```
000.001                1  .PIP.   EQU     1               DON'T ASSEMBLE AS PIP
000.000                2  ONECOPY EQU     0               ASSEMBLE AS ONECOPY
                       3
000.001                4          IF      .PIP.
                       5          TITLE   'PIP - PERIPHERAL INTERCHANGE PROGRAM'
                       6          ELSE
                       8          ENDIF
                       9
                      10
                      11  ***     PIP - PERIHPERAL INTERCHANGE PROGRAM.
                      12  *
                      13  *       J.G. LETWIN, 11/1977 FOR *HEATH* COMPANY
                      14  *
                      15  *       COPYRIGHT 1977 BY HEATH COMPANY
                      16  *
                      17  *       G. Chandler,    78/09   Maintenence Release
                      18  *                       79/04
                      19  *
                      20  *                       79/11   50.05.00
                      21  *


                      23  ***     USE:
                      24  *
                      25  *       DEST=SOURCE1 [,SOURCE2,...,SOURCEN] [/SWITCH1.../SWITCHN]
                      26  *
                      27  *       SWITCHES:
                      28  *
                      29  *       /R[ENAME]               RENAME
                      30  *       /DEL[ETE]       DELETE
                      31  *       /L[IST]         LIST
                      32  *       /B[RIEF]        BRIEF LIST
                      33  *       /S[YSTEM]       ENCLUDE SYSTEM FILES
                      34  *       /V[ERSION]      PIP VERSION NUMBER
                      35  *       /MOU[NT]        MOUNT DEVICE
                      36  *       /DIS[MOUNT]     DISMOUNT DEVICE
                      37  *       /RES[ET]        RESET DEVICE
                      38  *
                      39  *       /SU[PRESS]      SUPRESS
                      40  *       /JGL            WHO?


                      42  **      SYSTEM EQUIVALENCES
                      43
000.000               44  CN.SOU  EQU     0               SOURCE CHANNEL NUMBER
000.001               45  CN.DES  EQU     1               DESTINATION CHANNEL NUMBER
000.002               46  CN.DIR  EQU     2               DIRECTORY CHANNEL NUMBER
                      47
                      48  **      PROGRAM ERROR CODES
                      49
000.200               50  PEC.DF  EQU     200Q            DEVICE FORMAT ERROR
000.201               51  PEC.DNC EQU     201Q            DEVICES NOT CONSISTANT
```

```
000.203                 52  PEC.TFI EQU     203Q            TARGET FILE ILLEGAL
000.204                 53  PEC.CS  EQU     204Q            CONTRADICTORY SWITCHES
000.205                 54  PEC.IUW EQU     205Q            ILLEGAL USE OF WILDCARD
000.206                 55  PEC.IDF EQU     206Q            ILLEGAL DESTINATION FILE FORMAT
000.207                 56  PEC.SFI EQU     207Q            SOURCE FILE ILLEGAL
000.000                 57          IF      ONECOPY
000.210                 58  PEC.FCI EQU     210Q            FILE CONCATINATION ILLEGAL
                        59          ENDIF
                        60
000.000                 61          XTEXT   U8250


                       63X **       8250 UART CONTROL AND BIT DEFINITIONS.
                       64X
000.350                65X SC.ACE   EQU     350Q            SYSTEM CONSOLE PORT IF 8250 ACE
000.156                66X AC.DLY   EQU     110             220 MIL. SEC. DELAY FOR 8250
                       67X
000.000                68X UR.RBR   EQU     0               RECEIVER BUFFER REGISTER (READ ONLY)
                       69X
000.000                70X UR.THR   EQU     0               TRANSMITTER HOLDING REGISTER (WRITE ONLY)
                       71X
000.000                72X UR.DLL   EQU     0               DIVISOR LATCH (LEAST SIGNIFICANT)
                       73X
000.001                74X UR.DLM   EQU     1               DIVISOR LATCH (MOST SIGNIFICANT)
                       75X
000.001                76X UR.IER   EQU     1               INTERRUPT ENABLE REGISTER
000.001                77X UC.EDA   EQU     00000001B        ENABLE RECEIVED DATA AVAILABLE INTERRUPT
000.002                78X UC.TRE   EQU     00000010B        ENABLE TRANSMIT HOLD REGISTER EMPTY INTERRUPT
000.004                79X UC.RSI   EQU     00000100B        ENABLE RECEIVE STATUS INTERRUPT
000.010                80X UC.MSI   EQU     00001000B        ENABLE MODEM STATUS INTERRUPT
                       81X
000.002                82X UR.IIR   EQU     2               INTERRUPT IDENTIFICATION REGISTER
000.001                83X UC.IIP   EQU     00000001B        INVERTED INTERRUPT PENDING (0 MEANS PENDING)
000.006                84X UC.IID   EQU     00000110B        INTERRUPT ID
                       85X
000.003                86X UR.LCR   EQU     3               LINE CONTROL REGISTER
000.000                87X UC.5BW   EQU     00000000B        5 BIT WORDS
000.001                88X UC.6BW   EQU     00000001B        6 BIT WORDS
000.002                89X UC.7BW   EQU     00000010B        7 BIT WORDS
000.003                90X UC.8BW   EQU     00000011B        8 BIT WORDS
000.004                91X UC.2SB   EQU     00000100B        TWO STOP BITS SELECTED
000.010                92X UC.PEN   EQU     00001000B        PARITY COMPUTATION ENABLED
000.020                93X UC.EPS   EQU     00010000B        EVEN PARITY SELECT
000.040                94X UC.SKP   EQU     00100000B        STICK PARITY
000.100                95X UC.SB    EQU     01000000B        SET BREAK
000.200                96X UC.DLA   EQU     10000000B        DIVISOR LATCH ACCESS
                       97X
000.004                98X UR.MCR   EQU     4               MODEM CONTROL REGISTER
000.001                99X UC.DTR   EQU     00000001B        DATA TERMINAL READY
000.002               100X UC.RTS   EQU     00000010B        REQUEST TO SEND
000.004               101X UC.OU1   EQU     00000100B        OUT 1
000.010               102X UC.OU2   EQU     00001000B        OUT 2
000.020               103X UC.LOO   EQU     00010000B        LOOP
                      104X
```

```
000.005                 105X UR.LSR   EQU     5               LINE STATUS REGISTER
000.001                 106X UC.DR    EQU     00000001B         DATA READY
000.002                 107X UC.OR    EQU     00000010B         OVERRUN
000.004                 108X UC.PE    EQU     00000100B         PARITY ERROR
000.010                 109X UC.FE    EQU     00001000B         FRAMING ERROR
000.020                 110X UC.BI    EQU     00010000B         BREAK INTERRUPT
000.040                 111X UC.THE   EQU     00100000B         TRANSMITTER HOLDING REGISTER EMPTY
000.100                 112X UC.TSE   EQU     01000000B         TRANSMITTER SHIFT REGISTER EMPTY
                        113X
000.006                 114X UR.MSR   EQU     6               MODEM STATUS REGISTER
000.001                 115X UC.DCS   EQU     00000001B         DELTA CLEAR TO SEND
000.002                 116X UC.DDR   EQU     00000010B         DELTA DATA SET READY
000.004                 117X UC.TER   EQU     00000100B         TRAILING EDGE OF RING
000.010                 118X UC.DRL   EQU     00001000B         DELTA RECEIVE LINE SIGNAL DETECT
000.020                 119X UC.CTS   EQU     00010000B         CLEAR TO SEND
000.040                 120X UC.DSR   EQU     00100000B         DATA SET READY
000.100                 121X UC.RI    EQU     01000000B         RING INDICATOR
000.200                 122X UC.RLS   EQU     10000000B         RECEIVED LINE SIGNAL DETECT
000.000                 123           XTEXT   U8251
```

```
                        126X **     8251 USART BIT DEFINITIONS.
                        127X *
                        128X
                        129X **     PORT ADDRESSES
                        130X
000,000                 131X UDR    EQU     0               DATA REGISTER IS EVEN
000.001                 132X USR    EQU     1               STATUS REGISTER IS NEXT
                        133X
000.372                 134X SC.UART EQU    372Q            CONSOLE USART ADDRESS (IFF 8251)
                        135X
                        136X
                        137X **     MODE INSTRUCTION CONTROL BITS.
                        138X
000,100                 139X UMI.1B EQU     01000000B       1 STOP BIT
000.200                 140X UMI.HB EQU     10000000B       1 1/2 STOP BITS
000,300                 141X UMI.2B EQU     11000000B       2 STOP BITS
000.040                 142X UMI.PE EQU     00100000B       EVEN PARITY
000,020                 143X UMI.PA EQU     00010000B       USE PARITY
000.000                 144X UMI.L5 EQU     00000000B       5 BIT CHARACTERS
000,004                 145X UMI.L6 EQU     00000100B       6 BIT CHARACTERS
000.010                 146X UMI.L7 EQU     00001000B       7 BIT CHARACTERS
000,014                 147X UMI.L8 EQU     00001100B       8 BIT CHARACTERS
000.001                 148X UMI.1X EQU     00000001B       CLOCK X 1
000,002                 149X UMI.16X EQU    00000010B       CLOCK X 16
000.003                 150X UMI.64X EQU    00000011B       CLOCK X 64
                        151X
                        152X **     COMMAND INSTRUCTION BITS.
                        153X
000.100                 154X UCI.IR EQU     01000000B       INTERNAL RESET
000,040                 155X UCI.RO EQU     00100000B       READER-ON CONTROL FLAG
000.020                 156X UCI.ER EQU     00010000B       ERROR RESET
000,004                 157X UCI.RE EQU     00000100B       RECEIVE ENABLE
000.002                 158X UCI.IE EQU     00000010B       ENABLE INTERRUPTS FLAG
000,001                 159X UCI.TE EQU     00000001B       TRANSMIT ENABLE
                        160X
                        161X **     STATUS READ COMMAND BITS.
                        162X
000,040                 163X USR.FE EQU     00100000B       FRAMING ERROR
000.020                 164X USR.OE EQU     00010000B       OVERRUN ERROR
000,010                 165X USR.PE EQU     00001000B       PARITY ERROR
000.004                 166X USR.TXE EQU    00000100B       TRANSMITTER EMPTY
000,002                 167X USR.RXR EQU    00000010B       RECEIVER READY
000.001                 168X USR.TXR EQU    00000001B       TRANSMITTER READY
000,000                 169         XTEXT   DIRDEF


                        171X **     DIRECTORY ENTRY FORMAT.
                        172X
000,000                 173X        ORG     0
                        174X
                        175X
000.377                 176X DF.EMP EQU     377Q            FLAGS ENTRY EMPTY
000,376                 177X DF.CLR EQU     376Q            FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
                        178X
000,000                 179X DIR.NAM DS     8               NAME
```

```
000.010                 180X DIR.EXT DS      3          EXTENSION
000.013                 181X DIR.PRO DS      1          PROJECT
000.014                 182X DIR.VER DS      1          VERSION
000.015                 183X DIRIDL  EQU     *          FILE IDENTIFICATION LENGTH
                        184X
000.015                 185X DIR.CLU DS      1          CLUSTER FACTOR
000.016                 186X DIR.FLG DS      1          FLAGS
000.017                 187X         DS      1          RESERVED
000.020                 188X DIR.FGN DS      1          FIRST GROUP NUMBER
000.021                 189X DIR.LGN DS      1          LAST GROUP NUMBER
000.022                 190X DIR.LSI DS      1          LAST SECTOR INDEX (IN LAST GROUP)
000.023                 191X DIR.CRD DS      2          CREATION DATE
000.025                 192X DIR.ALD DS      2          LAST ALTERATION DATE
                        193X
000.027                 194X DIRELEN EQU     *          DIRECTORY ENTRY LENGTH
000.027                 195          XTEXT   DIFDEF


                        197X **      DIRECTORY FILE FLAGS.
                        198X
000.200                 199X DIF.SYS EQU     10000000B          SYSTEM FILE
000.100                 200X DIF.LOC EQU     01000000B          LOCKED FOR CHANGE
000.040                 201X DIF.WP  EQU     00100000B          WRITE PROTECTED
000.020                 202X DIF.CNT EQU     00010000B          CONTIGUOUS FILE
                        203X
000.027                 204          XTEXT   OVLDEF


                        206X **      OVERLAY TABLE ENTRYS.
                        207X
000.000                 208X         ORG     0
                        209X
000.000                 210X OVL.COD DS      2          FIRST SECTOR OF OVERLAY CODE
000.002                 211X OVL.SIZ DS      2          OVERLAY SIZE
000.004                 212X OVL.ENT DS      2          OVERLAY ENTRY POINT
000.006                 213X OVL.FLB DS      1          OVERLAY FLAG BYTE
000.007                 214X         DS      1          DUMMY BYTE TO ROUND TABLE SIZE UP TO 8
000.010                 215X OVL.ENS EQU     *          OVERLAY ENTRY SIZE
                        216X
                        217X *       OVERLAY INDICES
                        218X
000.000                 219X         ORG     0
                        220X
000.000                 221X OVL0    DS      1
000.001                 222X OVL1    DS      1
000.002                 223          XTEXT   DEVDEF
```

```
                        225X **     DEVICE TABLE ENTRYS.
                        226X
000.000                 227X        ORG     0
                        228X
000.000                 229X DEV.NAM DS     2           DEVICE NAME
000.000                 230X DV.EL   EQU    00000000B   END OF DEVICE LIST FLAG
000.001                 231X DV.NU   EQU    00000001B   DEVICE ENTRY NOT IN USE
                        232X
000.002                 233X DEV.RES DS     1           DRIVER RESIDENSE CODE
000.001                 234X DR.IM   EQU    00000001B   DRIVER IN MEMORY
000.002                 235X DR.PR   EQU    00000010B   DRIVER PERMINANTLY RESIDENT
                        236X
000.003                 237X DEV.JMP DS     1           JMP TO PROCESSOR
000.004                 238X DEV.DDA DS     2           DRIVER ADDRESS
000.006                 239X DEV.FLG DS     1           FLAG BYTE
000.001                 240X DT.DD   EQU    00000001B   DIRECTORY DEVICE
000.002                 241X DT.CR   EQU    00000010B   CAPABLE OF READ OPERATION
000.004                 242X DT.CW   EQU    00000100B   CAPABLE OF WRITE OPERATION
                        243X
000.007                 244X DEV.SPG DS     1           SECTORS PER GROUP THIS DEVICE
000.010                 245X DEV.MUM DS     1           MOUNTED UNIT MASK
000.011                 246X DEV.MNU DS     1           MAXIMUM NUMBER OF UNITS
000.012                 247X DEV.UNT DS     2           ADDRESS OF UNIT SPECIFIC DATA TABLE
                        248X
000.014                 249X DEV.DVL DS     2           DRIVER BYTE LENGTH
000.016                 250X DEV.DVG DS     1           DRIVER ROUTINE GROUP ADDRESS
                        251X
000.017                 252X DEVELEN EQU    *           DEVICE TABLE ENTRY LENGTH


                        254X **     UNIT SPECIFIC DEVICE DATA TABLE ENTRIES
                        255X
000.000                 256X        ORG     0
                        257X
000.000                 258X UNT.FLG DS     1           UNIT SPECIFIC  *DEV.FLG*
000.001                 259X UNT.GRT DS     2           ADDRESS OF GROUP RESERVATION TABLE (IF DT.DD)
000.003                 260X UNT.GTS DS     2           GRT SECTOR NUMBER
000.005                 261X UNT.DIS DS     2           DIRECTORY FIRST SECTOR NUMBER
                        262X
000.007                 263X UNT.SIZ EQU    *           SIZE OF UNIT SPECIFIC DATA TABLE PER UNIT
000.007                 264         XTEXT   IOCDEF


                        266X **     I/O CHANNEL DEFINITIONS.
                        267X
000.000                 268X        ORG     0
                        269X
000.000                 270X IOC.LNK DS     2           ADDRESS OF NEXT CHANNEL, =0 IF LAST
000.002                 271X IOC.DDA DS     2           THREAD JUMP TO DEVICE DRIVER (VIA DEV TABLE)
                        272X
000.004                 273X IOC.FLG DS     1           FILE TYPE FLAGS
000.001                 274X FT.DD   EQU    00000001B   =1 IF DIRECTORY DEVICE
000.002                 275X FT.OR   EQU    00000010B   =1 IF OPEN FOR READ
```

```
000.004                276X FT.OW    EQU     00000100B       =1 IF OPEN FOR WRITE
000.010                277X FT.OU    EQU     00001000B       =1 IF OPEN FOR UPDATE
000.003                278X IOC.SQL  EQU     *-IOC.DDA       LENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
                       279X
000.005                280X IOC.GRT  DS      2               ADDRESS OF GROUP RESERVATION TABLE
000.007                281X IOC.SPG  DS      1               SECTORS PER GROUP, THIS DEVICE
000.010                282X IOC.CGN  DS      1               CURRENT GROUP NUMBER
000.011                283X IOC.CSI  DS      1               CURRENT SECTOR INDEX (IN CURRENT GROUP)
000.012                284X IOC.LGN  DS      1               LAST GROUP NUMBER
000.013                285X IOC.LSI  DS      1               LAST SECTOR INDEX (IN LAST GROUP)
000.010                286X IOC.DRL  EQU     *-IOC.FLG       LENGTH OF INFO NORMALLY COPIED BACK TO
                       287X *                                THE CHANNEL TABLE
000.014                288X IOC.DTA  DS      2               DEVICE TABLE ADDRESS FOR THIS DEVICE
000.016                289X IOC.DES  DS      2               SECTOR NUMBER OF DIRECTORY ENTRY
000.020                290X IOC.DEV  DS      2               DEVICE CODE
000.022                291X IOC.UNI  DS      1               UNIT NUMBER (0-9)
000.021                292X IOC.DIL  EQU     *-IOC.DDA       LENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)
                       293X
000.023                294X IOC.DIR  DS      DIRELEN         DIRECTORY ENTRY
                       295X
000.052                296X IOCELEN  EQU     *               IOC ENTRY LENGTH
                       297X
000.001                298X IOCCTD   EQU     1               INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)
000.052                299           XTEXT   DISDEF


                       301X **       DIRECTORY BLOCK FORMAT.
                       302X
000.000                303X          ORG     0
                       304X
000.000                305X DIS.ENT  EQU     *               FIRST ENTRY ADDRESS
000.000                306X          DS      22*DIRELEN      22 DIRECTORY ENTRYS PER BLOCK
001.372                307X          DS      1               0 BYTE = END OF ENTRYS IN THIS BLOCK
                       308X
001.373                309X          ORG     512-5           AT END OF BLOCK
001.373                310X DIS.ENL  DS      1               LENGTH OF EACH ENTRY (=DIRELEN)
001.374                311X DIS.SEC  DS      2               BLOCK # OF THIS BLOCK,
001.376                312X DIS.LNK  DS      2               BLOCK # OF NEXT BLOCK, =0 IF THIS IS LAST
002.000                313           XTEXT   FBDEF


                       315X **       FILE BLOCK DEFINITIONS.
                       316X
000.000                317X          ORG     0
000.000                318X FB.CHA   DS      1               CHANNEL NUMBER
000.001                319X FB.FLG   DS      1               FLAGS
000.002                320X FB.FWA   DS      2               BUFFER FWA
000.004                321X FB.PTR   DS      2               BUFFER POINTER
000.006                322X FB.LIM   DS      2               LIMIT OF DATA IN BUFFER (READ OPERATIONS)
000.010                323X FB.LWA   DS      2               LWA OF BUFFER
000.012                324X FB.NAM   DS      4+8+4+1         NAME OF FILE
000.021                325X FB.NAML  EQU     *-FB.NAM
000.033                326X FBENL    EQU     *               ENTRY LENGTH
```

```
000.033                 327         XTEXT   ECDEF

                        329X **     ERROR CODE DEFINITIONS.
                        330X
000.000                 331X        ORG     0
000.000                 332X        DS      1               NO ERROR #0
000.001                 333X EC.EOF DS      1               END OF FILE
000.002                 334X EC.EOM DS      1               END OF MEDIA
000.003                 335X EC.ILC DS      1               ILLEGAL SYSCALL CODE
000.004                 336X EC.CNA DS      1               CHANNEL NOT AVAILABLE
000.005                 337X EC.DNS DS      1               DEVICE NOT SUITABLE
000.006                 338X EC.IDN DS      1               ILLEGAL DEVICE NAME
000.007                 339X EC.IFN DS      1               ILLEGAL FILE NAME
000.010                 340X EC.NRD DS      1               NO ROOM FOR DEVICE DRIVER
000.011                 341X EC.FNO DS      1               CHANNEL NOT OPEN
000.012                 342X EC.ILR DS      1               ILLEGAL REQUEST
000.013                 343X EC.FUC DS      1               FILE USAGE CONFLICT
000.014                 344X EC.FNF DS      1               FILE NAME NOT FOUND
000.015                 345X EC.UND DS      1               UNKNOWN DEVICE
000.016                 346X EC.ICN DS      1               ILLEGAL CHANNEL NUMBER
000.017                 347X EC.DIF DS      1               DIRECTORY FULL
000.020                 348X EC.IFC DS      1               ILLEGAL FILE CONTENTS
000.021                 349X EC.NEM DS      1               NOT ENOUGH MEMORY
000.022                 350X EC.RF  DS      1               READ FAILURE
000.023                 351X EC.WF  DS      1               WRITE FAILURE
000.024                 352X EC.WPV DS      1               WRITE PROTECTION VIOLATION
000.025                 353X EC.WP  DS      1               DISK WRITE PROTECTED
000.026                 354X EC.FAP DS      1               FILE ALREADY PRESENT
000.027                 355X EC.DDA DS      1               DEVICE DRIVER ABORT
000.030                 356X EC.FL  DS      1               FILE LOCKED
000.031                 357X EC.FAO DS      1               FILE ALREADY OPEN
000.032                 358X EC.IS  DS      1               ILLEGAL SWITCH
000.033                 359X EC.UUN DS      1               UNKNOWN UNIT NUMBER
000.034                 360X EC.FNR DS      1               FILE NAME REQUIRED
000.035                 361X EC.DIW DS      1               DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036                 362X EC.UNA DS      1               UNIT NOT AVAILABLE
000.037                 363X EC.ILV DS      1               ILLEGAL VALUE
000.040                 364X EC.ILO DS      1               ILLEGAL OPTION
000.041                 365X EC.VPM DS      1               VOLUME PRESENTLY MOUNTED ON DEVICE
000.042                 366X EC.NVM DS      1               NO VOLUME PRESENTLY MOUNTED
000.043                 367X EC.FOD DS      1               FILE OPEN ON DEVICE
000.044                 368X EC.NPM DS      1               NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045                 369X EC.DNI DS      1               DISK NOT INITIALIZED
000.046                 370X EC.DNR DS      1               DISK IS NOT READABLE
000.047                 371X EC.DSC DS      1               DISK STRUCTURE IS CORRUPT
000.050                 372X EC.NCV DS      1               NOT CORRECT VERSION OF HDOS
000.051                 373X EC.NOS DS      1               NO OPERATING SYSTEM MOUNTED
000.052                 374X EC.IOI DS      1               ILLEGAL OVERLAY INDEX
000.053                 375X EC.OTL DS      1               OVERLAY TO LARGE
000.054                 376         XTEXT   HOSEQU
```

```
                        378X **     HDOS SYSTEM EQUIVALENCES.
