query Flag
042.304  315 212 051                  CALL    SSL              Set the Sysgened flag in label
042.307  315 343 043                  CALL    COF              Copy Optional Files

                        *             Type File Count                                         /80.07.sc/

042.312  072 177 044                  LDA     OCOPYC           (A) = FILE COUNT
042.315  006 000                      MVI     B,0              (BC) = COUNT OF FILES COPIED
042.317  117                          MOV     C,A

042.320  076 003                      MVI     A,3
042.322  041 334 042                  LXI     H,SYSA
042.325  315 002 060                  CALL    $UDDN            UNPACK COUNT INTO MESSAGE
```

```
042.330  315 136 031   849            CALL    $TYPTX
042.333  012           850            DB      NL                                           /80.07.GC/
042.334  130 130 130   851  SYSA      DB      'XXX'
042.337  040 106 151   852            DB      ' Files Copied',ENL
                       853
                       854  *         Dismount all Disks                                   /80.07.GC/
                       855
042.355  041 303 060   856            LXI     H,DEST+DEVICE
042.360  377 203       857            SCALL   .DMNMS
042.362  041 324 060   858            LXI     H,SOURCE+DEVICE Ignore any possible errors
042.365  377 203       859            SCALL   .DMNMS
                       860
042.367  257           861            XRA     A
042.370  303 000 043   862            JMP     EXIT.           GRACEFUL EXIT
                       863
                       864  **        NO RESTARTING ALLOWED
                       865
042.373                866  RESTART EQU     *
                       867
042.373  303 376 042   868            JMP     EXIT            EXIT
                       869
                       870  *         CTL-D HIT
                       871
042.376  076 001       872  EXIT      MVI     A,1             FLAG ABORT
043.000  377 000       873  EXIT.     DB      SYSCALL,.EXIT   EXIT TO  *HDOS*


                       875  **        CCHIT - CTL-C HIT
                       876  *
                       877  *         ENTRY   FROM SYSTEM
                       878
                       879
043.002  315 136 031   880  CCHIT     CALL    $TYPTX
043.005  136 303       881            DB      '^','C'+200Q
043.007  303 376 042   882            JMP     EXIT            BOOT IT
```

```
                            ***     CRF     - Copy Required Files
                            *
                            *       CRF copies the required HDOS files accross to the
                            *       destination device.  Once they are copied accross,
                            *       they must be flagged illegal for subsequent operations
                            *       since we want to use the *.SYS wild-card specification
                            *       later on.
                            *
                            *       NOTE:   The files listed in CRFA must also be
                            *               specified in CSFA.
                            *

043.012  072 232 053        CRF     LDA     CSFB
043.015  365                        PUSH    PSW
043.016  257                        XRA     A
043.017  062 232 053                STA     CSFB                Flag these files valid for now

043.022  041 035 043                LXI     H,CRFA
043.025  315 065 044                CALL    OCOPY               Copy required files

043.030  361                        POP     PSW
043.031  062 232 053                STA     CSFB                Flag the files not valid
043.034  311                        RET

043.035  052 056 052        CRFA    DB      '*.*=HDOS.SYS,HDOSOVL0.SYS,HDOSOVL1.SYS,SYSCMD.SYS,'
043.117  120 111 120                DB      'PIP.ABS',0
043.127                             DS      16                  Patch area



                            ***     CSD     - Copy System Drivers
                            *
                            *       CSD copies the system device drivers accross.  It
                            *       would be nice if this could be combined with the
                            *       previous required files, however, to avoid rename
                            *       difficulties, these files are handled individually.
                            *
                            *       IF destination device name == source device name
                            *         THEN
                            *           Just copy SY.DVD
                            *         ELSE
                            *           DEST:SY.DVD=SOURCE:DEST.DVD
                            *           DEST:DEST.DVD=SOURCE:SY.DVD
                            *

043.147  052 324 060        CSD     LHLD    SOURCE+DEVICE
043.152  353                        XCHG                        DE = Source Device
043.153  052 303 060                LHLD    DEST+DEVICE         HL = Destination Device
000.000                             ERRNZ   IOC.UNI-IOC.DEV-2
043.156  315 216 030                CALL    $CDEHL
043.161  302 212 043                JNZ     CSD1                DE != HL

                            *       Devices are equal

043.164  072 333 053                LDA     CSFC
```

```
     043.167  365            939           PUSH     PSW
     043.170  257            940           XRA      A
     043.171  062 333 053    941           STA      CSFC            Flag this device valid temporarily
     043.174  062 350 053    942           STA      CSFD            Only 1 driver copied
                             943
     043.177  041 254 043    944           LXI      H,CSDA
     043.202  315 065 044    945           CALL     OCOPY
                             946
     043.205  361            947           POP      PSW
     043.206  062 333 053    948           STA      CSFC            Re-Flag SY: Illegal
     043.211  311            949           RET
                             950
                             951   *       Devices are not equal
                             952
     043.212  042 316 043    953   CSD1    SHLD     CSDC            Set Destination Device
     043.215  042 325 043    954           SHLD     CSDD
     043.220  042 350 053    955           SHLD     CSFD            Set Destination Driver Illegal for later
     043.223  072 333 053    956           LDA      CSFC
     043.226  365            957           PUSH     PSW
     043.227  257            958           XRA      A
     043.230  062 333 053    959           STA      CSFC            Flag these valid for now
                             960
     043.233  041 307 043    961           LXI      H,CSDB
     043.236  315 065 044    962           CALL     OCOPY
                             963
     043.241  041 325 043    964           LXI      H,CSDD
     043.244  315 065 044    965           CALL     OCOPY
                             966
     043.247  361            967           POP      PSW
     043.250  062 333 053    968           STA      CSFC
     043.253  311            969           RET
                             970
     043.254  052 056 052    971   CSDA    DB       '*.*=SY.DVD',0
     043.267                 972           DS       16
                             973
     043.307  123 131 056    974   CSDB    DB       'SY.DVD='               Copy System Device
     043.316  170 170 056    975   CSDC    DB       'xx.DVD',0
                             976
     043.325  170 170 056    977   CSDD    DB       'xx.DVD=SY.DVD',0       Copy destination device over


                             979   ***     COF      - Copy Optional Files
                             980   *
                             981   *       COF copies the optional files accross to the
                             982   *       destination diskette.  If  *MINIMAL*  has been
                             983   *       specified, no additional files are copied.
                             984   *       Otherwise, the default command line is used unless
                             985   *       a command line was specified at run time.
                             986   *
                             987
     043.343  072 244 060    988   COF     LDA      MINIMUM
     043.346  247            989           ANA      A
     043.347  300            990           RNZ                      Minimum switch is set
                             991
```

```
      043.350  072 240 060   992          LDA     CMDLIN
      043.353  247           993          ANA     A
      043.354  302 366 043   994          JNZ     COF1              Command line specified
                             995
                             996  *       No command line specified
                             997
      043.357  041 375 043   998          LXI     H,COFA
      043.362  315 065 044   999          CALL    OCOPY             Normal extra files
      043.365  311          1000          RET
                            1001
                            1002  *       Command Line specified
                            1003
      043.366  041 372 060  1004  COF1    LXI     H,OFILES
      043.371  315 065 044  1005          CALL    OCOPY             All files with query
      043.374  311          1006          RET
                            1007
      043.375  052 056 052  1008  COFA    DB      '*.*=*.SYS,SET.ABS,FLAGS,ONECOPY,*.DVD,'
      044.043  123 131 123  1009          DB      'SYSHELP.DOC,HELP.',0
```

```
                           ***     SYSGEN - 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.
                           *
                           *       OCOPY EXITS WITH THE DESTINATION DISK MOUNTED.
                           *
                           *       ENTRY:  HL       = Line Pointer                            /80.07.sc/
                           *
                           *       EXIT:   OCOPYC   = Number of files copied                  /80.07.sc/
                           *

044.065                    OCOPY   EQU     *
044.065  042 242 060               SHLD    LINEP           Initialize Line Pointer           /80.07.sc/
044.070  315 247 046               CALL    IFL             INITIALIZE  FDN LISTS             /80.07.GC/
044.073  315 004 054               CALL    DDF             DECODE DESTINATION FILE           /80.07.sc/
044.076  332 123 052               JC      ERROR           ERROR
044.101  062 176 044               STA     OCOPYA          SAVE DESTIONATION TYPE

044.104  315 262 047               CALL    MSD             Mount Source to build list        /80.08.sc/

044.107  257                       XRA     A               ALLOW *.*
044.110  315 360 052               CALL    BSL             BUILD SOURCE FILE LIST
044.113  332 123 052               JC      ERROR
044.116  315 214 057               CALL    $MOVEL
044.121  021 000                   DW      OCOPYDL
```

```
   044.123  343 060      1068           DW      DESTFB+FB.NAM
   044.125  200 044      1069           DW      OCOPYD          SAVE WILDCARD DESTINATION
   044.127  315 037 055  1070           CALL    EBM             EXPAND BUFFER TO MAX
                         1071
                         1072  *        START READ PHASE
                         1073
   044.132  072 235 060  1074  OCOPY1   LDA     BUFPTR+1        (A) = BUFFER FWA/256
   044.135  074          1075           INR     A               ROUND UP TO NEXT PAGE
   044.136  062 233 060  1076           STA     OBUFPTR         SET SECTOR BUFFER  FWA/256
                         1077
   044.141  315 262 047  1078           CALL    MSD             Mount Source Diskette           /80.07.sc/
   044.144  315 221 044  1079           CALL    RPH             Read Data                       /80.07.sc/
   044.147  315 041 047  1080           CALL    MDD             Mount Destination Diskette      /80.07.sc/
   044.152  315 260 045  1081           CALL    WPH             Write Data                      /80.07.sc/
                         1082
   044.155  052 070 060  1083           LHLD    FDNHED                                          /80.07.GC/
   044.160  174          1084           MOV     A,H                                             /80.07.GC/
   044.161  265          1085           ORA     L                                               /80.07.GC/
   044.162  302 132 044  1086           JNZ     OCOPY1          Source files in List            /80.07.sc/
                         1087
   044.165  052 364 060  1088           LHLD    NAMTLEN
   044.170  174          1089           MOV     A,H
   044.171  265          1090           ORA     L
   044.172  302 132 044  1091           JNZ     OCOPY1          MORE NAMES IN LIST
                         1092
                         1093  *        ALL DONE                                                /80.07.GC/
                         1094
   044.175  311          1095           RET                                                     /80.07.GC/
                         1096
   044.176  000          1097  OCOPYA   DB      0               DESTINATION FILE WILDCARD FLAG (=0 IF WC)
   044.177  000          1098  OCOPYC   DB      0               FILES COPIED COUNT
   044.200               1099  OCOPYD   DS      FB.NAML         HOLD AREA FOR WILDCARD DESTINATION
   000.021               1100  OCOPYDL  EQU     *-OCOPYD
```

```
                     **      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


044.221              RPH     EQU     *


                     *       SEE IF ANY MEMORY TO HAVE

044.221  315 230 046         CALL    CBR             COMPUTE BUFFER ROOM
044.224  310                 RZ                      NONE

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

044.225  052 070 060         LHLD    FDNHED          HL = Head of LIST               /80.07.sc/
044.230  174                 MOV     A,H                                             /80.07.GC/
044.231  265                 ORA     L                                               /80.07.GC/
044.232  312 252 044         JZ      RPH1            LIST is empty                   /80.07.sc/

044.235  315 100 057         CALL    $INDLB          A  = status                     /80.07.GC/
044.240  002 000             DW      FDN.STA                                         /80.07.sc/
044.242  346 002             ANI     ST.OPR
044.244  021 242 065         LXI     D,NAMTAB
044.247  302 372 044         JNZ     RPH2.5          FILE IS INCOMPLETELY READ

                     *       SEE IF ANY FREE FILE DESCRIPTOR NODES TO USE

044.252  052 066 060 RPH1    LHLD    FDNFREE         HL = Head of FREE list          /80.07.sc/
044.255  174                 MOV     A,H                                             /80.07.GC/
044.256  265                 ORA     L                                               /80.07.GC/
044.257  310                 RZ                      free list is empty              /80.07.sc/

                     *       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...
```

```
                          1160
044.260  001 021 000      1161            LXI    B,FB.NAML      (BC) = ENTRY SIZE IN NAMTAB
044.263  021 000 000      1162            LXI    D,0            (DE) = POINTER INTO NAMTAB       /80.07.GC/
044.266  041 070 060      1163            LXI    H,FDNHED       HL = head of FILE list           /80.07.sc/
                          1164
044.271  345              1165   RPH2     PUSH   H                                               /80.07.GC/
044.272  315 211 030      1166            CALL   $HLIHL         HL = next node                   /80.07.sc/
000.000                   1167            ERRNZ  FDN.LNK                                         /80.07.GC/
044.275  174              1168            MOV    A,H                                             /80.07.GC/
044.276  265              1169            ORA    L                                               /80.07.GC/
044.277  341              1170            POP    H                                               /80.07.sc/
044.300  312 314 044      1171            JZ     RPH2.2         At the end of the list           /80.07.sc/
                          1172
044.303  315 211 030      1173            CALL   $HLIHL         HL = next node                   /80.07.sc/
000.000                   1174            ERRNZ  FDN.LNK                                         /80.07.GC/
044.306  353              1175            XCHG
044.307  011              1176            DAD    B              ADVANCE POINTER INTO NAMTAB
044.310  353              1177            XCHG
044.311  303 271 044      1178            JMP    RPH2                                            /80.07.GC/
                          1179
044.314  345              1180   RPH2.2   PUSH   H              (HL) = ADDRESS OF LAST NODE
044.315  052 364 060      1181            LHLD   NAMTLEN
044.320  315 216 030      1182            CALL   $CDEHL         SEE IF HAVE ACCOUNTED FOR ALL NAMTAB ENTRYS
044.323  341              1183            POP    H
044.324  310              1184            RE                    FILES ALL USED UP
                          1185
                          1186   *        HAVE ROOM FOR DATA, HAVE A NODE FOR THE FILE COUNTS, AND
                          1187   *        HAVE A FILE NAME, ALL SET FOR BUSINESS..
                          1188   *
                          1189   *        (DE) = INDEX INTO NAMTAB FOR FILE
                          1190   *        (HL) = NODE ADDRESS OF LAST ENTRY IN LIST
                          1191   *
                          1192
                          1193   *        CHAIN THE FIRST FREE NODE ONTO THE END OF THE LIST      /80.07.sc/
                          1194
044.325  325              1195            PUSH   D                                               /80.07.sc/
044.326  345              1196            PUSH   H                                               /80.07.sc/
044.327  052 066 060      1197            LHLD   FDNFREE        HL = FREE node                   /80.07.sc/
                          1198
044.332  345              1199            PUSH   H                                               /80.07.sc/
044.333  315 211 030      1200            CALL   $HLIHL         HL = address of next node        /80.07.sc/
000.000                   1201            ERRNZ  FDN.LNK                                         /80.07.GC/
044.336  042 066 060      1202            SHLD   FDNFREE        Update FREE list head            /80.07.sc/
044.341  321              1203            POP    D              DE = address of new node         /80.07.sc/
                          1204
044.342  341              1205            POP    H              HL = address of TAIL of list     /80.07.sc/
044.343  315 121 057      1206            CALL   $INDS          tail points to the new node      /80.07.sc/
044.346  000 000          1207            DW     FDN.LNK                                         /80.07.sc/
044.350  353              1208            XCHG                  HL = address of new node         /80.07.sc/
044.351  321              1209            POP    D              DE = address in name table       /80.07.sc/
                          1210
044.352  006 014          1211            MVI    B,FDNELEN
044.354  345              1212            PUSH   H              SAVE NODE ADDRESS
044.355  315 212 031      1213            CALL   $ZERO          ZERO ENTIRE NODE, INCLUDING CHAIN (NOW AT END)
044.360  001 242 065      1214            LXI    B,NAMTAB
044.363  353              1215            XCHG
```

```
044.364  011                         DAD     B               (HL) = ADDRESS OF NAMTAB ENTRY
044.365  042 370 060                 SHLD    NAMTPTR         POINTER TO CURRENT NAMTAB ENTRY
044.370  353                         XCHG
044.371  341                         POP     H

                       *             READY TO OPEN FILE
                       *
                       *             (DE) = NAMTAB ENTRY ADDRESS
                       *             (HL) = #FDN.LNK OF ENTRY                                  /80.07.GC/

044.372  043           RPH2.5        INX     H                                                 /80.07.GC/
044.373  043                         INX     H               HL = &FDN.STA                     /80.07.GC/
000.000                              ERRNZ   FDN.STA-FDN.LNK-2                                 /80.07.GC/
044.374  345                         PUSH    H               SAVE ADDRESS
044.375  353                         XCHG
044.376  257                         XRA     A
000.000                              ERRNZ   CN.SOU          (A) = SOURCE CHANNEL NUMBER
044.377  377 042                     DB      SYSCALL,.OPENR  OPEN
045.001  332 267 051                 JC      NAMERR          ERROR
045.004  321                         POP     D
045.005  032                         LDAX    D               (A) = FDN.STA
045.006  346 002                     ANI     ST.OPR
045.010  325                         PUSH    D               SAVE ADDRESS
045.011  302 131 045                 JNZ     RPH3            ALREADY OPENED IN PREVIOUS PASSES

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

045.014  041 177 044                 LXI     H,OCOPYC
045.017  064                         INR     M
045.020  032                         LDAX    D
045.021  366 002                     ORI     ST.OPR          SET OPEN FOR READ
045.023  022                         STAX    D
045.024  325                         PUSH    D               SAVE #FDN.STA
045.025  052 352 040                 LHLD    S.CFWA          (HL) = CHANNEL 0 FWA
000.000                              ERRNZ   IOCCTD-1        MUST SKIP A CHANNEL FOR USER #0
045.030  315 211 030                 CALL    $HLIHL          (HL) = #USER CHANNEL 0
000.000                              ERRNZ   CN.SOU          ASSUME WE WANT CHANNEL 0
045.033  315 234 030                 CALL    $INDL
045.036  041 000                     DW      IOC.DIR+DIR.FLG
045.040  173                         MOV     A,E             (A) = DIR.FLG
045.041  321                         POP     D               (DE) = #FDN.STA
000.000                              ERRNZ   FDN.FLG-FDN.STA-1
045.042  023                         INX     D               (DE) = FDN.FLG
045.043  022                         STAX    D               SAVE FILE FLAGS
045.044  346 020                     ANI     DIF.CNT
045.046  312 131 045                 JZ      RPH3            NOT CONTIG

                       *             IS CONTIG. GET FILE SIZE

045.051  315 234 030                 CALL    $INDL
045.054  005 000                     DW      IOC.GRT
045.056  325                         PUSH    D               SAVE GRT ADDRESS
045.057  315 100 057                 CALL    $INDLB          A  = Last Sector Index            /80.08.sc/
045.062  045 000                     DW      IOC.DIR+DIR.LSI                                   /80.08.sc/
045.064  062 257 045                 STA     RPHA            Save Sector Index                 /80.08.sc/
045.067  315 100 057                 CALL    $INDLB          A  = DIR.FGN                      /80.07.GC/
```

```
045.072  043 000      1272          DW      IOC.DIR+DIR.FGN                                /80.07.GC/
045.074  341          1273          POP     H                 (HL) = GRT TABLE ADDRESS     /80.07.GC/
045.075  315 023 053  1274          CALL    CFS.              COMPUTE BLOCK SIZE           /80.07.GC/
045.100  033          1275          DCX     D                 Don't count last block       /80.08.sc/
045.101  072 246 060  1276          LDA     SRCSPG            A  = source volume SPG       /80.07.sc/
045.104  315 007 031  1277          CALL    $MU86             HL = DE * A                  /80.07.GC/
045.107  072 257 045  1278          LDA     RPHA              A = Sector Index             /80.08.sc/
045.112  315 072 030  1279          CALL    $DADA                                          /80.08.sc/
045.115  353          1280          XCHG                      DE = number of sectors       /80.07.GC/
045.116  341          1281          POP     H                 (HL) = ADDRESS OF FDN.STA
045.117  345          1282          PUSH    H
                      1283
045.120  176          1284          MOV     A,M               (A) = FDN.STA
045.121  366 020      1285          ORI     ST.CNT            FLAG CONTIG
045.123  167          1286          MOV     M,A
045.124  315 121 057  1287          CALL    $INDS             FDN.SIZ = number of sectors   /80.07.sc/
045.127  002 000      1288          DW      FDN.SIZ-FDN.STA                                /80.07.GC/
                      1289
                      1290  *       READY TO READ DATA. POSITION FILE (IN CASE SOME WAS READ IN
                      1291  *       PREVIOUS PASSES) AND COMPUTE THE MAX POSSIBLE READ COUNT
                      1292  *
                      1293  *       ((SP)) = ADDRESS OF FDN.STA FOR NODE
                      1294
045.131  341          1295  RPH3    POP     H                 (HL) = ADDRESS OF FDN.STA
045.132  345          1296          PUSH    H
045.133  315 234 030  1297          CALL    $INDL
045.136  004 000      1298          DW      FDN.AMR-FDN.STA (DE) = AMOUNT READ (IN SECTORS)
045.140  102          1299          MOV     B,D
045.141  113          1300          MOV     C,E               (BC) = AMOUNT READ
045.142  076 000      1301          MVI     A,CN.SOU
045.144  377 047      1302          DB      SYSCALL,.POSIT            POSIT
045.146  332 321 051  1303          JC      IERR3             POSIT BLEW UP
045.151  315 230 046  1304          CALL    CBR               COMPUTE BUFFER ROOM
045.154  353          1305          XCHG                      (D) = POINTER/256, (E) = LIMIT/256
045.155  341          1306          POP     H                 (HL) = #FDN.STA
045.156  001 010 000  1307          LXI     B,FDN.ADR-FDN.STA
045.161  011          1308          DAD     B                 (HL) = #FDN.ADR
045.162  162          1309          MOV     M,D               SET ADDRESS/256
045.163  345          1310          PUSH    H                 SAVE #FDN.ADR
045.164  036 000      1311          MVI     E,0               (DE) = ADDRESS
045.166  107          1312          MOV     B,A               (B) = SECTORS OF RAM AVAILABLE
045.167  113          1313          MOV     C,E               (C) = 0
045.170  305          1314          PUSH    B                 SAVE TRY COUNT
045.171  076 000      1315          MVI     A,CN.SOU
045.173  377 004      1316          DB      SYSCALL,.READ     READ THE STUFF
                      1317
                      1318  *       COMPUTE THE AMOUNT READ (IN CASE OF EOF)
                      1319
045.175  321          1320          POP     D                 (DE) = TRY COUNT
045.176  322 223 045  1321          JNC     RPH4              GOT  ALL WE TRYED
045.201  376 001      1322          CPI     EC.EOF
045.203  302 267 051  1323          JNE     NAMERR            NOT JUST EOF, GOT TROUBLES
045.206  172          1324          MOV     A,D
045.207  220          1325          SUB     B                 REMOVE AMOUNT WE DIDNT GET
045.210  127          1326          MOV     D,A
045.211  341          1327          POP     H                 (HL) = #FDN.ADR
```

```
045.212  345            1328          PUSH    H
045.213  001 370 377    1329          LXI     B,FDN.STA-FDN.ADR
045.216  011            1330          DAD     B
045.217  176            1331          MOV     A,M                (A) = FDN.STA
045.220  346 375        1332          ANI     377Q-ST.OPR        EOF, NOT OPEN FOR READ ANYMORE
045.222  167            1333          MOV     M,A                POST READ COMPLETE FOR THIS GUY
                        1334
                        1335  *       STORE RESULTS OF READ IN NODE
                        1336  *
                        1337  *       (D) = SECTORS READ
                        1338  *       ((SP)) = #FDN.ADR
                        1339
045.223  341            1340  RPH4    POP     H                  (HL) = #FDN.ADR
045.224  043            1341          INX     H
000.000                 1342          ERRNZ   FDN.AIM-FDN.ADR-1        (HL) = ADDRESS IF AMOUNT IN MEMORY BYTE
045.225  162            1343          MOV     M,D                STORE SECTORS IN MEMORY COUNT
045.226  001 373 377    1344          LXI     B,FDN.AMR-FDN.AIM
045.231  011            1345          DAD     B                  (HL) = #FDN.AMR (AMOUNT READ)
045.232  176            1346          MOV     A,M                (A) = AMOUNT READ BEFORE
045.233  202            1347          ADD     D                  ADD NEW AMOUNT
045.234  167            1348          MOV     M,A
045.235  043            1349          INX     H
045.236  176            1350          MOV     A,M
045.237  316 000        1351          ACI     0                  PROPIGATE FOR VERY LARGE FILES
045.241  167            1352          MOV     M,A
045.242  041 233 060    1353          LXI     H,OBUFPTR
045.245  176            1354          MOV     A,M
045.246  202            1355          ADD     D                  ADVANCE FREE RAM POINTER BY AMOUNT READ
045.247  167            1356          MOV     M,A
045.250  076 000        1357          MVI     A,CN.SQU
045.252  377 046        1358          DB      SYSCALL,.CLOSE     CLOSE FILE
045.254  303 221 044    1359          JMP     RPH                SEE IF MORE TO READ
                        1360
045.257  000            1361  RPHA    DB      0                  Saved Last Sector Index          /80.08.sc/


                        1363  **      WPH - WRITE PHASE.
                        1364  *
                        1365  *       WPH HANDLES THE WRITE PHASE PROCESSING. IT IS ENTERED WITH
                        1366  *       THE FDN CHAIN SETUP, THE NAMTAB SETUP, AND
                        1367  *       THE DESTINATION DISK MOUNTED.
                        1368  *
                        1369  *
                        1370  *       WRITE PHASE
                        1371  *
                        1372  *       DURING THE WRITE PHASE, THE DESTINATION DISK IS MOUNTED. THE NODES
                        1373  *       ARE TAKEN FROM THE ACTIVE CHAIN, AND PROCESSED. IF THE FILE HAD
                        1374  *       BEEN PARTIALLY WRITTEN THE LAST PASS, IT IS  RE-OPENED AND POSITIONED.
                        1375  *       IF THERE IS NOT MORE DATA TO READ FOR A PROCESSED
                        1376  *       NODE, IT IS REMOVED, AND THE CORRESPONDING ENTRY IN NAMTAB IS DELETED.
                        1377  *
                        1378  *       WRITE PHASE CONTINUES UNTIL
                        1379  *
                        1380  *               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


045.260                    WPH      EQU     *

                           *        SEE IF MORE TO WRITE

045.260  052 070 060                LHLD    FDNHED          HL = FILE list head             /80.07.sc/
045.263  174                        MOV     A,H                                             /80.07.GC/
045.264  265                        ORA     L                                               /80.07.GC/
045.265  310                        RZ                      FILE list is empty              /80.07.sc/

045.266  315 100 057                CALL    $INDLB          A  = amount in memory for file  /80.07.sc/
045.271  013 000                    DW      FDN.AIM                                         /80.07.sc/
045.273  247                        ANA     A                                               /80.07.sc/
045.274  302 313 045                JNZ     WPH0            GOT DATA

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

045.277  315 100 057                CALL    $INDLB          A  = FDN.STA                    /80.07.sc/
045.302  002 000                    DW      FDN.STA                                         /80.07.GC/
045.304  346 002                    ANI     ST.OPR
045.306  300                        RNZ                     STILL READING, GET MORE