                        379X *
                        380X
024.000                 381X S.GRT0 EQU     24000A          SYSTEM AREA FOR  GRT0
025.000                 382X S.GRT1 EQU     25000A          SYSTEM AREA FOR  GRT1
026.000                 383X S.GRT2 EQU     26000A          SYSTEM AREA FOR  GRT2
                        384X
030.000                 385X ROMBOOT EQU    30000A          ROM BOOT ENTRY
                        386X
040.100                 387X        ORG     40100A          FREE SPACE FROM PAM-8
                        388X
040.100                 389X        DS      8               JUMP TO SYSTEM EXIT
040.110                 390X D.CON  DS      16              DISK CONSTANTS
040.130                 391X SYDD   EQU     *               SYSTEM DISK ENTRY POINT
040.130                 392X D.VEC  DS      24*3            SYSTEM ROM ENTRY VECTORS
040.240                 393X D.RAM  DS      31              SYSTEM ROM WORK AREA
040.277                 394X S.VAL  DS      36              SYSTEM VALUES
040.343                 395X S.INT  DS      115             SYSTEM INTERNAL WORK AREAS
041.126                 396X        DS      16
041.146                 397X S.SOVR DS      2               STACK OVERFLOW WARNING
041.150                 398X        DS      42200A-*        SYSTEM STACK
001.032                 399X STACKL EQU     *-S.SOVR        STACK SIZE
                        400X
042.200                 401X STACK  EQU     *               LWA+1 SYSTEM STACK
042.200                 402X USERFWA EQU    *               USER FWA
042.200                 403         XTEXT   HOSDEF


                        405X **     HOSDEF - DEFINE HOS PARAMETER.
                        406X *
                        407X
                        408X
000.026                 409X VERS   EQU     1*16+6          VERSION 1.6
                        410X
000.377                 411X SYSCALL EQU    377Q            SYSCALL INSTRUCTION
                        412X
                        413X
000.000                 414X        ORG     0
                        415X
                        416X *      RESIDENT FUNCTIONS
                        417X
000.000                 418X .EXIT  DS      1               EXIT (MUST BE FIRST)
000.001                 419X .SCIN  DS      1               SCIN
000.002                 420X .SCOUT DS      1               SCOUT
000.003                 421X .PRINT DS      1               PRINT
000.004                 422X .READ  DS      1               READ
000.005                 423X .WRITE DS      1               WRITE
000.006                 424X .CONSL DS      1               SET/CLEAR CONSOLE OPTIONS
000.007                 425X .CLRCO DS      1               CLEAR CONSOLE BUFFER
000.010                 426X .LOADO DS      1               LOAD AN OVERLAY
000.011                 427X .VERS  DS      1               RETURN HDOS VERSION NUMBER
000.012                 428X .SYSRES DS     1               PRECEDING FUNCTIONS ARE RESIDENT
                        429X
                        430X
                        431X *      *HDOSOVLO.SYS*  FUNCTIONS
```

```
                        432X
000.040                 433X            ORG     40A
                        434X
000.040                 435X .LINK      DS      1               LINK  (MUST BE FIRST)
000.041                 436X .CTLC      DS      1               CTL-C
000.042                 437X .OPENR     DS      1               OPENR
000.043                 438X .OPENW     DS      1               OPENW
000.044                 439X .OPENU     DS      1               OPENU
000.045                 440X .OPENC     DS      1               OPENC
000.046                 441X .CLOSE     DS      1               CLOSE
000.047                 442X .POSIT     DS      1               POSITION
000.050                 443X .DELET     DS      1               DELETE
000.051                 444X .RENAM     DS      1               RENAME
000.052                 445X .SETTP     DS      1               SETTOP
000.053                 446X .DECODE    DS      1               NAME DECODE
000.054                 447X .NAME      DS      1               GET FILE NAME FROM CHANNEL
000.055                 448X .CLEAR     DS      1               CLEAR CHAN
000.056                 449X .CLEARA    DS      1               CLEAR ALL CHANS
000.057                 450X .ERROR     DS      1               LOOKUP ERROR
000.060                 451X .CHFLG     DS      1               CHANGE FLAGS
000.061                 452X .DISMT     DS      1               FLAG SYSTEM DISK DISMOUNTED
000.062                 453X .LOADD     DS      1               LOAD DEVICE DRIVER
                        454X
                        455X
                        456X *          *HDOSOVL1.SYS*  FUNCTIONS
                        457X
000.200                 458X            ORG     200Q
                        459X
000.200                 460X .MOUNT     DS      1               MOUNT  (MUST BE FIRST)
000.201                 461X .DMOUN     DS      1               DISMOUNT
000.202                 462X .MONMS     DS      1               MOUNT/NO MESSAGE
000.203                 463X .DMNMS     DS      1               DISMOUNT/NO MESSAGE
000.204                 464X .RESET     DS      1               RESET = DISMOUNT/MOUNT OF UNIT
000.205                 465             XTEXT   ASCII


                        467X **         ASCII CHARACTER EQUIVALENCES.
                        468X
000.015                 469X CR         EQU     13              CARRIAGE RETURN
000.012                 470X LF         EQU     10              LINE FEED
000.200                 471X NULL       EQU     200Q            PAD CHARACTER
000.000                 472X NUL2       EQU     0
000.007                 473X BELL       EQU     7               BELL CHARACTER
000.177                 474X RUBOUT     EQU     177Q
000.010                 475X BKSP       EQU     10Q             CTL-H
000.026                 476X C.SYN      EQU     26Q             SYNC
000.002                 477X C.STX      EQU     2               STX
000.047                 478X QUOTE      EQU     47Q
000.011                 479X TAB        EQU     11Q
000.033                 480X ESC        EQU     33Q
000.012                 481X NL         EQU     12Q             NEW LINE (HDOS SYSTEMS)
000.212                 482X ENL        EQU     NL+200Q         NL + END-OF-LINE-FLAG
000.014                 483X FF         EQU     14Q             FORM FEED
000.001                 484X CTLA       EQU     01Q             CTL-A
000.002                 485X CTLB       EQU     02Q             CTL-B
000.003                 486X CTLC       EQU     03Q             CTL-C
```

```
000.004                 487X CTLD   EQU     04Q             CTL-D
000.017                 488X CTLO   EQU     17Q             CTL-O
000.020                 489X CTLP   EQU     20Q             CTL-P
000.021                 490X CTLQ   EQU     21Q             CTL-Q
000.023                 491X CTLS   EQU     23Q             CTL-S
000.032                 492X CTLZ   EQU     32Q             CTL-Z
000.205                 493         XTEXT   EDRAM


                        495X **     EDRAM - DISK RAM WORKAREA DEFINITION.
                        496X *
                        497X *      ZEROED UPON BOOTING UP.
                        498X *
                        499X *      HOSEQU MUST BE CHANGED WHEN THIS DECK IS CHANGED.
                        500X
                        501X
040.240                 502X        ORG     D.RAM
                        503X
040.240                 504X D.TT   DS      1               TARGET TRACK (CURRENT OPERATION)
040.241                 505X D.TS   DS      1               TARGET SECTOR (CURRENT OPERATION)
                        506X
040.242                 507X D.DVCTL DS     1               DEVICE CONTROL BYTE
                        508X
040.243                 509X D.DLYMO DS     1               MOTOR ON DELAY COUNT
040.244                 510X D.DLYHS DS     1               HEAD SETTLE DELAY COUNTER
                        511X
040.245                 512X D.TRKPT DS     2               ADDRESS IN D.DRVTB FOR TRACK NUMBER
040.247                 513X D.VOLPT DS     2               ADDRESS IN D.DRVTB FOR VOLUME NUMBER
                        514X
040.251                 515X D.DRVTB DS     2*4             TRACK NUMBER AND VOLUME NUMBER FOR 4 DRIVES
                        516X
040.261                 517X D.HECNT DS     1               HARD ERROR COUNT
040.262                 518X D.SECNT DS     2               SOFT ERROR COUNT
040.264                 519X D.OECNT DS     1               OPERATION ERROR COUNT
                        520X
                        521X *      GLOBAL DISK ERROR COUNTERS
                        522X
040.265                 523X D.ERR  DS      0               BEGINNING OF ERROR BLOCK
040.265                 524X D.E.MDS DS     1               MISSING DATA SYNC
040.266                 525X D.E.HSY DS     1               MISSING HEADER SYNC
040.267                 526X D.E.CHK DS     1               DATA CHECKSUM
040.270                 527X D.E.HCK DS     1               HEADER CHECKSUM
040.271                 528X D.E.VOL DS     1               WRONG VOLUME NUMBER
040.272                 529X D.E.TRK DS     1               BAD TRACK SEEK
040.273                 530X D.ERRL DS      0               LIMIT OF ERROR COUNTERS
                        531X
                        532X *      I/O OPERATION COUNTS
                        533X
040.273                 534X D.OPR  DS      2
040.275                 535X D.OPW  DS      2
                        536X
000.037                 537X D.RAML EQU     *-D.RAM
040.277                 538         XTEXT   ESINT
```

```
                        540X **     S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
                        541X *
                        542X *      THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
                        543X *      MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
                        544X
                        545X
    040.343             546X        ORG     S.INT
                        547X
                        548X **     CONSOLE STATUS FLAGS
                        549X
    040.343             550X S.CDB  DS      1               CONSOLE DESCRIPTOR BYTE
    000.000             551X CDB.H85 EQU    00000000B
    000.001             552X CDB.H84 EQU    00000001B       =0 IF H8-5, =1 IF H8-4
    040.344             553X S.BAUD DS      2               [0-14]  H8-4 BAUD RATE, =0 IF H8-5
                        554X *                              [15]    =1 IF BAUD RATE => 2 STOP BITS
                        555X
                        556X **     TABLE ADDRESS WORDS
                        557X
    040.346             558X S.DLINK DS     2               ADDRESS OF DATA IN HDOS CODE
    040.350             559X S.OFWA DS      2               FWA  OVERLAY  TABLE
    040.352             560X S.CFWA DS      2               FWA  CHANNEL  TABLE
    040.354             561X S.DFWA DS      2               FWA  DEVICE   TABLE
    040.356             562X S.RFWA DS      2               FWA  RESIDENT HDOS CODE
                        563X
                        564X **     DEVICE DRIVER DELAYED LOAD FLAGS
                        565X
    040.360             566X S.DDLDA DS     2               DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
    040.362             567X S.DDLEN DS     2               CODE LENGTH IN BYTES
    040.364             568X S.DDGRP DS     1               GROUP NUMBER FOR DRIVER
    040.365             569X        DS      1               HOLD PLACE
                        570X *S.DDSEC       DS      2               SECTOR NUMBER FOR DRIVER ( * OBSOLETE ! * )
    040.366             571X S.DDDTA DS     2               DEVICE'S ADDRESS IN DEVLST +DEV.RES
    040.370             572X S.DDOPC DS     1               OPEN OPCODE PENDEDING
                        573X
                        574X **     OVERLAY MANAGEMENT FLAGS
                        575X
    000.001             576X OVL.IN EQU     00000001B       IN MEMORY
    000.002             577X OVL.RES EQU    00000010B       PERMINANTLY RESIDENT
    000.014             578X OVL.NUM EQU    00001100B       OVERLAY NUMBER MASK
    000.200             579X OVL.UCS EQU    10000000B       USER CODE SWAPPED FOR OVERLAY
                        580X
    040.371             581X S.OVLFL DS     1               OVERLAY FLAG
    040.372             582X S.UCSF DS      2               FWA SWAPPED USER CODE
    040.374             583X S.UCSL DS      2               LENGTH SWAPPED USER CODE
    040.376             584X S.OVLS DS      2               SIZE OF OVERLAY CODE
    041.000             585X S.OVLE DS      2               ENTRY POINT OF OVERLAY CODE
                        586X
    041.002             587X S.SSN  DS      2               SWAP AREA SECTOR NUMBER
    041.004             588X S.OSN  DS      2               OVERLAY SECTOR NUMBER
                        589X
                        590X *      SYSCALL PROCESSING WORK AREAS
                        591X
    041.006             592X S.CACC DS      1               (ACC) UPON SYSCALL
    041.007             593X S.CODE DS      1               SYSCALL INDEX IN PROGRESS
                        594X
                        595X *      JUMPS TO ROUTINES IN RESIDENT HDOS CODE
```

```
                        596X
041.010                 597X S.JUMPS DS      0              START OF DUMP VECTORS
041.010                 598X S.SDD   DS      3              JUMP TO STAND-IN DEVICE DRIVER
041.013                 599X S.FASER DS      3              JUMP TO FATSERR (FATAL SYSTEM ERROR)
041.016                 600X S.DIREA DS      3              JUMP TO DIREAD (DISK FILE READ)
041.021                 601X S.FCI   DS      3              JUMP TO FCI (FETCH CHANNEL INFO)
041.024                 602X S.SCI   DS      3              JUMP TO SCI (STORE CHANNEL INFO)
041.027                 603X S.GUP   DS      3              JUMP TO GUP (GET UNIT POINTER)
                        604X
041.032                 605X S.MOUNT DS      1              <>0 IF THE SYSTEM DISK IS MOUNTED
041.033                 606X S.DCS   DS      1              DEFAULT CLUSTER SIZE-1
                        607X
041.034                 608X S.BOOTF DS      1              BOOT FLAGS
000.001                 609X BOOT.P  EQU     00000001B      EXECUTE PROLOGUE UPON BOOTUP
                        610X
                        611X *       STACK VALUE SAVED FOR OVERLAY SYSCALLS
                        612X
041.035                 613X S.OVSTK DS      2              VALUE OF SP UPON SYSCALLS USING OVERLAY
                        614X
041.037                 615X         DS      1              RESERVED


                        617X **      ACTIVE I/O AREA.
                        618X *
                        619X *       THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION
                        620X *       CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM
                        621X *       THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.
                        622X *
                        623X *       NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY
                        624X *       FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE
                        625X *       8080 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY
                        626X *       COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND
                        627X *       BACKDATED AFTER PROCESSING.
                        628X
041.040                 629X AIO.VEC DS      3              JUMP INSTRUCTION
041.041                 630X AIO.DDA EQU     *-2            DEVICE DRIVER ADDRESS
041.043                 631X AIO.FLG DS      1              FLAG BYTE
041.044                 632X AIO.GRT DS      2              ADDRESS OF GROUP RESERV TABLE
041.046                 633X AIO.SPG DS      1              SECTORS PER GROUP
041.047                 634X AIO.CGN DS      1              CURRENT GROUP NUMBER
041.050                 635X AIO.CSI DS      1              CURRENT SECTOR INDEX
041.051                 636X AIO.LGN DS      1              LAST GROUP NUMBER
041.052                 637X AIO.LSI DS      1              LAST SECTOR INDEX
041.053                 638X AIO.DTA DS      2              DEVICE TABLE ADDRESS
041.055                 639X AIO.DES DS      2              DIRECTORY SECTOR
041.057                 640X AIO.DEV DS      2              DEVICE CODE
041.061                 641X AIO.UNI DS      1              UNIT NUMBER (0-9)
                        642X
041.062                 643X AIO.DIR DS      DIRELEN        DIRECTORY ENTRY
                        644X
041.111                 645X AIO.CNT DS      1              SECTOR COUNT
041.112                 646X AIO.EOM DS      1              END OF MEDIA FLAG
041.113                 647X AIO.EOF DS      1              END OF FILE FLAG
041.114                 648X AIO.TFP DS      2              TEMP FILE POINTERS
```

```
041.116                649X AIO.CHA DS      2              ADDRESS OF CHANNEL BLOCK (IOC.DDA)


041.120                651X S.SCR   DS      2              SYSTEM SCRATCH AREA ADDRESS
041.122                652          XTEXT   ESVAL


                       654X **      S.VAL - SYSTEM VALUE DEFINTIONS.
                       655X *
                       656X *       THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.
                       657X *
                       658X *       THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.
                       659X
                       660X
040.277                661X         ORG     S.VAL
                       662X
040.277                663X S.DATE  DS      9              SYSTEM DATE (IN ASCII)
040.310                664X S.DATC  DS      2              CODED DATE
040.312                665X S.TIME  DS      4              TIME FROM MIDNIGHT (IN TICS)
040.316                666X S.HIMEM DS      2              HARDWARE HIGH MEMORY ADRESS+1
                       667X
040.320                668X S.SYSM  DS      2              FWA RESIDENT SYSTEM
                       669X
040.322                670X S.USRM  DS      2              LWA USER MEMORY
                       671X
040.324                672X S.OMAX  DS      2              MAX OVERLAY SIZE FOR SYSTEM
                       673X
                       674X
                       675X **      THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL
                       676X
000.200                677X CSL.ECH EQU     10000000B      SUPPRESS ECHO
000.002                678X CSL.WRP EQU     00000010B      WRAP LINES AT WIDTH
000.001                679X CSL.CHR EQU     00000001B      OPERATE IN CHARACTER MODE
                       680X
000.000                681X I.CSLMD EQU     0              S.CSLMD IS FIRST BYTE
040.326                682X S.CSLMD DS      1              CONSOLE MODE
                       683X
000.200                684X CTP.BKS EQU     10000000B      TERMINAL PROCESSES BACKSPACES
000.040                685X CTP.MLI EQU     00100000B      MAP LOWER CASE TO UPPER ON INPUT
000.020                686X CTP.MLO EQU     00010000B      MAP LOWER CASE TO UPPER ON OUTPUT
000.010                687X CTP.2SB EQU     00001000B      TERMINAL NEEDS TWO STOP BITS
000.002                688X CTP.BKM EQU     00000010B      MAP BKSP (UPON INPUT) TO RUBOUT
000.001                689X CTP.TAB EQU     00000001B      TERMINAL SUPPORTS TAB CHARACTERS
                       690X
000.001                691X I.CONTY EQU     1              S.CONTY IS 2ND BYTE
000.000                692X         ERRNZ   *-S.CSLMD-I.CONTY
040.327                693X S.CONTY DS      1              CONSOLE TYPE FLAGS
000.002                694X I.CUSOR EQU     2              S.CUSOR IS 3RD BYTE
000.000                695X         ERRNZ   *-S.CSLMD-I.CUSOR
040.330                696X S.CUSOR DS      1              CURRENT CURSOR POSITION
000.003                697X I.CONWI EQU     3              S.CONWI IS 4TH BYTE
000.000                698X         ERRNZ   *-S.CSLMD-I.CONWI
```

```
040.331                 699X S.CONWI DS      1              CONSOLE WIDTH
                        700X
000.001                 701X CO.FLG  EQU     00000001B      CTL-O FLAG
000.200                 702X CS.FLG  EQU     10000000B      CTL-S FLAG
                        703X
000.004                 704X I.CONFL EQU     4              S.CONFL IS 5TH BYTE
000.000                 705X         ERRNZ   *-S.CSLMD-I.CONFL
040.332                 706X S.CONFL DS      1              CONSOLE FLAGS
                        707X
040.333                 708X S.CAADR DS      2              ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040.335                 709X S.CCTAB DS      6              ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
040.343                 710          XTEXT   DDDEF


                        712X **      DEVICE DRIVER COMMUNICATION FLAGS.
                        713X *
                        714X
000.000                 715X         ORG     0
                        716X
000.000                 717X DC.REA  DS      1              READ
000.001                 718X DC.WRI  DS      1              WRITE
000.002                 719X DC.RER  DS      1              READ REGARDLESS
000.003                 720X DC.OPR  DS      1              OPEN FOR READ
000.004                 721X DC.OPW  DS      1              OPEN FOR WRITE
000.005                 722X DC.OPU  DS      1              OPEN FOR UPDATE
000.006                 723X DC.CLO  DS      1              CLOSE
000.007                 724X DC.ABT  DS      1              ABORT
000.010                 725X DC.MOU  DS      1              MOUNT DEVICE
000.011                 726X DC.LOD  DS      1              LOAD DEVICE DRIVER
000.012                 727X DC.MAX  DS      1              MAXIMUM ENTRY INDEX
000.013                 728          XTEXT   MTR
```

```
                        731X **     MTR - PAM/8 EQUIVALENCES.
                        732X *
                        733X *      THIS DECK CONTAINS SYMBOLIC DEFINITIONS USED TO
                        734X *      MAKE USE OF THE PAM/8 CODE AND CONTROL BYTES.


                        736X **     IO PORTS
                        737X
000.360                 738X IP.PAD EQU     360Q        PAD INPUT PORT
000.360                 739X OP.CTL EQU     360Q        CONTROL OUTPUT PORT
000.360                 740X OP.DIG EQU     360Q        DIGIT SELECT OUTPUT PORT
000.361                 741X OP.SEG EQU     361Q        SEGMENT SELECT OUTPUT PORT


                        743X **     FRONT PANEL CONTROL BITS.
                        744X
000.020                 745X CB.SSI EQU     00010000B   SINGLE STEP INTERRUPT
000.040                 746X CB.MTL EQU     00100000B   MONITOR LIGHT
000.100                 747X CB.CLI EQU     01000000B   CLOCK INTERRUPT ENABLE
000.200                 748X CB.SPK EQU     10000000B   SPEAKER ENABLE


                        750X **     MONITOR MODE FLAGS.
                        751X
000.000                 752X DM.MR  EQU     0           MEMORY READ
000.001                 753X DM.MW  EQU     1           MEMORY WRITE
000.002                 754X DM.RR  EQU     2           REGISTER READ
000.003                 755X DM.RW  EQU     3           REGISTER WRITE


                        757X **     USER OPTION BITS.
                        758X *
                        759X *      THESE BITS ARE SET IN CELL .MFLAG.
                        760X
000.200                 761X UO.HLT EQU     10000000B   DISABLE HALT PROCESSING
000.100                 762X UO.NFR EQU     CB.CLI      NO REFRESH OF FRONT PANEL
000.002                 763X UO.DDU EQU     00000010B   DISABLE DISPLAY UPDATE
000.001                 764X UO.CLK EQU     00000001B   ALLOW PRIVATE INTERRUPT PROCESSING


                        766X **     MONITOR IDENTIFICATION FLAGS
                        767X *
                        768X *      THESE BYTES IDENTIFY THE ROM MONITOR.
                        769X *      THEY ARE THE VARIOUS VALUES OF LOCATION .IDENT
                        770X
000.021                 771X M.PAM8 EQU     021Q        'LXI' INSTRUCTION AT 000.000 IN PAM-8
000.303                 772X M.FOX  EQU     303Q        'JMP' INSTRUCTION AT 000.000 IN FOX ROM
```

```
                        774X **      ROUTINE ENTRY POINTS.
                        775X *
                        776X
000.000                 777X .IDENT  EQU     0000A          IDENTIFICATION LOCATION
000.053                 778X .DLY    EQU     0053A          DELAY
001.267                 779X .LOAD   EQU     1267A          TAPE LOAD
001.374                 780X .DUMP   EQU     1374A          TAPE DUMP
002.136                 781X .ALARM  EQU     2136A          ALARM ROUTINE
002.140                 782X .HORN   EQU     2140A          HORN
002.172                 783X .CTC    EQU     2172A          CHECK TAPE CHECKSUM
002.205                 784X .TPERR  EQU     2205A          TAPE ERROR ROUTINE
002.264                 785X .PCHL   EQU     2264A          PCHL INSTRUCTION
002.265                 786X .SRS    EQU     2265A          SCAN RECORD START
002.325                 787X .RNP    EQU     2325A          READ NEXT PAIR
002.331                 788X .RNB    EQU     2331A          READ NEXT BYTE
002.347                 789X .CRC    EQU     2347A          CRC-16 CALCULATOR
003.017                 790X .WNP    EQU     3017A          WRITE NEXT PAIR
003.024                 791X .WNB    EQU     3024A          WRITE NEXT BYTE
003.122                 792X .DOD    EQU     3122A          DECODE FOR OCTAL DISPLAY
003.260                 793X .RCK    EQU     3260A          READ CONSOLE KEYSET
003.356                 794X .DODA   EQU     3356A          SEGMENT CODE TABLE

                        796X **      RAM CELLS USED BY H8MTR.
                        797X *
                        798X
040.000                 799X .START  EQU     40000A         START DUMP ADDRESS
040.002                 800X .IOWRK  EQU     40002A         IN OR OUT INSTRUCTION
040.005                 801X .REGI   EQU     40005A         DISPLAYED REGISTER INDEX
040.006                 802X .DSPROT EQU     40006A         PERIOD FLAG BYTE
040.007                 803X .DSPMOD EQU     40007A         DISPLAY MODE
040.010                 804X .MFLAG  EQU     40010A         USER OPTION BYTE
040.011                 805X .CTLFLG EQU     40011A         PANEL CONTROL BYTE
040.013                 806X .ALEDS  EQU     40013A         ABUSS LEDS
040.021                 807X .DLEDS  EQU     40021A         DBUSS LEDS
040.024                 808X .ABUSS  EQU     40024A         ABUSS REGISTER
040.027                 809X .CRCSUM EQU     40027A         CRCSUM WORD
040.031                 810X .TPERRX EQU     40031A         TAPE ERROR EXIT VECTOR
040.033                 811X .TICCNT EQU     40033A         CLOCK TICK COUNTER
040.035                 812X .REGPTR EQU     40035A         REGISTER POINTER
040.037                 813X .UIVEC  EQU     40037A         USER INTERRUPT VECTORS
000.013                 814          XTEXT   DDFDEF

                        816X **      DIRECTORY DEVICE FORMAT DEFINITION.
                        817X *
                        818X
                        819X
000.002                 820X HOS.SPG EQU     2              2 SECTORS PER GROUP REQUIRED FOR NOW
                        821X
000.000                 822X         ORG     0
000.000                 823X DDF.BOO DS      9              2K BOOT PROGRAM
000.011                 824X DDF.BOL EQU     *              LENGTH OF BOOT
000.011                 825X DDF.LAB DS      1              LABEL SECTOR
```

```
000.012                826X DDF.RGT DS      2              RESERVED GROUP TABLE
000.014                827X DDF.USR DS      0              BEGINNING OF OPEN SPACE
000.014                828          XTEXT   LABDEF


                       830X **      DISK LABEL SECTOR FORMATS.
                       831X
000.000                832X         ORG     0
000.000                833X LAB.SER DS      1              SERIAL NUMBER OF VOLUME
000.001                834X LAB.IND DS      2              INITIALIZATION DATE
000.003                835X LAB.DIS DS      2              SECTOR NUMBER OF 1ST DIRECTORY SECTOR
000.005                836X LAB.GRT DS      2              INDEX OF GRT SECTOR
000.007                837X LAB.SPG DS      1              SECTORS PER GROUP
                       838X
000.000                839X LAB.DAT EQU     0              DATA VOLUME ONLY
000.001                840X LAB.SYS EQU     1              SYSTEM VOLUME
000.002                841X LAB.NOD EQU     2              => LAB.NOD MEANS VOLUME HAS NO DIRECTORY
                       842X
000.010                843X LAB.VLT DS      1              VOLUME TYPE
000.011                844X LAB.VER DS      1              VERSION OF INIT17 THAT INITED DISK
000.012                845X         DS      7              UNUSED
000.021                846X LAB.LAB DS      60             LABEL
000.074                847X LAB.LBL EQU     *-LAB.LAB      LABEL LENGTH
000.115                848          XTEXT   FILDEF


                       850X **      FILDEF - FILE TYPE DEFINITIONS.
                       851X *
                       852X *       DB      377Q,FT.XXX
                       853X
                       854X
000.000                855X FT.ABS  EQU     0              ABSOLUTE BINARY
000.001                856X FT.PIC  EQU     1              POSITION INDEPENDANT CODE
000.002                857X FT.REL  EQU     2              RELOCATABLE CODE
000.003                858X FT.BAC  EQU     3              COMPILED BASIC CODE
000.115                859          XTEXT   ABSDEF


                       861X **      ABS FORMAT EQUIVALENCES.
                       862X
000.000                863X         ORG     0
                       864X
000.000                865X ABS.ID  DS      1              377Q = BINARY FILE FLAG
000.001                866X         DS      1              FILE TYPE (FT.ABS)
000.002                867X ABS.LDA DS      2              LOAD ADDRESS
000.004                868X ABS.LEN DS      2              LENGTH OF ENTIRE RECORD
000.006                869X ABS.ENT DS      2              ENTRY POINT
                       870X
000.010                871X ABS.COD DS      0              CODE STARTS HERE
```

```
042.170                    874          ORG     USERFWA-ABS.COD
042.170  377 000           875          DB      377Q,FT.ABS
042.172  200 042           876          DW      USERFWA          LOAD ADDRESS
042.174  050 022           877          DW      MEML-USERFWA     SIZE
042.176  310 062           878          DW      ENTRY            ENTRY
                           879
042.200                    880  PIP     EQU     *
                           881
                           882  *       COMMAND INTERPRETATION COMES HERE
                           883
042.200                    884  RESTART EQU     *
                           885
042.200  072 220 062       886          LDA     MODE
042.203  247               887          ANA     A
042.204  302 337 042       888          JNZ     EXIT             ENTERED WITH COMMAND, WILL NOW EXIT
042.207  061 200 042       889  START   LXI     SP,STACK         CLEAN STACK
042.212  315 220 042       890          CALL    PIP1             EXECUTE COMMAND
                           891
                           892  *       COMMANDS EXIT HERE IF NO ERRORS FOUND
                           893
042.215  303 200 042       894          JMP     RESTART
                           895
                           896  *       GET READY TO PROCESS COMMAND
                           897
042.220  315 277 056       898  PIP1    CALL    SDD              SET DEFAULT DEFAULT
                           899
                           900  *       CLEAR CHANNELS AND FILE BUFFER
                           901
042.223  377 056           902          DB      SYSCALL,.CLEARA CLEAR CHANNELS
042.225  257               903          XRA     A
042.226  062 250 062       904          STA     DESTFB+FB.FLG    FLAG FILE NOT OPEN
                           905
                           906  *       CLEAR DYNAMIC BUFFERS
                           907
042.231  041 000 000       908          LXI     H,0
042.234  042 245 062       909          SHLD    BUFSIZ           EMPTY BUFFER
042.237  042 302 062       910          SHLD    NAMTLEN          CLEAR NAMTAB
042.242  042 304 062       911          SHLD    NAMTMAX          CLEAR NAMTAB AREA
042.245  041 132 063       912          LXI     H,BUFF
042.250  042 243 062       913          SHLD    BUFPTR           SET BUFFER AGAINST END OF NAMTAB
                           914
                           915  *       INPUT COMMAND LINE
                           916
042.253  315 027 057       917          CALL    $CCO             CLEAR CONTROL-O
042.256  072 220 062       918          LDA     MODE
042.261  247               919          ANA     A
042.262  314 173 043       920          CZ      ACL              ACCEPT COMMAND LINE (UNLESS WAS PASSED ONE BY CALLER)
042.265  332 337 042       921          JC      EXIT             EOF
042.270  041 012 063       922          LXI     H,LINE           (HL) = COMMAND ADDRESS
042.273  021 354 042       923          LXI     D,PIPA           (DE) = SWITCH LIST
000.000                    924          ERRNZ   I.COP
042.276  257               925          XRA     A                (A) = #I.COP
042.277  062 217 062       926          STA     COMAND           ASSUME COPY COMMAND
042.302  062 222 062       927          STA     SUPRES           CLEAR /SUP FLAG
042.305  074               928          INR     A                FLAG NO /S FLAG
042.306  062 223 062       929          STA     SYSTEM           CLEAR /S FLAG
```

```
042.311  315 317 060   930          CALL    $DRS            DETECT AND REMOVE SWITCHES
042.314  332 275 052   931          JC      ERROR           ERROR
042.317  072 217 062   932          LDA     COMAND
042.322  315 061 031   933          CALL    $TJMP           PROCESS COMMAND
```

```
                                    935  **      COMMAND LIST
                                    936
042.325                             937  PIPB    DS      0               COMMAND PROCESSOR TABLE
000.000                             938  I.COP   EQU     *-PIPB/2                COMMAND INDEX
042.325  254 043                    939          DW      COPY
000.001                             940  I.LIS   EQU     *-PIPB/2                COMMAND INDEX
042.327  371 046                    941          DW      LIST
000.002                             942  I.BRE   EQU     *-PIPB/2                COMMAND INDEX
042.331  377 046                    943          DW      BRIEF           /BR
000.003                             944  I.VER   EQU     *-PIPB/2                COMMAND INDEX
042.333  365 051                    945          DW      VERSN           /V
000.004                             946  I.MOU   EQU     *-PIPB/2        /MOU,/M
042.335  217 043                    947          DW      MOUNT
000.001                             948          IF      .PIP.
                                    949  I.DEL   EQU     *-PIPB/2
                                    950          DW      DELETE          /DEL
                                    951  I.REN   EQU     *-PIPB/2
                                    952          DW      RENAME          /RE
                                    953  I.DIS   EQU     *-PIPB/2
                                    954          DW      DISMOU          /DIS
                                    955  I.RES   EQU     *-PIPB/2
                                    956          DW      RESET           /RES
                                    957          ENDIF
                                    958
                                    959  *       CTL-D HIT
                                    960
042.337  257                        961  EXIT    XRA     A
042.340  377 000                    962          DB      SYSCALL,.EXIT   EXIT


                                    964  **      CCHIT - CTL-C HIT
                                    965  *
                                    966  *       ENTRY   FROM SYSTEM
                                    967
                                    968
042.342  315 136 031                969  CCHIT   CALL    $TYPTX
042.345  136 303                    970          DB      '^','C'+200Q
042.347  377 007                    971          DB      SYSCALL,.CLRCO  CLEAR CONSOLE TYPEAHEAD
042.351  303 200 042                972          JMP     RESTART         GET NEW COMMAND
```

```
                                    975  ***     SWITCH PROCESSING TABLES AND ROUTINES.
                                    976  *
                                    977  *       COMMAND SWITCHES ARE PROCESSED VIA THE ROUTINE $DRS, 'DECODE AND
                                    978  *       REMOVE SWITCHES', $DRS IS SUPPLIED WITH A SWITCH DESCRIPTION
                                    979  *       TABLE, WHICH CONTAINS THE ADDRESSES OF ROUTINES
                                    980  *       WHICH ARE ENVOKED WHEN THE SWITCHES ARE ENCOUNTERED.
                                    981
                                    982
                                    983  **      SWITCH TABLE
                                    984
042.354                             985  PIPA    DS      0               FWA SWITCH TABLE
000.001                             986          IF      .PIP.
                                    987          DB      'DEL'           /DELETE
                                    988          DB      'E'+200Q,'T'+200Q,'E'+200Q,200Q
                                    989          DW      SW.DEL          PROCESSING ROUTINES
                                    990
                                    991          DB      'R'             /RENAME
                                    992          DB      'E'+200Q,'N'+200Q,'A'+200Q,'M'+200Q,'E'+200Q,200Q
                                    993          DW      SW.REN          PROCESS RENAME
                                    994
                                    995          DB      'DIS'           /DISMOUNT
                                    996          DB      'M'+200Q,'O'+200Q,'U'+200Q,'N'+200Q,'T'+200Q,200Q
                                    997          DW      SW.DIS
                                    998
                                    999          DB      'RES'           /RESET
                                   1000          DB      'E'+200Q,'T'+200Q,200Q
                                   1001          DW      SW.RES
                                   1002          ENDIF
                                   1003
042.354  114                       1004          DB      'L'             /LIST
042.355  311 323 324               1005          DB      'I'+200Q,'S'+200Q,'T'+200Q,200Q
042.361  140 043                   1006          DW      SW.LIS          PROCESS LIST
                                   1007
042.363  102                       1008          DB      'B'             /BRIEF
042.364  322 311 305               1009          DB      'R'+200Q,'I'+200Q,'E'+200Q,'F'+200Q,200Q
042.371  115 043                   1010          DW      SW.BRE          PROCESS BRIEF
                                   1011
042.373  126                       1012          DB      'V'             /VERSION
042.374  305 322 323               1013          DB      'E'+200Q,'R'+200Q,'S'+200Q,'I'+200Q,'O'+200Q,'N'+200Q,200Q
043.003  161 043                   1014          DW      SW.VER          PROCESS VERSION
                                   1015
043.005  115 117 125               1016          DB      'MOU'           /MOUNT
043.010  316 324 200               1017          DB      'N'+200Q,'T'+200Q,200Q
043.013  166 043                   1018          DW      SW.MOU
                                   1019
043.015  123                       1020          DB      'S'             /SYSTEM
043.016  331 323 324               1021          DB      'Y'+200Q,'S'+200Q,'T'+200Q,'E'+200Q,'M'+200Q,200Q
043.024  065 043                   1022          DW      SW.SYS          PROCESS SYSTEM
                                   1023
043.026  123 125                   1024          DB      'SU'            /SUPRESS
043.030  320 322 305               1025          DB      'P'+200Q,'R'+200Q,'E'+200Q,'S'+200Q,'S'+200Q,200Q
043.036  072 043                   1026          DW      SW.SUP
                                   1027
043.040  112 107 114               1028          DB      'JGL'           /JGL INTERNAL SWITCH
043.043  200                       1029          DB      200Q
043.044  100 043                   1030          DW      SW.JGL
```

```
                    1031
043.046 000         1032        DB      0           END OF TABLE
```

```
000.001                 1034            IF      .PIP.
                        1035    SW.DEL  SPACE   3,10
                        1036    **      SW.DEL - /DELETE SWITCH DETECTED.
                        1037
                        1038    SW.DEL  MVI     A,I.DEL
                        1039            JMP     SWIT1           IS MAJOR FUNCTION
                        1040    SW.REN  SPACE   3,10
                        1041    **      SW.REN - /RENAME SWITCH DETECTED.
                        1042
                        1043    SW.REN  MVI     A,I.REN
                        1044            JMP     SWIT1           IS MAJOR FUNCTION
                        1045    SW.DIS  SPACE   3,10
                        1046    **      SW.DIS  -  /DISMOUNT SWITCH DETECTED
                        1047
                        1048    SW.DIS  MVI     A,I.DIS
                        1049            JMP     SWIT1           IS MAJOR FUNCTION
                        1050    SW.RES  SPACE   3,10
                        1051    **      SW.RES  -  /RESET SWITCH DETECTED.
                        1052
                        1053    SW.RES  MVI     A,I.RES
                        1054            JMP     SWIT1           IS MAJOR FUNCTION
                        1055            ENDIF


                        1057    *       SWIT1 - PROCESS MAJOR FUNCTION SWITCH.
                        1058    *
                        1059    *       SWIT1 IS ENTERED TO PROCESS SWITCHES WHICH DETERMINE THE FUNCTION
                        1060    *       PIP IS TO PERFORM. I.E. 'VERB' SWITCHES, SUCH
                        1061    *       AS /DELETE (AS OPOSED TO 'MODIFIER' SWITCHES, LIKE /SYSTEM)
                        1062
043.047 001 217 062     1063    SWIT1   LXI     B,COMAND
043.052 365             1064            PUSH    PSW             SAVE COMMAND
043.053 012             1065            LDAX    B               (A) = PREVIOUS COMMAND
043.054 247             1066            ANA     A
043.055 076 204         1067            MVI     A,PEC.CS        CONTRADICTORY SWITCHES
043.057 302 275 052     1068            JNZ     ERROR           IF SO
043.062 361             1069            POP     PSW             (A) = NEW CODE
043.063 002             1070            STAX    B               STORE IT
043.064 311             1071            RET


                        1073    **      SW.SYS - /SYSTEM SWITCH DETECTED.
                        1074
043.065 257             1075    SW.SYS  XRA     A               SET /S FLAG
043.066 062 223 062     1076            STA     SYSTEM
043.071 311             1077            RET
```

```
                                1079  **      SW.SUP - /SUPPRESS SWITCH.
                                1080
                                1081
043.072  076 001               1082  SW.SUP  MVI     A,1
043.074  062 222 062           1083          STA     SUPRES
043.077  311                   1084          RET


                                1086  **      SW.JGL - /JGL SYSTEM SWITCH.
                                1087
                                1088
043.100  076 001               1089  SW.JGL  MVI     A,1
043.102  062 221 062           1090          STA     JGL
043.105  076 103               1091          MVI     A,'C'
043.107  062 357 051           1092          STA     PFIB1           SET 'C' CHARACTER FOR FLAGS DISPLAY
043.112  303 065 043           1093          JMP     SW.SYS


                                1095  **      SW.BRE - /BRIEF SWITCH DETECTED.
                                1096
043.115  072 217 062           1097  SW.BRE  LDA     COMAND          ALLOW TO SUPERCEDE /LIST
043.120  247                   1098          ANA     A
043.121  312 132 043           1099          JZ      SW.BRE1         NO OTHER COMMAND
000.000                         1100          ERRNZ   I.LIS-1
043.124  075                   1101          DCR     A
043.125  076 204               1102          MVI     A,PEC.CS        ASSUME CONTRADICTORY SWITCHES
043.127  302 275 052           1103          JNZ     ERROR
043.132  076 002               1104  SW.BRE1 MVI     A,I.BRE         IS /BREIF
043.134  062 217 062           1105          STA     COMAND
043.137  311                   1106          RET


                                1108  **      SW.LST - /LIST SWITCH DETECTED.
                                1109
043.140  072 217 062           1110  SW.LIS  LDA     COMAND
043.143  247                   1111          ANA     A
043.144  312 153 043           1112          JZ      SW.LIS1         NO FUNCTION
000.000                         1113          ERRNZ   I.BRE-2
000.000                         1114          ERRNZ   I.LIS-1
043.147  326 003               1115          SUI     3
043.151  077                   1116          CMC
043.152  320                   1117          RNC                     ALREADY HAVE ONE SPECIFIED, I.BRE OVERRULES
043.153  076 001               1118  SW.LIS1 MVI     A,I.LIS         /LIST
043.155  062 217 062           1119          STA     COMAND
043.160  311                   1120          RET
```

```
                                1122  **      SW.VER  - /VERSION SWITCH DETECTED
                                1123
        043.161  076 003        1124  SW.VER  MVI     A,I.VER
        043.163  303 047 043    1125          JMP     SWIT1


                                1127  **      SW.MOU  - /MOUNT SWITCH DETECTED
                                1128
        043.166  076 004        1129  SW.MOU  MVI     A,I.MOU
        043.170  303 047 043    1130          JMP     SWIT1
```

```
                                    1134  ***     ACL - ACCEPT COMMAND LINE.
                                    1135  *
                                    1136  *       ACL PROMPTS FOR AND READS A COMMAND LINE FROM
                                    1137  *       THE CONSOLE.
                                    1138  *
                                    1139  *       ENTRY   NONE
                                    1140  *       EXIT    'C' CLEAR, GOT LINE
                                    1141  *                'LINE' = COMMAND LINE
                                    1142  *                'C' SET IF EOF
                                    1143  *       USES    ALL
                                    1144
                                    1145
043.173  315 044 057               1146  ACL     CALL    $GNL            GUARANTEE NEW LINE
043,176  315 136 031               1147          CALL    $TYPTX
000.001                             1148          IF      .PIP.
                                    1149          DB      ':P',':'+200Q
                                    1150          ELSE    ONECOPY
043.201  072 117 103               1151          DB      ':OC',':'+200Q
                                    1152          ENDIF
043.205  257                       1153          XRA     A
043.206  062 326 040               1154          STA     S.CSLMD         CLEAR SPECIAL MODES
043,211  041 012 063               1155          LXI     H,LINE
043.214  303 111 057               1156          JMP     $RTL.           READ UPPER CASE LINE AND EXIT
```

```
000.001  1159          IF      .PIP.           PIP USES 'COPY'
         1160  ***     COPY - PROCESS COPY COMMAND.
         1161  *
         1162  *       SYNTAX:
         1163  *
         1164  *       DEST=SOURCE1,...,SOURCEN
         1165  *
         1166  *       D'DEST' IS THE DESTINATION FILE DESIGNATOR. IF NULL
         1167  *       (IN WHICH CASE THE '=' MAY BE OMITTED) IT DEFAULTS  TO
         1168  *               KB:PIPDEST.JGL
         1169  *
         1170  *       THE 'SOURCE' FIELDS ARE THE SOURCE FILE DESIGNATORS. WILDCARDS
         1171  *       MAY BE USED FOR FILE NAME AND EXTENSION.
         1172  *       IF NO WILDCARDS ARE USED IN THE DESTINATION, MULTIPLE SOURCE FILES
         1173  *       ARE CONCATINATED TOGETHER.
         1174  *
         1175  *       IF WILDCARDS ARE PRESENT IN THE DESTINATION FILE DESCRIPTION,
         1176  *       THE SOURCE FILES ARE COPIED TO INDIVIDUAL OUTPUT FILES. THE
         1177  *       NAMES OF THE OUTPUT FILES ARE CREATED BY FILLING
         1178  *       THE 'WILD' SPOTS IN THE DESTINATION NAME WITH THE CORRESPONDING
         1179  *       CHARACTERS IN THE SOURCE NAME.