045.307  353                        XCHG                    (DE) = ADDRESS
045.310  303 140 046                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

045.313  345               WPH0     PUSH    H               SAVE NODE POINTER
045.314  043                        INX     H
045.315  043                        INX     H                                               /80.07.GC/
000.000                             ERRNZ   FDN.STA-FDN.LNK-2                               /80.07.GC/
045.316  176                        MOV     A,M             (A) = FDN.STA
045.317  346 001                    ANI     ST.OPW
045.321  302 033 046                JNZ     WPH2            OPENED BEFORE
000.000                             ERRNZ   ST.OPW-1
045.324  064                        INR     M               SET '1' BIT

                           *        BUILD NAME INTO DESTFB

045.325  345                        PUSH    H               SAVE NODE ADDRESS
045.326  001 200 044                LXI     B,OCOPYD
045.331  021 242 065                LXI     D,NAMTAB
045.334  041 343 060                LXI     H,DESTFB+FB.NAM
045.337  315 140 056                CALL    MWN             MERGE WILDCARD NAME
045.342  341                        POP     H

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

```

045.343  176                  MOV    A,M              (A) = FLAG BYTE
045.344  346 020              ANI    ST.CNT
045.346  302 366 045          JNZ    WPH1             IS CONTIG
045.351  041 343 060          LXI    H,DESTFB+FB.NAM
045.354  076 001              MVI    A,CN.DES
045.356  377 043              DB     SYSCALL,.OPENW   JUST OPEN FOR WRITE
045.360  332 301 051          JC     DESTERR          ERROR
045.363  303 065 046          JMP    WPH3             WRITE THE DATA

                        *     IS CONTIG FILE. OPEN IN CONTIG MODE

045.366  315 234 030    WPH1  CALL   $INDL                                                 /80.07.sc/
045.371  002 000              DW     FDN.SIZ-FDN.STA                                       /80.07.GC/
045.373  102                  MOV    B,D                                                   /80.07.sc/
045.374  113                  MOV    C,E              BC = Number of sectors required /80.07.sc/
045.375  041 343 060          LXI    H,DESTFB+FB.NAM                                       /80.07.sc/
046.000  076 001              MVI    A,CN.DES
046.002  305                  PUSH   B                SAVE COUNT
046.003  377 050              DB     SYSCALL,.DELET   DELETE OLD ONE
046.005  322 015 046          JNC    WPH1.5           DELETED
046.010  376 014              CPI    EC.FNF
046.012  302 123 052          JNE    ERROR            MUST BE WRITE PROTECTED, OR  SOMETHING...
046.015  301            WPH1.5 POP   B                (BC) = COUNT
046.016  041 343 060          LXI    H,DESTFB+FB.NAM
046.021  076 001              MVI    A,CN.DES
046.023  377 045              DB     SYSCALL,.OPENC   OPEN CONTIG
046.025  332 301 051          JC     DESTERR
046.030  303 065 046          JMP    WPH3

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

046.033  041 343 060    WPH2  LXI    H,DESTFB+FB.NAM
046.036  076 001              MVI    A,CN.DES
046.040  377 044              DB     SYSCALL,.OPENU   OPEN FOR UPDATE
046.042  332 301 051          JC     DESTERR          PROBLEMS
046.045  341                  POP    H
046.046  345                  PUSH   H                (HL) = #FDN.STA
046.047  315 234 030          CALL   $INDL
046.052  010 000              DW     FDN.AMW          (DE) = AMOUNT WRITTEN
046.054  102                  MOV    B,D
046.055  113                  MOV    C,E              (BC) = SECTORS WRITTEN
046.056  076 001              MVI    A,CN.DES
046.060  377 047              DB     SYSCALL,.POSIT   POSITION FOR EXTEND
046.062  332 307 051          JC     IERR1            COULDNT GET THERE!

                        *     FILE OPEN AND POSITIONED. WRITE DATA

046.065  341            WPH3  POP    H
046.066  345                  PUSH   H                (HL) = #FDN.LNK
046.067  315 234 030          CALL   $INDL
046.072  012 000              DW     FDN.ADR          (E) = ADDR/256, (D) = CNT/256
046.074  102                  MOV    B,D
046.075  123                  MOV    D,E
046.076  036 000              MVI    E,0              (DE) = ADDRESS
```

```
046.100  113          1493        MOV     C,E                 (BC) = COUNT
046.101  076 001      1494        MVI     A,CN.DES
046.103  305          1495        PUSH    B                   SAVE WRITE COUNT
046.104  377 005      1496        DB      SYSCALL,.WRITE      WRITE IT
046.106  332 301 051  1497        JC      DESTERR             PROBABLY OUT OF ROOM
046.111  076 001      1498        MVI     A,CN.DES
046.113  377 046      1499        DB      SYSCALL,.CLOSE      CLOSE IT
046.115  332 301 051  1500        JC      DESTERR
046.120  301          1501        POP     B                   (B) = SECTORS WRITTEN
046.121  341          1502        POP     H
046.122  345          1503        PUSH    H                   (HL) = #FDN.LNK
046.123  021 010 000  1504        LXI     D,FDN.AMW-FDN.LNK
046.126  031          1505        DAD     D                   (HL) = FDN.AMW
046.127  176          1506        MOV     A,M
046.130  200          1507        ADD     B
046.131  167          1508        MOV     M,A
046.132  043          1509        INX     H
046.133  176          1510        MOV     A,M
046.134  316 000      1511        ACI     0                   INCREMENT AMOUNT WRITTEN
046.136  167          1512        MOV     M,A
                      1513
                      1514  *     CLEAR 'IN MEMORY' COUNT IN NODE. IF THE FILE HAS NO MORE TO
                      1515  *     READ, REMOVE IT FROM THE CHAIN AND NAMTAB
                      1516
046.137  321          1517        POP     D                   (DE) = FDN.LNK
046.140  041 013 000  1518  WPH4  LXI     H,FDN.AIM
046.143  031          1519        DAD     D
046.144  066 000      1520        MVI     M,0                 CLEAR AMOUNT IN MEMORY
046.146  353          1521        XCHG                        (HL) = FDN.LNK
046.147  043          1522        INX     H
046.150  043          1523        INX     H                                                /80.07.GC/
000.000               1524        ERRNZ   FDN.STA-FDN.LNK-2                                /80.07.GC/
046.151  176          1525        MOV     A,M                 (A) = FDN.STA
046.152  346 002      1526        ANI     ST.OPR
046.154  300          1527        RNZ                         STILL READING, AM DONE FOR THIS PHASE
000.000               1528        ERRNZ   FDN.FLG-FDN.STA-1
046.155  043          1529        INX     H                   (HL) = #FDN.FLG
046.156  106          1530        MOV     B,M                 (B) = FILE FLAGS
046.157  305          1531        PUSH    B                   SAVE
                      1532
                      1533  *     UNLINK NODE FROM LIST
                      1534
046.160  053          1535        DCX     H
046.161  053          1536        DCX     H
046.162  053          1537        DCX     H                                                /80.07.GC/
000.000               1538        ERRNZ   FDN.LNK-FDN.FLG+3          (HL) = #FDN.LNK       /80.07.GC/
046.163  345          1539        PUSH    H                                                /80.07.GC/
046.164  315 211 030  1540        CALL    $HLIHL              HL = & next node             /80.07.sc/
000.000               1541        ERRNZ   FDN.LNK                                          /80.07.GC/
046.167  042 070 060  1542        SHLD    FDNHED              New FILE list head           /80.07.sc/
046.172  052 066 060  1543        LHLD    FDNFREE                                          /80.07.sc/
046.175  353          1544        XCHG                        DE = current FREE list head  /80.07.sc/
046.176  341          1545        POP     H                   HL = current node
046.177  315 121 057  1546        CALL    $INDS               Point to rest of free list   /80.07.GC/
046.202  000 000      1547        DW      FDN.LNK                                          /80.07.GC/
046.204  042 066 060  1548        SHLD    FDNFREE             Link new at head of FREE list /80.07.sc/
```

```

046.207  315 214 056                CALL    REN             REMOVE ENTRY FROM NAMTAB

                         *          FILE IS COMPLETED. NOW WE CAN
                         *          SET SPECIAL FLAGS: SWL

046.212  301                        POP     B               (B) = FLAGS
046.213  016 377                    MVI     C,377Q          SET AS MANY AS ALLOWED
046.215  041 343 060                LXI     H,DESTFB+FB.NAM
046.220  377 060                    DB      SYSCALL,.CHFLG  CHANGE FLAGS
046.222  332 301 051                JC      DESTERR
046.225  303 260 045                JMP     WFH             TRY TO WRITE THE NEXT GUY




                         **         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) = BUFPTR/256
                         *                  (L) = OBUFLIM/256
                         *          USES    A,F


046.230  052 232 060     CBR        LHLD    OBUFLIM
000.000                             ERRNZ   OBUFPTR-OBUFLIM-1
046.233  175                        MOV     A,L
046.234  224                        SUB     H
046.235  311                        RET




                         **         DMD     - Dismount Disk                              /80.07.sc/
                         *
                         *          DMD dismounts the source diskette
                         *
                         *          ENTRY:  NONE
                         *
                         *          EXIT:   To ERROR if problems
                         *
                         *          USES:   ALL
                         *

046.236  041 324 060     DMD        LXI     H,SOURCE+DEVICE

046.241  377 203         DMD.       SCALL   .DMNMS          Dismount without a message
046.243  332 123 052                JC      ERROR

046.246  311                        RET
```

```
                          **     IFL - INITIALIZE FDN LIST.                                  /80.07.sc/
                          *
                          *      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


046.247  041 072 060      IFL    LXI      H,FDN.1
046.252  042 066 060             SHLD     FDNFREE

046.255  006 007                 MVI      B,FDNCNT-1
046.257  021 106 060             LXI      D,FDN.1+FDNELEN

046.262  315 121 057      IFL1   CALL     $INDS            set link to next node
046.265  000 000                 DW       FDN.LNK
046.267  305                     PUSH     B
046.270  142                     MOV      H,D
046.271  153                     MOV      L,E              HL = DE
046.272  001 014 000             LXI      B,FDNELEN
046.275  011                     DAD      B                Advance Next node pointer
046.276  353                     XCHG                      HL = current ; DE = next
046.277  301                     POP      B
046.300  005                     DCR      B                Count node
046.301  302 262 046             JNZ      IFL1

046.304  021 000 000             LXI      D,0
046.307  315 121 057             CALL     $INDS            last element links to NIL
046.312  000 000                 DW       FDN.LNK

046.314  353                     XCHG
046.315  042 070 060             SHLD     FDNHED           FILE list is empty
046.320  311                     RET



                          **     MAD - MOUNT ALTERNATE DISK.                                 /80.07.GC/
                          *
                          *      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     PSW      = 'C' set   if   ERROR
                          *                        = 'C' clear if NO ERROR
                          *
                          *      USES     ALL


046.321                   MAD    EQU      *
```

```

                         *       DISMOUNT CURRENT DISK

046.321  325                     PUSH    D
046.322  305                     PUSH    B                SAVE ENTRY PARAMETERS OVER SYDD CALL
046.323  315 236 046             CALL    DMD              Dismount Source Diskette
046.326  301                     POP     B
046.327  321                     POP     D

                         *       SETUP PROMPT ON FP LEDS AND CONSOLE FOR NEW DISK

046.330  076 203                 MVI     A,UO.DDU+UO.CLK+UO.HLT
046.332  062 010 040             STA     .MFLAG           HALT DISPLAY UPDATE

046.335  305                     PUSH    B

046.336  041 013 040             LXI     H,.ALEDS
046.341  076 011                 MVI     A,9
046.343  160             MAD1    MOV     M,B              SET PATTERN
046.344  043                     INX     H
046.345  075                     DCR     A
046.346  302 343 046             JNZ     MAD1             IF MORE TO BLANK

046.351  041 016 040             LXI     H,.ALEDS+3
046.354  001 003 000             LXI     B,3
046.357  315 252 030             CALL    $MOVE            MOVE IN PROMPT PATTERN

046.362  353                     XCHG                     (HL) = PATTERN
046.363  377 003                 SCALL   .PRINT           CONSOLE PROMPT
046.365  315 136 031             CALL    $TYPTX
046.370  207                     DB      BELL+200Q        BEEP CONSOLE, TOO
046.371  076 144                 MVI     A,100
046.373  315 140 002             CALL    .HORN            BEEP A WARNING

046.376  076 012         MAD2    MVI     A,DC.RDY
047.000  315 327 056             CALL    SRCDRVR
047.003  322 376 046             JNC     MAD2             Wait for device NOT ready

047.006  076 012         MAD3    MVI     A,DC.RDY
047.010  315 327 056             CALL    SRCDRVR
047.013  332 006 047             JC      MAD3             Wait for device ready

                         *       ERASE FRONT PANEL DISLPLAY

047.016  301                     POP     B

047.017  041 013 040             LXI     H,.ALEDS
047.022  076 011                 MVI     A,9
047.024  160             MAD4    MOV     M,B              SET TO PATTERN
047.025  043                     INX     H
047.026  075                     DCR     A
047.027  302 024 047             JNZ     MAD4

047.032  315 342 056             CALL    $CRLF            Output Newline to Console

047.035  315 251 047             CALL    MND              MOUNT NEW DISK
```

```
       047.040  311          1708         RET


                             1710  **     MDD     - Mount Destination Diskette
                             1711  *
                             1712  *      MDD insures that the destination diskette is mounted.
                             1713  *
                             1714  *      Since MSD requires a sysgened label, if the disk passes
                             1715  *      the RDD test, we know that it is not the same diskette
                             1716  *      as the source since the volume types must be differant.
                             1717  *
                             1718  *      ENTRY:  Source Label in SRCLAB
                             1719  *
                             1720  *      EXIT:   Destination Diskette mounted
                             1721  *
                             1722  *      USES:   ALL
                             1723  *
                             1724
       047.041  072 247 060  1725  MDD    LDA     VOLFLAG
       047.044  247          1726         ANA     A
       047.045  300          1727         RNZ                     Destination is mounted
                             1728
       047.046  315 076 047  1729         CALL    MDD1
                             1730
       047.051  052 167 062  1731         LHLD    DSTLAB+LAB.SER                           /80.10.sc/
       047.054  046 000      1732         MVI     H,0             HL = Volume Number       /80.10.sc/
       047.056  076 010      1733         MVI     A,DC.MOU
       047.060  315 033 054  1734         CALL    DSTDRVR         Insure Good volume number of label stuff
       047.063  332 123 052  1735         JC      ERROR
                             1736
       047.066  072 247 060  1737         LDA     VOLFLAG
       047.071  057          1738         CMA
       047.072  062 247 060  1739         STA     VOLFLAG         Flag Destination mounted
                             1740
       047.075  311          1741         RET


                             1743  **     MDD1
                             1744  *
                             1745
       047.076  072 241 060  1746  MDD1   LDA     DRIVES2
       047.101  247          1747         ANA     A
       047.102  302 115 047  1748         JNZ     MDD3            2-drive sysgen
                             1749
                             1750  *      Mount the Diskette
                             1751
       047.105  006 177      1752  MDD2   MVI     B,177Q          Periods Mask
       047.107  021 221 047  1753         LXI     D,MDDA
       047.112  315 321 046  1754         CALL    MAD             Mount alternate disk
                             1755
       047.115  072 250 047  1756  MDD3   LDA     MDDB
       047.120  247          1757         ANA     A
```

```
047.121  312 153 047  1758         JZ      MDD4             Label not saved in DSTLAB
                      1759
047.124  072 241 060  1760         LDA     DRIVES2
047.127  247          1761         ANA     A
047.130  300          1762         RNZ                      2-drive sysgen means no change
                      1763
047.131  315 032 050  1764         CALL    GETDLB           LABEL = Destination Label
047.134  016 000      1765         MVI     C,0              256-Byte Compare
047.136  021 167 064  1766         LXI     D,LABEL
047.141  041 167 062  1767         LXI     H,DSTLAB
047.144  315 060 030  1768         CALL    $COMP
047.147  302 105 047  1769         JNZ     MDD2             Destination Label does not match Original
                      1770
047.152  311          1771         RET
                      1772
                      1773  *      Verify Diskette type and Save Label
                      1774
047.153  021 167 062  1775  MDD4   LXI     D,DSTLAB         DE = destination Label buffer
047.156  315 035 050  1776         CALL    GETDLB.          Read the label
                      1777
047.161  072 241 060  1778         LDA     DRIVES2
047.164  247          1779         ANA     A
047.165  302 206 047  1780         JNZ     MDD5             2-drive sysgen don't care if labels match
                      1781
047.170  016 000      1782         MVI     C,0
047.172  021 167 062  1783         LXI     D,DSTLAB
047.175  041 167 063  1784         LXI     H,SRCLAB
047.200  315 060 030  1785         CALL    $COMP            Compare Source Label to Destination Label
047.203  312 105 047  1786         JZ      MDD2             Source and Destination Labels Match
                      1787
047.206  315 130 050  1788  MDD5   CALL    RDD              Require Destination Data Diskette
                      1789
047.211  072 250 047  1790         LDA     MDDB
047.214  057          1791         CMA
047.215  062 250 047  1792         STA     MDDB             Flag label saved and verified
                      1793
047.220  311          1794         RET
                      1795
047.221  102 014 044  1796  MDDA   DB      102Q,014Q,44Q
047.224  012 111 156  1797         DB      NL,'Insert Destination',':'+200Q
                      1798
047.250  000          1799  MDDB   DB      0                != 0  If label saved and type verified



                      1801  **     MND - MOUNT SYSTEM DISK.                               /80.07.sc/
                      1802  *
                      1803  *      MND MOUNTS A NEW DISK INTO 'SY' UNIT 'UNIT'
                      1804  *
                      1805  *
                      1806  *      THE LABEL MUST ALREADY HAVE BEEN READ INTO 'LABEL'
                      1807  *
                      1808  *      ENTRY   NONE
                      1809  *
                      1810  *      EXIT    To ERROR if bad problems
```

```
                              1811  *
                              1812  *         USES    ALL
                              1813  *
                              1814
      047.251  041 324 060    1815  MND       LXI     H,SOURCE+DEVICE
                              1816
      047.254  377 202        1817  MND.      SCALL   .MONMS          Mount without message
      047.256  332 123 052    1818            JC      ERROR           IF ERROR
                              1819
      047.261  311            1820            RET



                              1822  **        MSD     - Mount System Diskette                         /80.07.sc/
                              1823  *
                              1824  *         MSD insures that the system diskette is mounted
                              1825  *
                              1826  *         ENTRY:  NONE
                              1827  *
                              1828  *         EXIT:   System diskette mounted
                              1829  *
                              1830  *         USES:   ALL
                              1831  *
                              1832
      047.262  072 247 060    1833  MSD       LDA     VOLFLAG
      047.265  247            1834            ANA     A
      047.266  310            1835            RZ                      System Diskette Mounted
                              1836
      047.267  315 317 047    1837            CALL    MSD1
                              1838
      047.272  052 167 063    1839            LHLD    SRCLAB+LAB.SER                                  /80.10.sc/
      047.275  046 000        1840            MVI     H,0                                             /80.10.sc/
      047.277  076 010        1841            MVI     A,DC.MOU
      047.301  315 327 056    1842            CALL    SRCDRVR         Insure good volume number after label stuff
      047.304  332 123 052    1843            JC      ERROR
                              1844
      047.307  072 247 060    1845            LDA     VOLFLAG
      047.312  057            1846            CMA     A
      047.313  062 247 060    1847            STA     VOLFLAG         Flag System Diskette Mounted
                              1848
      047.316  311            1849            RET



                              1851  **        MSD1
                              1852  *
                              1853
      047.317  072 241 060    1854  MSD1      LDA     DRIVES2
      047.322  247            1855            ANA     A
      047.323  302 336 047    1856            JNZ     MSD3            2-drive sysgen
                              1857
                              1858  *         Mount the Diskette
                              1859
      047.326  006 377        1860  MSD2      MVI     B,377Q          B  = periods mask
      047.330  021 007 050  1861            LXI     D,MSDA
      047.333  315 321 046  1862            CALL    MAD             Mount the other diskette
                            1863
      047.336               1864  MSD.      EQU     *               Used for Initial Mount
                            1865
      047.336  072 031 050  1866  MSD3      LDA     MSDB
      047.341  247          1867            ANA     A
      047.342  312 374 047  1868            JZ      MSD4            Source Label is not saved yet
                            1869
      047.345  072 241 060  1870            LDA     DRIVES2
      047.350  247          1871            ANA     A
      047.351  300          1872            RNZ                     2-drive sysgen => no change
                            1873
      047.352  315 071 050  1874            CALL    GETSLB          LABEL = Source Label
      047.355  016 000      1875            MVI     C,0             256-Byte compare
      047.357  021 167 064  1876            LXI     D,LABEL
      047.362  041 167 063  1877            LXI     H,SRCLAB
      047.365  315 060 030  1878            CALL    $COMP
      047.370  302 326 047  1879            JNZ     MSD2            This Source Label does not match original
                            1880
      047.373  311          1881            RET
                            1882
                            1883  *         Verify Diskette type and Save Label
                            1884
      047.374  315 045 051  1885  MSD4      CALL    RSD             Require Sysgened Source Diskette
                            1886
      047.377  072 031 050  1887            LDA     MSDB
      050.002  057          1888            CMA
      050.003  062 031 050  1889            STA     MSDB            Flag label saved and verified
                            1890
      050.006  311          1891            RET
                            1892
      050.007  244 306 307  1893  MSDA      DB      244Q,306Q,307Q
      050.012  012 111 156  1894            DB      NL,'Insert Source',':'+200Q
                            1895
      050.031  000          1896  MSDB      DB      0               != 0 If label saved and type verified



                            1898  **        GETXLB - GET LABEL                                          /80.07.sc/
                            1899  *
                            1900  *         GETXLB GETS THE LABEL FROM THE DISK
                            1901  *
                            1902  *         ENTRY   NONE
                            1903  *
                            1904  *         EXIT    (PSW)    = 'C' CLEAR IF NO ERROR
                            1905  *                          = 'C' SET   IF    ERROR
                            1906  *
                            1907  *         USES    ALL
                            1908  *
                            1909
      050.032  021 167 064  1910  GETDLB    LXI     D,LABEL         DE = buffer address
                            1911
      050.035  325          1912  GETDLB.   PUSH    D
      050.036  041 000 000  1913            LXI     H,0
```

```
050.041  076 010      1914             MVI    A,DC.MOU
050.043  315 033 054  1915             CALL   DSTDRVR         Mount as volume 0
050.046  321          1916             POP    D
050.047  332 123 052  1917             JC     ERROR
                      1918
050.052  041 011 000  1919             LXI    H,DDF.LAB
050.055  001 000 001  1920             LXI    B,256
050.060  076 002      1921             MVI    A,DC.RER        READ REGARDLESS
050.062  315 033 054  1922             CALL   DSTDRVR
050.065  332 123 052  1923             JC     ERROR           Bad Error
                      1924
050.070  311          1925             RET
                      1926
                      1927
050.071  021 167 064  1928  GETSLB     LXI    D,LABEL         DE = buffer address
                      1929
050.074  325          1930  GETSLB.    PUSH   D
050.075  041 000 000  1931             LXI    H,0
050.100  076 010      1932             MVI    A,DC.MOU
050.102  315 327 056  1933             CALL   SRCDRVR         Mount as volume 0
050.105  321          1934             POP    D
050.106  332 123 052  1935             JC     ERROR
                      1936
050.111  041 011 000  1937             LXI    H,DDF.LAB
050.114  001 000 001  1938             LXI    B,256
050.117  076 002      1939             MVI    A,DC.RER
050.121  315 327 056  1940             CALL   SRCDRVR         Read Source Label
050.124  332 123 052  1941             JC     ERROR           Bad Error
                      1942
050.127  311          1943             RET



                      1945  **         RDD - REQUIRE Destination DATA DISK.                 /80.07.sc/
                      1946  *
                      1947  *          RDD CHECKS THE VOLUME TYPE TO MAKE SURE THAT IT IS A VALID
                      1948  *          DATA DISK.
                      1949  *
                      1950  *          ENTRY  DSTLAB  = Destination Label
                      1951  *
                      1952  *          EXIT   TO CALLER IF OK
                      1953  *                 TO EXIT IF BAD
                      1954  *
                      1955  *          USES   ALL
                      1956  *
                      1957
050.130  072 200 062  1958  RDD        LDA    DSTLAB+LAB.VER  A = Version of INIT to initialize disk  /2.0b/
050.133  376 040      1959             CPI    VERS            Compare to SYSGEN Version               /2.0b/
050.135  302 332 050  1960             JNZ    RDD2            Not Equal                               /2.0b/
                      1961
050.140  072 177 062  1962             LDA    DSTLAB+LAB.VLT  (A) = VOLUME TYPE
000.000               1963             ERRNZ  LAB.DAT
050.143  247          1964             ANA    A
050.144  310          1965             RZ                     IS DATA DISK, OK
                      1966
```

```
000.000                         ERRNZ   LAB.SYS-1
050.145  075                    DCR     A                   SEE IF SYSTEM DISK
050.146  302 226 050            JNZ     RDD1                DISK NOT EVEN INITIALIZED
050.151  315 136 031            CALL    $TYPTX
050.154  012 007 124            DB      NL,BELL,'This Disk Has Already Been SYSGENed.',ENL
050.223  303 376 042            JMP     EXIT

                        *       DISK IS NOT PROPERLY INITIALIZED.
                        *       (THIS CODE MAY BE ENTERED FROM OTHER ROUTINES)

050.226  315 136 031    RDD1    CALL    $TYPTX
050.231  012 007 124            DB      NL,BELL,'This Disk Must be Re-Initialized Before It Can Be '
050.315  123 131 123            DB      'SYSGENed.',ENL
050.327  303 376 042            JMP     EXIT