         1180
         1181
         1182  COPY    EQU     *
         1183          XRA     A
         1184          STA     COPYC           CLEAR FILE COUNT
         1185          CALL    DDF             DECODE DESTINATION FILE
         1186          JC      ERROR           ERROR
         1187          STA     COPYA           SAVE DESTIONATION TYPE
         1188          CALL    SDD             RESET DEFAULT DEFAULTS
         1189          XRA     A               ALLOW *.*
         1190          CALL    BSL             BUILD SOURCE FILE LIST
         1191          JC      ERROR
         1192          CALL    $MOVEL
         1193          DW      COPYDL
         1194          DW      DESTFB+FB.NAM
         1195          DW      COPYD           SAVE WILDCARD DESTINATION
         1196
         1197  *       HAVE DESTINATION AND SOURCE FILE NAMES. DO THE COPYING.
         1198  *
         1199  *       IF NO DESTINATION WILD CARDS, THUS COPIING TO A SINGLE OUTPUT
         1200  *       FILE, OPEN THAT FILE NOW.
         1201
         1202          LDA     COPYA
         1203          ANA     A
         1204          JZ      COPY1           IS WILDCARDED
         1205          LXI     H,DESTFB+FB.NAM
         1206          MVI     A,CN.DES        (A) = DESTINATION CHANNEL
         1207          DB      SYSCALL,.OPENW          OPEN IT
         1208          LXI     H,DESTFB
         1209          JC      $FERROR         IF ERROR
         1210
         1211  *       OPEN NEXT SOURCE FILE
         1212
         1213  COPY1   LHLD    NAMTLEN
         1214          MOV     A,H
```

```
1215            ORA     L
1216            JZ      COPY5           NO MORE INPUT FILES
1217            LXI     H,COPYC
1218            INR     M               COUNT FILE
1219            LXI     H,NAMTAB        (HL) = NAME ADDRESS
1220            MVI     A,CN.SOU        SOURCE CHANNEL
1221            DB      SYSCALL,.OPENR  OPEN FOR READ
1222            JC      NAMERR          IF ERROR
1223
1224  *         OPEN DESTINATION FILE IFF WILDCARDS
1225
1226            LDA     COPYA
1227            ANA     A
1228            JNZ     COPY2           NOT WILDCARDS
1229            LXI     B,COPYD         (BC) = WILDCARD PATTERN ADDRESS
1230            LXI     D,NAMTAB        (DE) = SOURCE NAME
1231            LXI     H,DESTFB+FB.NAM (HL) = RESULT AREA
1232            PUSH    H               SAVE POINTER TO RESULT AREA
1233            CALL    MWN             MERGE WILDCARD NAME
1234            POP     H               (HL) = $DESTFB+FB.NAM
1235            MVI     A,CN.DES
1236            DB      SYSCALL,.OPENW
1237            LXI     H,DESTFB
1238            JC      $FERROR         CANT GET FILE OPEN
1239
1240  *         INPUT AND OUTPUT FILES OPEN, COPY
1241
1242  COPY2     CALL    EBM             EXPAND BUFFER TO MAX SIZE
1243  COPY3     LHLD    BUFSIZ
1244            MOV     B,H
1245            MOV     C,L             (BC) = LENGTH OF BUFFER
1246            LHLD    BUFPTR
1247            XCHG                    (DE) = BUFFER FWA
1248            MVI     A,CN.SOU
1249            PUSH    D
1250            DB      SYSCALL,.READ
1251            POP     D               (DE) = BUFFER FWA
1252            PUSH    PSW
1253            JNC     COPY4           GOT IT ALL
1254            CPI     EC.EOF
1255            JE      COPY4           IS EOF
1256            POP     PSW             RESTORE ERROR CODE
1257            JMP     NAMERR
1258
1259  COPY4     LDA     BUFSIZ+1        (A) = # OF SECTORS IN BUFFER
1260            SUB     B
1261            MOV     B,A             (B) = SECTORS READ
1262            MVI     C,0
1263            MVI     A,CN.DES
1264            DB      SYSCALL,.WRITE  WRITE IT OUT
1265            LXI     H,DESTFB
1266            JC      $FERROR         ERROR ON WRITE
1267            POP     PSW             (PSW) = STATUS FROM READ
1268            JNC     COPY3           NOT EOF
1269            CALL    SBE             SHRINK BUFFER TO MINIMUM SIZE
1270            MVI     A,CN.SOU
```

```
1271            DB      SYSCALL,.CLOSE  CLOSE SOURCE
1272            JC      NAMERR          ERROR ON CLOSE
1273            CALL    REN             REMOVE ENTRY FROM NAMTAB
1274
1275    *       IF DOING INDIVIDUAL FILE COPIES, CLOSE OUTPUT FILE.
1276
1277            LDA     COPYA
1278            ANA     A
1279            JNZ     COPY1           CONCATINATING
1280            MVI     A,CN.DES
1281            DB      SYSCALL,.CLOSE  CLOSE DESTINATION
1282            LXI     H,DESTFB
1283            JC      $FERROR         ERROR ON CLOSE
1284            JMP     COPY1           GET NEXT FILE
1285
1286    **      ALL COPIES COMPLETE. CLOSE FILES AND CLEAN UP
1287
1288    COPY5   LDA     COPYC
1289            ANA     A
1290            JNZ     COPY6
1291
1292    *       NO FILES COPIED
1293
1294            CALL    $TYPTX
1295            DB      BELL,'No Files Copied',ENL
1296            MVI     A,CN.DES
1297            DB      SYSCALL,.CLEAR  CLEAR CHANNEL
1298            RET
1299
1300    COPY6   MVI     B,0             (BC) = COUNT OF FILES COPIED
1301            MOV     C,A
1302            LDA     COPYA
1303            ANA     A
1304            JZ      COPY7           WILDCARDED
1305            PUSH    B               SAVE COUNT
1306            MVI     A,CN.DES
1307            DB      SYSCALL,.CLOSE  CLOSE DESTINATION
1308            POP     B               (BC) = FILES COPIED COUNT
1309            LXI     H,DESTFB
1310            JC      $FERROR         ERROR ON CLOSE
1311
1312    *       TYPE FILE COUNT
1313
1314    COPY7   LDA     SUPRES
1315            ANA     A
1316            RNZ                     SUPPRESS TRAIL MESSAGE
1317            MVI     A,3
1318            LXI     H,COPYE
1319            CALL    $UDDN           UNPACK COUNT INTO MESSAGE
1320            CALL    $TYPTX
1321            DB      NL
1322    COPYE   DB      'XXX'
1323            DB      ' Files Copied',ENL
1324            RET
1325
1326    COPYA   DB      0               DESTINATION FILE WILDCARD FLAG (=0 IF WC)
```

```
1327  COPYC   DB      0               FILES COPIED COUNT
1328  COPYD   DS      FB.NAML         HOLD AREA FOR WILDCARD DESTINATION
1329  COPYDL  EQU     *-COPYD
1330          STL     'MOUNT  -  MOUNT A NEW DISK'
1331          EJECT
1332  ***     MOUNT   -  MOUNT A NEW DISK
1333  *
1334  *       MOUNT MOUNTS A NEW DISK ON THE SPECIFIED UNIT OF THE SELECTED
1335  *       DEVICE.
1336  *
1337  *       DEV:/MOUNT]
1338  *
1339
1340  MOUNT   EQU     *
1341          MVI     A,.MOUNT
1342          CALL    MDR.            MOUNT/DISMOUNT/RESET
1343          RET
1344          STL     'DISMOU -  DISMOUNT CURRENT DISK'
1345          EJECT
1346  DISMOU  SPACE   4,10
1347  ***     DISMOU  -  DISMOUNT CURRENT DISK
1348  *
1349  *       DISMOU DISMOUNTS THE CURRENT DISK ON THE SPECIFIED UNIT OF THE
1350  *       SELECTED DEVICE.
1351  *
1352  *       DEV:/DISMOUNT]
1353  *
1354
1355  DISMOU  EQU     *
1356          MVI     A,.DMOUN
1357          CALL    MDR.            MOUNT/DISMOUNT/RESET
1358          RET
1359          STL     'RESET  -  RESET CURRENT DISK'
1360          EJECT
1361  RESET   SPACE   4,10
1362  ***     RESET   -  RESET THE CURRENT DISK
1363  *
1364  *       RESET RESETS THE SPECIFIED UNIT OF THE SELECTED DEVICE BY ISSUING
1365  *       THE HDOS RESET CALL, WHICH IN TURN ISSUES A DISMOUNT AND MOUNT
1366  *       ASKING THE USER TO OPEN THE DRIVE IN BETWEEN THE TWO.
1367  *
1368  *       DEV:/RESET]
1369  *
1370
1371  RESET   EQU     *
1372          MVI     A,.RESET
1373          CALL    MDR.            MOUNT/DISMOUNT/RESET
1374          RET
1375  MDR.    SPACE   4,10
1376  **      MDR.    -  MOUNT/DISMOUNT/RESET
1377  *
1378  *       MDR. PERFORMS THE SIMILAR FUNCTIONS OF MOUNT, DISMOUNT, AND RESET.
1379  *
1380  *
1381  *       ENTRY   (A)     =  SYSCALL CODE FOR OPERATION TO BE PERFORMED
1382  *
```

```
1383  *       EXIT    IF  NO ERROR
1384  *                   TO CALLER
1385  *                 ELSE
1386  *                   TO ERROR
1387  *
1388  *       USES    ALL
1389  *
1390
1391  MDR.    STA     MDRA            STORE SYSCALL VALUE
1392          CALL    CTS             CHECK FOR TARGET FILE SPECIFICATION
1393          STC
1394          JNZ     ERROR           THERE WAS A TARGET FILE
1395          LXI     H,LINE
1396          CALL    $DTB            DELETE TRAILING BLANKS
1397          CPI     1               (A) = LINE LENGTH INCLUDING <00> BYTE
1398          MVI     A,PEC.DF        DEVICE FORMAT ERROR
1399          JZ      ERROR           NULL DEVICE IS ILLEGAL, ONLY BYTE IS NULL
1400  MDR1    PUSH    H               SAVE SPEC. ADDRESS FOR RETRY
1401          DB      SYSCALL,0
1402  MDRA    EQU     *-1             SYSCALL VALUE
1403          POP     H
1404          RNC                     NO ERROR
1405          PUSH    H               SAVE SPEC. ADDRESS
1406          CPI     EC.NPM          NO PROVISIONS MADE FOR REMOUNT
1407          STC
1408          JNZ     ERROR           ALL ERRORS BUT 'EC.NPM' CONSIDERED FATAL
1409          MVI     A,OVL0
1410          DB      SYSCALL,.LOADO  LOAD *HDOSOVL0.SYS*
1411          JC      ERROR
1412          MVI     A,OVL1
1413          DB      SYSCALL,.LOADO  LOAD *HDOSOVL1.SYS*
1414          JC      ERROR           SYSCALL ERROR
1415          POP     H               RESTORE SPEC. ADDRESS
1416          JMP     MDR1            TRY AGAIN
1417          ELSE
```

```
                                1421  ***     MOUNT - MOUNT A DIFFERENT DISK.
                                1422  *
                                1423  *       MOUNT CAUSES A NEW DISK TO BE MOUNTED.
                                1424  *
                                1425  *       INSERT THE DISK IN SY0, THEN TYPE
                                1426  *
                                1427  *       /MOUNT
                                1428
                                1429
                                1430
        043.217                 1431  MOUNT   EQU     *
        043.217  021 230 043   1432          LXI     D,MOUNTA
        043.222  006 377       1433          MVI     B,377Q          OFF PERIODS
        043.224  315 130 046   1434          CALL    MAD             MOUNT ALTERNATE DISK
        043.227  311           1435          RET
                                1436
        043.230  244 306 307   1437  MOUNTA  DB      2440,306Q,307Q
        043.233  012 111 156   1438          DB      NL,'Insert New Disk',':'+200Q
```

```
                                1442  ***     ONECOPY - COPY FILES BETWEEN TWO VOLUMES, WITH ONLY ONE
                                1443  *       DRIVE.
                                1444  *
                                1445  *       (AND FOR MY NEXT TRICK...)
                                1446  *
                                1447  *       OPECOPY COPIES FILES BETWEEN TWO VOLUMES BY ALTERNATING BETWEEN
                                1448  *       TWO PHASES, THE READ PHASE AND THE WRITE PHASE.
                                1449  *
                                1450  *       READ PHASE:
                                1451  *
                                1452  *       DURING THE READ PHASE, THE SOURCE DISK IS MOUNTED. SOURCE FILES ARE
                                1453  *       OPENED IN THE ORDER OF THEIR APPEARANCE. FOR EACH OPENED
                                1454  *       FILE, A 'FILE DESCRIPTOR NODE' *FDN* IS ADDED TO THE ACTIVE
                                1455  *       CHAIN. THEN, AS MUCH AS THE FILE AS POSSIBLE IS READ INTO MEMORY.
                                1456  *
                                1457  *       THE PROCESS CONTINUES UNTIL
                                1458  *               1) THERE IS NO MORE FREE RAM
                                1459  *               2) OR, THERE ARE NO MORE FILE DESCRIPTOR NODES IN THE FREE CHAIN
                                1460  *               3) OR, THERE ARE NO MORE FILES IN NAMTAB (INPUT FILE LIST)
                                1461  *
                                1462  *
                                1463  *       WRITE PHASE
                                1464  *
                                1465  *       DURING THE WRITE PHASE, THE DESTINATION DISK IS MOUNTED. THE NODES
                                1466  *       ARE TAKEN FROM THE ACTIVE CHAIN, AND PROCESSED. IF THE FILE HAD
                                1467  *       BEEN PARTIALLY WRITTEN THE LAST PASS, IT IS  RE-OPENED AND POSITIONED.
                                1468  *       IF THERE IS NOT MORE DATA TO READ FOR A PROCESSED
                                1469  *       NODE, IT IS REMOVED, AND THE CORRESPONDING ENTRY IN NAMTAB IS DELETED.
                                1470  *
                                1471  *       WRITE PHASE CONTINUES UNTIL
                                1472  *
                                1473  *               1) THERE ARE NO MORE FILE NODES IN THE ACTIVE LIST
                                1474  *               2) OR, THE FIRST (AND ONLY) ENTRY IN THE LIST HAS NO
                                1475  *                  MORE DATA IN MEMORY, BUT HAS NOT BEEN COMPLETELY READ.
                                1476
                                1477
043.254                         1478  COPY    EQU     *               CALLED 'COPY' BY MAINLINE CODE
043.254                         1479  OCOPY   EQU     *
043.254  315 077 046            1480          CALL    IFL             INITIALIZE  FDN LISTS
043.257  257                   1481          XRA     A
043.260  062 110 044            1482          STA     OCOPYC          CLEAR FILE COUNT
043.263  062 131 062            1483          STA     VOLFLAG         FLAG SOURCE  VOLUME MOUNTED
043.266  072 252 040            1484          LDA     D.DRVTB+1
043.271  062 132 062            1485          STA     VOLSER          SET VOLUME SERIAL NUMBER
043.274  315 271 053            1486          CALL    DDF             DECODE DESTINATION FILE
043.277  332 275 052            1487          JC      ERROR           ERROR
043.302  062 107 044            1488          STA     OCOPYA          SAVE DESTIONATION TYPE
043.305  315 277 056            1489          CALL    SDD             RESET DEFAULT DEFAULTS
043.310  257                   1490          XRA     A               ALLOW *.*
043.311  315 122 053            1491          CALL    BSL             BUILD SOURCE FILE LIST
043.314  332 275 052            1492          JC      ERROR
043.317  315 252 060            1493          CALL    $MOVEL
043.322  021 000                1494          DW      OCOPYDL
043.324  261 062                1495          DW      DESTFB+FB.NAM
043.326  111 044                1496          DW      OCOPYD          SAVE WILDCARD DESTINATION
043.330  315 003 055            1497          CALL    EBM             EXPAND BUFFER TO MAX
```

```
                                1498
                                1499  *       MAKE SURE HE'S NOT TRYING TO CONCATINATE
                                1500
043.333  072 107 044           1501          LDA     OCOPYA
043.336  247                   1502          ANA     A
043.337  312 360 043           1503          JZ      OCOPY1          HAVE WILDCARDS
043.342  052 302 062           1504          LHLD    NAMTLEN         NO WILDCARDS, ONLY LET HIM SPEFICY ONE SOURCE
043.345  021 357 377           1505          LXI     D,-FB.NAML
043.350  031                   1506          DAD     D
043.351  174                   1507          MOV     A,H
043.352  265                   1508          ORA     L
043.353  076 210               1509          MVI     A,FEC.FCI       FILE CONCATINATION IS ILLEGAL
043.355  302 275 052           1510          JNZ     ERROR
                                1511
                                1512  *       START READ PHASE
                                1513
043.360  072 244 062           1514  OCOPY1  LDA     BUFPTR+1        (A) = BUFFER FWA/256
043.363  074                   1515          INR     A               ROUND UP TO NEXT PAGE
043.364  062 134 062           1516          STA     OBUFPTR         SET SECTOR BUFFER  FWA/256
043.367  072 131 062           1517          LDA     VOLFLAG
043.372  247                   1518          ANA     A
043.373  312 005 044           1519          JZ      OCOPY2          SOURCE IS MOUNTED
043.376  021 132 044           1520          LXI     D,OCOPYF
044.001  107                   1521          MOV     B,A             (B) = 377Q = PERIODS MASK
044.002  315 130 046           1522          CALL    MAD             MOUNT ALTERNATE DISK
044.005  315 203 044           1523  OCOPY2  CALL    RPH             READ PHASE
044.010  072 020 062           1524          LDA     FINHEAD
044.013  247                   1525          ANA     A
044.014  312 044 044           1526          JZ      OCOPY6          NO FILES ARE READ, ERGO NONE ARE LEFT
044.017  072 131 062           1527          LDA     VOLFLAG
044.022  247                   1528          ANA     A
044.023  302 036 044           1529          JNZ     OCOPY3
044.026  006 177               1530          MVI     B,177Q          (B) = PERIODS MASK
044.030  021 154 044           1531          LXI     D,OCOPYG
044.033  315 130 046           1532          CALL    MAD             MOUNT ALTERNATE DISK
044.036  315 156 045           1533  OCOPY3  CALL    WPH             WRITE PHASE
044.041  303 360 043           1534          JMP     OCOPY1
                                1535
                                1536  *       ALL DONE, FINISH MESSAGE
                                1537
044.044  072 110 044           1538  OCOPY6  LDA     OCOPYC          (A) = FILE COUNT
044.047  006 000               1539          MVI     B,0             (BC) = COUNT OF FILES COPIED
044.051  117                   1540          MOV     C,A
                                1541
                                1542  *       TYPE FILE COUNT
                                1543
044.052  076 003               1544          MVI     A,3
044.054  041 065 044           1545          LXI     H,OCOPYE
044.057  315 177 060           1546          CALL    $UDDN           UNPACK COUNT INTO MESSAGE
044.062  315 136 031           1547          CALL    $TYPTX
044.065  130 130 130           1548  OCOPYE  DB      'XXX'
044.070  040 106 151           1549          DB      ' Files Copied',ENL
044.106  311                   1550          RET
                                1551
044.107  000                   1552  OCOPYA  DB      0               DESTINATION FILE WILDCARD FLAG (=0 IF WC)
044.110  000                   1553  OCOPYC  DB      0               FILES COPIED COUNT
```

```
044.111                 1554  OCOPYD  DS      FB.NAML         HOLD AREA FOR WILDCARD DESTINATION
000.021                 1555  OCOPYDL EQU     *-OCOPYD
044.132  244 306 307    1556  OCOPYF  DB      244Q,306Q,307Q
044.135  012 111 156    1557          DB      NL,'Insert Source',':'+200Q
044.154  102 014 044    1558  OCOPYG  DB      102Q,014Q,44Q
044.157  012 111 156    1559          DB      NL,'Insert Destination',':'+200Q
```

```
                                1563  **      RPH - READ PHASE.
                                1564  *
                                1565  *       RPH  HANDLES THE READ PHASE OF THE COPY PROCESS.
                                1566  *
                                1567  *       IT IS ENTERED WITH THE NAMTAB AND FDN TABLE SETUP, AND
                                1568  *       WITH THE SOURCE DISK MOUNTED.
                                1569  *
                                1570  *       READ PHASE:
                                1571  *
                                1572  *       DURING THE READ PHASE, THE SOURCE DISK IS MOUNTED. SOURCE FILES ARE
                                1573  *       OPENED IN THE ORDER OF THEIR APPEARANCE. FOR EACH OPENED
                                1574  *       FILE, A 'FILE DESCRIPTOR NODE' *FDN* IS ADDED  TO  THE ACTIVE
                                1575  *       CHAIN. THEN, AS MUCH AS THE FILE AS POSSIBLE IS READ INTO MEMORY.
                                1576  *
                                1577  *       THE PROCESS CONTINUES UNTIL
                                1578  *               1) THERE IS NO MORE FREE RAM
                                1579  *               2) OR, THERE ARE NO MORE FILE DESCRIPTOR NODES IN THE FREE CHAIN
                                1580  *               3) OR, THERE ARE NO MORE FILES IN NAMTAB (INPUT FILE LIST)
                                1581  *
                                1582  *       ENTRY   NONE
                                1583  *       EXIT    NONE
                                1584  *       USES    ALL
                                1585
                                1586
044.203                         1587  RPH     EQU     *
                                1588
                                1589
                                1590  *       SEE IF ANY MEMORY TO HAVE
                                1591
044.203  315 071 046          1592          CALL    CBR             COMPUTE BUFFER ROOM
044.206  310                  1593          RZ                      NONE
                                1594
                                1595  *       SEE IF WE NEED TO READ SOME MORE INTO A PART-COPIED FILE
                                1596
044.207  041 020 062          1597          LXI     H,FDNHEAD
044.212  156                  1598          MOV     L,M             (HL) = ADDRESS IF FIRST NODE
044.213  175                  1599          MOV     A,L
044.214  247                  1600          ANA     A
044.215  312 232 044          1601          JZ      RPH1            IS NO FIRST NODE, ERGO NO FILE
044.220  043                  1602          INX     H
000.000                         1603          ERRNZ   FDN.STA-1
044.221  176                  1604          MOV     A,M             (A) = .STA
044.222  346 002              1605          ANI     ST.OPR
044.224  021 132 063          1606          LXI     D,NAMTAB
044.227  302 325 044          1607          JNZ     RPH2.5          FILE IS INCOMPLETELY READ
                                1608
                                1609  *       SEE IF ANY FREE FILE DESCRIPTOR NODES TO USE
                                1610
044.232  072 017 062          1611  RPH1    LDA     FDNFRE
044.235  247                  1612          ANA     A
044.236  310                  1613          RZ                      NO MORE
                                1614
                                1615  *       SEE IF THERE IS A FILE IN NAMTAB WITHOUT AN ENTRY IN FNDLIST.
                                1616  *       SINCE THE FIRST ENTRY IN FDNLIST CORRESPONDS TO THE FIRST IN
                                1617  *       NAMTAB, ETC., WE'LL JUST RUN DOWN FDNLIST UNTIL THE END, AND
                                1618  *       THE NEXT NAMTAB FILE WILL BE THE ONE WE WANT...
```

```
                                1619
044.237  001 021 000   1620           LXI     B,FB.NAML       (BC) = ENTRY SIZE IN NAMTAB
044.242  021 357 377   1621           LXI     D,-FB.NAML              (DE) = POINTER INTO NAMTAB
044.245  041 020 062   1622           LXI     H,FDNHEAD
044.250  175          1623           MOV     A,L             START WITH FDNHEAD
044.251  157          1624  RPH2     MOV     L,A             FOLLOW LINK
044.252  176          1625           MOV     A,M             (A) = NEXT NODE
044.253  353          1626           XCHG
044.254  011          1627           DAD     B               ADVANCE POINTER INTO NAMTAB
044.255  353          1628           XCHG
044.256  247          1629           ANA     A
044.257  302 251 044   1630           JNZ     RPH2            LINK SOME MORE
044.262  345          1631           PUSH    H               (HL) = ADDRESS OF LAST NODE
044.263  052 302 062   1632           LHLD    NAMTLEN
044.266  315 216 030   1633           CALL    $CDEHL          SEE IF HAVE ACCOUNTED FOR ALL NAMTAB ENTRYS
044.271  341          1634           POP     H
044.272  310          1635           RE                      FILES ALL USED UP
                      1636
                      1637  *        HAVE ROOM FOR DATA, HAVE A NODE FOR THE FILE COUNTS, AND
                      1638  *        HAVE A FILE NAME. ALL SET FOR BUSINESS..
                      1639  *
                      1640  *        (DE) = INDEX INTO NAMTAB FOR FILE
                      1641  *        (HL) = NODE ADDRESS OF LAST ENTRY IN LIST
                      1642  *
                      1643  *        CHAIN THE FIRST FREE NODE ONTO THE END OF THE LIST
                      1644
044.273  072 017 062   1645           LDA     FDNFRE
044.276  167          1646           MOV     M,A             CHAIN TO  NEW END  NODE
044.277  157          1647           MOV     L,A
044.300  176          1648           MOV     A,M             (A) =  NEXT NODE IN FREE CHAIN
044.301  062 017 062   1649           STA     FDNFRE
044.304  006 011      1650           MVI     B,FDNELEN
044.306  345          1651           PUSH    H               SAVE NODE ADDRESS
044.307  315 212 031   1652           CALL    $ZERO           ZERO ENTIRE NODE, ENCLUDING CHAIN (AT END,  NOW)
044.312  001 132 063   1653           LXI     B,NAMTAB
044.315  353          1654           XCHG
044.316  011          1655           DAD     B               (HL) = ADDRESS OF NAMTAB ENTRY
044.317  042 306 062   1656           SHLD    NAMTPTR         POINTER TO CURRENT NAMTAB ENTRY
044.322  353          1657           XCHG
044.323  341          1658           POP     H
000.000                1659           ERRNZ   FDN.STA-1
044.324  043          1660           INX     H               (HL) = ADDR OF FDN.STA OF NODE
                      1661
                      1662  *        READY TO OPEN FILE
                      1663  *
                      1664  *        (DE) = NAMTAB ENTRY ADDRESS
                      1665  *        (HL) = #FDN.STA OF ENTRY
                      1666
044.325  345          1667  RPH2.5   PUSH    H               SAVE ADDRESS
044.326  353          1668           XCHG
044.327  257          1669           XRA     A
000.000                1670           ERRNZ   CN.SOU          (A) = SOURCE CHANNEL NUMBER
044.330  377 042      1671           DB      SYSCALL,.OPENR  OPEN
044.332  332 044 052   1672           JC      NAMERR          ERROR
044.335  321          1673           POP     D
044.336  032          1674           LDAX    D               (A) = FDN.STA
```

```
044.337  346 002       1675          ANI     ST.OPR
044.341  325           1676          PUSH    D               SAVE ADDRESS
044.342  302 030 045   1677          JNZ     RPH3            ALREADY OPENED IN PREVIOUS PASSES
                       1678
                       1679  *       FIRST TIME THIS FILE HAS BEEN OPENED. SEE IF CONTIGUOUS
                       1680
044.345  345           1681          PUSH    H
044.346  041 110 044   1682          LXI     H,OCOPYC
044.351  064           1683          INR     M
044.352  341           1684          POP     H
044.353  032           1685          LDAX    D
044.354  366 002       1686          ORI     ST.OPR          SET OPEN FOR READ
044.356  022           1687          STAX    D
044.357  052 352 040   1688          LHLD    S.CFWA          (HL) = CHANNEL 0 FWA
000.000               1689          ERRNZ   IOCCTD-1        WE NEED TO CHAIN ONE TO GET TO USER #0
044.362  315 211 030   1690          CALL    $HLIHL
000.000               1691          ERRNZ   CN.SOU          ASSUME WE WANT CHANNEL 0
044.365  315 234 030   1692          CALL    $INDL
044.370  041 000       1693          DW      IOC.DIR+DIR.FLG
044.372  173           1694          MOV     A,E             (A) = DIR.FLG
044.373  346 000       1695          ANI     0 DIF.CNT       * * PATCH * *
044.375  312 030 045   1696          JZ      RPH3            NOT CONTIG
                       1697
                       1698  *       IS CONTIG. GET FILE SIZE
                       1699
045.000  315 234 030   1700          CALL    $INDL
045.003  005 000       1701          DW      IOC.GRT
045.005  325           1702          PUSH    D               SAVE GRT ADDRESS
045.006  315 234 030   1703          CALL    $INDL
045.011  043 000       1704          DW      IOC.DIR+DIR.FGN (E) = DIR.FGN
045.013  173           1705          MOV     A,E
045.014  341           1706          POP     H               (HL) = GRT TABLE ADDRESS
045.015  315 223 053   1707          CALL    CFS.            COMPUTE BLOCK SIZE
045.020  341           1708          POP     H               (HL) = ADDRESS OF FDN.STA
045.021  345           1709          PUSH    H
045.022  176           1710          MOV     A,M             (A) = FDN.STA
045.023  366 020       1711          ORI     ST.CNT          FLAG CONTIG
045.025  167           1712          MOV     M,A
045.026  043           1713          INX     H
000.000               1714          ERRNZ   FDN.SIZ-FDN.STA-1
045.027  163           1715          MOV     M,E             SET BLOCK COUNT
                       1716
                       1717  *       READY TO READ DATA. POSITION FILE (IN CASE SOME WAS READ IN
                       1718  *       PREVIOUS PASSES) AND COMPUTE THE MAX POSSIBLE READ COUNT
                       1719  *
                       1720  *       ((SP)) = ADDRESS OF FDN.STA FOR NODE
                       1721
045.030  341           1722  RPH3    POP     H               (HL) = ADDRESS OF FDN.STA
045.031  345           1723          PUSH    H
045.032  315 234 030   1724          CALL    $INDL
045.035  002 000       1725          DW      FDN.AMR-FDN.STA (DE) = AMOUNT READ (IN SECTORS)
045.037  102           1726          MOV     B,D
045.040  113           1727          MOV     C,E             (BC) = AMOUNT READ
045.041  076 000       1728          MVI     A,CN.SOU
045.043  377 047       1729          DB      SYSCALL,.POSIT          POSIT
045.045  332 076 052   1730          JC      IERR3           POSIT BLEW UP
```

```
045.050  315 071 046  1731          CALL    CBR             COMPUTE BUFFER ROOM
045.053  353          1732          XCHG                    (D) = POINTER/256, (E) = LIMIT/256
045.054  341          1733          POP     H               (HL) = #FDN.STA
045.055  001 006 000  1734          LXI     B,FDN.ADR-FDN.STA
045.060  011          1735          DAD     B               (HL) = #FDN.ADR
045.061  162          1736          MOV     M,D             SET ADDRESS/256
045.062  345          1737          PUSH    H               SAVE #FDN.ADR
045.063  036 000      1738          MVI     E,0             (DE) = ADDRESS
045.065  107          1739          MOV     B,A             (B) = SECTORS OF RAM AVAILABLE
045.066  113          1740          MOV     C,E             (C) = 0
045.067  305          1741          PUSH    B               SAVE TRY COUNT
045.070  076 000      1742          MVI     A,CN.SOU
045.072  377 004      1743          DB      SYSCALL,.READ   READ THE STUFF
                      1744
                      1745  *       COMPUTE THE AMOUNT READ (IN CASE OF EOF)
                      1746
045.074  321          1747          POP     D               (DE) = TRY COUNT
045.075  322 122 045  1748          JNC     RPH4            GOT  ALL WE TRYED
045.100  376 001      1749          CPI     EC.EOF
045.102  302 044 052  1750          JNE     NAMERR          NOT JUST EOF, GOT TROUBLES
045.105  172          1751          MOV     A,D
045.106  220          1752          SUB     B               REMOVE AMOUNT WE DIDNT GET
045.107  127          1753          MOV     D,A
045.110  341          1754          POP     H               (HL) = #FDN.ADR
045.111  345          1755          PUSH    H
045.112  001 372 377  1756          LXI     B,FDN.STA-FDN.ADR
045.115  011          1757          DAD     B
045.116  176          1758          MOV     A,M             (A) = FDN.STA
045.117  346 375      1759          ANI     377Q-ST.OPR     EOF, NOT OPEN FOR READ ANYMORE
045.121  167          1760          MOV     M,A             POST READ COMPLETE FOR THIS GUY
                      1761
                      1762  *       STORE RESULTS OF READ IN NODE
                      1763  *
                      1764  *       (D) = SECTORS READ
                      1765  *       ((SP)) = #FDN.ADR
                      1766
045.122  341          1767  RPH4    POP     H               (HL) = #FDN.ADR
045.123  043          1768          INX     H
000.000                1769          ERRNZ   FDN.AIM-FDN.ADR-1       (HL) = ADDRESS IF AMOUNT IN MEMORY BYTE
045.124  162          1770          MOV     M,D             STORE SECTORS IN MEMORY COUNT
045.125  001 373 377  1771          LXI     B,FDN.AMR-FDN.AIM
045.130  011          1772          DAD     B               (HL) = #FDN.AMR (AMOUNT READ)
045.131  176          1773          MOV     A,M             (A) = AMOUNT READ BEFORE
045.132  202          1774          ADD     D               ADD NEW AMOUNT
045.133  167          1775          MOV     M,A
045.134  043          1776          INX     H
045.135  176          1777          MOV     A,M
045.136  316 000      1778          ACI     0               PROPIGATE FOR VERY LARGE FILES
045.140  167          1779          MOV     M,A
045.141  041 134 062  1780          LXI     H,OBUFPTR
045.144  176          1781          MOV     A,M
045.145  202          1782          ADD     D               ADVANCE FREE RAM POINTER BY AMOUNT READ
045.146  167          1783          MOV     M,A
045.147  076 000      1784          MVI     A,CN.SOU
045.151  377 046      1785          DB      SYSCALL,.CLOSE  CLOSE FILE
045.153  303 203 044  1786          JMP     RPH             SEE IF MORE TO READ
```

```
                                1788  **      WPH - WRITE PHASE.
                                1789  *
                                1790  *       WPH HANDLES THE WRITE PHASE PROCESSING. IT IS ENTERED WITH
                                1791  *       THE FDN CHAIN SETUP, THE NAMTAB SETUP, AND
                                1792  *       THE DESTINATION DISK MOUNTED.
                                1793  *
                                1794  *
                                1795  *       WRITE PHASE
                                1796  *
                                1797  *       DURING THE WRITE PHASE, THE DESTINATION DISK IS MOUNTED, THE NODES
                                1798  *       ARE TAKEN FROM THE ACTIVE CHAIN, AND PROCESSED. IF THE FILE HAD
                                1799  *       BEEN PARTIALLY WRITTEN THE LAST PASS, IT IS  RE-OPENED AND POSITIONED.
                                1800  *       IF THERE IS NOT MORE DATA TO READ FOR A PROCESSED
                                1801  *       NODE, IT IS REMOVED, AND THE CORRESPONDING ENTRY IN NAMTAB IS DELETED.
                                1802  *
                                1803  *       WRITE PHASE CONTINUES UNTIL
                                1804  *
                                1805  *               1) THERE ARE NO MORE FILE NODES IN THE ACTIVE LIST
                                1806  *               2) OR, THE FIRST (AND ONLY) ENTRY IN THE LIST HAS NO
                                1807  *                  MORE DATA IN MEMORY, BUT HAS NOT BEEN COMPLETELY READ.
                                1808  *
                                1809  *       ENTRY   NONE
                                1810  *       EXIT    NONE
                                1811  *       USES    ALL
                                1812
                                1813
045.156                         1814  WPH     EQU     *
                                1815
                                1816  *       SEE IF MORE TO WRITE
                                1817
045.156  041 020 062           1818          LXI     H,FDNHEAD
045.161  156                   1819          MOV     L,M
045.162  175                   1820          MOV     A,L             (A) = FIRST NODE INDEX
045.163  247                   1821          ANA     A
045.164  310                   1822          RZ                      NO MORE
045.165  315 234 030           1823          CALL    $INDL
045.170  010 000               1824          DW      FDN.AIM         (E) = AMOUNT IN MEMORY FOR THIS GUY
045.172  173                   1825          MOV     A,E
045.173  247                   1826          ANA     A
045.174  302 211 045           1827          JNZ     WPH0            GOT DATA
                                1828
                                1829  *       NO DATA IN NODE, IF STILL READING, RETURN FOR MORE
                                1830
045.177  043                   1831          INX     H
045.200  176                   1832          MOV     A,M
045.201  053                   1833          DCX     H
045.202  346 002               1834          ANI     ST.OPR
045.204  300                   1835          RNZ                     STILL READING, GET MORE
045.205  353                   1836          XCHG                    (DE) = ADDRESS
045.206  303 032 046           1837          JMP     WPH4            REMOVE NODE, AM DONE WITH FILE
                                1838
                                1839  *       HAVE DATA TO WRITE, SEE IF WE HAVE OPENED THIS FILE BEFORE.,
                                1840  *       OR IF THIS IS THE FIRST TIME
                                1841
045.211  345                   1842  WPH0    PUSH    H               SAVE NODE POINTER
045.212  043                   1843          INX     H
```

```
000.000                 1844            ERRNZ   FDN.STA-1
045.213 176             1845            MOV     A,M             (A) = FDN.STA
045.214 346 001         1846            ANI     ST.OPW
045.216 302 325 045     1847            JNZ     WPH2            OPENED BEFORE
000.000                 1848            ERRNZ   ST.OPW-1
045.221 064             1849            INR     M               SET '1' BIT
                        1850
                        1851 *          BUILD NAME INTO DESTFB
                        1852
045.222 345             1853            PUSH    H               SAVE NODE ADDRESS
045.223 001 111 044     1854            LXI     B,OCOPYD
045.226 021 132 063     1855            LXI     D,NAMTAB
045.231 041 261 062     1856            LXI     H,DESTFB+FB.NAM
045.234 315 155 056     1857            CALL    MWN             MERGE WILDCARD NAME
045.237 341             1858            POP     H
                        1859
                        1860 *          IS 1ST TIME FOR THIS FILE. IF CONTIGUOUS FLAG, OPEN THE FILE
                        1861 *          FOR CONTIGUOUS
                        1862
045.240 176             1863            MOV     A,M             (A) = FLAG BYTE
045.241 346 020         1864            ANI     ST.CNT
045.243 302 263 045     1865            JNZ     WPH1            IS CONTIG
045.246 041 261 062     1866            LXI     H,DESTFB+FB.NAM
045.251 076 001         1867            MVI     A,CN.DES
045.253 377 043         1868            DB      SYSCALL,.OPENW  JUST OPEN FOR WRITE
045.255 332 056 052     1869            JC      DESTERR         ERROR
045.260 303 357 045     1870            JMP     WPH3            WRITE THE DATA
                        1871
                        1872 *          IS CONTIG FILE. OPEN IN CONTIG MODE
                        1873
045.263 043             1874 WPH1       INX     H
000.000                 1875            ERRNZ   FDN.SIZ-FDN.STA-1
045.264 116             1876            MOV     C,M             (C) = COUNT (IN BLOCKS)
045.265 006 000         1877            MVI     B,0
045.267 041 261 062     1878            LXI     H,DESTFB+FB.NAM
045.272 076 001         1879            MVI     A,CN.DES
045.274 305             1880            PUSH    B               SAVE COUNT
045.275 377 050         1881            DB      SYSCALL,.DELET  DELETE OLD ONE
045.277 322 307 045     1882            JNC     WPH1.5          DELETED
045.302 376 014         1883            CPI     EC.FNF
045.304 302 275 052     1884            JNE     ERROR           MUST BE WRITE PROTECTED, OR  SOMETHING...
045.307 301             1885 WPH1.5     POP     B               (BC) = COUNT
045.310 041 261 062     1886            LXI     H,DESTFB+FB.NAM
045.313 076 001         1887            MVI     A,CN.DES
045.315 377 045         1888            DB      SYSCALL,.OPENC  OPEN CONTIG
045.317 332 056 052     1889            JC      DESTERR
045.322 303 357 045     1890            JMP     WPH3
                        1891
                        1892 *          THIS FILE HAS ALREADY BEEN PARTIALLY WRITTEN. OPEN IN UPDATE MODE
                        1893 *          SO WE CAN EXTEND IT.
                        1894
045.325 041 261 062     1895 WPH2       LXI     H,DESTFB+FB.NAM
045.330 076 001         1896            MVI     A,CN.DES
045.332 377 044         1897            DB      SYSCALL,.OPENU  OPEN FOR UPDATE
045.334 332 056 052     1898            JC      DESTERR         PROBLEMS
045.337 341             1899            POP     H
```

```
045.340 345           1900          PUSH    H               (HL) = #FDN.STA
045.341 315 234 030   1901          CALL    $INDL
045.344 005 000       1902          DW      FDN.AMW         (DE) = AMOUNT WRITTEN
045.346 102           1903          MOV     B,D
045.347 113           1904          MOV     C,E             (BC) = SECTORS WRITTEN
045.350 076 001       1905          MVI     A,CN.DES
045.352 377 047       1906          DB      SYSCALL,.POSIT  POSITION FOR EXTEND
045.354 332 064 052   1907          JC      IERR1           COULDNT GET THERE!
                      1908
                      1909  *       FILE OPEN AND POSITIONED, WRITE DATA
                      1910
045.357 341           1911  WPH3    POP     H
045.360 345           1912          PUSH    H               (HL) = #FDN.LNK
045.361 315 234 030   1913          CALL    $INDL
045.364 007 000       1914          DW      FDN.ADR         (E) = ADDR/256, (D) = CNT/256
045.366 102           1915          MOV     B,D
045.367 123           1916          MOV     D,E
045.370 036 000       1917          MVI     E,0             (DE) = ADDRESS
045.372 113           1918          MOV     C,E             (BC) = COUNT
045.373 076 001       1919          MVI     A,CN.DES
045.375 305           1920          PUSH    B               SAVE WRITE COUNT
045.376 377 005       1921          DB      SYSCALL,.WRITE  WRITE IT
046.000 332 056 052   1922          JC      DESTERR         PROBABLY OUT OF ROOM
046.003 076 001       1923          MVI     A,CN.DES
046.005 377 046       1924          DB      SYSCALL,.CLOSE  CLOSE IT
046.007 332 056 052   1925          JC      DESTERR
046.012 301           1926          POP     B               (B) = SECTORS WRITTEN
046.013 341           1927          POP     H
046.014 345           1928          PUSH    H               (HL) = #FDN.LNK
046.015 021 005 000   1929          LXI     D,FDN.AMW-FDN.LNK
046.020 031           1930          DAD     D               (HL) = FDN.AMW
046.021 176           1931          MOV     A,M
046.022 200           1932          ADD     B
046.023 167           1933          MOV     M,A
046.024 043           1934          INX     H
046.025 176           1935          MOV     A,M
046.026 316 000       1936          ACI     0               INCREMENT AMOUNT WRITTEN
046.030 167           1937          MOV     M,A
                      1938
                      1939  *       CLEAR 'IN MEMORY' COUNT IN NODE. IF THE FILE HAS NO MORE TO
                      1940  *       READ, REMOVE IT FROM THE CHAIN AND NAMTAB
                      1941
046.031 321           1942          POP     D               (DE) = FDN.LNK
046.032 041 010 000   1943  WPH4    LXI     H,FDN.AIM
046.035 031           1944          DAD     D
046.036 066 000       1945          MVI     M,0             CLEAR AMOUNT IN MEMORY
046.040 353           1946          XCHG                    (HL) = FDN.LNK
046.041 043           1947          INX     H
000.000               1948          ERRNZ   FDN.STA-FDN.LNK-1
046.042 176           1949          MOV     A,M             (A) = FDN.STA
046.043 346 002       1950          ANI     ST.OPR
046.045 300           1951          RNZ                     STILL READING, AM DONE FOR THIS PHASE
                      1952
                      1953  *       UNLINK NODE FROM LIST
                      1954
046.046 053           1955          DCX     H
```

```
046.047 176             1956            MOV     A,M
046,050 062 020 062     1957            STA     FDNHEAD         UNLINK FROM ACTIVE LIST
046.053 072 017 062     1958            LDA     FDNFRE
046,056 167             1959            MOV     M,A             PUT THIS GUY ON HEAD OF FREE LIST
046.057 175             1960            MOV     A,L
046,060 062 017 062     1961            STA     FDNFRE
046.063 315 231 056     1962            CALL    REN             REMOVE ENTRY FROM NAMTAB
046,066 303 156 045     1963            JMP     WPH             TRY TO WRITE THE NEXT GUY


                        1965  **        CBR - COMPUTE BUFFER ROOM.
                        1966  *
                        1967  *         CBR COMPUTES THE NUMBER OF SECTORS WORTH OF RAM
                        1968  *         STILL FREE.
                        1969  *
                        1970  *         ENTRY   NONE
                        1971  *         EXIT    (A) = SECTORS OF RAM FREE
                        1972  *                 'Z' SET IFF (A) = 0
                        1973  *                 (H) = BUFPTR/256
                        1974  *                 (L) = OBUFLIM/256
                        1975  *         USES    A,F
                        1976
                        1977
046.071 052 133 062     1978  CBR       LHLD    OBUFLIM
000.000                 1979            ERRNZ   OBUFPTR-OBUFLIM-1
046,074 175             1980            MOV     A,L
046.075 224             1981            SUB     H
046.076 311             1982            RET


                        1984  **        IFL - INITIALIZE FDN LIST.
                        1985  *
                        1986  *         IFL CHAINS ALL THE FDN NODES TO THE FREE LIST. THIS
                        1987  *         CLEANUP IS NECESSARY IN CASE A CTL-C OR SOMETHING
                        1988  *         LEFT THE LIST GARBAGED.