                        *       Not Initialized by the correct version of HDOS                  /2.0b/

050.332  315 136 031    RDD2    CALL    $TYPTX
050.335  012 007 124            DB      NL,BELL,'This disk has not been initialized by the correct '
051.021  166 145 162            DB      'version of INIT.',ENL
051.042  303 226 050            JMP     RDD1


                        **      RSD - REQUIRE SYSGENED Source DISK.                    /80.07.sc/
                        *
                        *       RSD CHECKS TO SEE IF THE MOUNTED VOLUME HAS BEEN SYSGENED.
                        *
                        *       ENTRY   NONE
                        *       EXIT    TO CALLER IF OK
                        *               TO EXIT IF ERROR
                        *       USES    ALL


051.045  021 167 063    RSD     LXI     D,SRCLAB            HL = Source Label Save Area
051.050  315 074 050            CALL    GETSLB.

051.053  072 177 063            LDA     SRCLAB+LAB.VLT  (A) = VOLUME TYPE
000.000                         ERRNZ   LAB.SYS-1
051.056  326 001                SUI     1
051.060  310                    RZ                          IS OK

051.061  322 226 050            JNC     RDD1                MUST BE INITIALIZED

051.064  315 136 031            CALL    $TYPTX
051.067  012 007 124            DB      NL,BELL,'This Disk Must be SYSGENed Before It Can be Used'
051.151  012 101 163            DB      NL,'As Input For Another SYSGEN.',ENL
051.207  303 376 042            JMP     EXIT
```

```
                     **      SSL     - Set Sysgened Flag in Label                 /80.07.sc/
                     *
                     *       SSL sets the sysgened flag in the label
                     *
                     *       ENTRY:  Destination Diskette dismounted
                     *
                     *       EXIT:   To ERROR  if problems
                     *

051.212  315 041 047   SSL     CALL    MDD               Mount Destination Diskette
051.215  041 303 060           LXI     H,DEST+DEVICE
051.220  315 241 046           CALL    DMD.              Dismount Destination Diskette

051.223  056 000               MVI     L,0
051.225  076 010               MVI     A,DC.MOU
051.227  315 033 054           CALL    DSTDRVR           Mount Diskette as volume 0

051.232  076 001               MVI     A,LAB.SYS
051.234  062 177 062           STA     DSTLAB+LAB.VLT    SET VOLUME TYPE

051.237  021 167 062           LXI     D,DSTLAB
051.242  041 011 000           LXI     H,DDF.LAB
051.245  001 000 001           LXI     B,256
051.250  076 001               MVI     A,DC.WRI
051.252  315 033 054           CALL    DSTDRVR           Write Label Back
051.255  332 123 052           JC      ERROR             BAD TROUBLE

051.260  041 303 060           LXI     H,DEST+DEVICE
051.263  315 254 047           CALL    MND.              Re-Mount the diskette
051.266  311                   RET
```

```
                          2046  **      ERROR PROCESSING ROUTINES
                          2047  *


                          2049  ***     NAMERR - FILE TYPE ERROR, OCCOURED ON FILE WHOSE NAME
                          2050  *       IS NEXT UP IN NAMTAB.
                          2051  *
                          2052  *       PROCESS VIA $FERROR
                          2053
051.267  052 370 060      2054  NAMERR  LHLD    NAMTPTR
051.272  001 366 377      2055          LXI     B,-FB.NAM
051.275  011              2056          DAD     B
051.276  303 011 057      2057          JMP     $FERROR


                          2059  **      ERROR ON FILE IN DESTFB
                          2060
051.301  041 331 060      2061  DESTERR LXI     H,DESTFB
051.304  303 011 057      2062          JMP     $FERROR


                          2064  **      INTERNAL ERRORS. SHOULD NOT OCCOUR.
                          2065
051.307  076 061          2066  IERR1   MVI     A,'1'
051.311  303 326 051      2067          JMP     INTERR
                          2068
051.314  076 062          2069  IERR2   MVI     A,'2'
051.316  303 326 051      2070          JMP     INTERR
051.321  076 063          2071  IERR3   MVI     A,'3'
051.323  303 326 051      2072          JMP     INTERR
                          2073
                          2074
051.326  365              2075  INTERR  PUSH    PSW                     SAVE CODE
051.327  315 136 031      2076          CALL    $TYPTX
051.332  007 012 123      2077          DB      BELL,NL,'SYSGEN Internal Error ','#'+200Q
051.363  361              2078          POP     PSW
051.364  315 245 057      2079          CALL    $WCHAR
051.367  315 136 031      2080          CALL    $TYPTX
051.372  012 124 150      2081          DB      NL,'This Error Should not Occur. Contact HEATH Technical'
052.057  012 103 157      2082          DB      NL,'Correspondence for Assistance.',NL
052.117  076 001          2083          MVI     A,1
052.121  377 000          2084          DB      SYSCALL,.EXIT               ABORT
```

```
                      **      ERROR - GENERAL AND SYNTAX ERRORS NOT DIRECTLY ASSOCIATED
                      *       WITH A VALID FILE NAME.


052.123  365          ERROR   PUSH    PSW               SAVE CODE
052.124  315 136 031          CALL    $TYPTX
052.127  007 105 122          DB      BELL,'ERROR -',' '+200Q
052.140  361                  POP     PSW
052.141  247                  ANA     A
052.142  372 154 052          JM      ERROR1            IS PRODUCT ERROR
052.145  046 012              MVI     H,NL              USE NL AS MESSAGE TRAIL CHAR
052.147  377 057              DB      SYSCALL,.ERROR    LOOK UP SYSTEM ERROR
052.151  303 373 042          JMP     RESTART

                      *       IS PRODUCT ERROR

052.154  041 214 052  ERROR1  LXI     H,ERRORA
052.157  276          ERROR2  CMP     M
052.160  043                  INX     H
052.161  302 157 052          JNE     ERROR2            FIND ERROR MESSAGE
052.164  315 136 031          CALL    $TYPTX
052.167  007 123 131          DB      BELL,'SYSGEN Error #',' '+200Q
052.207  377 003              DB      SYSCALL,.PRINT    PRINT MESSAGE
052.211  303 373 042          JMP     RESTART

052.214               ERRORA  DS      0                 ERROR MESSAGES
052.214  200 060 061          DB      PEC.DF,'01',ENL
052.220  201 060 062          DB      PEC.DNC,'02',ENL
052.224  202 060 063          DB      PEC.RSE,'03',ENL
052.230  203 060 064          DB      PEC.TFI,'04',ENL
052.234  204 060 065          DB      PEC.CS,'05',ENL
052.240  205 060 066          DB      PEC.IUW,'06',ENL
052.244  206 060 067          DB      PEC.IDF,'07',ENL
052.250  207 060 070          DB      PEC.CO,'08',ENL
```

```
                             2123  **      AEN - ADD ENTRY TO 'NAMTAB'
                             2124  *
                             2125  *       AEN EXPANDS THE FILE INFO IN PIO.XXX INTO A FILE DESCRIPTOR
                             2126  *       AND ENTERS IT IN THE NAMTAB TABLE.
                             2127  *
                             2128  *       If the QUERY flag is set, the user is interrogated
                             2129  *       before the file is actually copied.  If a yes response
                             2130  *       is given, then the file is actually added, otherwise,
                             2131  *       the file is skipped.                                          /80.07.sc/
                             2132  *
                             2133  *
                             2134  *       ENTRY   NONE
                             2135  *       EXIT    'C' SET IF WILDCARD
                             2136  *       USES    ALL
                             2137
                             2138
052.254  041 337 052  2139  AEN     LXI     H,AENA
052.257  315 063 055  2140          CALL    CDA             CONVERT DIRECTORY FORMAT TO ASCII FORMAT
052.262  326 001      2141          SUI     1               'C' SET IF WILDCARD
052.264  365          2142          PUSH    PSW             SAVE FLAG
                      2143
052.265  332 276 052  2144          JC      AEN1            Ignore query for wild-carded files /80.07.sc/
052.270  315 036 053  2145          CALL    CQF             Check Query Flag                   /80.07.sc/
052.273  302 335 052  2146          JNZ     AEN2            Don't Copy this file               /80.07.sc/
                      2147
052.276  052 364 060  2148  AEN1    LHLD    NAMTLEN                                            /80.07.sc/
052.301  001 021 000  2149          LXI     B,FB.NAML
052.304  011          2150          DAD     B               INCREASE SIZE
052.305  042 364 060  2151          SHLD    NAMTLEN
052.310  353          2152          XCHG                    (DE) = NEW LENGTH
052.311  052 366 060  2153          LHLD    NAMTMAX
052.314  175          2154          MOV     A,L             SEE IF WILL OVERFLOW
052.315  223          2155          SUB     E
052.316  174          2156          MOV     A,H
052.317  232          2157          SBB     D
052.320  334 066 056  2158          CC      INA             INCREASE NAMTAB ALLOCATION
052.323  041 221 065  2159          LXI     H,NAMTAB-FB.NAML
052.326  031          2160          DAD     D               (HL) = *TO* ADDRESS
052.327  021 337 052  2161          LXI     D,AENA          (DE) = *FROM* ADDRESS
052.332  315 252 030  2162          CALL    $MOVE           MOVE ENTRY IN
                      2163
052.335  361          2164  AEN2    POP     PSW             (PSW) = WILDCARD FLAG              /80.07.GC/
052.336  311          2165          RET
                      2166
052.337               2167  AENA    DS      FB.NAML



                      2169  **      BSL - BUILD SOURCE FILE LIST.
                      2170  *
                      2171  *       BSL CRACKS THE LIST OF THE SOURCE FILES FROM THE COMMAND LINE AND
                      2172  *       BUILDS THEM INTO THE NAMTAB MANAGED TABLE.
                      2173  *       WILD CARDS ENCOUNTERED ARE EXPANDED.
                      2174  *
                      2175  *       ENTRY   (A) <> 0 IF TO ASK ABOUT '*.*' USE
```

```
                            *       EXIT    'C' CLEAR IF OK
                            *               'C' SET IF ERROR
                            *                (A) = CODE
                            *       USES    ALL


052.360  062 022 053        BSL     STA     BSLA                SAVE ASK FLAG
052.363  315 120 056                CALL    LSN                 LOCATE SOURCE NAME

                            *       GO THROUGH SOURCE LIST CRACKING NAMES

052.366  176                BSL1    MOV     A,M
052.367  247                        ANA     A
052.370  310                        RZ                          ALL DONE
052.371  021 310 060                LXI     D,SOURCE+DEFAULT         Source default definition /80.07.sc/
052.374  315 046 054                CALL    CAD                 CONVERT ASCII NAME TO DIRECTORY FORMAT
052.377  330                        RC                          ERROR
053.000  315 277 056                CALL    SND                 SET NEW DEFAULTS
053.003  345                        PUSH    H                   SAVE LINE ADDRESS
053.004  315 154 055                CALL    EWS                 EXPAND WILDCARD SPECIFICATION
053.007  332 012 053                JC      BSL2                IF ERROR
053.012  341                BSL2    POP     H                   RESTORE LINE ADDRESS
053.013  330                        RC                          USER REFUSED *.*
053.014  315 262 056                CALL    SFS                 SKIP FILE SEPERATOR (BLANKS AND/OR COMMA)
053.017  303 366 052                JMP     BSL1                DO MORE

053.022  000                BSLA    DB      0                   <>0 IF TO CHECK FOR *.*



                            **      CFS - COMPUTE FILE SIZE
                            *
                            *       CFS COMPUTES THE SIZE OF A FILE. THE DEVICE'S GRT MUST BE IN
                            *       THE 'GRT' BUFFER.
                            *
                            *       ENTRY   (A) = FIRST GROUP NUMBER
                            *       EXIT    (DE) = SIZE
                            *       USES    ALL


053.023  021 000 000        CFS.    LXI     D,0
053.026  247                CFS1    ANA     A
053.027  310                        RZ                          ALL DONE
053.030  157                        MOV     L,A
053.031  176                        MOV     A,M                 (A) = NEXT GRT
053.032  023                        INX     D
053.033  303 026 053                JMP     CFS1                TRY AGAIN
```

```
                           2222  **     CQF      - Check Query Flag                               /80.07.sc/
                           2223  *
                           2224  *      CQF checks the query flag, and if it is set, asks
                           2225  *      the user if the file is to be transfered by typing
                           2226  *      the filename followed by a question mark.  If the
                           2227  *      response begins with a 'Y' the file is transfered,
                           2228  *      else, it is to be ignored.
                           2229  *
                           2230  *      ENTRY:  PIO.xxx = File specification
                           2231  *
                           2232  *      EXIT:   PSW      = 'Z'  if     to Copy
                           2233  *                         'NZ' if NOT to Copy
                           2234  *
                           2235  *      USES:   ALL
                           2236  *
                           2237
  053.036  072 245 060     2238  CQF    LDA     QUERY
  053.041  247             2239         ANA     A
  053.042  310             2240         RZ                      NO Query, so transfer file
                           2241
  053.043  315 054 053     2242         CALL    CQF.
  053.046  365             2243         PUSH    PSW
  053.047  315 342 056     2244         CALL    $CRLF
  053.052  361             2245         POP     PSW
  053.053  311             2246         RET
                           2247
  053.054  041 213 065     2248  CQF.   LXI     H,PIO.DIR+DIR.NAM
  053.057  315 271 057     2249         CALL    $TFN.           Type the File Name
  053.062  315 136 031     2250         CALL    $TYPTX
  053.065  077 240         2251         DB      '?',' '+200Q
                           2252
  053.067  377 007         2253         SCALL   .CLRCO          Clear the console
  053.071  315 237 057     2254         CALL    $RCHAR
  053.074  315 203 057     2255         CALL    $MCU
  053.077  376 131         2256         CPI     'Y'
  053.101  310             2257         RZ                      Copy the file
                           2258
  053.102  376 116         2259         CPI     'N'
  053.104  302 112 053     2260         JNZ     CQF1            Illegal
  053.107  366 001         2261         ORI     1               'NZ' => DON'T copy the file
  053.111  311             2262         RET
                           2263
  053.112  315 136 031     2264  CQF1   CALL    $TYPTX
  053.115  012 207         2265         DB      NL,BELL+200Q    Bell user for illegal character
  053.117  303 054 053     2266         JMP     CQF.



                           2268  **     CSF - CHECK FOR SPECIAL FILE.                              /80.07.sc/
                           2269  *
                           2270  *      CSF CHECKS TO SEE IF THE FILE NAME (IN DIRECTORY FORMAT)
                           2271  *      SUPPLIED MATCHES ONE OF A LIST OF 'NOT-TO-BE-PROCESSED'
                           2272  *      FILES. THE LIST IS:
                           2273  *
                           2274  *      Since the table is terminated by a zero byte,
                          2275  *       the files to be copied may be modified by
                          2276  *       zeroing appropriately:
                          2277  *
                          2278  *                 CSFB            No GRT.SYS, RGT.SYS, DIRECT.SYS
                          2279  *                 CSFC            or HDOS.SYS, HDOSOVL0.SYS, HDOSOVL1.SYS
                          2280  *                                    SYSCMD.SYS, PIP.ABS
                          2281  *                 CSFD            or SY.DVD
                          2282  *                                 or xx.DVD
                          2283  *
                          2284  *       GRT.SYS
                          2285  *       RGT.SYS
                          2286  *       DIRECT.SYS
                          2287  *
                          2288  *       ENTRY     (DE) = ADDRESS OF DIRECTORY BLOCK
                          2289  *                 CSFB    = 0 If not to search extra files        /80.07.GC/
                          2290  *                         = CSFC if to search extra files         /80.07.sc/
                          2291  *
                          2292  *       EXIT      'Z' SET IF MATCH
                          2293  *                 'Z' CLEAR OTHERSIZE
                          2294  *
                          2295  *       USES      A,F
                          2296  *
                          2297
    053.122  305          2298  CSF     PUSH      B
    053.123  325          2299          PUSH      D
    053.124  345          2300          PUSH      H               SAVE POINTERS
                          2301
    053.125  041 163 053  2302          LXI       H,CSFA          (A) = START OF LIST
    053.130  325          2303  CSF1    PUSH      D               SAVE NAME
    053.131  345          2304          PUSH      H               SAVE LIST ADDRESS
    053.132  016 015      2305          MVI       C,DIRIDL
    053.134  315 060 030  2306          CALL      $COMP           SEE IF MATCH
    053.137  341          2307          POP       H
    053.140  321          2308          POP       D
    053.141  312 157 053  2309          JE        CSF2            GOT MATCH
    053.144  076 015      2310          MVI       A,DIRIDL
    053.146  315 101 030  2311          CALL      $DADA.          POINT TO NEXT ENTRY
    053.151  176          2312          MOV       A,M
    053.152  247          2313          ANA       A
    053.153  302 130 053  2314          JNZ       CSF1            MORE TO CHECK
                          2315
                          2316  *       NO MATCH
                          2317
    053.156  074          2318          INR       A               CLEAR 'Z'
    053.157  341          2319  CSF2    POP       H
    053.160  321          2320          POP       D               RESTORE REGS
    053.161  301          2321          POP       B
    053.162  311          2322          RET
                          2323
    053.163  107 122 124  2324  CSFA    DB        'GRT',0,0,0,0,0,'SYS',0,0          GRT.SYS
    000.000               2325          ERRNZ     *-CSFA-DIRIDL    ENTRYS MUST BE  DIRIDL  LONG
    053.200  122 107 124  2326          DB        'RGT',0,0,0,0,0,'SYS',0,0          RGT.SYS
    053.215  104 111 122  2327          DB        'DIRECT',0,0,'SYS',0,0
    053.232  110 104 117  2328  CSFB    DB        'HDOS',0,0,0,0,'SYS',0,0           HDOS.SYS        /80.07.GC/
    053.247  110 104 117  2329          DB        'HDOSOVL0SYS',0,0                  HDOSOVL0.SYS    /80.07.GC/
    053.264  110 104 117  2330          DB        'HDOSOVL1SYS',0,0                  HDOSOVL1.SYS    /80.07.GC/
```

```
053.301  123 131 123              DB      'SYSCMD',0,0,'SYS',0,0         SYSCMD.SYS        /80.07.GC/
053.316  120 111 120              DB      'PIP',0,0,0,0,0,'ABS',0,0      PIP.ABS           /80.07.GC/
053.333  123 131 000        CSFC  DB      'SY',0,0,0,0,0,0,'DVD',0,0     SY.DVD            /80.07.GC/
053.350  170 170 000        CSFD  DB      'xx',0,0,0,0,0,0,'DVD',0,0     xx.DVD            /80.07.sc/
053.365  000                      DB      0                New end of table                /80.07.sc/



                            **    CWM - CHECK WILDCARD MATCH.
                            *
                            *     CWM CHECKS TO SEE IF A WILDCARDED FIELD MATCHES A NON-WILDCARDED
                            *     FIELD.
                            *
                            *     ENTRY   (DE) = ADDRESS OF WC NAME
                            *             (HL) = ADDRESS OF NON/WC NAME
                            *             (B) = NUMBER OF CHARACTERS TO CHECK
                            *     EXIT    'Z' SET IF MATCH
                            *              (HL) = (HL)+(B)
                            *              (DE) = (DE) = (B)
                            *             'Z' CLEAR IF NO MATCH
                            *     USES    A,F,B,D,E,H,L


053.366  032                CWM   LDAX    D
053.367  247                      ANA     A
053.370  372 375 053              JM      CWM1                IS MATCH
053.373  276                      CMP     M
053.374  300                      RNE                         NO MATCH
053.375  023                CWM1  INX     D
053.376  043                      INX     H                   ADVANCE ADDRESSES
053.377  005                      DCR     B
054.000  302 366 053              JNZ     CWM                 GO FOR MORE
054.003  311                      RET                         GOT MATCH



                            **    DDF - DECODE DESTINATION FILE.                               /80.07.GC/
                            *
                            *     DDF DECODES THE DESTINATION FILE NAME FROM THE COMMAND LINE.
                            *
                            *     Must have default specification.
                            *
                            *     ENTRY   NONE
                            *     EXIT    'C' CLEAR IF OK
                            *              (A) = 0 IF NAME HAS WILDCARDS
                            *              (A) = 1 IF NO WILDCARD USED
                            *              DESTFB+FB.NAM CONTAINS A COMPLETE DESTINATION FILE NAME
                            *              (HL) = COMMAND LINE POINTER UPDATED
                            *             'C' SET IF ERROR
                            *              (A) = CODE
                            *     USES    ALL


054.004  052 242 060        DDF   LHLD    LINEP                                                /80.07.GC/
```

```

             *       (HL) = ADDRESS FOR NAME

054.007  021 267 060  DDF2    LXI     D,DEST+DEFAULT                                     /80.07.GC/
054.012  315 046 054          CALL    CAD             CONVERT ASCII NAME TO DIRECTORY FORMAT
054.015  330                  RC                      ERROR

054.016  176                  MOV     A,M
054.017  376 075              CPI     '='
054.021  076 206              MVI     A,PEC.IDF       ASSUME ILLEGAL DESTINATION FORMAT
054.023  067                  STC
054.024  300                  RNE                     MUST HAVE '='

             *       HAVE NAME DECODED. EXPAND INTO DESTFB+FB.NAM

054.025  041 343 060          LXI     H,DESTFB+FB.NAM
054.030  303 063 055          JMP     CDA             CONVERT DIRECTORY FORMAT TO ASCII



             **      DSTDRVR - Destination Driver
             *
             *       DSTDRVR invokes the DESTination device driver.
             *
             *       ENTRY:  NONE
             *
             *       EXIT:   NONE
             *
             *       USES:   NONE
             *

054.033  365          DSTDRVR PUSH    PSW
054.034  072 302 060          LDA     DEST+UNIT
054.037  062 061 041          STA     AIO.UNI
054.042  361                  POP     PSW
054.043  303 277 060          JMP     DEST+DRIVER



             **      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
```

```
                      2431  *               (HL) = POINTER TO TEXT
                      2432  *       EXIT    'C' SET IF ERROR
                      2433  *                (A) = ERROR CODE
                      2434  *               'C' CLEAR IF OK
                      2435  *                (HL) = POINTS PAST FILE NAME
                      2436  *                'Z' SET IF NULL NAME
                      2437  *                'Z' CLEAR IF NON-NULL
                      2438  *                 PIO.DIR.NAM = NAME
                      2439  *                 PIO.DIR.EXT = EXTENSION
                      2440  *                 PIO.DEV = DEVICE CODE
                      2441  *                 PIO.UNI = UNIT NUMBER (ASCII DIGIT)
                      2442  *       USES    ALL
                      2443
                      2444
054.046  257          2445  CAD     XRA     A               SET TO NULLS
054.047  303 054 054  2446          JMP     CAD0
                      2447
054.052  076 200      2448  CAD.    MVI     A,200Q
054.054  345          2449  CAD0    PUSH    H
054.055  062 313 054  2450          STA     CADA            SAVE DEFAULT VALUE
                      2451
                      2452  *       SET DEFAULTS IN PIO.xxx
                      2453
054.060  041 210 065  2454          LXI     H,PIO.DEV
054.063  001 003 000  2455          LXI     B,3
054.066  315 252 030  2456          CALL    $MOVE           SET DEFALUT DEVICE
054.071  001 003 000  2457          LXI     B,3
054.074  041 223 065  2458          LXI     H,PIO.DIR+DIR.EXT
054.077  315 252 030  2459          CALL    $MOVE           SET DEFAULT EXTENSION
054.102  341          2460          POP     H
054.103  315 250 057  2461          CALL    $SOB            SKIP BLANKS
054.106  006 000      2462          MVI     B,0
054.110  376 077      2463          CPI     '?'
054.112  312 141 054  2464          JE      CAD1            IS '?'
054.115  376 052      2465          CPI     '*'
054.117  312 141 054  2466          JE      CAD1            IS '*'
054.122  376 056      2467          CPI     '.'
054.124  312 141 054  2468          JE      CAD1            IS '.'
054.127  376 101      2469          CPI     'A'
054.131  332 301 054  2470          JC      CAD4            NOT NAME
054.134  376 133      2471          CPI     'Z'+1
054.136  322 301 054  2472          JNC     CAD4            NOT NAME
                      2473
                      2474  *       HAVE ALPHA STRING. CRACK IT
                      2475
054.141  315 314 054  2476  CAD1    CALL    DNT             DECODE NEXT TOKEN
054.144  332 307 054  2477          JC      CAD5            ERROR
054.147  376 072      2478          CPI     ':'
054.151  302 204 054  2479          JNE     CAD2            NOT DEVICE
                      2480
                      2481  *       HAVE EXPLICIT DEVICE
                      2482
054.154  043          2483          INX     H               SKIP ':'
054.155  076 003      2484          MVI     A,3
054.157  271          2485          CMP     C
054.160  332 307 054  2486          JC      CAD5            TOO MANY CHARACTERS
```

```
054.163  001 003 000  2487              LXI     B,3
054.166  345          2488              PUSH    H                SAVE (HL)
054.167  041 210 065  2489              LXI     H,PIO.DEV
054.172  315 252 030  2490              CALL    $MOVE            SET EXPLICIT DEVICE
054.175  341          2491              POP     H
054.176  315 314 054  2492              CALL    DNT              DECODE NEXT TOKEN
054.201  332 307 054  2493              JC      CAD5             ERROR
                      2494
                      2495    *         DECODE NAME
                      2496
054.204  001 010 000  2497    CAD2      LXI     B,8              (BC) = COUNT
054.207  345          2498              PUSH    H                SAVE TEXT ADDR
                      2499
                      2500    *         SEE IF NAME IS  UNSPECIFIED
                      2501
054.210  041 213 065  2502              LXI     H,PIO.DIR+DIR.NAM
054.213  345          2503              PUSH    H                SAVE ADDRESS OF DIR.NAM
054.214  315 252 030  2504              CALL    $MOVE            MOVE IN NAME
054.217  341          2505              POP     H                (HL) = #PIO.DIR+DIR.NAM
054.220  176          2506              MOV     A,M
054.221  247          2507              ANA     A
054.222  302 240 054  2508              JNZ     CAD2.6           IS SPECIFIED
054.225  072 313 054  2509              LDA     CADA             (A) = FILL CHARACTER
054.230  016 010      2510              MVI     C,8              (C) = COUNT
054.232  167          2511    CAD2.4    MOV     M,A
054.233  043          2512              INX     H
054.234  015          2513              DCR     C
054.235  302 232 054  2514              JNZ     CAD2.4
054.240  341          2515    CAD2.6    POP     H
054.241  176          2516              MOV     A,M              (A) = DELIMITER
054.242  376 056      2517              CPI     '.'
054.244  302 277 054  2518              JNE     CAD3             NOT EXTENSION
                      2519
                      2520    *         HAVE EXPLICIT EXTENSION
                      2521
054.247  043          2522              INX     H
054.250  315 314 054  2523              CALL    DNT
054.253  332 307 054  2524              JC      CAD5             ERROR
054.256  076 003      2525              MVI     A,3
054.260  271          2526              CMP     C
054.261  332 307 054  2527              JC      CAD5             TOO LONG
054.264  001 003 000  2528              LXI     B,3
054.267  345          2529              PUSH    H                SAVE TEXT POINTER
054.270  041 223 065  2530              LXI     H,PIO.DIR+DIR.EXT
054.273  315 252 030  2531              CALL    $MOVE            MOVE EXTENSION
054.276  341          2532              POP     H
                      2533