                        1989  *
                        1990  *         ENTRY   NONE
                        1991  *         EXIT    NONE
                        1992  *         USES    ALL
                        1993
                        1994
046,077 041 021 062     1995  IFL       LXI     H,FDN,1
046.102 175             1996            MOV     A,L             (A) = FIRST LINK
046.103 062 017 062     1997            STA     FDNFRE
046.106 257             1998            XRA     A
046,107 062 020 062     1999            STA     FDNHEAD         NONE IN LIST
046.112 006 007         2000            MVI     B,FDNCNT-1      (B) = NUMBER OF NODES-1
046.114 076 011         2001  IFL1      MVI     A,FDNELEN
046.116 205             2002            ADD     L               (A) = #ADDR OF NEXT NODE
046.117 167             2003            MOV     M,A             SET LINK
046.120 157             2004            MOV     L,A             FORWARD TO NEXT LINK
046.121 005             2005            DCR     B
```

```
046.122  302 114 046  2006          JNZ     IFL1            MORE TO GO
046.125  066 000      2007          MVI     M,0             LAST ONE CHAINS NOWHERE
046.127  311          2008          RET


                      2010  **      MAD - MOUNT ALTERNATE DISK.
                      2011  *
                      2012  *       MAD DISMOUNTES THE CURRENT DISK, HAS THE USER INSERT THE
                      2013  *       OTHER DISK, AND MOUNTS IT.
                      2014  *
                      2015  *       ENTRY   (B) = FRONT PANEL LED PATTERN
                      2016  *               (DE) = PROMPT PATTERNS FOR PANEL AND CONSOLE
                      2017  *       EXIT    (HL) = $VOLFLAG
                      2018  *       USES    ALL
                      2019
                      2020
046.130               2021  MAD     EQU     *
                      2022
                      2023  *       DISMOUNT CURRENT DISK
                      2024
046.130  325          2025          PUSH    D
046.131  305          2026          PUSH    B               SAVE ENTRY PARAMETERS IN CASE OF RETRY
046.132  325          2027          PUSH    D
046.133  305          2028          PUSH    B               SAVE ENTRY PARAMETERS OVER SYDD CALL
046.134  041 342 046  2029          LXI     H,MNDA          DEVICE SPECIFICATION
046.137  377 203      2030          DB      SYSCALL,.DMNMS  DISMOUNT WITHOUT MESSAGE
046.141  332 275 052  2031          JC      ERROR           IF ERROR
                      2032
                      2033  *       SETUP PROMPT ON FP LEDS AND CONSOLE FOR NEW DISK
                      2034
046.144  363          2035  MAD0    DI
046.145  041 243 040  2036          LXI     H,D.DLYMO
046.150  176          2037          MOV     A,M
046.151  247          2038          ANA     A
046.152  312 157 046  2039          JZ      MAD1            DISK ALREADY STOPPED
046.155  066 001      2040          MVI     M,1             STOP DISK  VERY SOON
046.157  373          2041  MAD1    EI
046.160  076 203      2042          MVI     A,UO.DDU+UO.CLK+UO.HLT
046.162  062 010 040  2043          STA     .MFLAG          HALT DISPLAY UPDATE
046.165  041 013 040  2044          LXI     H,.ALEDS
046.170  076 011      2045          MVI     A,9
046.172  301          2046          POP     B               (B) = PERIOD PATTERN
046.173  160          2047  MAD2    MOV     M,B             SET PATTERN
046.174  043          2048          INX     H
046.175  075          2049          DCR     A
046.176  302 173 046  2050          JNZ     MAD2            IF MORE TO BLANK
046.201  041 016 040  2051          LXI     H,.ALEDS+3
046.204  001 003 000  2052          LXI     B,3
046.207  321          2053          POP     D               (DE) = PROMPT LIST
046.210  315 252 030  2054          CALL    $MOVE           MOVE IN PROMPT PATTERN
046.213  353          2055          XCHG                    (HL) = PATTERN
046.214  377 003      2056          DB      SYSCALL,.PRINT  CONSOLE PROMPT
046.216  315 136 031  2057          CALL    $TYPTX
046.221  207          2058          DB      BELL+200Q       BEEP CONSOLE, TOO
```

```
046.222  076 144        2059           MVI     A,100
046.224  315 140 002    2060           CALL    .HORN           BEEP A WARNING
                        2061
                        2062  *        WAIT FOR SIGNAL THAT NEW DISK IS IN
                        2063
046.227  377 001        2064  MAD3     DB      SYSCALL,.SCIN
046.231  322 242 046    2065           JNC     MAD4            GOT A CHARACTER
046.234  333 360        2066           IN      IP.PAD
046.236  074            2067           INR     A
046.237  312 227 046    2068           JZ      MAD3            NO REPLY THERE, EITHER
                        2069
                        2070  *        GOT REPLY. GOBBLE EXTRA CHARACTERS FROM CONSOLE
                        2071
046.242  377 001        2072  MAD4     DB      SYSCALL,.SCIN
046.244  322 242 046    2073           JNC     MAD4
                        2074
                        2075  *        READ NEW DISK'S LABEL
                        2076
046.247  315 347 046    2077           CALL    GETLAB
046.252  332 275 052    2078           JC      ERROR
                        2079
                        2080  *        SEE IF LABEL CHANGED FROM BEFORE
                        2081
046.255  301            2082           POP     B
046.256  321            2083           POP     D               RESTORE ENTRY PARAMETERS
046.257  041 132 062    2084           LXI     H,VOLSER
046.262  072 000 027    2085           LDA     LABEL+LAB.SER
046.265  276            2086           CMP     M
046.266  302 300 046    2087           JNE     MAD4.5          IS THE RIGHT DISK
046.271  325            2088           PUSH    D               SAVE PARAMS AS IN BEGINNING
046.272  305            2089           PUSH    B
046.273  325            2090           PUSH    D               SAVE FOR RETRY
046.274  305            2091           PUSH    B
046.275  303 144 046    2092           JMP     MAD0            IT WAS NOT THE RIGHT DISK
                        2093
046.300  167            2094  MAD4.5   MOV     M,A             SET NEW SERIAL
046.301  041 131 062    2095           LXI     H,VOLFLAG
046.304  176            2096           MOV     A,M
046.305  057            2097           CMA
046.306  167            2098           MOV     M,A             COMPLEMENT VOLUME FLAG
                        2099
                        2100  *        ERASE FRONT PANEL DISLPLAY
                        2101
046.307  041 013 040    2102           LXI     H,.ALEDS
046.312  076 011        2103           MVI     A,9
046.314  160            2104  MAD5     MOV     M,B             SET TO PATTERN
046.315  043            2105           INX     H
046.316  075            2106           DCR     A
046.317  302 314 046    2107           JNZ     MAD5
046.322  315 326 046    2108           CALL    MND             MOUNT NEW DISK
046.325  311            2109           RET
```

```
                                2111  **      MND     - MOUNT NEW DISK
                                2112  *
                                2113  *       MOUNT NEW DISK ONTO DEVICE SSECIFIED IN MNDA
                                2114  *
                                2115  *
                                2116  *       ENTRY   NONE
                                2117  *
                                2118  *       EXIT    LABEL   = LABEL SECTOR
                                2119  *
                                2120  *       USES    ALL
                                2121  *
                                2122
046.326  041 342 046            2123  MND     LXI     H,MNDA
046.331  377 202                2124          DB      SYSCALL,.MONMS  MOUNT WITHOUT MESSAGE
046.333  332 275 052            2125          JC      ERROR           IF ERROR IN MOUNT
046.336  315 347 046            2126          CALL    GETLAB          GET LABEL
046.341  311                    2127          RET
                                2128
046.342  123 131 060            2129  MNDA    DB      'SY0:',0


                                2131  **      GETLAB  - GET LABEL
                                2132  *
                                2133  *       GETLAB READS THE DISK LABEL
                                2134  *
                                2135  *       ENTRY   NONE
                                2136  *
                                2137  *       EXIT    LABEL IN LABEL
                                2138  *               (PSW)   = 'C' CLEAR IF NO ERROR
                                2139  *                       = 'C' SET   IF    ERROR
                                2140  *                         (A)  = ERROR CODE
                                2141  *
                                2142  *       USES    ALL
                                2143  *
                                2144
046.347  041 011 000            2145  GETLAB  LXI     H,DDF.LAB
046.352  021 000 027            2146          LXI     D,LABEL
046.355  001 000 001            2147          LXI     B,256
046.360  315 241 031            2148          CALL    $WER            WRITE ENABLE RAM
046.363  076 002                2149          MVI     A,DC.RER
046.365  315 130 040            2150          CALL    SYDD
046.370  311                    2151          RET
                                2152          ENDIF
```

```
                    2155  ***     DELETE - PROCESS DELETE COMMAND.
                    2156  *
                    2157  *       SYNTAX:
                    2158  *
                    2159  *       SOURCE1,...,SOURCEN/DELETE
                    2160  *
                    2161  *       AT LEAST ONE SOURCE FILE MUST BE SPECIFIED.
                    2162  *       IF *.* IS SPECIFIED, DELETE ASKS,
                    2163  *               DELETE ALL ?!? ARE YOU SURE?
                    2164
                    2165
      000.001       2166          IF      .PIP.
                    2167  DELETE  EQU     *
                    2168          LXI     H,LINE
                    2169
                    2170  *       SEE IF A DESTINATION FILE SPECIFIED
                    2171
                    2172  DEL1    MOV     A,M
                    2173          INX     H
                    2174          ANA     A
                    2175          JZ      DEL2            END OF LINE
                    2176          CPI     '='
                    2177          JNE     DEL1
                    2178
                    2179  *       HE SPECIFIED A DESTINATION FILE
                    2180
                    2181          MVI     A,PEC.TFI       TARGET FILE ILLEGAL
                    2182          JMP     ERROR           FORMAT ERROR
                    2183
                    2184  *       NO TARGET FILE SPECIFIED
                    2185
                    2186  DEL2    MVI     A,1             CHECK FOR *.*
                    2187          CALL    BSL             BUILD SOURCE FILE LIST
                    2188          JC      ERROR           NO GOOD
                    2189
                    2190  *       DELETE FILES ONE BY ONE
                    2191
                    2192  DEL5    LHLD    NAMTLEN
                    2193          MOV     A,H
                    2194          ORA     L
                    2195          RZ                      END OF LIST
                    2196          LXI     H,NAMTAB
                    2197          DB      SYSCALL,.DELET  REMOVE IT
                    2198          JC      NAMERR          ERROR ON DELETE
                    2199          CALL    REN             REMOVE ENTRY FROM NAMTAB
                    2200          JMP     DEL5            DELETE THE NEXT ONE
                    2201          STL     'RENAME - PROCESS RENAME COMMAND'
                    2202          EJECT
                    2203  ***     RENAME - RENAME FILES.
                    2204  *
                    2205  *       SYNTAX:
                    2206  *
                    2207  *       DEST = SOURCE1,...,SOURCEN
                    2208  *
                    2209  *       RENAME IS PROCESSED IN A MANNER SIMILAR TO COPY, EXCEPT THAT THE
                    2210  *       FILE IS RENAMED, RATHER THAN COPIED.
```

```
2211
2212
2213  RENAME  EQU     *
2214          CALL    DDF             DECODE DESTINATION FILE
2215          JC      ERROR
2216          XRA     A               ALLOW *.*
2217          CALL    BSL             BUILD SOURCEFILE LIST
2218          JC      ERROR
2219
2220  *       DO MULTIPLE RENAMES
2221
2222  REN1    LXI     B,DESTFB+FB.NAM (BC) = WILDCARDED TARGET NAME
2223          LXI     D,NAMTAB        (DE) = NORMAL SOURCE NAME
2224          LXI     H,RENA          (HL) = BUFFER FOR RESULT NAME
2225          PUSH    B               SAVE #DESTFB+FB.NAM
2226          PUSH    D               SAVE #NAMTAB
2227          CALL    MWN             MERGE WILDCARD NAME
2228          POP     D               (DE) = #NAMTAB
2229          POP     H               (HL) = #DESTFB+FB.NAM
2230
2231
2232  *       SEE IF SOURCE AND DEST FILE ON SAME DEVICE
2233
2234          PUSH    D               SAVE #NAMTAB (SOURCE NAME)
2235          MVI     C,3
2236          CALL    $COMP           COMPARE DEVICES
2237          MVI     A,EC.DNC        DEVICES NOT CONSISTANT
2238          JNE     ERROR
2239
2240  *       SEE IF TARGET ALREADY EXISTS
2241
2242          LXI     H,RENA
2243          MVI     A,CN.SOU
2244          DB      SYSCALL,.OPENR
2245          LXI     H,RENA-FB.NAM
2246          JC      REN2            HAVE AN ERROR (AS WE SHOULD)
2247          MVI     A,EC.FAP        FILE AALREADY PRESENT
2248          JMP     $FERROR         ALREADY THERE
2249
2250  REN2    CPI     EC.FNF          MUST BE NOT FOUND
2251          JNE     $FERROR         OTHER ERROR
2252          POP     H               (HL) = SOURCE NAME
2253          LXI     B,RENA          (BC) = NEW (TARGET) NAME
2254          DB      SYSCALL,.RENAM  RENAME IT
2255          JC      NAMERR          ERROR ON RENAME
2256
2257  *       REMOVE NAME FROM NAMTAB
2258
2259          CALL    REN             REMOVE ENTRY FROM NAMTAB
2260          LHLD    NAMTLEN
2261          MOV     A,H
2262          ORA     L
2263          JNZ     REN1
2264          RET
2265
2266  RENA    DS      FB.NAML         FILE NAME WORK AREA
```

```
                        2267            ENDIF
```

```
                                2270  ***     LIST - INDEX DIRECTORY.
                                2271  *
                                2272  *       DEST=SOURCE/LIST
                                2273  *                 /BRIEF
                                2274  *
                                2275  *       THESE SWITCHES CAUSE THE DIRECTORY CONTENTS OF THE SPECIFIED FILE(S)
                                2276  *       TO BE LISTED
                                2277  *
                                2278  *       IN /LI FIRM, THE OUTPUT IS:
                                2279  *
                                2280  *       NAME    EXT     SIZE      DATE          FLAGS
                                2281  *       XXX     .XXX     NNN    DD-MMM-YY       CWS
                                2282  *        .       .       .         .             .
                                2283  *        .       .       .         .             .
                                2284  *        .       .       .         .             .
                                2285  *              NNN FILES USING MMM SECTORS, XXX FREE
                                2286  *
                                2287  *       IN /BR FORM, ONLY THE NAME AND EXTENSION ARE LISTED,
                                2288  *       4 ACROSS THE PAGE.
                                2289  *
                                2290  *       SPECIAL CONSIDERATIONS:
                                2291  *
                                2292  *       A NULL NAME OR EXTENSION IS TAKEN AS '*' (WILDCARD)
                                2293  *
                                2294  *       IMPLIMENTATION:
                                2295  *
                                2296  *       A FILE LIST OF SOURCE FILES IS BUILT, THE DEVICE DIRECTORY FILE
                                2297  *       IS THEN READ, AND EACH FILE IN IT IS CHECKED FOR A MATCH
                                2298  *       AGAINST ANY SOURCE SPECIFICATIONS. ELIGIBLE FILES ARE LISTED.
                                2299
                                2300
046.371  041 000 000           2301  LIST    LXI     H,0
046.374  303 002 047           2302          JMP     LIST1
                                2303
046.377  041 001 000           2304  BRIEF   LXI     H,1
                                2305  *       JMP     LIST1
                                2306
047.002  042 107 050           2307  LIST1   SHLD    LSTA            (LSTA) = 0 IF LIST, 1 IF /BRIEF
000.000                         2308          ERRNZ   LSTB-LSTA-1     LSTB = FILE COUNT
047.005  041 000 000           2309          LXI     H,0
047.010  042 111 050           2310          SHLD    LSTC            CLEAR SECTORS USED COUNT
047.013  315 252 060           2311          CALL    $MOVEL
047.016  011 000 277           2312          DW      9,S.DATE,LSTG1  SET DATE IN HEADING
                                2313
                                2314  *       CRACK DESTINATION FILE NAMES
                                2315
000.001                         2316          IF      .PIP.
                                2317          CALL    DDF             DECODE DEST FILE NAME
                                2318          JC      ERROR           FILE NAME ERROR
                                2319          ANA     A
                                2320          MVI     A,FEC.IUW       ILLEGAL USE OF WILDCARD IN DEST
                                2321          JZ      ERROR
                                2322          ENDIF
                                2323
                                2324  *       BUILD LIST OF SPECIFICATIONS
                                2325
```

```
047.024  315 273 050  2326           CALL    BLS             BUILD LIST OF SOURCE SPECS
047.027  332 275 052  2327           JC      ERROR           ERROR IN LIST
047.032  001 003 000  2328           LXI     B,3
047.035  041 224 062  2329           LXI     H,DIRNAM
047.040  315 252 030  2330           CALL    $MOVE           MOVE DEVICE CODE INTO DIRECT.SYS NAME
047.043  041 226 062  2331           LXI     H,DIRNAM+2
047.046  176          2332           MOV     A,M             SEE IF UNIT NUMBER OMITTED
047.047  247          2333           ANA     A
047.050  302 055 047  2334           JNZ     LIST1.5         SPECIFIED
047.053  066 060      2335           MVI     M,'0'           DONT ALLOW NULL NUMBER
                      2336
                      2337  *        GET ADDRESS OF DEVICE'S GRT
                      2338
047.055  041 224 062  2339  LIST1.5  LXI     H,DIRNAM        (HL) = # OF XXX:DIRECT.SYS (XXX = DEVICE)
047.060  001 113 050  2340           LXI     B,LSTD          (BC) = ADDRESS FOR RETURN INFO
047.063  377 053      2341           DB      SYSCALL,.DECODE         DECODE NAME
047.065  332 275 052  2342           JC      ERROR           UNKNOWN DEVICE
047.070  072 113 050  2343           LDA     LSTD+0
047.073  346 001      2344           ANI     DT.DD
047.075  076 005      2345           MVI     A,EC.DNS
047.077  312 275 052  2346           JZ      ERROR           NOT DIRECTORY DEVICE
047.102  052 134 050  2347           LHLD    LSTD+17         (HL) = DEV TBL ADDR
                      2348
047.105  315 307 057  2349           CALL    $INDLB
047.110  007 000      2350           DW      DEV.SPG
047.112  062 145 050  2351           STA     LSTF            SAVE  SECTORS PER GROUP
                      2352
047.115  021 012 000  2353           LXI     D,DEV.UNT
047.120  031          2354           DAD     D
047.121  072 116 050  2355           LDA     LSTD+3
047.124  315 027 041  2356           CALL    S.GUP           HL = UNIT TABLE POINTER
                      2357
047.127  315 234 030  2358           CALL    $INDL
047.132  001 000      2359           DW      UNT.GRT
047.134  353          2360           XCHG
047.135  042 143 050  2361           SHLD    LSTE            SAVE  GRT ADDRESS
047.140  353          2362           XCHG
                      2363
                      2364  *        OPEN DEVICE'S DIRECTORY
                      2365
047.141  041 224 062  2366           LXI     H,DIRNAM
047.144  076 002      2367           MVI     A,CN.DIR
047.146  377 042      2368           DB      SYSCALL,.OPENR
047.150  076 200      2369           MVI     A,PEC.DF        DEVICE FORMAT ERROR
047.152  332 275 052  2370           JC      ERROR           CANT OPEN DIRECTORY
                      2371
                      2372
                      2373  *        OPEN OUTPUT FILE
                      2374
000.001               2375           IF      .PIP.
                      2376           LXI     H,DESTFB
                      2377           CALL    $FOPEW          OPEN FOR WRITE
                      2378           ENDIF
                      2379
                      2380  *        GENERATE HEADING
                      2381
```

```
047.155  001 001 000  2382           LXI     B,1             (BC) = TEXT COUNT
047.160  021 146 050  2383           LXI     D,LSTG          (DE) = TEXT ADDRESS
047.163  072 107 050  2384           LDA     LSTA
047.166  247          2385           ANA     A
047.167  302 174 047  2386           JNZ     LIST2           IS SHORT
047.172  016 051      2387           MVI     C,LSTGL         PRINT FULL HEADING
000.001               2388           IF      .PIP.
                      2389  LIST2    CALL    $FWRIB          WRITE HEADING
                      2390           ELSE
047.174  171          2391  LIST2    MOV     A,C
047.175  353          2392           XCHG                    (HL) = LINE ADDRESS
047.176  315 013 057  2393           CALL    $TYPCC          PRINT ON CONSOLE
                      2394           ENDIF
                      2395
                      2396  *        READ DIRECTORY BLOCKS, LOOKING FOR FILE MATCHES
                      2397
047.201  001 000 002  2398  LIST3    LXI     B,512
047.204  315 071 056  2399           CALL    GDWP            DE = DIRECTORY WORKSPACE POINTER  /79.11.GC/
047.207  076 002      2400           MVI     A,CN.DIR
047.211  325          2401           PUSH    D                                                 /79.11.GC/
047.212  377 004      2402           DB      SYSCALL,.READ
047.214  321          2403           POP     D               DE = DIRECOTRY WORKSPACE          /79.11.GC/
047.215  332 367 047  2404           JC      LIST9           ALL DONE
                      2405
                      2406  *        CHECK NEXT ENTRY IN NAMTAB AGAINST DIRECTORY ENTRY.