                      2534    *         DONE WITH NAME. MUST HAVE LEGIT DELIMITER
                      2535
054.277  006 001      2536    CAD3      MVI     B,1              (B) = NAME PRESENT FLAG
                      2537
                      2538    *         END OF NAME. EXIT
                      2539    *         (B) = 0 IF NULL, (B) <> 0 IF NON-NULL
                      2540
054.301  315 250 057  2541    CAD4      CALL    $SOB             SKIP BLANKS
054.304  170          2542              MOV     A,B
```

```
054.305  247           2543          ANA     A               SET 'Z' IF NULL
054.306  311           2544          RET
                       2545
                       2546  *       ERROR
                       2547
054.307  076 007       2548  CAD5    MVI     A,EC.IFN        ILLEGAL FILE NAME
054.311  067           2549          STC
054.312  311           2550          RET
                       2551
054.313  000           2552  CADA    DB      0               FILL CHARACTER FOR OMITTED NAME FIELD



                       2554  **      DNT - DECODE NEXT TOKEN.
                       2555  *
                       2556  *       DNT COPIES THE NEXT ALPHANUMERIC FIELD INTO A ZERO-FILLED WORK AREA.
                       2557  *
                       2558  *       ENTRY   (HL) = TEXT POINTER
                       2559  *       EXIT    'C' SET IF ERROR
                       2560  *               'C' CLEAR IF OK
                       2561  *               (A) = DELIMTER CHARACTER
                       2562  *               (HL) UPDATED TO DELIMITER CHARACTER
                       2563  *               (DNTA) = STRING
                       2564  *               (C) = LENGTH
                       2565  *               (DE) = #DNTA
                       2566  *       USES    ALL
                       2567
                       2568
054.314  021 026 055   2569  DNT     LXI     D,DNTA
054.317  016 011       2570          MVI     C,9             (C) = SIZE OF DNTA
054.321  101           2571          MOV     B,C             (B) = MAX ALLOWED +1
054.322  257           2572          XRA     A
054.323  022           2573  DNT1    STAX    D               ZERO BUFFER
054.324  023           2574          INX     D
054.325  015           2575          DCR     C
054.326  302 323 054   2576          JNZ     DNT1
054.331  021 026 055   2577          LXI     D,DNTA
                       2578
                       2579  *       COPY CHARACTERS
                       2580
054.334  176           2581  DNT2    MOV     A,M
054.335  376 077       2582          CPI     '?'
054.337  076 200       2583          MVI     A,200Q
054.341  312 376 054   2584          JE      DNT3            IS MATCHONE
054.344  176           2585          MOV     A,M
054.345  376 052       2586          CPI     '*'
054.347  312 010 055   2587          JE      DNT5            IS WILDCARD
054.352  376 060       2588          CPI     '0'
054.354  332 021 055   2589          JC      DNT4            NOT ALPHANUMERIC
054.357  376 072       2590          CPI     '9'+1
054.361  332 376 054   2591          JC      DNT3            NUMERIC
054.364  376 101       2592          CPI     'A'
054.366  332 021 055   2593          JC      DNT4            DELIMITER
054.371  376 133       2594          CPI     'Z'+1
054.373  322 021 055   2595          JNC     DNT4            DELIMITER
```

```
                          2596
                          2597  *       HAVE GOOD CHARACTER
                          2598
    054.376  022          2599  DNT3    STAX    D               STORE CHAR
    054.377  023          2600          INX     D
    055.000  043          2601          INX     H
    055.001  014          2602          INR     C               COUNT
    055.002  005          2603          DCR     B               LIMIT DECREMENT
    055.003  302 334 054  2604          JNZ     DNT2            NOT OVERFLOW
                          2605
                          2606  *       OVERFLOW
                          2607
    055.006  067          2608          STC                     FLAG ERR
    055.007  311          2609          RET
                          2610
                          2611  *       IS '*' WILDCARD
                          2612
    055.010  076 200      2613  DNT5    MVI     A,200Q
    055.012  022          2614          STAX    D
    055.013  023          2615          INX     D
    055.014  005          2616          DCR     B
    055.015  302 010 055  2617          JNZ     DNT5            FILL WITH MATCH ONE
    055.020  043          2618          INX     H               SKIP '*'
                          2619
                          2620  *       END OF STRING
                          2621
    055.021  247          2622  DNT4    ANA     A               CLEAR 'C'
    055.022  021 026 055  2623          LXI     D,DNTA          SET POINTER
    055.025  311          2624          RET
                          2625
    055.026               2626  DNTA    DS      9               WORK AREA



                          2628  **      EBM - EXPAND BUFFER TO MAXIMUM.
                          2629  *
                          2630  *       EBM IS CALLED TO EXPAND THE BUFFER 'BUF' TO THE MAXIMUM SIZE.
                          2631  *       WHICH DOES NOT REQUIRE THE OVERLAYING OF THE SYSTEM.
                          2632  *
                          2633  *       ENTRY   NONE
                          2634  *       EXIT    (BUFSIZ) = BUFFER SIZE (MULTIPLE OF 256)
                          2635  *       USES    ALL
                          2636
                          2637
    055.037  052 320 040  2638  EBM     LHLD    S.SYSM
    055.042  021 366 377  2639          LXI     D,-10
    055.045  031          2640          DAD     D               THROW IN SOME SLOP
    055.046  377 052      2641          DB      SYSCALL,.SETTP
    055.050  332 307 051  2642          JC      IERR1           NOT ENOUGH MEMORY
    055.053  052 322 040  2643          LHLD    S.USRM
                          2644
    055.056  174          2645          MOV     A,H             (A) = LIMIT/256
    055.057  062 232 060  2646          STA     OBUFLIM         SET LIMIT
    055.062  311          2647          RET
```

```
                           **       CDA - CONVERT DIRECTORY FORMAT TO ASCII.
                           *
                           *        CDA COPIES A DIRECTORY ENTRY FROM PIO.XXX TO A TARGET FIELD.
                           *        THE DEVICE SPECIFICATION (IN PIO.DEV AND PIO.UNI) IS ALSO ENCODED.
                           *        THE TARGET FIELD IS LEFT IN THE FORM:
                           *
                           *        DEV:NAME.XXX <00>
                           *
                           *        ENTRY   (HL) = FWA NAME FIELD
                           *        EXIT    (A) = 0, HAVE WILDCARD
                           *                    = 1, NO WILDCARDS USED
                           *                'C' CLEAR
                           *        USES    ALL


 055.063  001 000 003      CDA      LXI     B,3*256             (B) = CHARACTER COUNT, (C) = WILDCARD FLAG
 055.066  021 210 065               LXI     D,PIO.DEV
 055.071  315 127 055               CALL    CDA5                COPY IT
 055.074  066 072                   MVI     M,':'
 055.076  043                       INX     H
 055.077  006 010                   MVI     B,8
 055.101  021 213 065               LXI     D,PIO.DIR+DIR.NAM
 055.104  315 127 055               CALL    CDA5                COPY IT
 055.107  066 056                   MVI     M,'.'
 055.111  043                       INX     H
 055.112  006 003                   MVI     B,3
 000.000                            ERRNZ   DIR.EXT-DIR.NAM-8
 055.114  315 127 055               CALL    CDA5                COPY IT
 055.117  066 000                   MVI     M,0                 FLAG END OF NAME
 055.121  171                       MOV     A,C                 (A) (BIT 7) = 1 IF WILDCARDS
 055.122  007                       RLC
 055.123  057                       CMA
 055.124  346 001                   ANI     1                   =0 IF WILDCARD
 055.126  311                       RET


                           **       CDA5 - CONVERT DIRECTORY FIELD TO ASCII.
                           *
                           *        ZEROS ARE IGNORED, 200Q WILDCARDS ARE MAPPED TO '?'
                           *
                           *        ENTRY   (DE) = FROM
                           *                (HL) = TO
                           *                (B) = COUNT
                           *                (C) = ORA ACCUMULATOR
                           *        EXIT    (DE) ADVANCED
                           *                (HL) = (HL)+(B)
                           *                (C) = (C) .OR. (FROM CHARACTERS PROCESSED)
                           *        USES    ALL


 055.127  032              CDA5     LDAX    D                   (A) = CHARACTER
 055.130  261                       ORA     C
 055.131  117                       MOV     C,A
 055.132  032                       LDAX    D
 055.133  023                       INX     D
 055.134  247                       ANA     A
```

```
055.135  312 147 055  2704           JZ      CDA7           IS 00
055.140  362 145 055  2705           JP      CDA6           NOT 200Q
055.143  076 077      2706           MVI     A,'?'
055.145  167          2707  CDA6     MOV     M,A
055.146  043          2708           INX     H              INCREMENT TO
055.147  005          2709  CDA7     DCR     B
055.150  302 127 055  2710           JNZ     CDA5           IF MORE TO GO
055.153  311          2711           RET



                      2713  **       EWS - EXPAND WILDCARD SPECIFICATION.
                      2714  *
                      2715  *        DWS ENTERS THE FILE NAME IN FIO.XXX INTO THE MANAGED TABLE
                      2716  *        NAMTAB. IF THE FILE NAME CONTAINS WILDCARDS, THE DIRECTORY
                      2717  *        IS READ FOR ELIGIBLE FILES.
                      2718  *
                      2719  *        ENTRY   FIO.XXX = FILE NAME
                      2720  *        EXIT    'C' CLEAR IF OK
                      2721  *                'C' SET IF ERROR
                      2722  *        USES    ALL
                      2723
                      2724
055.154  315 254 052  2725  EWS      CALL    AEN            TRY TO ENTER IT
055.157  320          2726           RNC                    NO WILDCARDS, AM DONE
                      2727
                      2728  *        IS WILDCARD. LOOK UP DEVICE TYPE
                      2729
055.160  052 364 060  2730           LHLD    NAMTLEN
055.163  021 221 065  2731           LXI     D,NAMTAB-FB.NAML
055.166  031          2732           DAD     D              (HL) = ADDRESS OF LAST ENTRY
055.167  315 046 054  2733           CALL    CAD            CONVERT ASCII NAME TO DIRECTORY FORMAT
055.172  052 364 060  2734           LHLD    NAMTLEN
055.175  021 357 377  2735           LXI     D,-FB.NAML
055.200  031          2736           DAD     D
055.201  042 364 060  2737           SHLD    NAMTLEN        REMOVE WILDCARD FROM TABLE
055.204  315 214 057  2738           CALL    $MOVEL
055.207  003 000 210  2739           DW      3,FIO.DEV,DIRNAM        SET DIRECTORY NAME IN XXX:DIRECT.SYS
055.215  315 214 057  2740           CALL    $MOVEL
055.220  013 000 213  2741           DW      8+3,FIO.DIR+DIR.NAM,EWSC        SAVE WILDCARD PATTERN
055.226  001 015 056  2742           LXI     B,EWSB
055.231  041 250 060  2743           LXI     H,DIRNAM
055.234  377 053      2744           DB      SYSCALL,.DECODE GET INFORMATION ABOUT DEVICE
055.236  330          2745           RC                     ERROR
055.237  072 015 056  2746           LDA     EWSB           SEE IF A DIRECTORY DEVICE
055.242  346 001      2747           ANI     DT.DD
055.244  076 005      2748           MVI     A,EC.DNS                ASSUME DEVICE NOT SUITABLE
055.246  067          2749           STC
055.247  310          2750           RZ                     ERROR
                      2751
                      2752  *        IS DIRECTORY DEVICE. OPEN DIRECTORY
                      2753
055.250  041 250 060  2754           LXI     H,DIRNAM
055.253  076 002      2755           MVI     A,CN.DIR
055.255  377 042      2756           DB      SYSCALL,.OPENR
```

```
055.257  076 200               MVI     A,FEC.DF
055.261  330                   RC                      DEVICE FORMAT FAILURE

                *       READ DIRECTORY ENTRYS FOR MATCH

055.262  052 121 041    EWS1   LHLD    DIRWRKP                                       /79.12.GC/
055.265  353                   XCHG                    DE = POINTER TO THE SCRATCH   /79.12.GC/
055.266  001 000 002           LXI     B,512
055.271  076 002               MVI     A,CN.DIR
055.273  325                   PUSH    D               SAVE ADDRESS
055.274  377 004               DB      SYSCALL,.READ   READ BLOCK
055.276  341                   POP     H               (HL) = DIRECTORY ADDRESS
055.277  332 002 056           JC      EWS7            ALL DONE

                *       LOOK AT DIRECTORY BLOCK FOR MATCHES

055.302  345                   PUSH    H
055.303  052 121 041           LHLD    DIRWRKP                                       /79.12.GC/
055.306  021 373 001           LXI     D,DIS.ENL                                     /79.12.GC/
055.311  031                   DAD     D                                             /79.12.GC/
055.312  116                   MOV     C,M             C  = LENGTH                   /79.12.GC/
055.313  341                   POP     H                                             /79.12.GC/

                *       CHECK NEXT ENTRY

055.314  176            EWS3   MOV     A,M             (A) = 1ST CHAR THIS ENTRY
055.315  247                   ANA     A
055.316  312 262 055           JZ      EWS1            END OF BLOCK
000.000                        ERRNZ   DF.EMP-377Q
055.321  074                   INR     A
055.322  312 374 055           JZ      EWS6            ENTRY EMPTY
000.000                        ERRNZ   DF.CLR-376Q
055.325  074                   INR     A
055.326  312 002 056           JZ      EWS7            END OF LIST
055.331  345                   PUSH    H
055.332  021 053 056           LXI     D,EWSC
055.335  006 013               MVI     B,8+3
055.337  315 366 053           CALL    CWM             CHECK WILDCARD MATCH
055.342  302 373 055           JNZ     EWS4            NO MATCH

                *       HAVE MATCH, ADD TO LSIT

055.345  321                   POP     D               (DE) = FROM
055.346  325                   PUSH    D
055.347  315 122 053           CALL    CSF             CHECK FOR SPECIAL FILE
055.352  312 373 055           JZ      EWS4            IS SPECIAL FILE, DONT ENTER
055.355  305                   PUSH    B               SAVE (C)
055.356  001 013 000           LXI     B,8+3
055.361  041 213 065           LXI     H,PIO.DIR+DIR.NAM
055.364  315 252 030           CALL    $MOVE
055.367  315 254 052           CALL    AEN             ADD TO TABLE
055.372  301                   POP     B               RESTORE (C)

                *       LOOKUP NEXT ENTRY

055.373  341            EWS4   POP     H
```

```
055.374  006 000        EWS6    MVI     B,0
055.376  011                    DAD     B               POINT TO NEXT
055.377  303 314 055            JMP     EWS3

                        *       ALL DONE. CLOSE DIRECTORY FILE

056.002  076 002        EWS7    MVI     A,CN.DIR
056.004  377 046                DB      SYSCALL,.CLOSE
056.006  311                    RET

056.007  123 131 060    EWSA    DB      'SY0',200Q,200Q,200Q

056.015                 EWSB    DS      30

056.053                 EWSC    DS      8+3             WILDCARD PATTERN FOR DIRECTORY SEARCH



                        **      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

056.066  041 367 060    INA     LXI     H,NAMTMAX+1
056.071  064                    INR     M               INCREMENT LENGTH
056.072  041 235 060            LXI     H,BUFPTR+1
056.075  064                    INR     M               MOVE BUFFER
056.076  052 236 060            LHLD    BUFSIZ
056.101  174                    MOV     A,H
056.102  265                    ORA     L
056.103  076 021                MVI     A,EC.NEM        FLAG OUT OF MEMORY IF BUFFER NOT EMPTY
056.105  302 123 052            JNZ     ERROR
056.110  305                    PUSH    B
056.111  325                    PUSH    D
056.112  315 241 056            CALL    SBE             NOTIFY SYSTEM
056.115  321                    POP     D
056.116  301                    POP     B
056.117  311                    RET



                        **      LSN - LOCATE SOURCE NAME                        /80.07.GC/
                        *
                        *       LSN SCANS THE COMMAND LINE FOR THE FIRST SOURCE FILE NAME.
                        *
                        *       ENTRY   NONE
                        *       EXIT    (HL) = 1ST FILE NAME FWA
                        *       USES    A,F,H,L

056.120  052 242 060    LSN     LHLD    LINEP           HL = Line Pointer       /80.07.sc/
```

```
                        2863
056.123  176            2864  LSN1    MOV     A,M
056.124  043            2865          INX     H
056.125  376 075        2866          CPI     '='
056.127  310            2867          RE                      GOT IT
                        2868
056.130  247            2869          ANA     A
056.131  302 123 056    2870          JNZ     LSN1            MORE LINE
                        2871
056.134  052 242 060    2872          LHLD    LINEP           Is no '='                      /80.07.sc/
056.137  311            2873          RET


                        2875  **      MWN - MERGE WILDCARD NAMES.
                        2876  *
                        2877  *       MWN MERGES A COMPLETELY SPECIFIED FILENAME WITH A WILDCARDED COMPLETELY
                        2878  *       SPECIFIED FILE NAME.
                        2879  *
                        2880  *       BOTH FILE NAMES SHOULD HAVE THE SAME DEVICE SPECIFICATION.
                        2881  *
                        2882  *       FILE NAME FORMAT:
                        2883  *
                        2884  *       DEV:NAMEXXXX.EXT 00
                        2885  *
                        2886  *       ENTRY   (BC) = ADDRESS OF WILDCARDED ASCII NAME
                        2887  *               (DE) = ADDRESS OF NON-WC ASCII NAME
                        2888  *               (HL) = ADDRESS FOR RESULTANT ASCII NAME
                        2889  *       EXIT    NONE
                        2890  *       USES    ALL
                        2891
                        2892
056.140  345            2893  MWN     PUSH    H               SAVE TARGET ADDRESS
056.141  305            2894          PUSH    B               SAVE WC PATTERN
056.142  353            2895          XCHG                    (HL) = MASTER NAME
056.143  315 046 054    2896          CALL    CAD             CONVERT TO DIRECTORY FORMAT
056.146  315 214 057    2897          CALL    $MOVEL
056.151  013 000 213    2898          DW      8+3,FIO.DIR,MWNA        (MWNA) = DECODED MASTER
056.157  341            2899          POP     H               (HL) = WC PATTERN
056.160  315 046 054    2900          CALL    CAD             (FIO.DIR) = WC PATTERN
056.163  021 167 065    2901          LXI     D,MWNA          (DE) = MASTER PATTERN
056.166  041 213 065    2902          LXI     H,FIO.DIR       (DE) = WC PATTERN ADDRESS
056.171  016 013        2903          MVI     C,8+3           MERGE NAME AND EXTENSION
                        2904
                        2905  *       MERGE NAMES
                        2906
056.173  176            2907  MWN1    MOV     A,M             (A) = WC PATTERN
056.174  247            2908          ANA     A
056.175  362 201 056    2909          JP      MWN2            USE THIS
056.200  032            2910          LDAX    D               IS MATCH CHARACTER, USE MASTER INSTEAD
056.201  167            2911  MWN2    MOV     M,A             STORE CHARACTER
056.202  023            2912          INX     D
056.203  043            2913          INX     H
056.204  015            2914          DCR     C
056.205  302 173 056    2915          JNZ     MWN1            MERGE TILL DONE
```

```
056.210  341          2916           POP     H               (HL) = TARGET ADDRESS
056.211  303 063 055  2917           JMP     CDA             CONVERT DIRECTORY FORMAT TO ASCII


                      2919  **       REN - REMOVE ENTRY FROM *NAMTAB*
                      2920  *
                      2921  *        REN REMOVES THE FIRST 'FB.NAML' BYTES FROM NAMTAB.
                      2922  *
                      2923  *        THE AMOUNT (FB.NAML) IS REMOVED FROM THE SIZE OF THE TABLE. THE
                      2924  *        TABLE IS NOT CHECKED FOR UNDERFLOW, THE CALLER MUST GUARANTEE THE
                      2925  *        PRESENSE OF AT LEAST FB.NAML BYTES IN NAMTAB.
                      2926  *
                      2927  *        ENTRY   NONE
                      2928  *        EXIT    NONE
                      2929  *        USES    ALL
                      2930
                      2931
056.214  052 364 060  2932  REN      LHLD    NAMTLEN
056.217  021 357 377  2933           LXI     D,-FB.NAML
056.222  031          2934           DAD     D               REMOVE COUNT FROM LEN
056.223  042 364 060  2935           SHLD    NAMTLEN
056.226  104          2936           MOV     B,H
056.227  115          2937           MOV     C,L             (BC) = REMAINING LENGTH
056.230  021 263 065  2938           LXI     D,NAMTAB+FB.NAML        (DE) = START OF 2ND ENTRY
056.233  041 242 065  2939           LXI     H,NAMTAB
056.236  303 252 030  2940           JMP     $MOVE           MOVE DOWN AND RETURN


                      2942  **       SBE - SET BUFFER EMPTY.
                      2943  *
                      2944  *        THE SYSTEM IS NOTIFIED.
                      2945  *
                      2946  *        ENTRY   NONE
                      2947  *        EXIT    NONE
                      2948  *        USES    ALL
                      2949
                      2950
056.241  041 000 000  2951  SBE      LXI     H,0
056.244  042 236 060  2952           SHLD    BUFSIZ
056.247  052 234 060  2953           LHLD    BUFPTR          (HL) = BUFFER FWA (AND LWA!)
056.252  043          2954           INX     H
056.253  043          2955           INX     H
056.254  377 052      2956           DB      SYSCALL,.SETTP
056.256  320          2957           RNC                     OK
056.257  303 123 052  2958           JMP     ERROR           NOT ENOUGH ROOM
```

```
                       **      SFS - SKIP FILE SEPERATOR.
                       *
                       *       SFS IS CALLED TO SKIP OVER THE CHARACTERS SEPERATING ONE
                       *       FILE NAME FROM ANOTHER ON THE LINE. THE FILES MAY BE SEPERATED
                       *       BY BLANKS OR A COMMA ALONE, OR BY BLANKS WITH A COMMA. THE
                       *       SYNTAX IS
                       *
                       *       <BLANKS> <,> <BLANKS>
                       *
                       *       ONE, TWO OR ALL THREE FIELDS MAY BE PRESENT.
                       *
                       *       ENTRY   (HL) = POINT TO START OF SEP FIELD
                       *       EXIT    (HL) ADVANCED PAST SEPERATOR FIELD
                       *       USES    A,F,H,L


056.262  315 250 057   SFS     CALL    $SOB            SKIP BLANKS
056.265  176                   MOV     A,M
056.266  376 054               CPI     ','
056.270  302 274 056           JNE     SFS1            NOT ,
056.273  043                   INX     H               SKIP ,
056.274  303 250 057   SFS1    JMP     $SOB            GET ANY MORE BLANKS AND EXIT



                       **      SND - SET NEW DEFAULTS.
                       *
                       *       SND IS CALLED TO SET A NEW DEFAULT DEVICE AND EXTENSION
                       *       IN THE 'DEFALT' AREA.
                       *
                       *       ENTRY   PIO.DEV = DEVICE CODE
                       *               PIO.UNI = UNIT #
                       *               PIO.DIR+DIR.EXT = EXTENSION
                       *       EXIT    NONE
                       *       USES    NONE


056.277  315 054 031   SND     CALL    $SAVALL         SAVE REGS
000.000                        ERRNZ   PIO.UNI-PIO.DEV-2
056.302  315 214 057           CALL    $MOVEL
056.305  003 000               DW      3
056.307  210 065               DW      PIO.DEV
056.311  310 060               DW      SOURCE+DEFAULT                          /80.07.GC/

056.313  315 214 057           CALL    $MOVEL
056.316  003 000               DW      3
056.320  223 065               DW      PIO.DIR+DIR.EXT
056.322  313 060               DW      SOURCE+DEFAULT+3                        /80.07.GC/
056.324  303 047 031           JMP     $RSTALL         RETURN
```

```
                      3008 **       SRCDRVR - Source Device Driver
                      3009 *
                      3010 *        SRCDRVR invokes the Source device Driver
                      3011 *
                      3012 *        ENTRY:  NONE
                      3013 *
                      3014 *        EXIT:   NONE
                      3015 *
                      3016 *        USES:   NONE
                      3017 *
                      3018
056.327  365          3019 SRCDRVR PUSH    PSW
056.330  072 323 060  3020         LDA     SOURCE+UNIT
056.333  062 061 041  3021         STA     AIO.UNI
056.336  361          3022         POP     PSW
056.337  303 320 060  3023         JMP     SOURCE+DRIVER
```

```
     056.342             3026            XTEXT   CDEHL



                         3028X **        $CDEHL - COMPARE (DE) TO (HL)
                         3029X *
                         3030X *         $CDEHL COMPARES (DE) TO (HL) FOR EQUALITY.
                         3031X *
                         3032X *         ENTRY   NONE
                         3033X *         EXIT    'Z' SET IF (DE) = (HL)
                         3034X *         USES    A,F
                         3035X
                         3036X
     030.216             3037X $CDEHL    EQU     30216A          IN H17 ROM
     056.342             3038            XTEXT   CHL



                         3040X **        $CHL - COMPLEMENT (HL).
                         3041X *
                         3042X *         (HL) = -(HL)            TWO'S COMPLEMENT
                         3043X *
                         3044X *         ENTRY   NONE
                         3045X *         EXIT    NONE
                         3046X *         USES    A,F,H,L
                         3047X
                         3048X
     030.224             3049X $CHL      EQU     30224A          IN H17 ROM
     056.342             3050            XTEXT   COMP



                         3052X **        $COMP - COMPARE TWO CHARACTER STRINGS.
                         3053X *
                         3054X *         $COMP COMPARES TWO BYTE STRINGS.
                         3055X *
                         3056X *         ENTRY   (C) = COMPARE COUNT
                         3057X *                 (DE) = FWA OF STRING #1
                         3058X *                 (HL) = FWA OF STRING #2
                         3059X *         EXIT    'Z' CLEAR; IS MIS-MATCH
                         3060X *                  (C) = LENGTH REMAINING
                         3061X *                  (DE) = ADDRESS OF MISMATCH IN STRING#1
                         3062X *                  (HL) = ADDRESS OF MISMATCH IN STRING #2
                         3063X *                 'C' SET; HAVE MATCH
                         3064X *                  (C) = 0
                         3065X *                  (DE) = (DE) + (OC)
                         3066X *                  (HL) = (HL) + (OC)
                         3067X *         USES    A,F,C,D,E,H,L
                         3068X
                         3069X
     030.060             3070X $COMP     EQU     30060A          IN H17 ROM
     056.342             3071            XTEXT   CRLF
```

```
                        3073X **        $CRLF - TYPE CARRIAGE RETURN/ LINE FEED
                        3074X *
                        3075X *         $CRLF IS USED TO GENERATE PADDED CRLF'S.
                        3076X *
                        3077X *         ENTRY   NONE
                        3078X *         EXIT    (A) = 0
                        3079X *         USES    A,F
                        3080X
                        3081X
  056.342  076 012      3082X $CRLF     MVI     A,NL
  056.344  377 002      3083X           DB      SYSCALL,.SCOUT
  056.346  257          3084X           XRA     A
  056.347  311          3085X           RET
  056.350               3086            XTEXT   DADA



                        3088X **        $DADA - PERFORM (H,L) = (H,L) + (0,A)
                        3089X *
                        3090X *         ENTRY   (H,L) = BEFORE VALUE
                        3091X *                 (A) = BEFORE VALUE
                        3092X *         EXIT    (H,L) = (H,L) + (0,A)
                        3093X *                 'C' SET IF OVERFLOW
                        3094X *         USES    F,H,L
                        3095X
                        3096X
  030.072               3097X $DADA     EQU     30072A          IN H17 ROM
  056.350               3098            XTEXT   DADA2



                        3100X **        $DADA. - ADD (0,A) TO (H,L)
                        3101X *