                      2407  *        (DE) = DIRECTORY BUFFER POINTER
                      2408
047.220  032          2409  LIST4    LDAX    D               (A) = FIRST CHARACTER OF NAME
047.221  247          2410           ANA     A
047.222  312 201 047  2411           JZ      LIST3           END OF THIS BUFFER
047.225  074          2412           INR     A
000.000               2413           ERRNZ   DF.EMP-377Q
047.226  312 321 047  2414           JZ      LIST7           THIS ENTRY IS EMPTY
047.231  074          2415           INR     A
047.232  312 367 047  2416           JZ      LIST9           NO MORE ENTRYS IN DIRECTORY
047.235  353          2417           XCHG
047.236  315 200 053  2418           CALL    CFE             CHECK FILE ELIGIBILITY
047.241  353          2419           XCHG
047.242  302 321 047  2420           JNE     LIST7           NOT ELIGIBLE
047.245  041 132 063  2421           LXI     H,NAMTAB
                      2422
047.250  345          2423  LIST5    PUSH    H
047.251  325          2424           PUSH    D               SAVE ADDRESS OF FILE AND PATTERN
047.252  315 005 054  2425           CALL    CAD             CONVERT ASCII NAMTAB ENTRY TO DIRECTORY FORMAT
047.255  021 342 062  2426           LXI     D,FIO.DIR+DIR.NAM       (DE) = NAMTAB PATTERN
047.260  341          2427           POP     H
047.261  345          2428           PUSH    H               (HL) = DIRECTORY PATTERN
047.262  006 013      2429           MVI     B,8+3           CHECK FOR MATCH
047.264  315 253 053  2430           CALL    CWM             CHECK FOR WILDCARD MATCH
047.267  321          2431  LIST6    POP     D
047.270  341          2432           POP     H
047.271  312 350 047  2433           JE      LIST8           GOT FILE TO LIST
047.274  001 021 000  2434           LXI     B,FB.NAML
047.277  011          2435           DAD     B               ADVANCE PAST ENTRY IN NAMTAB
                      2436
                      2437  *        SEE IF AT END OF NAMTAB
```

```
                                2438
047.300  325                   2439            PUSH    D
047.301  353                   2440            XCHG                    (DE) = NEW ADDRESS
047.302  052 302 062           2441            LHLD    NAMTLEN
047.305  001 132 063           2442            LXI     B,NAMTAB
047.310  011                   2443            DAD     B               (HL) = LWA+1 OF TABLE
047.311  353                   2444            XCHG
047.312  315 216 030           2445            CALL    $CDEHL          COMPARE
047.315  321                   2446            POP     D
047.316  302 250 047           2447            JNE     LIST5           MORE IN TABLE
                                2448
                                2449 *          FILE DOESNT MATCH ANY SELECTED FILE, PASS TO NEXT ONE
                                2450
047.321  353                   2451 LIST7      XCHG                    (HL) = DIR BUFFER ADDRESS
                                2452
047.322  345                   2453            PUSH    H                                       /79.11.GC/
047.323  315 077 056           2454            CALL    GDWP.           HL = DIRECTORY WORKSPACE PTR.   /79.11.GC/
047.326  315 307 057           2455            CALL    $INDLB          A  = DIR. ENTRY LENGTH       /79.11.GC/
047.331  373 001               2456            DW      DIS.ENL                                 /79.11.GC/
047.333  341                   2457            POP     H                                       /79.11.GC/
                                2458
047.334  315 101 030           2459            CALL    $DADA.          ADVANCE
047.337  176                   2460            MOV     A,M
047.340  247                   2461            ANA     A
047.341  353                   2462            XCHG
047.342  302 220 047           2463            JNZ     LIST4           TRY THIS ONE
047.345  303 201 047           2464            JMP     LIST3           READ ANOTHER BLOCK
                                2465
                                2466 *          HAVE FILE TO LIST
                                2467
047.350  325                   2468 LIST8      PUSH    D               SAVE DIR POINTER
047.351  072 145 050           2469            LDA     LSTF            (A) = SECTORS PER GROUP THIS DEVICE
047.354  315 022 051           2470            CALL    PFI             PRINT FILE INFO
047.357  321                   2471            POP     D
047.360  041 110 050           2472            LXI     H,LSTB
047.363  064                   2473            INR     M               COUNT FILE
047.364  303 321 047           2474            JMP     LIST7           ADVANCE TO NEXT FILE
                                2475
                                2476 *          ALL DONE, CLOSE DIRECTORY FILE
                                2477
047.367  076 002               2478 LIST9      MVI     A,CN.DIR
047.371  377 046               2479            DB      SYSCALL+.CLOSE  CLOSE FILE
047.373  001 001 000           2480            LXI     B,1             ASSUME SHORT FORM, JUST WRITE NL
047.376  072 107 050           2481            LDA     LSTA            (A) = FORM FLAG
050.001  247                   2482            ANA     A
050.002  302 072 050           2483            JNZ     LIST10          IS SHORT, NO TRAILER
                                2484
                                2485 *          PRINT SUMMARY:
                                2486 *
                                2487 *          NNN FILES, USING XXX SECTORS, YYY FREE
                                2488
050.005  072 110 050           2489            LDA     LSTB
050.010  117                   2490            MOV     C,A
050.011  006 000               2491            MVI     B,0             (BC) = FILE COUNT
050.013  076 003               2492            MVI     A,3
050.015  041 223 050           2493            LXI     H,LSTH1
```

```
050.020  315 177 060  2494          CALL    $UDDN           FILE COUNT
050.023  052 111 050  2495          LHLD    LSTC
050.026  104          2496          MOV     B,H
050.027  115          2497          MOV     C,L             (BC) = SECTOR COUNT
050.030  041 244 050  2498          LXI     H,LSTH2
050.033  076 003      2499          MVI     A,3
050.035  315 177 060  2500          CALL    $UDDN           USED COUNT
050.040  052 143 050  2501          LHLD    LSTE
050.043  176          2502          MOV     A,M
050.044  315 220 053  2503          CALL    CFS             FOLLOW GRT CHAIN
050.047  072 145 050  2504          LDA     LSTF
050.052  315 007 031  2505          CALL    $MU86           (HL) = SECTORS FREE
050.055  104          2506          MOV     B,H
050.056  115          2507          MOV     C,L
050.057  041 261 050  2508          LXI     H,LSTH3
050.062  076 003      2509          MVI     A,3
050.064  315 177 060  2510          CALL    $UDDN           UNPACK FREE
050.067  001 054 000  2511          LXI     B,LSTHL
050.072  021 217 050  2512  LIST10  LXI     D,LSTH
050.075  072 222 062  2513          LDA     SUPRES
050.100  247          2514          ANA     A
000.001               2515          IF      .PIP.
                      2516          LXI     H,DESTFB
                      2517          JNZ     $FCLO           CLOSE AND EXIT, SUMMARY SUPPRESSED
                      2518          CALL    $FWRIB          WRITE TRAILER
                      2519
                      2520  *       ALL DONE. CLOSE OUTPUT FILE
                      2521
                      2522          JMP     $FCLO           CLOSE AND EXIT
                      2523          ELSE
050.101  300          2524          RNZ                     NOT TO SUMMARYIZE
050.102  171          2525          MOV     A,C             (A) = COUNT
050.103  353          2526          XCHG                    (HL) = ADDRESS
050.104  303 013 057  2527          JMP     $TYPCC          TYPE TEXT AND EXIT
                      2528          ENDIF
                      2529
050.107  000          2530  LSTA    DB      0               <>0 IFF SHORT FORM
                      2531
050.110  000          2532  LSTB    DB      0               FILE COUNT
050.111  000 000      2533  LSTC    DW      0               SECTORS USED
050.113               2534  LSTD    DS      24              FILE NAME DECODE AREA
050.143  000 000      2535  LSTE    DW      0               GRT ADDRESS
050.145  000          2536  LSTF    DB      0               SECTORS PER GROUP FOR THIS DEVICE
050.146  012 116 141  2537  LSTG    DB      NL,'Name',TAB,'.Ext',TAB,'Size',TAB,'  Date',TAB,TAB,'Flags',TAB
050.204               2538  LSTG1   DS      9               DATE
050.215  012 012      2539          DB      NL,NL
000.051               2540  LSTGL   EQU     *-LSTG
                      2541
050.217  012 040 040  2542  LSTH    DB      NL,'   '                FIRST CHARACTER MUST BE <NL>
050.223  116 116 116  2543  LSTH1   DB      'NNN Files, Usins '
050.244  115 115 115  2544  LSTH2   DB      'MMM Sectors ('
050.261  130 130 130  2545  LSTH3   DB      'XXX Free)',NL
000.054               2546  LSTHL   EQU     *-LSTH
```

```
                                2548  **      BLS - BUILD LIST OF SOURCE FILES.
                                2549  *
                                2550  *       BLS BUILDS A LIST OF SOURCE FILES INTO *NAMTAB*
                                2551  *       NULL FIELDS ARE SET TO WILDCARDS. BLS REQUIRES THAT ALL
                                2552  *       FILES SPECIFIED HAVE THE SAME DEVICE.
                                2553  *
                                2554  *       IF THE COMMAND LINE CONTAINS NO FILES, BUT CONTAINS AT LEAST
                                2555  *       ONE BLANK (AS WOULD BE THE CASE IN PROCESSING THE /LIST SWITCH, SINCE
                                2556  *       THE '/LIST' IS REPLACED WITH BLANKS) A FILE NAME OF ????????.???
                                2557  *       IS DECODED.
                                2558  *       ENTRY   NAMTAB EMPTY
                                2559  *       EXIT    'C' CLEAR IF OK
                                2560  *                (DE) = #BLSA = 3 CHARACTER DEVICE NAME
                                2561  *               'C' SET IF ERROR
                                2562  *                (A) = ERROR MESSAGE
                                2563  *       USES    ALL
                                2564
                                2565
050.273  315 252 060            2566  BLS     CALL    $MOVEL
050.276  003 000 015            2567          DW      3,BLSC,BLSA     SET INITIAL DEFAULT DEVICE
050.304  041 000 000            2568          LXI     H,0
050.307  042 302 062            2569          SHLD    NAMTLEN         CLEAR NAMTAB
050.312  076 377                2570          MVI     A,377Q
050.314  062 014 051            2571          STA     BLSB            FLAG PROCESSING OF FIRST FILE NAME
050.317  315 135 056            2572          CALL    LSN             LOCATE SOURCE NAMES
                                2573
                                2574  *       CRACK THE NEXT NAME
                                2575
050.322  176                   2576  BLS1    MOV     A,M
050.323  021 006 051            2577          LXI     D,BLSA          (DE) = DEFAULT ADDRESS
050.326  247                   2578          ANA     A
050.327  310                   2579          RZ                      NO MORE NAMES
050.330  315 156 057            2580          CALL    $SOB            SEE IF ALL NULL
050.333  176                   2581          MOV     A,M
050.334  247                   2582          ANA     A
050.335  302 343 050            2583          JNZ     BLS2            NOT ALL NULL
050.340  041 015 051            2584          LXI     H,BLSC          USE DEFAULT DEVICE
050.343  315 011 054            2585  BLS2    CALL    CAD.            CONVERT ASCII NAME TO DIRECTORY FORMAT
050.346  330                   2586          RC                      ERROR
                                2587
                                2588  *       IF FIRST NAME, RECORD DEVICE
                                2589  *       IF NOT FIRST, COMPARE DEVICE AGAINST FIRST DEVICE
                                2590
050.347  345                   2591          PUSH    H
050.350  021 337 062            2592          LXI     D,PIO.DEV
050.353  041 006 051            2593          LXI     H,BLSA
050.356  001 003 000            2594          LXI     B,3             SETUP COUNT, FROM AND TO
000.001                         2595          IF      .PIP.
                                2596          LDA     BLSB
                                2597          ANA     A
                                2598          JP      BLS3            NOT 1ST FILE
                                2599          CALL    $MOVE           MOVE IN REQUIRED DEVICE FOR REMAINING FILES
                                2600          XRA     A
                                2601          STA     BLSB            FLAG 1ST NAME PROCESSED
                                2602          JMP     BLS4
                                2603          ENDIF
```

```
                                2604
050.361  315 060 030  2605  BLS3    CALL    $COMP           SEE IF THIS DEVICE SAME AS PREVIOUS
050.364  312 374 050  2606          JE      BLS4            OK
050.367  076 201      2607          MVI     A,PEC.DNC       MULTIPLE DEVICES ARE ILLEGAL
050.371  067          2608          STC
050.372  341          2609          POP     H
050.373  311          2610          RET                     RETURN WITH ERROR
                                2611
                                2612  *       GOT NAME DECODED, ENTER IN NAMTAB
                                2613
050.374  315 027 053  2614  BLS4    CALL    AEN             ADD ENTRY TO NAMTAB
050.377  341          2615          POP     H
051.000  315 324 056  2616          CALL    SFS             SKIP FILE SEPERATOR (BLANKS AND/OR COMMA)
051.003  303 322 050  2617          JMP     BLS1            SEE IF MORE
                                2618
051.006  123 131 060  2619  BLSA    DB      'SY0',200Q,200Q,200Q
051.014  000          2620  BLSB    DB      0               FIRST FILE NAME FLAG
051.015  123 131 060  2621  BLSC    DB      'SY0:',0                DEFAULT DEVICE
```

```
                                2623  **      PFI - PRINT FILE INFO.
                                2624  *
                                2625  *       PFI DECODES A DIRECTORY ENTRY INTO A CODED LINE, THEN
                                2626  *       WRITES IT TO 'DESTFB'.
                                2627  *
                                2628  *       THE PRODUCED FORMAT DEPENDS UPON THE LISTING FORMAT FLAG,
                                2629  *       LSTA.
                                2630  *
                                2631  *       SHORT FORM:
                                2632  *
                                2633  *       NAME    .EXT    (TAB)
                                2634  *
                                2635  *       LONG FORM:
                                2636  *
                                2637  *       NAME    .EXT    SIZE    DATE    FLAGS   (NL)
                                2638  *
                                2639  *       ENTRY   (A) = SECTORS PER GROUP FOR THIS DEVICE
                                2640  *               (DE) = DIRECTORY ENTRY POINTER
                                2641  *       EXIT    IF LONG FORM, SECTOR COUNT IS ACCUMULATED IN LSTC
                                2642  *       USES    ALL
                                2643
                                2644
051.022  062 364 051  2645  PFI     STA     PFIC            SAVE SECTORS PER GROUP
051.025  041 302 051  2646          LXI     H,PFIA
051.030  016 010      2647          MVI     C,8
051.032  315 264 051  2648          CALL    PFI20           COPY NAME
051.035  312 043 051  2649          JZ      PFI1            ALL 8 CHARACTERS
051.040  066 011      2650          MVI     M,TAB
051.042  043          2651          INX     H
051.043  066 056      2652  PFI1    MVI     M,'.'
051.045  043          2653          INX     H
051.046  016 003      2654          MVI     C,3
051.050  315 264 051  2655          CALL    PFI20           COPY EXTENSION
051.053  066 011      2656          MVI     M,TAB
```

```
051.055  043           2657          INX     H
051.056  072 107 050   2658          LDA     LSTA
051.061  247           2659          ANA     A
051.062  312 107 051   2660          JZ      PFI3            IS LONG FORM
                       2661
                       2662  *       IS SHORT FORM, SEE IF NEED TO END LINE
                       2663
051.065  074           2664          INR     A
051.066  376 005       2665          CPI     5
051.070  302 101 051   2666          JNE     PFI2            NOT TIME YET
051.073  053           2667          DCX     H
051.074  066 012       2668          MVI     M,NL
051.076  043           2669          INX     H               TIME TO END LINE
051.077  076 001       2670          MVI     A,1
051.101  062 107 050   2671  PFI2    STA     LSTA            RESET COUNT
051.104  303 244 051   2672          JMP     PFI6            OUTPUT TO FILE
                       2673
                       2674  *       IS LONG FORM.
                       2675
051.107  001 005 000   2676  PFI3    LXI     B,DIR.FGN-DIR.EXT-3
051.112  353           2677          XCHG                    (DE) = LINE ADDR, (HL) = #PIO.DIR+DIR.EXT+3
051.113  011           2678          DAD     B               (HL) = #DIR.FGN
051.114  176           2679          MOV     A,M             (A) = (DIR.FGN)
051.115  043           2680          INX     H
051.116  043           2681          INX     H
051.117  116           2682          MOV     C,M             (C) = DIR.LSI = SECTORS USED IN LAST GROUP
000.000               2683          ERRNZ   DIR.LSI-DIR.FGN-2
051.120  353           2684          XCHG                    (DE) = ADDRESS OF LSI
051.121  325           2685          PUSH    D               SAVE #DIR.LSI
051.122  345           2686          PUSH    H               SAVE LINE ADDRESS
051.123  052 143 050   2687          LHLD    LSTE
051.126  157           2688          MOV     L,A
051.127  176           2689          MOV     A,M
051.130  315 220 053   2690          CALL    CFS             COMPUTE FILE ISZE
051.133  072 364 051   2691          LDA     PFIC            (A) = SECTORS PER GROUP
051.136  315 007 031   2692          CALL    $MU86           (HL) = SECTORS USED (EXCEPT FOR THOSE IN LAST GROUP)
051.141  006 000       2693          MVI     B,0
051.143  011           2694          DAD     B               (HL) = SECTORS USED
051.144  104           2695          MOV     B,H
051.145  115           2696          MOV     C,L             (BC) = SECTORS USED COUNT
051.146  052 111 050   2697          LHLD    LSTC
051.151  011           2698          DAD     B
051.152  042 111 050   2699          SHLD    LSTC            ACCUMULATE COUNT OF SECTORS
051.155  341           2700          POP     H               (HL) = LINE ADDRESS
051.156  076 003       2701          MVI     A,3             3 DIGITS MAX
051.160  315 177 060   2702          CALL    $UDDN           UNPACK COUNT
051.163  066 011       2703          MVI     M,TAB
051.165  043           2704          INX     H
051.166  321           2705          POP     D               (DE) = #DIR.LSI
                       2706
                       2707  *       TYPE DATE
                       2708
051.167  353           2709          XCHG
000.000               2710          ERRNZ   DIR.CRD-DIR.LSI-1
051.170  043           2711          INX     H               (HL) = #DIR.CRD
051.171  345           2712          PUSH    H
```

```
051.172  315 211 030  2713          CALL    $HLIHL
051.175  353          2714          XCHG
051.176  315 012 060  2715          CALL    $DAD            DECODE AUGUSTAN DATE
                      2716
                      2717  *       CODE FLAGS
                      2718
051.201  353          2719          XCHG                    (DE) = LINE ADDRESS
051.202  341          2720          POP     H               (HL) = $DIR.CRD
051.203  001 373 377  2721          LXI     B,DIR.FLG-DIR.CRD
051.206  011          2722          DAD     B               (HL) = ADDRESS OF DIRFLG
051.207  176          2723          MOV     A,M             (A) = FLAGS
051.210  353          2724          XCHG                    (HL) = LINE ADDRESS
051.211  247          2725          ANA     A
051.212  312 241 051  2726          JZ      PFI5.5          NO FLAGS
051.215  066 011      2727          MVI     M,TAB           TAB BEFORE FLAGS
051.217  043          2728          INX     H
051.220  021 354 051  2729          LXI     D,PFIB
051.223  207          2730  PFI4    ADD     A
051.224  322 234 051  2731          JNC     PFI5            NOT SET
051.227  365          2732          PUSH    PSW             SAVE FLAGS
051.230  032          2733          LDAX    D
051.231  167          2734          MOV     M,A
051.232  361          2735          POP     PSW             RESTORE FLAGS
051.233  043          2736          INX     H
051.234  023          2737  PFI5    INX     D               SET FLAG
051.235  247          2738          ANA     A
051.236  302 223 051  2739          JNZ     PFI4            MORE FLAGS SET
051.241  066 012      2740  PFI5.5  MVI     M,NL
051.243  043          2741          INX     H
                      2742
                      2743  *       LINE ALL BUILT. WRITE TO DESTFB
                      2744
051.244  021 076 326  2745  PFI6    LXI     D,-PFIA
051.247  031          2746          DAD     D
000.001               2747          IF      .PIP.
                      2748          MOV     B,H
                      2749          MOV     C,L             (BC) = LEN
                      2750          LXI     D,PFIA          (DE) = DATA FWA
                      2751          LXI     H,DESTFB
                      2752          JMP     $FWRIB          WRITE AND EXIT
                      2753          ELSE
051.250  175          2754          MOV     A,L             (A) = COUNT
051.251  041 302 051  2755          LXI     H,PFIA
051.254  303 013 057  2756          JMP     $TYPCC          TYPE LINE AND EXIT
                      2757          ENDIF


                      2759  **      PFI20 - COPY FILE NAME.
                      2760  *
                      2761  *       PFI20 COPIES A NAME FILED FROM THE DIRECTORY ENTRY TO A CODED
                      2762  *       LINE
                      2763  *
                      2764  *       EENTRY  (DE) = DIRECTORY ADDRESS
                      2765  *               (C) = NAME LENGTH
                      2766  *               (HL) = LINE ADDRESS
```

```
                                2767  *       EXIT    (DE) = (DE) + (C)
                                2768  *               'Z' SET IF MAX CHARACTERS COPIED
                                2769  *       USES    A,F,C,D,E,H,L
                                2770
                                2771
051.257  167                   2772  PFI19   MOV     M,A             COPY
051.260  043                   2773          INX     H
051.261  023                   2774          INX     D
051.262  015                   2775          DCR     C
051.263  310                   2776          RZ                      ALL COPIED
051.264  032                   2777  PFI20   LDAX    D
051.265  247                   2778          ANA     A
051.266  302 257 051           2779          JNZ     PFI19           GOT CHAR
                                2780
                                2781  *       NO NAME, (C) = COUNT LEFT
                                2782
051.271  173                   2783          MOV     A,E
051.272  201                   2784          ADD     C
051.273  137                   2785          MOV     E,A
051.274  172                   2786          MOV     A,D
051.275  316 000               2787          ACI     0
051.277  127                   2788          MOV     D,A
051.300  263                   2789          ORA     E               CLEAR 'Z'
051.301  311                   2790          RET
                                2791
051.302                         2792  PFIA    DS      0               BUFFER AREA FOR LINE BUILD
051.302  130 130 130            2793          DB      'XXXXXXXX.YYY   NNN     DD-MMM-YY'
051.334  011 011 106            2794          DB      '               FLAGS         '
051.354  123 114 127            2795  PFIB    DB      'SLW'           CODES
051.357  040 061 062            2796  PFIB1   DB      ' 1234'          ('C' FOR CONTIGUOUS IS OPTIONAL)
000.000                         2797          ERRNZ   DIF.SYS-200Q
000.000                         2798          ERRNZ   DIF.LOC-100Q
000.000                         2799          ERRNZ   DIF.WP-400
000.000                         2800          ERRNZ   DIF.CNT-20Q
051.364  000                   2801  PFIC    DB      0               SECTORS PER GROUP FOR THIS DEVICE
```

```
                                2804  ***     VERSN   - PIP VERSION INFORMATION
                                2805  *
                                2806  *       DEST=/VERSION]
                                2807  *
                                2808  *       PRINT THE 'PIP' VERSION INFORMATION TO THE 'DEST' FILE.
                                2809  *
                                2810
        051.365                 2811  VERSN   EQU     *
                                2812
        051.365  315 236 053   2813          CALL    CTS             CHECK FOR TARGET FILE SPECIFICATION
        051.370  067            2814          STC
        051.371  302 275 052   2815          JNZ     ERROR           TARGET FILE SPECIFICATION ILLEGAL
        051.374  041 012 063   2816          LXI     H,LINE
        051.377  315 156 057   2817          CALL    $SOB            SKIP OVER ALL THE BLANKS ($DRS TURNS SWITCHES
        052.002  176            2818          MOV     A,M              TO BLANKS)
        052.003  247            2819          ANA     A
        052.004  076 207        2820          MVI     A,PEC.SFI       SOURCE FILE ILLEGAL
        052.006  067            2821          STC
        052.007  302 275 052   2822          JNZ     ERROR           ONLY ALLOW SWITCH ON LINE
        052.012  315 136 031   2823          CALL    $TYPTX
                                2824
        000.001                 2825          IF      .PIP.