                        3102X *         ENTRY   NONE
                        3103X *         EXIT    (HL) = (HL) + (0A)
                        3104X *         USES    A,F,H,L
                        3105X
                        3106X
  030.101               3107X $DADA.    EQU     30101A          IN H17 ROM
  056.350               3108            XTEXT   DTB



                        3110X **        $DTB - DELETE TRAILING BLANKS.
                        3111X *
                        3112X *         $DTB DELETES THE TRAILING BLANKS FROM A CODED LINE.
                        3113X *
                        3114X *         ENTRY   (HL) = LINE FWA
                        3115X *         EXIT    (A) = LENGTH OF RESULT (ENCLUDING 00 TERMINATOR BYTE)
                        3116X *         USES    A,F
                        3117X
                        3118X
  056.350  325          3119X $DTB      PUSH    D               SAVE (DE)
```

```
056.351  124          3120X          MOV     D,H
056.352  135          3121X          MOV     E,L             (DE) = FWA
056.353  033          3122X          DCX     D               (DE) = FWA-1
056.354  176          3123X $DTB1    MOV     A,M
056.355  043          3124X          INX     H
056.356  247          3125X          ANA     A               FIND END OF LINE
056.357  302 354 056  3126X          JNZ     $DTB1
056.362  053          3127X          DCX     H               (HL) = ADDRESS OF TERMINATING ZERO BYTE
                      3128X
                      3129X *        GOT END OF LINE, DELETE TRAILING BLANKS
                      3130X
056.363  053          3131X $DTB2    DCX     H               BACKUP ONE CHARACTER
056.364  315 216 030  3132X          CALL    $CDEHL
056.367  312 000 057  3133X          JE      $DTB3           GONE PAST FRONT OF LINE, MUST BE ALL BLANKS
056.372  176          3134X          MOV     A,M
056.373  376 040      3135X          CPI     ' '
056.375  312 363 056  3136X          JE      $DTB2           GOT BLANK
                      3137X
                      3138X *        HAVE TRIMED LINE, COMPUTE LENGTH
                      3139X
057.000  043          3140X $DTB3    INX     H
057.001  066 000      3141X          MVI     M,0             TERMINATE LINE
057.003  175          3142X          MOV     A,L
057.004  223          3143X          SUB     E               (A) = LENGTH +1 (FOR 00 BYTE)
057.005  353          3144X          XCHG
057.006  043          3145X          INX     H               (HL) = LINE FWA
057.007  321          3146X          POP     D               RESTORE (DE)
057.010  311          3147X          RET
057.011               3148           XTEXT   DU66



                      3150X **       $DU66 - UNSIGNED 16 / 16 DIVIDE.
                      3151X *
                      3152X *        (HL) = (BC)/(DE)
                      3153X *
                      3154X *        ENTRY   (BC), (DE) PRESET
                      3155X *        EXIT    (HL) = RESULT
                      3156X *                (DE) = REMAINDER
                      3157X *        USES    ALL
                      3158X
                      3159X
030.106               3160X $DU66    EQU     30106A          IN H17 ROM
057.011               3161           XTEXT   FERROR



                      3163X **       $FERROR - PROCESS FILE ERRORS.
                      3164X *
                      3165X *        $FERROR IS CALLED TO COMPLAIN ABOUT AN ERROR ENCOUNTERED
                      3166X *        WHEN PROCESSING FILES.
                      3167X *
                      3168X *        ENTRY   (A) = ERROR CODE
                      3169X *                (HL) = ADDRESS OF FILE NAME - FB.NAM
```

```
                              3170X *        EXIT     TO RESTART
                              3171X *        USES     ALL
                              3172X
                              3173X
  057.011  365                3174X $FERROR PUSH     PSW              SAVE CODE
  057.012  315 136 031        3175X          CALL     $TYPTX
  057.015  012 007 105        3176X          DB       NL,BELL,'ERROR ON FILE',' '+200Q
  057.035  021 012 000        3177X          LXI      D,FB.NAM
  057.040  031                3178X          DAD      D
                              3179X
                              3180X *        PRINT FILE NAME
                              3181X
  057.041  176                3182X $FERR1   MOV      A,M
  057.042  043                3183X          INX      H                ADVANCE MESSAGE
  057.043  247                3184X          ANA      A
  057.044  312 055 057        3185X          JZ       $FERR2
  057.047  315 245 057        3186X          CALL     $WCHAR
  057.052  303 041 057        3187X          JMP      $FERR1
                              3188X
                              3189X *        TYPE ERROR MESSAGE
                              3190X
  057.055  315 136 031        3191X $FERR2   CALL     $TYPTX
  057.060  040 055 240        3192X          DB       ' -',' '+200Q
  057.063  046 012            3193X          MVI      H,NL
  057.065  361                3194X          POP      PSW              (A) = CODE
  057.066  377 057            3195X          DB       SYSCALL,.ERROR
  057.070  303 373 042        3196X          JMP      RESTART          EXIT
  057.073                     3197           XTEXT    HLIHL



                              3199X **       $HLIHL - LOAD HL INDIRECT THROUGH HL.
                              3200X *
                              3201X *        (HL) = ((HL))
                              3202X *
                              3203X *        ENTRY    NONE
                              3204X *        EXIT     NONE
                              3205X *        USES     A,H,L
                              3206X
  030.211                     3207X $HLIHL  EQU      030211A          IN H17 ROM
  057.073                     3208           XTEXT    ILDEHL



                              3210X **       ILDEHL  -  INDEXED LOAD OF DE FROM HL
                              3211X *
                              3212X *        'DE' GET THE FULL WORD VALUE POINTED TO BY 'HL', AND 'HL' IS
                              3213X *        INCREMENTED BY TWO.
                              3214X *
                              3215X *        ENTRY:   HL       = ADDRESS OF FULL WORD VALUE
                              3216X *
                              3217X *        EXIT:    DE       = (HL)
                              3218X *                 HL       = HL + 2
                              3219X *
```

```
                          3220X *        USES:    DE
                          3221X *
                          3222X
      057.073  136        3223X ILDEHL  MOV      E,M
      057.074  043        3224X         INX      H
      057.075  126        3225X         MOV      D,M
      057.076  043        3226X         INX      H
      057.077  311        3227X         RET
      057.100             3228          XTEXT    INDL



                          3230X **       $INDL - INDEXED LOAD.
                          3231X *
                          3232X *        $INDL LOADS DE WITH THE TWO BYTES AT (HL)+DISPLACMENT
                          3233X *
                          3234X *        THIS ACTS AS AN INDEXED FULL WORD LOAD.
                          3235X *
                          3236X *        (DE) = ( (HL) + DSPLACEMENT )
                          3237X *
                          3238X *        ENTRY    ((RET)) = DISPLACMENT (FULL WORD)
                          3239X *                 (HL) = TABLE ADDRESS
                          3240X *        EXIT     TO (RET+2)
                          3241X *        USES     A,F,D,E
                          3242X
                          3243X
      030.234             3244X $INDL   EQU      30234A          IN H17 ROM
      057.100             3245          XTEXT    INDXX



                          3247X **       $INDLB  -  INDEXED LOAD BYTE
                          3248X *
                          3249X *        BYTE INDEXED LOAD PRIMITIVE
                          3250X *
                          3251X *        ENTRY:   HL       = BASE ADDRESS
                          3252X *                 (RET)    = FULL WORD RELOCATION
                          3253X *
                          3254X *        EXIT:    A        = ( HL + (RET) )
                          3255X *
                          3256X *        USES:    A
                          3257X *
                          3258X
      057.100  353        3259X $INDLB  XCHG                     DE = BASE
      057.101  343        3260X         XTHL                     SAVE  .DE.
      057.102  325        3261X         PUSH     D               SAVE  BASE
      057.103  305        3262X         PUSH     B               SAVE  .BC.
                          3263X
      057.104  116        3264X         MOV      C,M
      057.105  043        3265X         INX      H
      057.106  106        3266X         MOV      B,M             BC = OFFSET
      057.107  043        3267X         INX      H               HL =  .RET.
                          3268X
      057.110  353        3269X         XCHG                     HL = BASE
```

```
     057.111  011             3270X           DAD      B                HL = BASE + OFFSET
     057.112  176             3271X           MOV      A,M              A  = ( BASE + OFFSET )
     057.113  353             3272X           XCHG                      HL =  .RET.
                              3273X
     057.114  301             3274X           POP      B                RESTORE  .BC.
     057.115  321             3275X           POP      D                RESTORE  BASE
     057.116  343             3276X           XTHL                      HL = .DE. ; (SP) = .RET.
     057.117  353             3277X           XCHG                      DE = .DE. ; HL = BASE
     057.120  311             3278X           RET



                              3280X **        $INDS    -  INDEXED STORE
                              3281X *
                              3282X *         INDEXED STORE PRIMITIVE.
                              3283X *
                              3284X *         ENTRY:   HL = BASE ADDRESS
                              3285X *                  DE = VALUE TO STORE
                              3286X *
                              3287X *         EXIT:    ( HL + (RET) ) = DE
                              3288X *
                              3289X *         USES:    NONE
                              3290X *
                              3291X
     057.121  315 055 060     3292X $INDS     CALL     XCHGBC
     057.124  343             3293X           XTHL                      SAVE  .BC.
     057.125  325             3294X           PUSH     D
     057.126  315 073 057     3295X           CALL     ILDEHL           DE = OFFSET
     057.131  315 055 060     3296X           CALL     XCHGBC           BC = .RET.
     057.134  353             3297X           XCHG                      DE = BASE ; HL = OFFSET
     057.135  031             3298X           DAD      D                HL = BASE + OFFSET
     057.136  353             3299X           XCHG
     057.137  343             3300X           XTHL                      SAVE  BASE
     057.140  353             3301X           XCHG                      DE = VALUE
     057.141  315 176 057     3302X           CALL     ISDEHL
     057.144  341             3303X           POP      H                HL = BASE
     057.145  315 055 060     3304X           CALL     XCHGBC
     057.150  343             3305X           XTHL                      RESTORE  .BC.
     057.151  315 055 060     3306X           CALL     XCHGBC
     057.154  311             3307X           RET



                              3309X **        $INDSB   -  INDEXED BYTE STORE
                              3310X *
                              3311X *         INDEXED BYTE STORE.
                              3312X *
                              3313X *         ENTRY:   A        = VALUE TO STORE
                              3314X *                  HL       = BASE ADDRESS
                              3315X *                  (RET)    = OFFSET
                              3316X *
                              3317X *         EXIT:    NONE
                              3318X *
                              3319X *         USES:    PSW
```

```
                           3320X *
                           3321X
     057.155  353          3322X $INDSB  XCHG                        DE = BASE
     057.156  343          3323X         XTHL                        SAVE  .DE.
     057.157  325          3324X         PUSH    D                   SAVE  BASE
     057.160  305          3325X         PUSH    B                   SAVE  .BC.
                           3326X
     057.161  116          3327X         MOV     C,M
     057.162  043          3328X         INX     H
     057.163  106          3329X         MOV     B,M                 BC = OFFSET
     057.164  043          3330X         INX     H                   HL =  .RET.
                           3331X
     057.165  353          3332X         XCHG                        HL = BASE
     057.166  011          3333X         DAD     B                   HL = BASE + OFFSET
     057.167  167          3334X         MOV     M,A                 ( BASE + OFFSET ) = A
     057.170  353          3335X         XCHG
                           3336X
     057.171  301          3337X         POP     B                   RESTORE  .BC.
     057.172  321          3338X         POP     D                   RESTORE  BASE
     057.173  343          3339X         XTHL                        HL = .DE. ; (SP) = .RET.
     057.174  353          3340X         XCHG                        DE = .DE. ; HL = BASE
     057.175  311          3341X         RET
     057.176               3342          XTEXT   ISDEHL



                           3344X **      ISDEHL  -  INDEXED STORE OF DE AT HL
                           3345X *
                           3346X *       STORE 'DE' AT THE ADDRESS POINTED TO BY 'HL', AND INCREMENT 'HL'
                           3347X *       BY 2.
                           3348X *
                           3349X *       ENTRY:  DE       = VALUE
                           3350X *               HL       = ADDRESS OF VALUE
                           3351X *
                           3352X *       EXIT:   (HL)     = DE
                           3353X *               HL       = HL + 2
                           3354X *
                           3355X *       USES:   HL
                           3356X *
                           3357X
     057.176  163          3358X ISDEHL  MOV     M,E
     057.177  043          3359X         INX     H
     057.200  162          3360X         MOV     M,D
     057.201  043          3361X         INX     H
     057.202  311          3362X         RET
     057.203               3363          XTEXT   MCU
```

```
                          3365X **       MCU - MAP LOWER CASE TO UPPER CASE.
                          3366X *
                          3367X *        MCU MAPS A LOWER CASE ALPHABETIC TO UPPER
                          3368X *        CASE.
                          3369X *
                          3370X *        ENTRY   (A) = CHARACTER
                          3371X *        EXIT    (A) = CHARACTER RESULT
                          3372X *        USES    A,F
                          3373X
                          3374X
      057.203  376 141    3375X $MCU     CPI     'a'
      057.205  330        3376X          RC                      NOT LOWER CASE
      057.206  376 173    3377X          CPI     'z'+1
      057.210  320        3378X          RNC                     NOT LOWER CASE
      057.211  326 040    3379X          SUI     'a'-'A'
      057.213  311        3380X          RET
      057.214             3381           XTEXT   MOVE


                          3383X **       $MOVE - MOVE DATA
                          3384X *
                          3385X *        $MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                          3386X *        IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                          3387X *        FIRST TO LAST.
                          3388X *
                          3389X *        IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                          3390X *        LAST TO FIRST.
                          3391X *
                          3392X *        THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                          3393X *
                          3394X *        ENTRY   (BC) = COUNT
                          3395X *                (DE) = FROM
                          3396X *                (HL) = TO
                          3397X *        EXIT    MOVED
                          3398X *                (DE) = ADDRESS OF NEXT FROM BYTE
                          3399X *                (HL) = ADDRESS OF NEXT *TO* BYTE
                          3400X *                'C' CLEAR
                          3401X *        USES    ALL
                          3402X
                          3403X
      030.252             3404X $MOVE    EQU     30252A          IN H17 ROM
      057.214             3405           XTEXT   MOVEL


                          3407X **       $MOVEL - MOVE DATA
                          3408X *
                          3409X *        $MOVEL MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                          3410X *        IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                          3411X *        FIRST TO LAST.
                          3412X *
                          3413X *        IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                          3414X *        LAST TO FIRST.
                          3415X *
                          3416X *        THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                          3417X *
                          3418X *        CALL    $MOVEL
                          3419X *        DW      COUNT
                          3420X *        DW      FROM
                          3421X *        DW      TO
                          3422X *
                          3423X *        ENTRY   ((SP)) = RET
                          3424X *                (RET+0) = COUNT (WORD VALUE)
                          3425X *                (RET+2) = FROM
                          3426X *                (RET+4) = TO
                          3427X *        EXIT    TO (RET+6)
                          3428X *                (DE) = ADDRESS OF NEXT FROM BYTE
                          3429X *                (HL) = ADDRESS OF NEXT *TO* BYTE
                          3430X *                'C' CLEAR
                          3431X *        USES    ALL
                          3432X
                          3433X
057.214  341              3434X $MOVEL   POP     H               (HL) = RET
057.215  116              3435X          MOV     C,M
057.216  043              3436X          INX     H
057.217  106              3437X          MOV     B,M             (BC) = COUNT
057.220  043              3438X          INX     H
057.221  136              3439X          MOV     E,M
057.222  043              3440X          INX     H
057.223  126              3441X          MOV     D,M             (DE) = FROM
057.224  043              3442X          INX     H
057.225  325              3443X          PUSH    D               ((SP)) = FROM
057.226  136              3444X          MOV     E,M
057.227  043              3445X          INX     H
057.230  126              3446X          MOV     D,M             (DE) = TO
057.231  043              3447X          INX     H
057.232  343              3448X          XTHL                    ((SP)) = RET, (HL) = FROM
057.233  353              3449X          XCHG                    (DE) = FROM , (HL) = TO
057.234  303 252 030      3450X          JMP     $MOVE           MOVE IT
057.237                   3451           XTEXT   MU86



                          3453X **       $MU86 - MULTIPLY 8X16 UNSIGNED.
                          3454X *
                          3455X *        $MU86 MULTIPLIES A 16 BIT VALUE BY A 8
                          3456X *        BIT VALUE.
                          3457X *
                          3458X *        ENTRY   (A) = MULTIPLIER
                          3459X *                (DE) = MULTIPLICAND
                          3460X *        EXIT    (HL) = RESULT
                          3461X *                'Z' SET IF NOT OVERFLOW
                          3462X *        USES    A,F,H,L
                          3463X
                          3464X
031.007                   3465X $MU86    EQU     31007A          IN H17 ROM
057.237                   3466           XTEXT   RCHAR
```

```
                                3468X **      $RCHAR - READ SINGLE CHARACTER FROM CONSOLE.
                                3469X *
                                3470X *       ENTRY    NONE
                                3471X *       EXIT     (A) = CHARACTER
                                3472X *       USES     A,F
                                3473X
                                3474X
   057.237  377 001             3475X $RCHAR  DB       SYSCALL,.SCIN
   057.241  332 237 057         3476X         JC       $RCHAR          NOT READY
   057.244  311                 3477X         RET
                                3478X
   057.245  377 002             3479X $WCHAR  DB       SYSCALL,.SCOUT
   057.247  311                 3480X         RET
   057.250                      3481          XTEXT    SAVALL


                                3483X **      $RSTALL - RESTORE ALL REGISTERS.
                                3484X *
                                3485X *       $RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND
                                3486X *       RETURNS TO THE PREVIOUS CALLER.
                                3487X *
                                3488X *       ENTRY    (SP) = PSW
                                3489X *                (SP+2) = BC
                                3490X *                (SP+4) = DE
                                3491X *                (SP+6) = HL
                                3492X *                (SP+8) = RET
                                3493X *       EXIT     TO *RET*, REGISTERS RESTORED
                                3494X *       USES     ALL
                                3495X
                                3496X
   031.047                      3497X $RSTALL EQU      31047A          IN H17 ROM


                                3499X **      $SAVALL - SAVE ALL REGISTERS ON STACK.
                                3500X *
                                3501X *       $SAVALL SAVES ALL THE REGISTERS ON THE STACK.
                                3502X *
                                3503X *       ENTRY    NONE
                                3504X *       EXIT     (SP) = PSW
                                3505X *                (SP+2) = BC
                                3506X *                (SP+4) = DE
                                3507X *                (SP+6) = HL
                                3508X *       USES     H,L
                                3509X
                                3510X
   031.054                      3511X $SAVALL EQU      31054A          IN H17 ROM
   057.250                      3512          XTEXT    SOB
```

```
                      3514X **       $SOB - SKIP OVER BLANKS.
                      3515X *
                      3516X *        $SOB IS CALLED TO SKIP AN ARBITRARILY LONG STRING OF BLANKS AND TABS.
                      3517X *
                      3518X *        ENTRY    (HL) = FWA OF (POSSIBLE) BLANK STRING
                      3519X *        EXIT     (HL) = LWA+1 OF BLANK STRING (UNCHANGED IF NO BLANKS)
                      3520X *                 (A) = FIRST NON-BLANK, NON-TAB CHARACTER EEN
                      3521X *        USES     A,F,H,L
                      3522X
                      3523X
057.250  053          3524X $SOB     DCX      H                    PRE-DECREMENT
057.251  043          3525X $SOB1    INX      H
057.252  176          3526X          MOV      A,M
057.253  376 040      3527X          CPI      ' '
057.255  312 251 057  3528X          JE       $SOB1                GOT BLANK
057.260  376 011      3529X          CPI      TAB
057.262  312 251 057  3530X          JE       $SOB1                GOT TAB
057.265  311          3531X          RET
057.266               3532           XTEXT    TFN



                      3534X **       $TFN - TYPE FILE NAME.
                      3535X *
                      3536X *        $TFN TYPES THE FILE WHOSE NAME APPEARS IN AIO.XXX
                      3537X *
                      3538X *        ENTRY    NONE
                      3539X *        EXIT     NONE
                      3540X *        USES     A,F,B,H,L
                      3541X
                      3542X
057.266  041 062 041  3543X $TFN     LXI      H,AIO.DIR+DIR.NAM
057.271  006 010      3544X $TFN.    MVI      B,8                                                /80.07.GC/
057.273  315 304 057  3545X          CALL     $TFN1                TYPE NAME
057.276  315 333 057  3546X          CALL     $TYPCH
057.301  056          3547X          DB       '.'
057.302  006 003      3548X          MVI      B,3
                      3549X
057.304  176          3550X $TFN1    MOV      A,M
057.305  247          3551X          ANA      A
057.306  304 337 057  3552X          CNZ      $TYPC.
057.311  043          3553X          INX      H
057.312  005          3554X          DCR      B
057.313  302 304 057  3555X          JNZ      $TFN1
057.316  311          3556X          RET
057.317               3557           XTEXT    TJMP
```

```
                           3559X **       $TJMP - TABLE JUMP.
                           3560X *
                           3561X *        USAGE
                           3562X *
                           3563X *        CALL      $TJMP            (A) = INDEX
                           3564X *        DW        ADDR1
                           3565X *        .          .
                           3566X *        .          .
                           3567X *        .          .
                           3568X *        DW        ADDRN
                           3569X *
                           3570X *        ENTRY     (A) = INDEX
                           3571X *        EXIT      TO PROCESSOR
                           3572X *                  (A) = INDEX*2
                           3573X *        USES      NONE.
                           3574X
                           3575X
  031.061                  3576X $TJMP    EQU      31061A           IN H17 ROM, (A) = INDEX*2
                           3577X
  031.062                  3578X $TJMP.   EQU      31062A           IN H17 ROM
  057.317                  3579           XTEXT    TYPCC


                           3581X **       $TYPCC - TYPE A CHARACTER STRING BY COUNT.
                           3582X *
                           3583X *        $TYPCC TYPES A STRING OF CHARACTERS. THE CALLER SUPPLIES
                           3584X *        THE CHARACTER ADDRESS AND COUNT.
                           3585X *
                           3586X *        ENTRY    (HL) = ADDRESS
                           3587X *                 (A) = COUNT
                           3588X *        EXIT     (HL) = LAST CHARACTER ADDRESS+1
                           3589X *        USES     A,F,H,L
                           3590X
                           3591X
  057.317                  3592X $TYPCC   EQU      *
  057.317  247             3593X          ANA      A
  057.320  310             3594X          RZ                        NOTHING TO TYPE
  057.321  365             3595X          PUSH     PSW              SAVE COUNT
  057.322  176             3596X          MOV      A,M              (A) = CHARACTER
  057.323  043             3597X          INX      H
  057.324  377 002         3598X          DB       SYSCALL,.SCOUT
  057.326  361             3599X          POP      PSW
  057.327  075             3600X          DCR      A
  057.330  303 317 057     3601X          JMP      $TYPCC
  057.333                  3602           XTEXT    TYPCH
```

```
                        3604X **       $TYPCH - TYPE SINGLE CHARACTER.
                        3605X *
                        3606X *        ENTRY   (RET) = CHARACTER
                        3607X *        EXIT    TO (RET)+1
                        3608X *                (A) = CHARACTER TYPED
                        3609X
                        3610X
 057.333  343           3611X $TYPCH   XTHL                    (HL) = RETURN ADDRESS
 057.334  176           3612X          MOV     A,M             (A) = CHARACTER
 057.335  043           3613X          INX     H
 057.336  343           3614X          XTHL                    RESTORE ADVANCED EXIT ADDRESS
                        3615X
                        3616X **       $TYPC. - TYPE SINGLE CHARACTER.
                        3617X *
                        3618X *        ENTRY   (A) = CHARACTER
                        3619X *        EXIT    TO (RET)
                        3620X
 057.337  377 002       3621X $TYPC.   DB      SYSCALL,.SCOUT
 057.341  311           3622X          RET
 000.001                3623  $CMP$    EQU     1
 057.342                3624           XTEXT   TYPLN


                        3626X **       $TYPLN - TYPE LINE.
                        3627X *
                        3628X *        $TYPLN IS CALLED TO TYPE A LINE OF TEXT. ZERO BYTES ARE
                        3629X *        TAKEN AS CRLF (WITH THE PROPER PADDING)
                        3630X *
                        3631X *        CALL    $TYPLN
                        3632X *        DB      N               BYTE COUNT OF FOLLOWING MESSAGE
                        3633X *        DB      'N-CHARACTER MESSAGE'
                        3634X *
                        3635X *        ENTRY   (RET) = TEXT COUNT
                        3636X *                (RET)+1 - (RET)+N = TEXT
                        3637X *        EXIT    TO (RET)+N+1
                        3638X *        USES    A,F
                        3639X *
                        3640X
                        3641X
 057.342  343           3642X $TYPLN.  XTHL                    (H,L) = COUNT ADDRESS
 057.343  176           3643X          MOV     A,M             (A) = COUNT
 057.344  043           3644X          INX     H               (H,L)  = TEXT ADDRESS
 057.345  345           3645X          PUSH    H               SAVE TEXT FWA
 057.346  315 072 030  3646X          CALL    $DADA           CALCULATE RETURN ADDRESS
 057.351  343           3647X          XTHL                    (HL) = TEXT ADDRE
 057.352  315 360 057  3648X          CALL    $TYPL.          OUTPUT LINE
 057.355  341           3649X          POP     H               (HL) = RETURN ADDRESS
 057.356  343           3650X          XTHL                    RESTORE (HL), SET RETURN ADDRESS
 057.357  311           3651X          RET
                        3652X
                        3653X **       $TYPL. - TYPE LINE.
                        3654X *
                        3655X *        ENTRY   (HL) = ADDRESS
                        3656X *                (A) = COUNT
```

```
                              3657X *         EXIT     NONE
                              3658X *         USES     A,F,H,L
                              3659X
       057.360                3660X $TYPL.    EQU      *
       057.360  247           3661X           ANA      A
       057.361  310           3662X           RZ                         NOTHING TO TYPE
       057.362  365           3663X           PUSH     PSW               SAVE COUNT
       057.363  176           3664X           MOV      A,M               (A) = CHARACTER
       057.364  043           3665X           INX      H
       057.365  247           3666X           ANA      A
       000.001                3667X           IF       $CMP$             IF HAVE COMPRESSED SPACES
                              3668X           JM       TPL2              IS COMPRESSED SPACE
                              3669X           ENDIF
       057.366  314 342 056   3670X           CZ       $CRLF
       057.371  315 337 057   3671X           CALL     $TYPC.            TYPE CHARACTER
       057.374  361           3672X TPL1      POP      PSW
       057.375  075           3673X           DCR      A
       057.376  302 360 057   3674X           JNZ      $TYPL.
       060.001  311           3675X           RET
       000.001                3676X           IF       $CMP$             IF COMPRESSED TEXT
                              3677X
                              3678X *         HAVE COMPRESSED SPACE.
                              3679X
                              3680X TPL2      DCR      A
                              3681X           CP       $TYPCH            TYPE 00 IF CHARACTER WAS 200Q
                              3682X           DB       0
                              3683X           ANA      A                 SET CODES
                              3684X TPL3      JP       TPL1              ALL EXPANDED
                              3685X           PUSH     PSW               SAVE COUNT
                              3686X           CALL     $TYPCH
                              3687X           DB       ' '
                              3688X           POP      PSW
                              3689X           DCR      A
                              3690X           JMP      TPL3
                              3691X           ENDIF
       060.002                3692            XTEXT    TYPT2



                              3694X **        $TYPTX - TYPE TEXT.
                              3695X *
                              3696X *         $TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
                              3697X *
                              3698X *         IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED,
                              3699X *         A BYTE WITH THE 200Q BIT SET IS THE LAST BYTE IN THE MESSAGE.
                              3700X *
                              3701X *         ENTRY    (RET) = TEXT
                              3702X *         EXIT     TO (RET+LENGTH)
                              3703X *         USES     A,F
                              3704X
                              3705X
       031.136                3706X $TYPTX    EQU      31136A            IN H17 ROM
                              3707X
       031.144                3708X $TYPTX.   EQU      31144A            IN H17 ROM
       060.002                3709            XTEXT    UDD
```

```
                                3711X **       $UDD - UNPACK DECIMAL DIGITS.
                                3712X *
                                3713X *        UDD CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
                                3714X *        DECIMAL DIGITS. THE RESULT IS ZERO FILLED.
                                3715X *
                                3716X *        ENTRY    (B,C) = ADDRESS VALUE
                                3717X *                 (A) = DIGIT COUNT
                                3718X *                 (H,L) = MEMORY ADDRESS
                                3719X *        EXIT     (HL) = (HL) + (A)
                                3720X *        USES     ALL
                                3721X
                                3722X
     031.157                    3723X $UDD     EQU      31157A          IN H17 ROM
     060.002                    3724           XTEXT    UDDN


                                3726X **       $UDDN - UNPACK DECIMAL DIGITS.
                                3727X *
                                3728X *        UDDN CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
                                3729X *        DECIMAL DIGITS. THE RESULT IS NULL FILLED TO THE LEFT.
                                3730X *
                                3731X *        ENTRY    (B,C) = ADDRESS VALUE
                                3732X *                 (A) = DIGIT COUNT
                                3733X *                 (H,L) = MEMORY ADDRESS
                                3734X *        EXIT     (HL) = (HL) + (A)
                                3735X *        USES     ALL
                                3736X
                                3737X
     060.002                    3738X $UDDN    EQU      *
     060.002  315 072 030       3739X          CALL     $DADA
     060.005  345               3740X          PUSH     H               SAVE FINAL (H,L) VALUE
                                3741X
     060.006  365               3742X UDDN1    PUSH     PSW
     060.007  345               3743X          PUSH     H
     060.010  021 012 000       3744X          LXI      D,10
     060.013  315 106 030       3745X          CALL     $DU66           (H,L) = VALUE/10
     060.016  104               3746X          MOV      B,H
     060.017  115               3747X          MOV      C,L             (BC) = QUOTIENT
     060.020  341               3748X          POP      H
     060.021  076 060           3749X          MVI      A,'0'
     060.023  203               3750X          ADD      E               ADD REMAINDER
     060.024  053               3751X          DCX      H
     060.025  167               3752X          MOV      M,A             STORE DIGIT
     060.026  170               3753X          MOV      A,B
     060.027  261               3754X          ORA      C
     060.030  312 042 060       3755X          JZ       UDDN2           ALL ZEROS
     060.033  361               3756X          POP      PSW
     060.034  075               3757X          DCR      A
     060.035  302 006 060       3758X          JNZ      UDDN1           IF MORE TO GO
                                3759X
                                3760X *        ALL DONE. EXIT
                                3761X
     060.040  341               3762X UDDN1.5  POP      H               RESTORE H
     060.041  311               3763X          RET                         RETURN
```

```
                        3764X
                        3765X *       DIGITS LEADING THIS ONE ARE ZERO. STORE NULLS INSTEAD.
                        3766X
    060.042  361        3767X UDDN2   POP     PSW
    060.043  075        3768X UDDN3   DCR     A
    060.044  312 040 060 3769X        JE      UDDN1.5         ALL DONE
    060.047  053        3770X         DCX     H
    060.050  066 000    3771X         MVI     M,0
    060.052  303 043 060 3772X        JMP     UDDN3
    060.055             3773          XTEXT   XCHGBC


                        3775X **      XCHGBC  -  XCHG BC
                        3776X *
                        3777X *       EXCHANGE THE 'BC' REGISTER PAIR WITH THE 'HL' REGISTER PAIR.
                        3778X *
                        3779X *       ENTRY:  BC      = ORIGINAL BC
                        3780X *               HL      = ORIGINAL HL
                        3781X *
                        3782X *       EXIT:   BC      = ORIGINAL HL
                        3783X *               HL      = ORIGINAL BC
                        3784X *
                        3785X *       USES:   BC,HL
                        3786X *
                        3787X
    060.055  365        3788X XCHGBC  PUSH    PSW
    060.056  170        3789X         MOV     A,B
    060.057  104        3790X         MOV     B,H
    060.060  147        3791X         MOV     H,A
    060.061  171        3792X         MOV     A,C
    060.062  115        3793X         MOV     C,L
    060.063  157        3794X         MOV     L,A
    060.064  361        3795X         POP     PSW
    060.065  311        3796X         RET
    060.066             3797          XTEXT   ZERO


                        3799X **      $ZERO - ZERO MEMORY
                        3800X *
                        3801X *       $ZERO ZEROS A BLOCK OF MEMORY.
                        3802X *
                        3803X *       ENTRY   (HL) = ADDRESS
                        3804X *               (B) = COUNT
                        3805X *       EXIT    (A) = 0
                        3806X *       USES    A,B,F,H,L
                        3807X
                        3808X
    031.212             3809X $ZERO   EQU     31212A          IN H17 ROM
```

```
                         3812
                         3813
                         3814 **      FDN - FILE DESCRIPTOR NODES.
                         3815 *
                         3816 *       THESE NODES ARE USED TO KEEP TRACK OF FILES WHICH ARE BEING
                         3817 *       HELD IN MEMORY WHILE TRANSFERING.
                         3818
060.066                  3819 FDN     DS      0               START OF TYPICAL NODE
                         3820
000.000                  3821 FDN.LNK EQU     *-FDN           LINK TO NEXT NODE IN CHAIN
060.066                  3822         DS      2               Full word link                  /80.07.sc/
                         3823
000.002                  3824 FDN.STA EQU     *-FDN           STATUS BYTE
000.020                  3825 ST.CNT  EQU     DIF.CNT         IS CONTIGUOUS
000.002                  3826 ST.OPR  EQU     00000010B       IS BEING READ
000.001                  3827 ST.OPW  EQU     00000001B       OPEN FOR WRITE
060.070                  3828         DS      1
                         3829
000.003                  3830 FDN.FLG EQU     *-FDN           FLAG BITS SET ON SOURCE FILE
060.071                  3831         DS      1
                         3832
000.004                  3833 FDN.SIZ EQU     *-FDN           TOTAL SIZE OF FILE (IF ST.CNT SET)
060.072                  3834         DS      2                 In Sectors                    /80.07.sc/
                         3835
000.006                  3836 FDN.AMR EQU     *-FDN           AMOUNT ALREADY READ
060.074                  3837         DS      2                IN SECTORS
                         3838
000.010                  3839 FDN.AMW EQU     *-FDN           AMOUNT ALREADY WRITTEN
060.076                  3840         DS      2                IN SECTORS
                         3841
000.012                  3842 FDN.ADR EQU     *-FDN           ADDRESS IN BUFFER
060.100                  3843         DS      1                ADDRESS/256 (MUST BE EVEN PAGE)
                         3844
000.013                  3845 FDN.AIM EQU     *-FDN           AMOUNT IN MEMORY
060.101                  3846         DS      1                 IN SECTORS
                         3847
000.014                  3848 FDNELEN EQU     *-FDN           ENTRY  LENGTH
                         3849
060.066                  3850         ORG     FDN             ORG BACK OVER DEFINITION AREA


                         3852 **      TABLE. A LINK OF 0 IS A NULL LINK.                      /80.07.GC/
                         3853 *
                         3854
060.066  000 000         3855 FDNFREE DW      0               HEAD of FREE List               /80.07.sc/
060.070  000 000         3856 FDNHED  DW      0               HEAD of FILE List               /80.07.sc/
                         3857
060.072                  3858 FDN.1   DS      0
060.072                  3859         DS      FDNCNT*FDNELEN  Reserve space for nodes         /80.07.sc/
                         3860
                         3861
060.232  000             3862 OBUFLIM DB      0               BUFFER LIMIT/256
060.233  000             3863 OBUFPTR DB      0               NEXT FREE PAGE IN BUFFER/256
                         3864
```

```
                   3865
```

```
060.234  242 065      BUFPTR  DW      BUFF              POINTER TO START  OF BUFFER
060.236  000 000      BUFSIZ  DW      0                 BUFFER LENGTH
060.240  000          CMDLIN  DB      0                 != 0  =>  Command Line specified /80.07.sc/
060.241  000          DRIVES2 DB      0                 != 0  =>  Two-Drive system      /80.07.sc/
060.242  000 000      LINEP   DW      0                 Line Pointer                    /80.07.sc/
060.244  000          MINIMUM DB      0                 != 0  =>  Minimal Syssen        /80.07.sc/
060.245  000          QUERY   DB      0                 != 0  =>  Query extra files     /80.07.sc/
060.246  000          SRCSPG  DB      0                 Source volume Sectors per Group /80.07.sc/
060.247  000          VOLFLAG DB      0                 == 0  =>  System Vol. Mounted   /80.07.sc/
                      *                                 == 377Q => Dest. Vol. Mounted   /80.07.sc/

060.250  130 130 130  DIRNAM  DB      'XXX:DIRECT.SYS',0        DIRECTORY FILE NAME


060.267               DEST    EQU     *

000.000                       ERRNZ   *-DEFAULT-DEST
060.267  123 131 061          DB      'SY1',0,0,0

000.000                       ERRNZ   *-DEVTAB-DEST
060.275  000 000              DW      0

000.000                       ERRNZ   *-DRIVER-DEST
060.277  303 000 000          JMP     0

000.000                       ERRNZ   *-UNIT-DEST
060.302  001                  DB      1

000.000                       ERRNZ   *-DEVICE-DEST
060.303  123 131 061          DB      'SY1:',0

000.000                       ERRNZ   *-DVCLEN-DEST


060.310               SOURCE  EQU     *

000.000                       ERRNZ   *-DEFAULT-SOURCE
060.310  123 131 060          DB      'SY0',0,0,0

000.000                       ERRNZ   *-DEVTAB-SOURCE
060.316  000 000              DW      0

000.000                       ERRNZ   *-DRIVER-SOURCE
060.320  303 000 000          JMP     0

000.000                       ERRNZ   *-UNIT-SOURCE
060.323  000                  DB      0

000.000                       ERRNZ   *-DEVICE-SOURCE
060.324  123 131 060          DB      'SY0:',0

000.000                       ERRNZ   *-DVCLEN-SOURCE
```

```
                              3923  **       FILE BLOCKS
                              3924
   060.331                    3925  DESTFB  DS      0                DUMY BUFFER
   060.331  310               3926          DB      200              ILLEGAL CHANNEL NUMBER
   060.332  000               3927          DB      0                FLAGS
   060.333  000 000           3928          DW      0
   060.335  000 000           3929          DW      0
   060.337  000 000           3930          DW      0
   060.341  000 000           3931          DW      0                END OF BLOCK
   060.343                    3932          DS      FB.NAML          NAME AREA


   060.364  000 000           3934  NAMTLEN DW      0                NAME TABLE POINTER
   060.366  000 000           3935  NAMTMAX DW      0                MAXIMUM SIZE OF NAME TABLE
   060.370  000 000           3936  NAMTPTR DW      0                POINTER TO ACTIVE ELEMENT IN NAMTAB
                              3937
   060.372  052 056 052       3938  OFILES  DB      '*.*='
   060.376                    3939  OFILESA DS      80               Optional File List          /80.07.sc/
   000.124                    3940  OFILESL EQU     *-OFILES                                     /80.07.GC/
   061.116  000               3941          DB      0
                              3942
                              3943
   061.117                    3944  PATCH   DS      64
```

```
                         **      PCL     - Parse Command Line                              /80.07.gc/
                         *
                         *       PCL parses the command line.  Valid switches are:
                         *
                         *               M[INIMUM]       Minimal Sysgen
                         *               Q[UERY]         Query user for optional files
                         *               File list specifying files other than internal default
                         *
                         *       ENTRY:  Command line pushed on the stack, followed by return address
                         *
                         *       EXIT:   Command line parsed
                         *
                         *       USES:   ALL
                         *

061.217  041 002 000     PCL     LXI     H,2
061.222  071                     DAD     SP              PRS must directly call this routine
061.223  021 200 042             LXI     D,STACK
061.226  315 216 030             CALL    $CDEHL
061.231  310                     RZ                      Nothing on Stack other than RET address

061.232  345                     PUSH    H
061.233  021 221 065             LXI     D,SWTFWA
061.236  315 321 064             CALL    $DRS
061.241  341                     POP     H               Restore pointer
061.242  332 123 052             JC      ERROR           Bad error

061.245  315 250 057             CALL    $SOB
061.250  176                     MOV     A,M
061.251  247                     ANA     A
061.252  310                     RZ

                         *       Process File List

061.253  062 240 060             STA     CMDLIN          Flag Command Line specified

061.256  072 244 060             LDA     MINIMUM
061.261  247                     ANA     A
061.262  076 204                 MVI     A,PEC.CS
061.264  302 123 052             JNZ     ERROR           /min and command list together are illegal

061.267  016 124                 MVI     C,OFILESL
061.271  021 376 060             LXI     D,OFILESA
061.274  176             PCL1    MOV     A,M             Copy the file list to the optional table
061.275  022                     STAX    D
061.276  023                     INX     D
061.277  043                     INX     H
061.300  247                     ANA     A
061.301  310                     RZ                      End of List

061.302  015                     DCR     C
061.303  302 274 061             JNZ     PCL1

061.306  076 207                 MVI     A,PEC.CO
061.310  303 123 052             JMP     ERROR           COMMAND OVERFLOW
```

```
                               4004  **      PDN     - Parse Device Name                                    /80.07.sc/
                               4005  *
                               4006  *       PDN parses a device name, assuming a default of SY0:.
                               4007  *
                               4008  *       ENTRY:  NONE
                               4009  *
                               4010  *       EXIT:   To EXIT if CTL-D was hit
                               4011  *               DEST and SOURCE initialized
                               4012  *               DRIVES2 = 0 iff 1-drive sysgen
                               4013  *                       = 1 iff 2-drive sysgen
                               4014  *
                               4015  *       USES:   ALL
                               4016  *
                               4017
    061.313  315 015 064  4018  PDN     CALL    $CCO
    061.316  315 136 031  4019          CALL    $TYPTX
    061.321  012          4020          DB      NL
    061.322  104 145 163  4021          DB      'Destination Device<SY0:>?',' '+200Q
                          4022
    061.354  315 120 065  4023          CALL    $ITL.
    061.357  332 376 042  4024          JC      EXIT                Exit if CTL-D is hit
                          4025
    061.362  041 234 065  4026          LXI     H,ITLA              HL = Address of device specification
    061.365  001 153 062  4027  PDN.    LXI     B,PDNA              Decode Area
    061.370  021 156 062  4028          LXI     D,PDNC              Default Device
    061.373  315 032 064  4029          CALL    DDS
    061.376  322 045 062  4030          JNC     PDN1                No Problems with Device Specification
                          4031
                          4032  *       Illegal Device Specification
                          4033
    062.001  315 136 031  4034          CALL    $TYPTX
    062.004  012          4035          DB      NL
    062.005  111 154 154  4036          DB      'Illegal Device Specification',ENL
                          4037
    062.042  303 313 061  4038          JMP     PDN
                          4039
                          4040  *       Set up Flags and Pointers
                          4041
    062.045  257          4042  PDN1    XRA     A
    062.046  062 241 060  4043          STA     DRIVES2             Default to 1-Drive Sysgen
    062.051  016 003      4044          MVI     C,PDNAL
    062.053  021 153 062  4045          LXI     D,PDNA
    062.056  041 164 062  4046          LXI     H,PDND
    062.061  315 060 030  4047          CALL    $COMP               Compare Destination to Source
    062.064  312 123 062  4048          JZ      PDN2                Source == Destination
                          4049
                          4050  *       Source != Destination, Initialize Source
                          4051
    062.067  076 001      4052          MVI     A,1
    062.071  062 241 060  4053          STA     DRIVES2             Flag 2-drive sysgen
                          4054
    062.074  052 153 062  4055          LHLD    PDNA
    062.077  042 267 060  4056          SHLD    DEST+DEFAULT        Initialize Name
    062.102  042 303 060  4057          SHLD    DEST+DEVICE
    062.105  072 155 062  4058          LDA     PDNB
    062.110  062 302 060  4059          STA     DEST+UNIT           Initialize Binary Unit
```

```
   062.113  306 060      4060            ADI     '0'              Convert to ASCII
   062.115  062 271 060  4061            STA     DEST+DEFAULT+IOC.UNI-IOC.DEV
   062.120  062 305 060  4062            STA     DEST+DEVICE+IOC.UNI-IOC.DEV
                         4063
                         4064  *         Initialize Source
                         4065
   062.123  052 164 062  4066  PDN2      LHLD    PDND
   062.126  042 310 060  4067            SHLD    SOURCE+DEFAULT   Initialize Device Name
   062.131  042 324 060  4068            SHLD    SOURCE+DEVICE
   062.134  072 166 062  4069            LDA     PDND+IOC.UNI-IOC.DEV
   062.137  062 323 060  4070            STA     SOURCE+UNIT
   062.142  306 060      4071            ADI     '0'              Convert to ASCII
   062.144  062 312 060  4072            STA     SOURCE+DEFAULT+IOC.UNI-IOC.DEV
   062.147  062 326 060  4073            STA     SOURCE+DEVICE+IOC.UNI-IOC.DEV
                         4074
   062.152  311          4075            RET
                         4076
   062.153  170 170      4077  PDNA      DB      'xx'
   062.155  000          4078  PDNB      DB      0
   000.003               4079  PDNAL     EQU     *-PDNA
                         4080
   000.000               4081            ERRNZ   IOC.UNI-IOC.DEV-2        2-Byte Device
   000.000               4082            ERRNZ   IOC.DIR-IOC.UNI-1        1-Byte Unit
                         4083
   062.156  123 131 060  4084  PDNC      DB      'SY0',0,0,0              Default Destination Device
                         4085
   062.164  123 131      4086  PDND      DB      'SY'                     Source Device Specification
   062.166  000          4087            DB      0


                         4089  ***       PRS - PRESET PIP PROGRAM.                                       /80.07.GC/
                         4090  *
                         4091  *         PRS IS CALLED TO PERFORM ONE-TIME-ONLY PRESETTING OF
                         4092  *         THE PROGRAM ENVIRONMENT.
                         4093  *
                         4094  *         THE CODE IS OVERLAID BY BUFFERS AND WORK AREAS WHEN PIP IS RUNNING.
                         4095  *
                         4096  *         ENTRY   NONE
                         4097  *         EXIT    NONE
                         4098  *         USES    ALL
                         4099
                         4100
   062.167               4101  ENTRY     EQU     *                INITIAL ENTRY POINT
                         4102
   062.167  377 011      4103  PRS       SCALL   .VERS
   062.171  332 371 063  4104            JC      PRSERR           NO .VERS SYSTEM CALL
   062.174  376 040      4105            CPI     VERS
   062.176  302 371 063  4106            JNZ     PRSERR
                         4107
   062.201  076 377      4108            MVI     A,377Q
   062.203  377 046      4109            SCALL   .CLOSE           CLOSE THE CHANNEL THAT WE CAME IN ON
   062.205  041 242 065  4110            LXI     H,RMEML          (HL) = RUN-TIME HIGH MEMORY
   062.210  377 052      4111            SCALL   .SETTP           SET HI MEMORY
   062.212  332 123 052  4112            JC      ERROR
```

```

062.215  257                     XRA     A
062.216  062 240 060             STA     CMDLIN          Initialize to NO command line
062.221  062 244 060             STA     MINIMUM         Initialize to Maximal Sysgen
062.224  062 245 060             STA     QUERY           Initialize to no query

062.227  041 002 043             LXI     H,CCHIT
062.232  076 003                 MVI     A,CTLC
062.234  377 041                 SCALL   .CTLC           SET CTL-C PROCESSING

062.236  315 136 031             CALL    $TYPTX
062.241  012 011 011             DB      NL,TAB,TAB,TAB,'    ','SYSGEN'
062.256  012 011 011             DB      NL,TAB,TAB,TAB,'Version:  ',VERS/16+'0','.',VERS&0FH+'0'
062.277  012 011 011             DB      NL,TAB,TAB,'       ','Issue:  #50.06.00'
062.331  012                     DB      NL
062.332  212                     DB      ENL

062.333  315 217 061             CALL    PCL             Parse Command Line
062.336  315 313 061             CALL    PDN             Parse Device Name

062.341  041 324 060             LXI     H,SOURCE+DEVICE
062.344  377 062                 SCALL   .LOADD          Load Source Device Driver
062.346  332 123 052             JC      ERROR
062.351  042 316 060             SHLD    SOURCE+DEVTAB   Save Table Address
062.354  021 003 000             LXI     D,DEV.JMP
062.357  031                     DAD     D               HL = driver jump address
062.360  042 321 060             SHLD    SOURCE+DRIVER+1

062.363  072 241 060             LDA     DRIVES2
062.366  247                     ANA     A
062.367  312 014 063             JZ      PRS1            1-Drive Sysgen

062.372  041 303 060             LXI     H,DEST+DEVICE
062.375  377 062                 SCALL   .LOADD          Load Device Driver for Destination
062.377  332 123 052             JC      ERROR
063.002  042 275 060             SHLD    DEST+DEVTAB     Save Device Table Address
063.005  021 003 000             LXI     D,DEV.JMP
063.010  031                     DAD     D               HL = Device Jump Address
063.011  042 300 060             SHLD    DEST+DRIVER+1

063.014  315 152 064     PRS1    CALL    $DOS            DISMOUNT OPERATING SYSTEM
063.017  332 123 052             JC      ERROR

                         *       Mount the Source Diskette

063.022  315 214 057             CALL    $MOVEL          Stuff Source specification
000.000                          ERRNZ   IOC.DIR-IOC.DEV-3
063.025  003 000                 DW      3
063.027  324 060                 DW      SOURCE+DEVICE
063.031  075 063                 DW      PRSA

063.033  315 136 031             CALL    $TYPTX
063.036  012                     DB      NL
063.037  111 156 163             DB      'Insert the Source Diskette in '
063.075  170 170 156     PRSA    DB      'xxn:.  Hit Return when Ready!',' '+200Q

```

```
063.133  315 237 057  4169  PRS2    CALL    $RCHAR
063.136  376 012      4170          CPI     NL
063.140  302 133 063  4171          JNZ     PRS2                    Wait for Newline
                      4172
063.143  041 324 060  4173          LXI     H,SOURCE+DEVICE
063.146  315 254 047  4174          CALL    MND.
                      4175
063.151  052 316 060  4176          LHLD    SOURCE+DEVTAB
063.154  021 011 000  4177          LXI     D,DEV.UNT
063.157  031          4178          DAD     D                       HL = Address of Unit Pointer
063.160  257          4179          XRA     A                       Use unit 0
063.161  315 027 041  4180          CALL    S.GUP                   HL = Base address of SY0: Unit Table
063.164  315 100 057  4181          CALL    $INDLB                  A  = Sectors Per Group
063.167  001 000      4182          DW      UNT.SPG
063.171  062 246 060  4183          STA     SRCSPG                  Save Source SPG for later
                      4184
063.174  072 241 060  4185          LDA     DRIVES2
063.177  247          4186          ANA     A
063.200  302 217 063  4187          JNZ     PRS3                    2-Drive Sysgen
                      4188
                      4189  *       1-Drive Sysgen
                      4190
063.203  315 214 057  4191          CALL    $MOVEL
063.206  021 000      4192          DW      DVCLEN                  Count
063.210  310 060      4193          DW      SOURCE                  From
063.212  267 060      4194          DW      DEST                    To
                      4195
063.214  303 353 063  4196          JMP     PRS5
                      4197
                      4198  *       2-Drive Sysgen, Mount Destination
                      4199
063.217  315 214 057  4200  PRS3    CALL    $MOVEL                  Stuff Device Specification into message
000.000               4201          ERRNZ   IOC.DIR-IOC.DEV-3
063.222  003 000      4202          DW      3
063.224  303 060      4203          DW      DEST+DEVICE
063.226  277 063      4204          DW      PRSB
                      4205
063.230  315 136 031  4206          CALL    $TYPTX
063.233  012          4207          DB      NL
063.234  111 156 163  4208          DB      'Insert the Destination Diskette in '
063.277  170 170 156  4209  PRSB    DB      'xxn:.  Hit Return when Ready:',' '+200Q
                      4210
063.335  315 237 057  4211  PRS4    CALL    $RCHAR
063.340  376 012      4212          CPI     NL
063.342  302 335 063  4213          JNZ     PRS4                    Wait for NL
                      4214
063.345  041 303 060  4215          LXI     H,DEST+DEVICE
063.350  315 254 047  4216          CALL    MND.                    Mount a new Diskette
                      4217
000.000               4218          ERRNZ   *-PRS5
                      4219
063.353  076 000      4220  PRS5    MVI     A,I.CSLMD
063.355  006 001      4221          MVI     B,CSL.CHR
063.357  016 001      4222          MVI     C,CSL.CHR               Character Mode
063.361  377 006      4223          SCALL   .CONSL
063.363  332 123 052  4224          JC      ERROR
```

```

063.366  303 200 042          JMP     START

063.371  076 050      PRSERR  MVI     A,EC.NCV        NOT CORRECT VERSION
063.373  067                  STC
063.374  303 123 052          JMP     ERROR
```

```
                               4234  **       P.MIN    - Process Minimum
                               4235  *
                               4236  *        P.MIN processes the minimum switch
                               4237  *
                               4238  *        This switch will transfer the minimal set
                               4239  *        of HDOS working files.
                               4240  *
                               4241
       063.377  076 001        4242  P.MIN    MVI      A,1
       064.001  062 244 060    4243           STA      MINIMUM
       064.004  247            4244           ANA      A                 Clear 'C'
       064.005  311            4245           RET



                               4247  **       P.QUE    - Process Query
                               4248  *
                               4249  *        P.QUE processes the query switch
                               4250  *
                               4251  *        This switch will query the user as to whether
                               4252  *        each non-essential file is to be transfered.
                               4253  *
                               4254
       064.006  076 001        4255  P.QUE    MVI      A,1
       064.010  062 245 060    4256           STA      QUERY
       064.013  247            4257           ANA      A                 Clear 'C'
       064.014  311            4258           RET
```

```
064.015               4261          XTEXT  CCO


                      4263X **      $CCO - CLEAR CONTROL-O
                      4264X *
                      4265X *       $CCO IS CALLED TO CLEAR THE EFFECT OF THE CTL-O CHARACTER.
                      4266X *
                      4267X *       ENTRY  NONE
                      4268X *       EXIT   NONE
                      4269X *       USES   NONE
                      4270X
                      4271X
064.015  315 054 031  4272X $CCO    CALL   $SAVALL         SAVE REGISTERS
064.020  076 004      4273X         MVI    A,I.CONFL
064.022  001 001 000  4274X         LXI    B,CO.FLG        CLEAR CO.FLG
064.025  377 006      4275X         DB     SYSCALL,.CONSL
064.027  303 047 031  4276X         JMP    $RSTALL         RESTORE REGISTERS AND RETURN
064.032               4277          XTEXT  DDS


                      4279X **      DDS    -  Decode Device Specification                  /80.05.sc/
                      4280X *
                      4281X *       DDS decodes the device specification, returning a two character
                      4282X *       device name, and one byte unit number.
                      4283X *
                      4284X *
                      4285X *       ENTRY:  BC      = Address of destination fields
                      4286X *               DE      = Address of default
                      4287X *               HL      = Address of string specifier
                      4288X *
                      4289X *       EXIT:   PSW     = 'C' SET   if ERROR
                      4290X *                         'C' CLEAR if NO ERROR
                      4291X *
                      4292X *       USES:   ALL
                      4293X *
                      4294X
064.032               4295X DDS     EQU    *
                      4296X
                      4297X *       Initialize the fields to the defaults
                      4298X
064.032  305          4299X         PUSH   B
064.033  315 142 064  4300X         CALL   DDS3
064.036  315 142 064  4301X         CALL   DDS3
064.041  032          4302X         LDAX   D
064.042  326 060      4303X         SUI    '0'
064.044  002          4304X         STAX   B
064.045  301          4305X         POP    B
                      4306X
064.046  176          4307X         MOV    A,M
064.047  247          4308X         ANA    A
064.050  310          4309X         RZ                     took the default
                      4310X
```

```
                                4311X *       Check the supplied name
                                4312X
      064.051  315 250 057     4313X         CALL      $SOB                       skip the whitespace
      064.054  315 123 064     4314X         CALL      DDS2
      064.057  330             4315X         RC                                   Not alpha
      064.060  315 123 064     4316X         CALL      DDS2
      064.063  330             4317X         RC                                   Not alpha
                                4318X
      064.064  176             4319X         MOV       A,M
      064.065  376 072         4320X         CPI       ':'
      064.067  076 000         4321X         MVI       A,0                        assume unit 0
      064.071  312 105 064     4322X         JZ        DDS1                       default to unit 0
                                4323X
                                4324X *       Check for a valid digit
                                4325X
      064.074  176             4326X         MOV       A,M
      064.075  326 060         4327X         SUI       '0'
      064.077  330             4328X         RC                                   Not digit
      064.100  376 010         4329X         CPI       7+1
      064.102  077             4330X         CMC
      064.103  330             4331X         RC                                   digit too large
      064.104  043             4332X         INX       H
                                4333X
      064.105  002             4334X DDS1    STAX      B
      064.106  003             4335X         INX       B
      064.107  176             4336X         MOV       A,M
      064.110  043             4337X         INX       H
      064.111  376 072         4338X         CPI       ':'
      064.113  067             4339X         STC
      064.114  300             4340X         RNZ                                  requires ':'
                                4341X
      064.115  176             4342X         MOV       A,M
      064.116  247             4343X         ANA       A
      064.117  067             4344X         STC
      064.120  300             4345X         RNZ                                  require 'NULL'
                                4346X
      064.121  247             4347X         ANA       A                          Clear ERROR flag
      064.122  311             4348X         RET
                                4349X
      064.123  176             4350X DDS2    MOV       A,M
      064.124  043             4351X         INX       H
      064.125  315 203 057     4352X         CALL      $MCU
      064.130  376 101         4353X         CPI       'A'
      064.132  330             4354X         RC                                   Not alpha
                                4355X
      064.133  376 133         4356X         CPI       'Z'+1
      064.135  077             4357X         CMC
      064.136  330             4358X         RC                                   Not alpha
                                4359X
      064.137  002             4360X         STAX      B
      064.140  003             4361X         INX       B                          replace the default char
      064.141  311             4362X         RET
                                4363X
      064.142  032             4364X DDS3    LDAX      D
      064.143  023             4365X         INX       D
      064.144  315 203 057     4366X         CALL      $MCU                       Map to upper case
```

```
064.147  002           4367X          STAX    B
064.150  003           4368X          INX     B
064.151  311           4369X          RET
000.000                4370X          ERRNZ   IOC.UNI-IOC.DEV-2        2 byte device
000.000                4371X          ERRNZ   IOC.DIR-IOC.UNI-1        1 byte unit
064.152                4372           XTEXT   DOS


                       4374X **       $DOS - DISMOUNT OPERATING SYSTEM.
                       4375X *
                       4376X *        $DOS disounts all units of all directory devices       /80.04.sc/
                       4377X *
                       4378X *        THE USER IS MESSAGED ABOUT THE DISKS, AND THE OPERATING
                       4379X *        SYSTEM IS NOTIFIED.
                       4380X *
                       4381X *
                       4382X *        ENTRY   NONE
                       4383X *
                       4384X *        EXIT    (PSW)   = 'C' CLEAR IF NO ERROR
                       4385X *                          'C' SET   IF    ERROR
                       4386X *                           (A)  = ERROR CODE
                       4387X *
                       4388X *        USES    ALL
                       4389X *
                       4390X
064.152  315 136 031   4391X $DOS     CALL    $TYPTX
064.155  012 007 104   4392X          DB      NL,BELL,'Dismounting All Disks!',NL,ENL
                       4393X
064.207  315 304 064   4394X          CALL    $DOS.
064.212  330           4395X          RC
                       4396X
064.213  315 136 031   4397X          CALL    $TYPTX
064.216  012 122 145   4398X          DB      NL,'Remove the Disk(s). Hit RETURN when ready:',' '+200Q
                       4399X
064.272  315 237 057   4400X DOS1     CALL    $RCHAR           READ CHARACTER
064.275  376 012       4401X          CPI     NL
064.277  302 272 064   4402X          JNE     DOS1
                       4403X
064.302  247           4404X          ANA     A                CLEAR CARRY
064.303  311           4405X          RET


064.304  076 000       4407X $DOS.    MVI     A,OVL0
064.306  377 010       4408X          SCALL   .LOADO
064.310  330           4409X          RC
                       4410X
064.311  076 001       4411X          MVI     A,OVL1
064.313  377 010       4412X          SCALL   .LOADO
064.315  330           4413X          RC
                       4414X
064.316  377 206       4415X          SCALL   .DAD             Dismount all Disks                /80.09.sc/
064.320  311           4416X          RET
```

```
      064.321              4417          XTEXT   DRS


                           4419X **      $DRS - DECODE AND REMOVE SWITCHES.
                           4420X *
                           4421X *       $DRS IS CALLED TO DECODE COMMAND SWITCHES FROM A LINE
                           4422X *       OF TEXT. SWITCHES TAKE THE FORM:
                           4423X *
                           4424X *       /XXXXX
                           4425X *
                           4426X *       AFTER A SWITCH HAS BEEN LOCATED, IT (AND THE PRECEDING '/')
                           4427X *       ARE REPLACED WITH BLANKS.
                           4428X *
                           4429X *       VALID SWITCH DESCRIPTIONS ARE ENCODED INTO A TABLE
                           4430X *       SUPPLIED BY THE CALLER, IN THE FORMAT:
                           4431X *
                           4432X *       DB      'X...X'         REQUIRED SWITCH CHARACTERS
                           4433X *       DB      'C'+200Q,...,'C'+200Q   OPTIONAL CHARACTERS
                           4434X *       DB      200Q            END OF CHARACTERS
                           4435X *       DW      ADDR            PROCESSOR ADDRESS (CALLED WHEN SWITCH DETECTED)
                           4436X *
                           4437X *       DB      'Y...Y'         NEXT SWITCH
                           4438X *       .        .
                           4439X *       .        .
                           4440X *       .        .
                           4441X *
                           4442X *       DB      0               FLAGS END OF TABLE
                           4443X *
                           4444X *       SWITCHES MUST BE FOLLOWED BY A ':', A '/' (ANOTHER SWITCH)
                           4445X *       A ',', OR A 00 BYTE.
                           4446X *
                           4447X *       UPON DETECTION OF A VALID SWITCH, $DRS CALLS THE USER PROCESS
                           4448X *       ROUTINE. UPON ENTRY,
                           4449X *       (HL) = ADDRESS OF THE FIRST BYTE FOLLOWING THE SWTICH
                           4450X *       'Z' CLEAR IF CHARACTER = '/', ',', OR 00
                           4451X *       'Z' SET IF CHARACTER = ':'
                           4452X *
                           4453X *       THE USER ROUTINE CAN DECODE SWITCH SUB-OPTIONS, IF DESIRED.
                           4454X *       THE USER ROUTINE MAY USE ALL REGISTERS.
                           4455X *
                           4456X *       ENTRY   (DE) = SWITCH TABLE FWA
                           4457X *               (HL) = LINE FWA
                           4458X *       EXIT    'C' CLEAR IF OK
                           4459X *               'C' SET IF ERROR
                           4460X *                (HL) = ADDRESS OF START OF BAD SWITCH
                           4461X *                (A) = ERROR CODE
                           4462X *       USES    ALL
                           4463X
                           4464X
      064.321              4465X $DRS    EQU     *
                           4466X
                           4467X *       LOOK FOR SWITCHES
                           4468X
      064.321  176         4469X $DRS1   MOV     A,M
```

```
064.322  247          4470X          ANA    A
064.323  310          4471X          RZ                     END OF LINE
064.324  043          4472X          INX    H
064.325  376 057      4473X          CPI    '/'
064.327  302 321 064  4474X          JNE    $DRS1           NOT A SWITCH
064.332  042 116 065  4475X          SHLD   $DRSB           ($DRSB) = SWITCH FWA (AFTER '/')
                      4476X
                      4477X *        GOT A SWITCH. LOOK FOR A MATCH IN THE CALLER'S TABLE
                      4478X
064.335  325          4479X          PUSH   D               SAVE TABLE FWA
064.336  052 116 065  4480X $DRS2    LHLD   $DRSB           (HL) = SWITCH FWA
064.341  032          4481X $DRS3    LDAX   D               (A) = TABLE ENTRY
064.342  346 177      4482X          ANI    177Q
064.344  312 014 065  4483X          JZ     $DRS6           GOT A MATCH
064.347  276          4484X          CMP    M
064.350  302 360 064  4485X          JNE    $DRS4           NO MATCH
064.353  023          4486X          INX    D
064.354  043          4487X          INX    H
064.355  303 341 064  4488X          JMP    $DRS3           SEE IF MORE MATCH
                      4489X
                      4490X *        HAVE MIS-MATCH. SEE IF THE MISSING CHARACTER IS SIGNIFICANT
                      4491X
064.360  176          4492X $DRS4    MOV    A,M             (A) = LINE CHARACTER WE COULDNT MATCH
064.361  315 065 065  4493X          CALL   $DRS15          SEE IF OK TERMINATOR
064.364  302 374 064  4494X          JNE    $DRS4.5         NO MATCH ON THIS SWITCH
064.367  032          4495X          LDAX   D               (A) = NEXT CHARACTER IN SWITCH PATTERN
064.370  247          4496X          ANA    A
064.371  372 014 065  4497X          JM     $DRS6           HAVE SUFFICIENT MATCH
064.374  315 100 065  4498X $DRS4.5  CALL   $DRS20          SKIP TABLE ENTRY
064.377  032          4499X          LDAX   D
065.000  247          4500X          ANA    A
065.001  302 336 064  4501X          JNZ    $DRS2           MORE SWITCHES IN TABLE TO CHECK
                      4502X
                      4503X *        BAD SWITCH
                      4504X
065.004  321          4505X $DRS5    POP    D               RESTORE STACK
065.005  052 116 065  4506X          LHLD   $DRSB           POINT TO BAD SWITCH
065.010  067          4507X          STC
065.011  076 032      4508X          MVI    A,EC.IS         ILLEGAL SWITCH
065.013  311          4509X          RET
                      4510X
                      4511X *        HAVE SWITCH. CHECK IT'S FOLLOWING CHARACTER
                      4512X
065.014  315 250 057  4513X $DRS6    CALL   $SOB            SKIP OVER BLANKS
065.017  176          4514X          MOV    A,M
065.020  315 065 065  4515X          CALL   $DRS15          CHECK CHARACTER
065.023  302 004 065  4516X          JNE    $DRS5           IN ERROR
065.026  315 100 065  4517X          CALL   $DRS20          GET PROCESSOR ADDRESS
065.031  021 043 065  4518X          LXI    D,$DRS7
065.034  345          4519X          PUSH   H               SAVE (HL)
065.035  325          4520X          PUSH   D               SET RETURN ADDRESS FOR TABLE CODE
065.036  305          4521X          PUSH   B               SAVE PROCESSOR ADDRESS
065.037  176          4522X          MOV    A,M             (A) = NEXT CHARACTER
065.040  376 072      4523X          CPI    ':'             SET CONDITION CODES
065.042  311          4524X          RET                    CALL USER PROCESS
                      4525X
```

```
                          4526X *        USER PROCESS RETURNS HERE
                          4527X
065.043  321              4528X $DRS7    POP      D                   (DE) = LAST CHARACTER OF SWITCH+1
065.044  052 116 065      4529X          LHLD     $DRSB               (HL) = FIRST CHARACTER OF SWITCH AFTER /
065.047  053              4530X          DCX      H                   (HL) = ADDRESS OF '/'
                          4531X
                          4532X *        REPLACE SWITCH WITH BLANKS
                          4533X
065.050  066 040          4534X $DRS8    MVI      M,' '
065.052  043              4535X          INX      H
065.053  315 216 030      4536X          CALL     $CDEHL
065.056  302 050 065      4537X          JNE      $DRS8               NOT THERE YET
065.061  321              4538X          POP      D                   (DE) = SWITCH TABLE FWA
065.062  303 321 064      4539X          JMP      $DRS1               LOOK FOR MORE SWITCHES


                          4541X **       $DRS15 - CHECK FOR VALID DELIMITER CHARACTER.
                          4542X *
                          4543X *        $DRS15 CHECKS THE NEXT TEXT CHARACTER TO SEE IF IT IS
                          4544X *
                          4545X *        00, '/', ',', ':'
                          4546X *
                          4547X *        ENTRY    (A) = CHARACTER
                          4548X *        EXIT     'Z' SET IFF CHARACTER IS ONE OF THE ABOVE
                          4549X *        USES     F
                          4550X
065.065  247              4551X $DRS15   ANA      A
065.066  310              4552X          RZ                           IS 00
065.067  376 057          4553X          CPI      '/'
065.071  310              4554X          RE
065.072  376 054          4555X          CPI      ','
065.074  310              4556X          RE
065.075  376 072          4557X          CPI      ':'
065.077  311              4558X          RET


                          4560X **       $DRS20 - GET PROCESSOR ADDRESS.
                          4561X *
                          4562X *        $DRS20 IS CALLED TO GET THE PROCESSOR ADDRESS FIELD OUT OF
                          4563X *        AN ENTRY IN THE SWITCH TABLE. THE CALLER SUPPLIES A POINTER
                          4564X *        TO SOMEWHERE IN THE TEXT PART OF THE SWITCH DESCRIPTION;
                          4565X *        $DRS20 ADVANCES THE POINTER TO THE PROCESSOR ADDRESS.
                          4566X *
                          4567X *        ENTRY    (DE) = POINTER TO TEXT PART OF SWITCH ENTRY
                          4568X *        EXIT     (DE) = POINTER TO 1ST BYTE OF NEXT SWITCH TABLE ENTRY
                          4569X *                 (BC) = PROCESSOR ADDRESS FROM TABLE
                          4570X *        USES     A,F,B,C,D,E
                          4571X
                          4572X
065.100  032              4573X $DRS20   LDAX     D
065.101  023              4574X          INX      D
065.102  376 200          4575X          CPI      200Q
065.104  302 100 065      4576X          JNE      $DRS20
065.107  032              4577X          LDAX     D                   (A) = LOW BYTE OF PROCESSOR ADDRESS
065.110  117              4578X          MOV      C,A
065.111  023              4579X          INX      D
```

```
065.112  032             4580X          LDAX    D
065.113  107             4581X          MOV     B,A                (BC) = PROCESSOR ADDRESS
065.114  023             4582X          INX     D
065.115  311             4583X          RET
                         4584X
065.116  000 000         4585X $DRSB    DW      0                  POINTER TO SWITCH BEING PROCESSED
065.120                  4586           XTEXT   ITL



                         4588X **       $ITL - INPUT TEXT LINE.
                         4589X *
                         4590X *        $ITL INPUTS A LINE FROM THE TERMINAL.
                         4591X *
                         4592X *        CHARACTER ARE ACCEPTED FROM THE TERMINAL, RUBOUT AND BACKSPACE
                         4593X *        CHARACTERS ARE PROCESSED. WHEN A CARRIAGE RETURN IS ENTERED,
                         4594X *        $ITL RETURNS.
                         4595X *
                         4596X *        ENTRY   NONE
                         4597X *        EXIT    (HL) = $ITLA
                         4598X *                (A) = TEXT LENGTH
                         4599X *        USES    A,F,H,L
                         4600X
                         4601X
065.120  315 126 065     4602X $ITL.    CALL    $ITL               INPUT LINE IN UPPER CASE
065.123  303 134 065     4603X          JMP     $MLU               MAP LINE TO UPPER
                         4604X
065.126  041 234 065     4605X $ITL     LXI     H,ITLA
065.131  303 163 065     4606X          JMP     $RTL               READ TEXT LINE
065.134                  4607           XTEXT   MLU



                         4609X **       MLU - MAP LOWER CASE LINE TO UPPER CASE.
                         4610X *
                         4611X *        MLU MAPS THE LOWER CASE ALPHABETICS IN A LINE TO UPPER CASE.
                         4612X *
                         4613X *        ENTRY   (HL)  = LINE FWA
                         4614X *        EXIT    NONE
                         4615X *        USES    NONE
                         4616X
                         4617X
065.134  365             4618X $MLU     PUSH    PSW                SAVE (PSW)
065.135  345             4619X          PUSH    H                  SAVE FWA
065.136  053             4620X          DCX     H                  ANTICIPATE INX H
065.137  043             4621X $MLU1    INX     H
065.140  176             4622X          MOV     A,M                (A)= CHARACTER
065.141  315 203 057     4623X          CALL    $MCU               MAP CHAR TO UPPER
065.144  167             4624X          MOV     M,A
065.145  247             4625X          ANA     A
065.146  302 137 065     4626X          JNZ     $MLU1              MORE TO GO
065.151  341             4627X          POP     H                  RESTORE (HL)
065.152  361             4628X          POP     PSW                RESTORE (PSW)
065.153  311             4629X          RET
```

```
     065.154                     4630        XTEXT   RTL



                                 4632X **       $RTL - READ TEXT LINE.
                                 4633X *
                                 4634X *        $RTL READS A LINE FROM THE TERMINAL.
                                 4635X *
                                 4636X *        CHARACTER ARE ACCEPTED FROM THE TERMINAL, RUBOUT AND BACKSPACE
                                 4637X *        CHARACTERS ARE PROCESSED. WHEN A CARRIAGE RETURN IS ENTERED,
                                 4638X *        $RTL RETURNS.
                                 4639X *
                                 4640X *        ENTRY   (HL) = BUFFER FWA
                                 4641X *        EXIT    'C' CLEAR IF OK
                                 4642X *                 DATA IN BUFFER
                                 4643X *                 (A) = TEXT LENGTH
                                 4644X *                'C' SET IF CTL-D STRUCK
                                 4645X *        USES    A,F
                                 4646X
                                 4647X
     065.154  315 163 065        4648X $RTL.    CALL    $RTL             $RTL IN UPPER CASE
     065.157  330                4649X          RC                       CTL-D
     065.160  303 134 065        4650X          JMP     $MLU             MAP LINE TO UPPER CASE
                                 4651X
     065.163                     4652X $RTL     EQU     *
     065.163  345                4653X          PUSH    H                SAVE FWA
     065.164  315 237 057        4654X $RTL1    CALL    $RCHAR
     065.167  376 004            4655X          CPI     CTLD
     065.171  312 216 065        4656X          JE      $RTL2            CTL-D STRUCK
     065.174  167                4657X          MOV     M,A
     065.175  043                4658X          INX     H
     065.176  376 012            4659X          CPI     NL
     065.200  302 164 065        4660X          JNE     $RTL1
     065.203  053                4661X          DCX     H
     065.204  066 000            4662X          MVI     M,0
     065.206  043                4663X          INX     H
                                 4664X
                                 4665X *        ALL DONE, COMPUTE LENGTH
                                 4666X
     065.207  353                4667X          XCHG                     (DE) = LWA+1
     065.210  343                4668X          XTHL                     (HL) = FWA
     065.211  173                4669X          MOV     A,E
     065.212  225                4670X          SUB     L                (A) = LENGTH
     065.213  247                4671X          ANA     A                CLEAR CARRY
     065.214  321                4672X          POP     D                RESTORE (DE)
     065.215  311                4673X          RET
                                 4674X
                                 4675X *        CTL-D STRUCK
                                 4676X
     065.216  341                4677X $RTL2    POP     H                (HL) = FWA
     065.217  067                4678X          STC
     065.220  311                4679X          RET
```

```
                           4682  **      Overlaid Buffers and Data
                           4683  *
                           4684
  065.221                  4685  SWTFWA  DS      0                Switch Table FWA
                           4686
  065.221  115             4687          DB      'M'              /M[INIMAL]
  065.222  311 316 311     4688          DB      'I'+200Q,'N'+200Q,'I'+200Q,'M'+200Q,'A'+200Q,'L'+200Q,200Q
  065.231  377 063         4689          DW      P.MIN
                           4690
  000.001                  4691          IF      QUERYF
                           4692          DB      'Q'              /Q[UERY]
                           4693          DB      'U'+200Q,'E'+200Q,'R'+200Q,'Y'+200Q,200Q
                           4694          DW      P.QUE
                           4695          ENDIF
                           4696
  065.233  000             4697          DB      0
                           4698
                           4699
  065.234                  4700  ITLA    DS      80               Line buffer
                           4701
                           4702
  065.354                  4703  MEML    EQU     *                MEMORY LENGTH
```

```
                       4706  **      THE FOLLOWING BUFFERS AND AREAS OVERLAY THE PRS CODE.
                       4707
    062.167            4708          ORG      PRS
                       4709
                       4710
                       4711
    062.167            4712  DSTLAB  DS       256               Saved Destination Label
    063.167            4713  SRCLAB  DS       256               Saved Source Label
    064.167            4714  LABEL   DS       256               Transient Label
                       4715
    065.167            4716  MWNA    DS       FB.NAML           MWN WORK AREA
                       4717
                       4718
                       4719  **      * * NOTE * *
                       4720  *       DIRWORK  USES THE SYSTEM SCRATCH AREA, SECSCR. DIRWORK WILL NOT
                       4721  *       BE PRESERVED DURING A SYSCALL !!
                       4722
    041.121            4723  DIRWRKP EQU      S.SCR             POINTER TO THE SCRATCH AREA


                       4725  **      PIO.XXX - IMAGE OF SYSTEM AIO.XXX AREA
                       4726  *
                       4727  *       THESE CELLS MIRROR THE SYSTEM AIO.XXX AREA
                       4728
                       4729
    065.210            4730  PIO.DEV DS       2                 DEVICE CODE
    065.212            4731  PIO.UNI DS       1                 UNIT NUMBER (0-9)
                       4732
    065.213            4733  PIO.DIR DS       DIRELEN           DIRECTORY ENTRY
                       4734
                       4735
    065.242            4736  NAMTAB  DS       0                 NAME TABLE
                       4737
                       4738
    002.000            4739  BUFMINL EQU      512               MINIMUM SIZE FOR BUFFER (WHEN IN USE)
    065.242            4740  BUFF    EQU      *                 BUFFER AREA STARTS AFTER NAMTAB
                       4741
    065.242            4742  RMEML   EQU      *                 INITIAL RUNNING MEMORY LENGTH
                       4743
                       4744
                       4745
    065.242            4746          END
ASSEMBLY COMPLETE
  4746 STATEMENTS
     0 ERRORS DETECTED
  9484 BYTES FREE
```

```
$CCO    064015   4018     4272L
$CDEHL  030216    933     1182     3037E    3132     3966     4536
$CHL    030224   3049E
$CMP$   000001   3623E    3667     3676
$COMP   030060   1768     1785     1878     2306     3070E    4047
$CRLF   056342   1705     2244     3082L    3670
$DADA   030072   1279     3097E    3646     3739
$DADA.  030101   2311     3107E
$DOS    064152   4153     4391L
$DOS.   064304   4394     4407L
$DRS    064321   3971     4465E
$DRS1   064321   4469L    4474     4539
$DRS15  065065   4493     4515     4551L
$DRS2   064336   4480L    4501
$DRS20  065100   4498     4517     4573L    4576
$DRS3   064341   4481L    4488
$DRS4   064360   4485     4492L
$DRS4.5 064374   4494     4498L
$DRS5   065004   4505L    4516
$DRS6   065014   4483     4497     4513L
$DRS7   065043   4518     4528L
$DRS8   065050   4534L    4537
$DRSB   065116   4475     4480     4506     4529     4585L
$DTB    056350   3119L
$DTB1   056354   3123L    3126
$DTB2   056363   3131L    3136
$DTB3   057000   3133     3140L
$DU66   030106   3160E    3745
$FERR1  057041   3182L    3187
$FERR2  057055   3185     3191L
$FERROR 057011   2057     2062     3174L
$HLIHL  030211   1166     1173     1200     1251     1540     3207E
$INDL   030234   1253     1265     1297     1449     1476     1488     3244E
$INDLB  057100   1143     1268     1271     1398     1405     3259L    4181
$INDS   057121   1206     1287     1546     1616     1629     3292L
$INDSB  057155   3322L
$ITL    065126   4602     4605L
$ITL.   065120   4023     4602L
$MCU    057203   2255     3375L    4352     4366     4623
$MLU    065134   4603     4618L    4650
$MLU1   065137   4621L    4626
$MOVE   030252   1677     2162     2456     2459     2490     2504     2531     2806     2940     3404E    3450
$MOVEL  057214   1066     2738     2740     2897     2997     3002     3434L    4158     4191     4200
$MU86   031007   1277     3465E
$RCHAR  057237   2254     3475L    3476     4169     4211     4400     4654
$RSTALL 031047   3006     3497E    4276
$RTL    065163   4606     4648     4652E
$RTL.   065154   4648L
$RTL1   065164   4654L    4660
$RTL2   065216   4656     4677L
$SAVALL 031054   2995     3511E    4272
$SOB    057250   2461     2541     2976     2981     3524L    3975     4313     4513
$SOB1   057251   3525L    3528     3530
$TFN    057266   3543L
$TFN.   057271   2249     3544L
$TFN1   057304   3545     3550L    3555
$TJMP   031061   3576E
$TJMP.  031062   3578E
```

```
$TYPC.  057337     3552       3621L      3671
$TYPCC  057317     3592E      3601
$TYPCH  057333     3546       3611L
$TYPL.  057360     3648       3660E      3674
$TYPLN. 057342     3642L
$TYPTX  031136      849        880       1681       1970       1977       1984       2009       2076       2080       2091       2106       2250
                   2264       3175       3191       3706E      4019       4034       4123       4164       4206       4391       4397
$TYPTX. 031144     3708E
$UDD    031157     3723E
$UDDN   060002      848       3738E
$WCHAR  057245     2079       3186       3479L
$ZERO   031212     1213       3809E
.ABUSS  040024      690E
.ALARM  002136      663E
.ALEDS  040013      688E      1668       1675       1698
.CHFLG  000060      341L      1558
.CLEAN  000205      356L
.CLEAR  000055      338L
.CLEARA 000056      339L       818
.CLOSE  000046      331L      1358       1499       2820       4109
.CLRCO  000007      315L      2253
.CONSL  000006      314L      4223       4275
.CRC    002347      671E
.CRCSUM 040027      691E
.CTC    002172      665E
.CTL2FL 040066      697E
.CTLC   000041      326L      4121
.CTLFLG 040011      687E
.DAD    000206      357L      4415
.DECODE 000053      336L      2744
.DELET  000050      333L      1456
.DISMT  000061      342L
.DLEDS  040021      689E
.DLY    000053      660E
.DMNMS  000203      354L       857        859       1594
.DMOUN  000201      352L
.DOD    003122      674E
.DODA   003356      676E
.DSPMOD 040007      685E
.DSPROT 040006      684E
.DUMP   001374      662E
.ERROR  000057      340L      2097       3195
.EXIT   000000      308L       873       2084
.HORN   002140      664E      1684
.IDENT  000000      659E
.IOWRK  040002      682E
.LINK   000040      325L
.LOAD   001267      661E
.LOADD  000062      343L      4134       4146
.LOADO  000010      316L      4408       4412
.MFLAG  040010      686E      1664
.MONMS  000202      353L      1817
.MOUNT  000200      351L
.NAME   000054      337L
.NMIRET 040064      696E
.OPEN   000063      344L
.OPENC  000045      330L      1463
.OPENR  000042      327L      1233       2756
```

```
.OPENU 000044     329L     1472
.OPENW 000043     328L     1443
.PCHL  002264     667E
.POSIT 000047     332L     1302   1481
.PRINT 000003     311L     1680   2108
.RCK   003260     675E
.READ  000004     312L     1316   2767
.REGI  040005     683E
.REGPTR 040035    694E
.RENAM 000051     334L
.RESET 000204     355L
.RNB   002331     670E
.RNP   002325     669E
.SCIN  000001     309L     3475
.SCOUT 000002     310L     3083   3479   3598   3621
.SETTP 000052     335L     2641   2956   4111
.SRS   002265     668E
.START 040000     681E
.SYSRES 000012    318L
.TICCNT 040033    693E
.TPERR 002205     666E
.TPERRX 040031    692E
.UIVEC 040037     695E
.VERS  000011     317L     4103
.WNB   003024     673E
.WNP   003017     672E
.WRITE 000005     313L     1496
ABS.COD 000010    773L      793
ABS.ENT 000006    771L
ABS.ID 000000     767L
ABS.LDA 000002    769L
ABS.LEN 000004    770L
AEN    052254    2139L     2725   2807
AEN1   052276    2144      2148L
AEN2   052335    2146      2164L
AENA   052337    2139      2161   2167L
AIO.CGN 041047    482L
AIO.CHA 041116    497L
AIO.CNT 041111    493L
AIO.CSI 041050    483L
AIO.DDA 041041    478E
AIO.DES 041055    487L
AIO.DEV 041057    488L
AIO.DIR 041062    491L     3543
AIO.DTA 041053    486L
AIO.EOF 041113    495L
AIO.EOM 041112    494L
AIO.FLG 041043    479L
AIO.GRT 041044    480L
AIO.LGN 041051    484L
AIO.LSI 041052    485L
AIO.SPG 041046    481L
AIO.TFP 041114    496L
AIO.UNI 041061    489L     2412   3021
AIO.VEC 041040    477L
BELL   000007     366E     1682   1971   1978   1985   2010   2077   2092   2107   2265   3176   4392
BKSP   000010     368E
BOOT.P 000001     457E
```

```
BSL      052360    1064      2182L
BSL1     052366    2187L     2200
BSL2     053012    2196      2197L
BSLA     053022    2182      2202L
BUFF     065242     824      3868     4740E
BUFMINL  002000    4739E
BUFPTR   060234     825      1074     2840     2953     3868L
BUFSIZ   060236     821      2842     2952     3869L
C.STX    000002     370E
C.SYN    000026     369E
CAD      054046    2191      2385     2445L    2733     2896     2900
CAD.     054052    2448L
CAD0     054054    2446      2449L
CAD1     054141    2464      2466     2468     2476L
CAD2     054204    2479      2497L
CAD2.4   054232    2511L     2514
CAD2.6   054240    2508      2515L
CAD3     054277    2518      2536L
CAD4     054301    2470      2472     2541L
CAD5     054307    2477      2486     2493     2524     2527     2548L
CADA     054313    2450      2509     2552L
CB.CLI   000100     605E      628
CB.MTL   000040     604E
CB.SPK   000200     606E
CB.SSI   000020     603E
CB2.CLI  000002     609E
CB2.ORG  000040     610E
CB2.SID  000100     611E
CB2.SSI  000001     608E
CBR      046230    1133      1304     1575L
CCHIT    043002     880L     4119
CDA      055063    2140      2397     2664L    2917
CDA5     055127    2666      2671     2676     2698L    2710
CDA6     055145    2705      2707L
CDA7     055147    2704      2709L
CDB.H84  000001     400E
CDB.H85  000000     399E
CFS.     053023    1274      2214L
CFS1     053026    2215L     2220
CMDLIN   060240     992      3870L    3982     4115
CN.170M  000014     646E
CN.174M  000003     645E
CN.ABO   000200     650E
CN.BAU   000100     649E
CN.DES   000001      40E     1442     1454     1462     1471     1480     1494     1498
CN.DIR   000002      41E     2755     2765     2819
CN.MEM   000040     648E
CN.PRI   000020     647E
CN.SOU   000000      39E     1232     1252     1301     1315     1357
CND.H17  000000     652E
CND.H47  000001     654E
CND.NDI  000000     653E
CO.FLG   000001     552E     4274
COF      043343     838       988L
COF1     043366     994      1004L
COFA     043375     998      1008L
CQF      053036    2145      2238L
CQF.     053054    2242      2248L    2266
```

```
CQF1      053112       2260      2264L
CR        000015       362E
CRF       043012        833       898L
CRFA      043035        903       910L
CS.FLG    000200       553E
CSD       043147        834       929L
CSD1      043212        934       953L
CSDA      043254        944       971L
CSDB      043307        961       974L
CSDC      043316        953       975L
CSDD      043325        954       964    977L
CSF       053122      2298L      2801
CSF1      053130      2303L      2314
CSF2      053157       2309      2319L
CSFA      053163       2302      2324L   2325
CSFB      053232        898       901     907   2328L
CSFC      053333        938       941     948     956     959     968   2333L
CSFD      053350        942       955   2334L
CSL.CHR   000001       529E      4221    4222
CSL.ECH   000200       526E
CSL.RAW   000004       527E
CSL.WRP   000002       528E
CTLA      000001       377E
CTLB      000002       378E
CTLC      000003       379E      4120
CTLD      000004       380E      4655
CTLO      000017       381E
CTLP      000020       382E
CTLQ      000021       383E
CTLS      000023       384E
CTLZ      000032       385E
CTP.2SB   000010       538E
CTP.BKM   000002       539E
CTP.BKS   000200       534E
CTP.FF    000100       535E
CTP.MLI   000040       536E
CTP.MLO   000020       537E
CTP.TAB   000001       540E
CWM       053366      2352L      2360    2794
CWM1      053375       2354      2357L
D.CON     040110       280L
D.RAM     040240       283L
D.VEC     040130       282L
DC.ABT    000007       575L
DC.CLO    000006       574L
DC.LOD    000011       577L
DC.MAX    000013       579L
DC.MOU    000010       576L       808    1733    1841    1914    1932    2028
DC.OPR    000003       571L
DC.OPU    000005       573L
DC.OPW    000004       572L
DC.RDY    000012       578L      1686    1690
DC.REA    000000       568L
DC.RER    000002       570L      1921    1939
DC.WRI    000001       569L      2037
DDF       054004       1057      2380L
DDF.BOL   000011       710E
DDF.BOO   000000       709L
DDF.LAB  000011      711L      1919      1937      2035
DDF.USR  000012      712L
DDF2     054007     2384L
DDS      064032     4029       4295E
DDS1     064105     4322       4334L
DDS2     064123     4314       4316      4350L
DDS3     064142     4300       4301      4364L
DEFAULT  000000      780L      2190      2384      3000      3005      3884      3904      4056      4061      4067      4072
DEST     060267      856        931      2024      2041      2384      2411      2414      3882E     3884      3887      3890      3893
                    3896       3899      4056      4057      4059      4061      4062      4145      4148      4151      4194      4203      4215
DESTERR  051301     1444       1464      1473      1497      1500      1559      2061L
DESTFB   060331     1068       1431      1441      1453      1461      1470      1557      2061      2396      3925L
DEV.DDA  000004      106L
DEV.DVG  000015      119L
DEV.DVL  000013      118L
DEV.FLG  000006      107L
DEV.JMP  000003      105L      4137      4149
DEV.MNU  000010      115L
DEV.MUM  000007      114L
DEV.NAM  000000       97L
DEV.RES  000002      101L
DEV.UNT  000011      116L      4177
DEVELEN  000016      121E
DEVICE   000014      788L       856       858       929       931      1592      1815      2024      2041      3896      3916      4057
                    4062       4068      4073      4133      4145      4161      4173      4203      4215
DEVTAB   000006      782L      3887      3907      4136      4148      4176
DF.CLR   000376       64E      2788
DF.EMP   000377       63E      2785
DIF.CNT  000020       89E      1260      3825
DIF.LOC  000100       87E
DIF.SYS  000200       86E
DIF.WP   000040       88E
DIR.ALD  000025       79L
DIR.CLU  000015       72L
DIR.CRD  000023       78L
DIR.EXT  000010       67L      2458      2530      2675      3004
DIR.FGN  000020       75L      1272
DIR.FLG  000016       73L      1254
DIR.LGN  000021       76L
DIR.LSI  000022       77L      1269
DIR.NAM  000000       66L      2248      2502      2670      2675      2741      2805      3543
DIR.PRO  000013       68L
DIR.VER  000014       69L
DIRELEN  000027       81E       165       177       491      4733
DIRIDL   000015       70E      2305      2310      2325
DIRNAM   060250     2739       2743      2754      3879L
DIRWRKP  041121     2762       2774      4723E
DIS.ENC  001373      181L      2775
DIS.ENT  000000      176E
DIS.LNK  001376      183L
DIS.SEC  001374      182L
DM.MR    000000      618E
DM.MW    000001      619E
DM.RR    000002      620E
DM.RW    000003      621E
DMD      046236     1592L      1657
DMD.     046241     1594L      2025
DNT      054314     2476       2492      2523      2569L
```

```
DNT1     054323     2573L     2576
DNT2     054334     2581L     2604
DNT3     054376     2584      2591      2599L
DNT4     055021     2589      2593      2595      2622L
DNT5     055010     2587      2613L     2617
DNTA     055026     2569      2577      2623      2626L
DOS1     064272     4400L     4402
DR.IM    000001      102E
DR.PR    000002      103E
DRIVER   000010      784L     2414      3023      3890      3910      4139      4151
DRIVES2  060241      812      1746      1760      1778      1854      1870      3871L     4043      4053      4141      4185
DSTDRVR  054033     1734      1915      1922      2029      2038      2410L
DSTLAB   062167     1731      1767      1775      1783      1958      1962      2032      2034      4712L
DT.CH    000020      112E
DT.CR    000002      109E
DT.CW    000004      110E
DT.DD    000001      108E     2747
DT.RN    000010      111E
DV.EL    000000       98E
DV.NU    000001       99E
DVCLEN   000021      790E     3899      3919      4192
EBM      055037     1070      2638L
EC.CNA   000004      207L
EC.DDA   000027      226L
EC.DIF   000017      218L
EC.DIW   000035      232L
EC.DNI   000045      240L
EC.DNR   000046      241L
EC.DNS   000005      208L     2748
EC.DSC   000047      242L
EC.EOF   000001      204L     1322
EC.EOM   000002      205L
EC.FAO   000031      228L
EC.FAF   000026      225L
EC.FL    000030      227L
EC.FNF   000014      215L     1458
EC.FNO   000011      212L
EC.FNR   000034      231L
EC.FOD   000043      238L
EC.FOC   000013      214L
EC.ICN   000016      217L
EC.IDN   000006      209L
EC.IFC   000020      219L
EC.IFN   000007      210L     2548
EC.ILC   000003      206L
EC.ILO   000040      235L
EC.ILR   000012      213L
EC.ILV   000037      234L
EC.IOI   000052      245L
EC.IS    000032      229L     4508
EC.NCV   000050      243L     4228
EC.NEM   000021      220L     2845
EC.NOS   000051      244L
EC.NFM   000044      239L
EC.NRD   000010      211L
EC.NVM   000042      237L
EC.OTL   000053      246L
EC.RF    000022      221L
```

```
  EC.UNA  000036       233L
  EC.UND  000015       216L
  EC.UUN  000033       230L
  EC.VPM  000041       236L
  EC.WF   000023       222L
  EC.WP   000025       224L
  EC.WPV  000024       223L
  ENL     000212       375E      852    1971    1979    1986    2011    2112    2113    2114    2115    2116    2117
                      2118      2119    4036    4128    4392
  ENTRY   062167       797      4101E
  ERROR   052123       810     1058    1065    1459    1595    1735    1818    1843    1917    1923    1935    1941
                      2039     2090L   2846    2958    3973    3987    4002    4112    4135    4147    4154    4224    4230
  ERROR1  052154      2095     2102L
  ERROR2  052157      2103L    2105
  ERRORA  052214      2102     2111L
  ESC     000033       373E
  EWS     055154      2195     2725L
  EWS1    055262      2762L    2784
  EWS3    055314      2782L    2815
  EWS4    055373      2795     2802    2812L
  EWS6    055374      2787     2813L
  EWS7    056002      2769     2790    2819L
  EWSA    056007      2823L
  EWSB    056015      2742     2746    2825L
  EWSC    056053      2741     2792    2827L
  EXIT    042376       868      872L    882    1972    1980    2012    4024
  EXIT.   043000       862      873L
  FB.CHA  000000       189L
  FB.FLG  000001       190L
  FB.FWA  000002       191L
  FB.LIM  000006       193L
  FB.LWA  000010       194L
  FB.NAM  000012       195L      196    1068    1431    1441    1453    1461    1470    1557    2055    2396    3177
  FB.NAML 000021       196E     1099    1161    2149    2159    2167    2731    2735    2933    2938    3932    4716
  FB.PTR  000004       192L
  FBENL   000033       197E
  FDN     060066      3819L    3821    3824    3830    3833    3836    3839    3842    3845    3848    3850
  FDN.1   060072      1610     1614    3858L
  FDN.ADR 000012      1307     1329    1342    1489    3842E
  FDN.AIM 000013      1342     1344    1399    1518    3845E
  FDN.AMR 000006      1298     1344    3836E
  FDN.AMW 000010      1477     1504    3839E
  FDN.FLG 000003      1257     1528    1538    3830E
  FDN.LNK 000000      1167     1174    1201    1207    1228    1419    1504    1524    1538    1541    1547    1617
                      1630     3821E
  FDN.SIZ 000004      1288     1450    3833E
  FDN.STA 000002      1144     1228    1257    1288    1298    1307    1329    1406    1419    1450    1524    1528
                      3824E
  FDNCNT  000010        34E    1613    3859
  FDNELEN 000014      1211     1614    1621    3848E   3859
  FDNFREE 060066      1151     1197    1202    1543    1548    1611    3855L
  FDNHED  060070      1083     1138    1163    1393    1542    1633    3856L
  FF      000014       376E
  FT.ABS  000000       757E      794
  FT.BAC  000003       760E
  FT.DD   000001       144E
  FT.OC   000020       148E
  FT.OR   000002       145E
```

```
FT.OU    000010    147E
FT.OW    000004    146E
FT.PIC   000001    758E
FT.REL   000002    759E
GETDLB   050032   1764      1910L
GETDLB.  050035   1776      1912L
GETSLB   050071   1874      1928L
GETSLB.  050074   1930L     2000
I.CONFL  000004    555E      556     4273
I.CONTY  000001    542E      543
I.CONWI  000003    548E      549
I.CSLMD  000000    531E     4220
I.CUSOR  000002    545E      546
IERR1    051307   1482      2066L    2642
IERR2    051314   2069L
IERR3    051321   1303      2071L
IFL      046247   1056      1610L
IFL1     046262   1616L     1626
ILDEHL   057073   3223L     3295
INA      056066   2158      2838L
INTERR   051326   2067      2070     2072     2075L
IOC.CGN  000010    153L
IOC.CSI  000011    154L
IOC.DDA  000002    141L      149      163
IOC.DES  000016    160L
IOC.DEV  000020    161L      788      932     4061     4062     4069     4072     4073     4081     4159     4201     4370
IOC.DIL  000021    163E
IOC.DIR  000023    165L      788     1254     1269     1272     4082     4159     4201     4371
IOC.DRL  000010    157E
IOC.DTA  000014    159L
IOC.FLG  000004    143L      157
IOC.GRT  000005    151L     1266
IOC.LGN  000012    155L
IOC.LNK  000000    140L
IOC.LSI  000013    156L
IOC.SFG  000007    152L
IOC.SQL  000003    149E
IOC.UNI  000022    162L      932     4061     4062     4069     4072     4073     4081     4082     4370     4371
IOCCTD   000001    169E     1250
IOCELEN  000052    167E
IP.CON   000362    594E
IP.PAD   000360    590E
ISDEHL   057176   3302      3358L
ITLA     065234   4026      4605     4700L
LAB.AUX  000117    747E      749
LAB.AXL  000001    749E
LAB.DAT  000000    724E     1963
LAB.DIS  000003    720L
LAB.GRT  000005    721L
LAB.IND  000001    719L
LAB.LAB  000021    743L      744
LAB.LBL  000074    744E
LAB.NOD  000002    726E
LAB.PSS  000016    735L
LAB.RGT  000012    731L
LAB.SER  000000    718L      806     1731     1839
LAB.SIZ  000014    734L
LAB.SFG  000007    722L
LAB.SFT  000117       748L
LAB.SYS  000001       725E       1967       2003       2031
LAB.VER  000011       729L       1958
LAB.VFL  000020       736L
LAB.VLT  000010       728L       1962       2002       2032
LAB.VPL  000005       738E        740        741
LAB.VPR  000014       733E        738
LABEL    064167      1766        1876       1910       1928       4714L
LF       000012       363E
LINEP    060242      1055        2380       2862       2872       3872L
LSN      056120      2183        2862L
LSN1     056123      2864L       2870
M.FOX    000303       638E
M.PAMB   000021       637E
MAD      046321      1651E       1754       1862
MAD1     046343      1670L       1673
MAD2     046376      1686L       1688
MAD3     047006      1690L       1692
MAD4     047024      1700L       1703
MDD      047041       814        1080       1725L      2023
MDD1     047076      1729        1746L
MDD2     047105      1752L       1769       1786
MDD3     047115      1748        1756L
MDD4     047153      1758        1775L
MDD5     047206      1780        1788L
MDDA     047221      1753        1796L
MDDB     047250      1756        1790       1792       1799L
MEML     065354       796        4703E
MINIMUM  060244       988        3873L      3984       4116       4243
MND      047251      1707        1815L
MND.     047254      1817L       2042       4174       4216
MSD      047262      1061        1078       1833L
MSD.     047336       805        1864E
MSD1     047317      1837        1854L
MSD2     047326      1860L       1879
MSD3     047336      1856        1866L
MSD4     047374      1868        1885L
MSDA     050007      1861        1893L
MSDB     050031      1866        1887       1889       1896L
MWN      056140      1432        2893L
MWN1     056173      2907L       2915
MWN2     056201      2909        2911L
MWNA     065167      2898        2901       4716L
NAMERR   051267      1234        1323       2054L
NAMTAB   065242      1146        1214       1430       2159       2731       2938       2939       4736L
NAMTLEN  060364       822        1088       1181       2148       2151       2730       2734       2737       2932       2935       3934L
NAMTMAX  060366       823        2153       2838       3935L
NAMTPTR  060370      1217        2054       3936L
NL       000012       374E        375        850       1797       1894       1971       1978       1985       2010       2011       2077       2081
                     2082        2082       2096       2265       3082       3176       3193       4020       4035       4124       4125       4126       4127
                     4165        4170       4207       4212       4392       4392       4398       4401       4659
NUL2     000000       365E
NULL     000200       364E
OBUFLIM  060232      1575        1576       2646       3862L
OBUFPTR  060233      1076        1353       1576       3863L
OCOPY    044065       904         945        962        965        999       1005       1054E
OCOPY1   044132      1074L       1086       1091
OCOPYA   044176      1059        1097L
```

```
OCOPYC  044177     842     1098L   1243
OCOPYD  044200    1069     1099L   1100    1429
OCOPYDL 000021    1067     1100E
OFILES  060372    1004     3938L   3940
OFILESA 060376    3939L    3990
OFILESL 000124    3940E    3989
OP.CTL  000360     591E
OP.DIG  000360     592E
OP.SEG  000361     593E
OP2.CTL 000362     595E
OVL.COD 000000     253L
OVL.ENS 000010     258E
OVL.ENT 000004     255L
OVL.FLB 000006     256L
OVL.IN  000001     424E
OVL.NUM 000014     426E
OVL.RES 000002     425E
OVL.SIZ 000002     254L
OVL.UCS 000200     427E
OVL0    000000     264L    4407
OVL1    000001     265L    4411
P.MIN   063377    4242L    4689
P.QUE   064006    4255L
PATCH   061117    3944L
PCL     061217    3963L    4130
PCL1    061274    3991L    3999
PDN     061313    4018L    4038    4131
PDN.    061365    4027L
PDN1    062045    4030     4042L
PDN2    062123    4048     4066L
PDNA    062153    4027     4045    4055    4077L   4079
PDNAL   000003    4044     4079E
PDNB    062155    4058     4078L
PDNC    062156    4028     4084L
PDND    062164    4046     4066    4069    4086L
PEC.CO  000207      53E    2119    4001
PEC.CS  000204      50E    2116    3986
PEC.DF  000200      46E    2112    2757
PEC.DNC 000201      47E    2113
PEC.IDF 000206      52E    2118    2390
PEC.IUW 000205      51E    2117
PEC.RSE 000202      48E    2114
PEC.TFI 000203      49E    2115
PIO.DEV 065210    2454     2489    2665    2739    2996    2999    4730L
PIO.DIR 065213    2248     2458    2502    2530    2670    2741    2805    2898    2902    3004    4733L
PIO.UNI 065212    2996     4731L
PRS     062167    4103L    4708
PRS1    063014    4143     4153L
PRS2    063133    4169L    4171
PRS3    063217    4187     4200L
PRS4    063335    4211L    4213
PRS5    063353    4196     4218    4220L
PRSA    063075    4162     4167L
PRSB    063277    4204     4209L
PRSERR  063371    4104     4106    4228L
QUERY   060245     829      832     836    2238    3874L   4117    4256
QUERYF  000001       1E    4691
QUOTE   000047     371E
```

```
RDD      050130      1788       1958L
RDD1     050226      1969       1977L   1987    2007
RDD2     050332      1960       1984L
REN      056214      1550       2932L
RESTART  042373       866E      2098    2109    3196
RMEML    065242      4110       4742E
ROMBOOT  030000       275E
RPH      044221      1079       1128E   1359
RPH1     044252      1141       1151L
RPH2     044271      1165L      1178
RPH2.2   044314      1171       1180L
RPH2.5   044372      1147       1226L
RPH3     045131      1239       1261    1295L
RPH4     045223      1321       1340L
RPHA     045257      1270       1278    1361L
RSD      051045      1885       1999L
RUBOUT   000177       367E
S.BAUD   040344       401L
S.BDA    041120       499L
S.BOOTF  041034       456L
S.CAADR  040333       559L
S.CACC   041006       440L
S.CCTAB  040335       560L
S.CDB    040343       398L
S.CFWA   040352       408L      1249
S.CODE   041007       441L
S.CONFL  040332       557L
S.CONTY  040327       544L
S.CONWI  040331       550L
S.CSLMD  040326       532L       543     546     549     556
S.CUSOR  040330       547L
S.DATC   040310       513L
S.DATE   040277       512L
S.DCS    041033       454L
S.DDDTA  040366       419L
S.DDGRP  040364       416L
S.DDLDA  040360       414L
S.DDLEN  040362       415L
S.DDOPC  040370       420L
S.DFWA   040354       409L
S.DIREA  041016       448L
S.DLINK  040346       406L
S.FASER  041013       447L
S.FCI    041021       449L
S.GRT0   024000       271E
S.GRT1   025000       272E
S.GRT2   026000       273E
S.GUF    041027       451L      4180
S.HIMEM  040316       515L
S.INT    040343       285L       394
S.JUMPS  041010       445L
S.MOUNT  041032       453L
S.OFWA   040350       407L
S.OMAX   040324       521L
S.OSN    041004       436L
S.OVLE   041000       433L
S.OVLFL  040371       429L
S.OVLS   040376       432L
```

```
S.OVSTK 041035      461L
S.RFWA  040356      410L
S.SCI   041024      450L
S.SCR   041121      500L      4723
S.SDD   041010      446L
S.SOVR  041146      287L       289
S.SSN   041002      435L
S.SYSM  040320      517L      2638
S.TIME  040312      514L
S.UCSF  040372      430L
S.UCSL  040374      431L
S.USRM  040322      519L      2643
S.VAL   040277      284L       510
SBE     056241     2849       2951L
SFS     056262     2199       2976L
SFS1    056274     2979       2981L
SND     056277     2193       2995L
SOURCE  060310      858        929      1592     1815     2190     3000     3005     3020     3023     3902E    3904     3907
                   3910       3913      3916     3919     4067     4068     4070     4072     4073     4133     4136     4139     4161
                   4173       4176      4193
SRCDRVR 056327      809       1687      1691     1842     1933     1940     3019L
SRCLAB  063167      806       1784      1839     1877     1999     2002     4713L
SRCSFG  060246     1276       3875L     4183
SSL     051212      837       2023L
ST.CNT  000020     1285       1439      3825E
ST.OPR  000002     1145       1237      1246     1332     1407     1526     3826E
ST.OPW  000001     1421       1423      3827E
STACK   042200      291E       801      3965
STACKL  001032      289E
START   042200      801L      4226
SWTFWA  065221     3970       4685L
SYDD    040130      281E
SYSA    042334      847        851L
SYSCALL 000377      301E       873      1233     1302     1316     1358     1443     1456     1463     1472     1481     1496
                   1499       1558      2084     2097     2108     2641     2744     2756     2767     2820     2956     3083     3195
                   3475       3479      3598     3621     4275
TAB     000011      372E      3529      4124     4124     4124     4125     4125     4125     4126     4126
TPL1    057374     3672L
UDDN1   060006     3742L      3758
UDDN1.5 060040     3762L      3769
UDDN2   060042     3755       3767L
UDDN3   060043     3768L      3772
UNIT    000013      786L      2411      3020     3893     3913     4059     4070
UNT.DIS 000006      131L
UNT.FLG 000000      127L
UNT.GRT 000002      129L
UNT.GTS 000004      130L
UNT.SIZ 000010      133E
UNT.SPG 000001      128L      4182
UO.CLK  000001      630E      1663
UO.DDU  000002      629E      1663
UO.HLT  000200      627E      1663
UO.NFR  000100      628E
USERFWA 042200      292E       793       795      796
VERS    000040      299E      1959      4105     4125     4125
VFL.NSD 000001      737E
VOLFLAG 060247      804       1725      1737     1739     1833     1845     1847     3876L
WPH     045260     1081       1389E     1560
```

```
    WPH0     045313       1401        1416L
    WPH1     045366       1440        1449L
    WPH1.5   046015       1457        1460L
    WPH2     046033       1422        1470L
    WPH3     046065       1445        1465      1486L
    WPH4     046140       1411        1518L
    XCHGBC   060055       3292        3296      3304      3306      3788L

  14324 BYTES FREE
```