                                2826          DB      'PIP'
                                2827          ELSE
        052.015  117 116 105   2828          DB      'ONECOPY'
                                2829          ENDIF
                                2830
        052.024  011 126 145   2831          DB      TAB,'Version:  '
        052.037  061 056 066   2832          DB      VERS/16+'0','.',VERS&00001111B+'0'
        052.042  212            2833          DB      ENL
                                2834
        052.043  311            2835          RET
```

```
                                2838  **      ERROR PROCESSING ROUTINES
                                2839  *
                                2840
                                2841  ***     NAMERR - FILE TYPE ERROR, OCCURRED ON FILE WHOSE NAME
                                2842  *       IS NEXT UP IN NAMTAB.
                                2843  *
                                2844  *       PROCESS VIA $FERROR
                                2845
000.001                         2846          IF      .PIP.
                                2847  NAMERR  LXI     H,NAMTAB-FB.NAM
                                2848          JMP     $FERROR
                                2849          ELSE
052.044  052 306 062            2850  NAMERR  LHLD    NAMTPTR
052.047  001 366 377            2851          LXI     B,-FB.NAM
052.052  011                   2852          DAD     B
052.053  303 135 062            2853          JMP     $FERROR
                                2854
                                2855  **      ERROR ON FILE IN DESTFB
                                2856
052.056  041 247 062            2857  DESTERR LXI     H,DESTFB
052.061  303 135 062            2858          JMP     $FERROR
                                2859          ENDIF
                                2860
                                2861  **      INTERNAL ERRORS. SHOULD NOT OCCUR.
                                2862
052.064  076 061                2863  IERR1   MVI     A,'1'
052.066  303 103 052            2864          JMP     INTERR
                                2865
052.071  076 062                2866  IERR2   MVI     A,'2'
052.073  303 103 052            2867          JMP     INTERR
052.076  076 063                2868  IERR3   MVI     A,'3'
052.100  303 103 052            2869          JMP     INTERR
                                2870
                                2871
052.103  365                    2872  INTERR  PUSH    PSW             SAVE CODE
052.104  315 136 031            2873          CALL    $TYPTX
052.107  007 012 120            2874          DB      BELL,NL,'PIP INTERNAL ERROR ','$'+200Q
052.135  361                    2875          POP     PSW
052.136  315 303 060            2876          CALL    $WCHAR
052.141  315 136 031            2877          CALL    $TYPTX
052.144  012 124 110            2878          DB      NL,'THIS ERROR SHOULD NOT OCCUR. CONTACT HEATH TECHNICAL'
052.231  012 103 117            2879          DB      NL,'CORRESPONDENCE FOR ASSISTANCE.',NL
052.271  076 001                2880          MVI     A,1
052.273  377 000                2881          DB      SYSCALL,.EXIT           ABORT
```

```
                                2883  **      ERROR - GENERAL AND SYNTAX ERRORS NOT DIRECTLY ASSOCIATED
                                2884  *       WITH A VALID FILE NAME.
                                2885
                                2886
052.275  365                   2887  ERROR   PUSH    PSW             SAVE CODE
052.276  315 136 031            2888          CALL    $TYPTX
052.301  007 105 122            2889          DB      BELL,'ERROR -',' '+200Q
052.312  361                    2890          POP     PSW
052.313  247                    2891          ANA     A
052.314  372 326 052            2892          JM      ERROR1          IS PRODUCT ERROR
052.317  046 012                2893          MVI     H,NL            USE NL AS MESSAGE TRAIL CHAR
052.321  377 057                2894          DB      SYSCALL,.ERROR  LOOK UP SYSTEM ERROR
052.323  303 200 042            2895          JMP     RESTART
                                2896
                                2897  *       IS PRODUCT ERROR
                                2898
052.326  041 367 052            2899  ERROR1  LXI     H,ERRORA
052.331  276                    2900  ERROR2  CMP     M
052.332  043                    2901          INX     H
052.333  302 331 052            2902          JNE     ERROR2          FIND ERROR MESSAGE
000.000                         2903          IF      ONECOPY
052.336  315 136 031            2904          CALL    $TYPTX
052.341  007 117 116            2905          DB      BELL,'ONECOPY Error #',' '+200Q
                                2906          ENDIF
052.362  377 003                2907          DB      SYSCALL,.PRINT  PRINT MESSAGE
052.364  303 200 042            2908          JMP     RESTART
                                2909
052.367                         2910  ERRORA  DS      0               ERROR MESSAGES
000.001                         2911          IF      .PIP.
                                2912          DB      PEC.DF,'Device Format Error',ENL
                                2913          DB      PEC.DNC,'All Files Must Reside on the Same Device',ENL
                                2914          DB      PEC.TFI,'Destination File Specification is Illegal',ENL
                                2915          DB      PEC.CS,'Contradictory Switches Specified',ENL
                                2916          DB      PEC.IUW,'Illegal Use of Wildcard',ENL
                                2917          DB      PEC.IDF,'Illegal Destination File Format',ENL
                                2918          DB      PEC.SFI,'Source File Specification is Illegal',ENL
                                2919          ELSE
052.367  200 060 061            2920          DB      PEC.DF,'01',ENL
052.373  201 060 062            2921          DB      PEC.DNC,'02',ENL
052.377  203 060 063            2922          DB      PEC.TFI,'03',ENL
053.003  204 060 064            2923          DB      PEC.CS,'04',ENL
053.007  205 060 065            2924          DB      PEC.IUW,'05',ENL
053.013  206 060 066            2925          DB      PEC.IDF,'06',ENL
053.017  207 060 067            2926          DB      PEC.SFI,'07',ENL
053.023  210 060 070            2927          DB      PEC.FCI,'08',ENL
                                2928          ENDIF
```

```
                                2932  **      AEN - ADD ENTRY TO 'NAMTAB'
                                2933  *
                                2934  *       AEN EXPANDS THE FILE INFO IN PIO.XXX INTO A FILE DESCRIPTOR
                                2935  *       AND ENTERS IT IN THE NAMTAB TABLE.
                                2936  *
                                2937  *       ENTRY   NONE
                                2938  *       EXIT    'C' SET IF WILDCARD
                                2939  *       USES    ALL
                                2940
                                2941
053.027  041 101 053            2942  AEN     LXI     H,AENA
053.032  315 065 055            2943          CALL    CDA             CONVERT DIRECTORY FORMAT TO ASCII FORMAT
053.035  326 001                2944          SUI     1               'C' SET IF WILDCARD
053.037  365                    2945          PUSH    PSW             SAVE FLAG
053.040  052 302 062            2946          LHLD    NAMTLEN
053.043  001 021 000            2947          LXI     B,FB.NAML
053.046  011                    2948          DAD     B               INCREASE SIZE
053.047  042 302 062            2949          SHLD    NAMTLEN
053.052  353                    2950          XCHG                    (DE) = NEW LENGTH
053.053  052 304 062            2951          LHLD    NAMTMAX
053.056  175                    2952          MOV     A,L             SEE IF WILL OVERFLOW
053.057  223                    2953          SUB     E
053.060  174                    2954          MOV     A,H
053.061  232                    2955          SBB     D
053.062  334 103 056            2956          CC      INA             INCREASE NAMTAB ALLOCATION
053.065  041 111 063            2957          LXI     H,NAMTAB-FB.NAML
053.070  031                    2958          DAD     D               (HL) = *TO* ADDRESS
053.071  021 101 053            2959          LXI     D,AENA          (DE) = *FROM* ADDRESS
053.074  315 252 030            2960          CALL    $MOVE           MOVE ENTRY IN
053.077  361                    2961          POP     PSW             (PSW) = WILDCARD FLAG
053.100  311                    2962          RET
                                2963
053.101                         2964  AENA    DS      FB.NAML


                                2966  **      BSL - BUILD SOURCE FILE LIST.
                                2967  *
                                2968  *       BSL CRACKS THE LIST OF THE SOURCE FILES FROM THE COMMAND LINE AND
                                2969  *       BUILDS THEM INTO THE NAMTAB MANAGED TABLE.
                                2970  *       WILD CARDS ENCOUNTERED ARE EXPANDED.
                                2971  *
                                2972  *       ENTRY   (A) <> 0 IF TO ASK ABOUT '*.*' USE
                                2973  *       EXIT    'C' CLEAR IF OK
                                2974  *               'C' SET IF ERROR
                                2975  *                (A) = CODE
                                2976  *       USES    ALL
                                2977
                                2978
053.122  062 173 053            2979  BSL     STA     BSLA            SAVE ASK FLAG
053.125  315 135 056            2980          CALL    LSN             LOCATE SOURCE NAME
                                2981
                                2982  *       GO THROUGH SOURCE LIST CRACKING NAMES
                                2983
053.130  176                    2984  BSL1    MOV     A,M
```

```
053.131  247            2985          ANA     A
053.132  310            2986          RZ                      ALL DONE
053.133  021 310 062     2987          LXI     D,DEFALT
053.136  315 005 054     2988          CALL    CAD             CONVERT ASCII NAME TO DIRECTORY FORMAT
053.141  330            2989          RC                      ERROR
053.142  315 341 056     2990          CALL    SND             SET NEW DEFAULTS
053.145  345            2991          PUSH    H               SAVE LINE ADDRESS
053.146  072 173 053     2992          LDA     BSLA
053.151  247            2993          ANA     A
053.152  304 174 053     2994          CNZ     CCW             CHECK FOR COMPLETE WILDCARD (*.*)
053.155  332 200 042     2995          JC      RESTART         USER CHICKENED OUT                /79.12.GC/
053.160  315 156 055     2996          CALL    EWS             EXPAND WILDCARD SPECIFICATION
053.163  341            2997  BSL2    POP     H               RESTORE LINE ADDRESS
053.164  330            2998          RC                      USER REFUSED *.*
053.165  315 324 056     2999          CALL    SFS             SKIP FILE SEPERATOR (BLANKS AND/OR COMMA)
053.170  303 130 053     3000          JMP     BSL1            DO MORE
                        3001
053.173  000            3002  BSLA    DB      0               <>0 IF TO CHECK FOR *.*



                        3004  **      CCW - CHECK FOR COMPLETE WILDCARD.
                        3005  *
                        3006  *       CCW IS CALLED WITH A NAME CRACKED INTO PIO.XXX, TO SEE IF
                        3007  *       IT IS A *.* SPECIFICATION.
                        3008  *
                        3009  *       IF SO, CCW ASKS,
                        3010  *
                        3011  *       DELETE ALL FILES ON DEV: ?!? (Y/N)
                        3012  *
                        3013  *       THE USER REPLY IS ACCEPTED AND DECODED.
                        3014  *
                        3015  *       ENTRY   NONE
                        3016  *       EXIT    'C' CLEAR IF NOT *.*, OR 'Y' REPLIED
                        3017  *               'C' SET IF *.* AND NOT 'Y'
                        3018  *       USES    A,F,B,H,L
                        3019
                        3020
053.174  041 342 062    3021  CCW     LXI     H,PIO.DIR+DIR.NAM
000.001                 3022          IF      .PIP.
                        3023          MVI     B,8+3
                        3024          MVI     A,200Q
                        3025  CCW1    ANA     M               SEE IF ALL HAVE 200Q BIT SET
                        3026          INX     H
                        3027          DCR     B
                        3028          JNZ     CCW1
                        3029          ANA     A
                        3030          RP                      NOT *.*
                        3031
                        3032  *       IS *.*
                        3033
                        3034          CALL    $TYPTX
                        3035          DB      BELL,'!?! DELETE ALL FILES ON',' '+200Q
                        3036          LXI     H,PIO.DEV
                        3037          MVI     A,3
```

```
                        3038            CALL    $TYPCC          TYPE DEVICE NAME
                        3039            CALL    $TYPTX
                        3040            DB      ': (Y/N)?',' '+200Q
                        3041            LXI     H,DESTBUF
                        3042            CALL    $RTL.           READ REPLY
                        3043            LDA     DESTBUF
                        3044            CPI     'Y'
                        3045            RE                      IS OK
                        3046            STC
                        3047            MVI     A,PEC.IUW       FLAG ILLEGAL USE OF WILDCARD
                        3048            ENDIF
     053.177  311      3049            RET                     FORGET IT


                        3051    **      CFE - CHECK FILE ELIGIBILITY.
                        3052    *
                        3053    *       CFE CHECKS TO SEE IF A WILDCARD-SELECTED FILE IS ELIGIBLE
                        3054    *       FOR PROCESSING. IF THE FILE IS FLAGGED SYSTEM, AND /S IS NOT
                        3055    *       SPECIFIED, THE FILE IS NOT ELIGIBLE.
                        3056    *
                        3057    *       ENTRY   (HL) = DIRECTORY ENTRY POINTER
                        3058    *       EXIT    'Z' SET IF ELIGIBLE
                        3059    *       USES    A,F
                        3060
                        3061
     053.200  345       3062    CFE     PUSH    H
     053.201  076 016   3063            MVI     A,DIR.FLG
     053.203  315 101 030  3064         CALL    $DADA.
     053.206  176       3065            MOV     A,M             (A) = FLAG
     053.207  346 200   3066            ANI     DIF.SYS
     053.211  341       3067            POP     H
     053.212  310       3068            RZ                      ELIGIBLE
     053.213  072 223 062  3069        LDA     SYSTEM          CHECK /S FLAG
     053.216  247       3070            ANA     A
     053.217  311       3071            RET


                        3073    **      CFS - COMPUTE FILE SIZE
                        3074    *
                        3075    *       CFS COMPUTES THE SIZE OF A FILE. THE DEVICE'S GRT MUST BE IN
                        3076    *       THE 'GRT' BUFFER.
                        3077    *
                        3078    *       ENTRY   (A) = FIRST GROUP NUMBER
                        3079    *       EXIT    (DE) = SIZE
                        3080    *       USES    ALL
                        3081
                        3082
     053.220  052 143 050  3083  CFS     LHLD    LSTE
     053.223  021 000 000  3084  CFS.    LXI     D,0
     053.226  247       3085    CFS1    ANA     A
     053.227  310       3086            RZ                      ALL DONE
     053.230  157       3087            MOV     L,A
```

```
053.231  176          3088           MOV     A,M             (A) = NEXT GRT
053.232  023          3089           INX     D
053.233  303 226 053  3090           JMP     CFS1            TRY AGAIN


                      3092   **      CTS     -  CHECK TARGET FILE SPECIFICATION
                      3093   *
                      3094   *       CTS CHECKS FOR A TARGET FILE SPECIFICATION
                      3095   *
                      3096   *
                      3097   *       ENTRY   NONE
                      3098   *
                      3099   *       EXIT    (PSW)   = 'Z' SET IF NO TARGET FILE
                      3100   *                       = 'Z' CLEAR IF    TARGET FILE
                      3101   *                          (A)  = PEC.TFI ERROR CODE
                      3102   *
                      3103   *       USES    (PSW),(HL)
                      3104   *
                      3105
053.236  315 135 056  3106   CTS     CALL    LSN             (HL) = ADDRESS OF FIRST SOURCE NAME
053.241  021 366 314  3107           LXI     D,-LINE
053.244  031          3108           DAD     D               (HL) == 0 IF NO '=' IN COMMAND LINE
053.245  175          3109           MOV     A,L
053.246  264          3110           ORA     H
053.247  310          3111           RZ                      NO TARGET FILE
053.250  076 203       3112           MVI     A,PEC.TFI       TARGET FILE ILLEGAL
053.252  311          3113           RET                     TARGET FILE SPECIFIED


                      3115   **      CWM - CHECK WILDCARD MATCH.
                      3116   *
                      3117   *       CWM CHECKS TO SEE IF A WILDCARDED FIELD MATCHES A NON-WILDCARDED
                      3118   *       FIELD.
                      3119   *
                      3120   *       ENTRY   (DE) = ADDRESS OF WC NAME
                      3121   *               (HL) = ADDRESS OF NON/WC NAME
                      3122   *               (B) = NUMBER OF CHARACTERS TO CHECK
                      3123   *       EXIT    'Z' SET IF MATCH
                      3124   *                (HL) = (HL)+(B)
                      3125   *                (DE) = (DE) = (B)
                      3126   *               'Z' CLEAR IF NO MATCH
                      3127   *       USES    A,F,B,D,E,H,L
                      3128
                      3129
053.253  032          3130   CWM     LDAX    D
053.254  247          3131           ANA     A
053.255  372 262 053  3132           JM      CWM1            IS MATCH
053.260  276          3133           CMP     M
053.261  300          3134           RNE                     NO MATCH
053.262  023          3135   CWM1    INX     D
053.263  043          3136           INX     H               ADVANCE ADDRESSES
053.264  005          3137           DCR     B
```

```
053.265  302 253 053  3138           JNZ     CWM             GO FOR MORE
053.270  311          3139           RET                     GOT MATCH


                      3141  **       DDF - DECODE DESTINATION FILE.
                      3142  *
                      3143  *        DDF DECODES THE DESTINATION FILE NAME FROM THE COMMAND LINE.
                      3144  *
                      3145  *        IF NO DESTINATION NAME IS SPECIFIED, IT DEFAULTS TO
                      3146  *
                      3147  *        KB:PIPDEST.JGL
                      3148  *
                      3149  *        ENTRY   NONE
                      3150  *        EXIT    'C' CLEAR IF OK
                      3151  *                 (A) = 0 IF NAME HAS WILDCARDS
                      3152  *                 (A) = 1 IF NO WILDCARD USED
                      3153  *                 DESTFB+FB.NAM CONTAINS A COMPLETE DESTINATION FILE NAME
                      3154  *                 (HL) = COMMAND LINE POINTER UDPATED
                      3155  *                'C' SET IF ERROR
                      3156  *                 (A) = CODE
                      3157  *        USES    ALL
                      3158
                      3159
053.271  021 012 063  3160  DDF      LXI     D,LINE
053.274  142          3161           MOV     H,D
053.275  153          3162           MOV     L,E             (HL) = COMMAND POINTER
053.276  032          3163  DDF1     LDAX    D
053.277  023          3164           INX     D
053.300  376 075      3165           CPI     '='
053.302  312 314 053  3166           JE      DDF2            HAVE A SOURCE FILE
053.305  247          3167           ANA     A
053.306  302 276 053  3168           JNZ     DDF1            MORE TO CHECK
053.311  041 371 053  3169  DDF1.0   LXI     H,DDFA          USE DEFAULT
                      3170
                      3171  *        (HL) = ADDRESS FOR NAME
                      3172
053.314  021 310 062  3173  DDF2     LXI     D,DEFALT
053.317  315 005 054  3174           CALL    CAD             CONVERT ASCII NAME TO DIRECTORY FORMAT
053.322  330          3175           RC                      ERROR
053.323  312 311 053  3176           JZ      DDF1.0          NO FILE NAME SPECIFIED, USE DEFAULT
053.326  176          3177           MOV     A,M
053.327  376 075      3178           CPI     '='
053.331  076 206      3179           MVI     A,PEC.IDF       ASSUME ILLEGAL DESTINATION FORMAT
053.333  067          3180           STC
053.334  300          3181           RNE                     MUST HAVE '='
                      3182
                      3183  *        HAVE NAME DECODED. EXPAND INTO DESTFB+FB.NAM
                      3184
053.335  041 261 062  3185           LXI     H,DESTFB+FB.NAM
000.001               3186           IF      .PIP.
                      3187           JMP     CDA             CONVERT DIRECTORY FORMAT TO ASCII FORMAT
                      3188           ELSE    ONECOPY
053.340  315 065 055  3189           CALL    CDA             CONVERT DIRECTORY FORMAT TO ASCII FORMAT
053.343  365          3190           PUSH    PSW             SAVE CODE
```

```
053.344  016 003      3191          MVI     C,3
053.346  021 002 054  3192          LXI     D,DDFB
053.351  041 261 062  3193          LXI     H,DESTFB+FB.NAM
053.354  315 060 030  3194          CALL    $COMP           SEE IF DEVICE IS SY0
053.357  302 364 053  3195          JNE     DDF3            IS ERROR
053.362  361          3196          POP     PSW
053.363  311          3197          RET                     RETURN WITH 'C' CLEAR
                      3198
053.364  361          3199  DDF3    POP     PSW             ERROR, ILLEGAL DEVICE CODE
053.365  076 005      3200          MVI     A,EC.DNS
053.367  067          3201          STC
053.370  311          3202          RET
                      3203
053.371  123 131 060  3204  DDFA    DB      'SY0:*.*=',0    DEFAULT TARGET FOR ONECOPY
054.002  123 131 060  3205  DDFB    DB      'SY0'           REQUIRED DEVICE SPECIFICATION FOR ONECOPY
                      3206          ELSE
                      3207
                      3208  DDFA    DB      'TT:PIPDEST.JGL=',0
                      3209          ENDIF


                      3211  **      CAD - CONVERT ASCII FILE NAME INTO DIRECTORY FORMAT.
                      3212  *
                      3213  *       CAD CRACKS AN ALPHANUMERIC FILE DESCRIPTION, OF THE FORM
                      3214  *
                      3215  *       DEV:NAME.EXT
                      3216  *
                      3217  *       INTO THE PIO.XXX FIELDS.
                      3218  *
                      3219  *       THE DEFAULT BLOCK DETERMINES THE VALUES FOR THE DEVICE AND EXTENSION
                      3220  *       FIELDS, IF THEY ARE UNSPECIFIED.  IF *CAD* IS ENTERED
                      3221  *       AT *CAD*, AN UNSPECIFIED NAME FIELD IS RETURNED AS ZERO BYTES.
                      3222  *       IF ENTERED AT *CAD.*, AN UNSPECIFIED NAME FIELD IS
                      3223  *       RETURNED AS 200Q (MATCH-ONE) BYTES.
                      3224  *
                      3225  *       ENTRY   (DE) = POINT TO DEFAULT BLOCK
                      3226  *               (HL) = POINTER TO TEXT
                      3227  *       EXIT    'C' SET IF ERROR
                      3228  *                (A) = ERROR CODE
                      3229  *               'C' CLEAR IF OK
                      3230  *                (HL) = POINTS PAST FILE NAME
                      3231  *                'Z' SET IF NULL NAME
                      3232  *                'Z' CLEAR IF NON-NULL
                      3233  *                 PIO.DIR.NAM = NAME
                      3234  *                 PIO.DIR.EXT = EXTENSION
                      3235  *                 PIO.DEV = DEVICE CODE
                      3236  *                 PIO.UNI = UNIT NUMBER (ASCII DIGIT)
                      3237  *       USES    ALL
                      3238
                      3239
054.005  257          3240  CAD     XRA     A               SET TO NULLS
054.006  303 013 054  3241          JMP     CAD0
                      3242
054.011  076 200      3243  CAD.    MVI     A,200Q
```

```
054.013  345            3244  CADO    PUSH    H
054.014  062 257 054     3245          STA     CADA            SAVE DEFAULT VALUE
                         3246
                         3247  *       SET DEFAULTS IN FIO.xxx
                         3248
054.017  041 337 062     3249          LXI     H,FIO.DEV
054.022  001 003 000     3250          LXI     B,3
054.025  315 252 030     3251          CALL    $MOVE           SET DEFALUT DEVICE
054.030  001 003 000     3252          LXI     B,3
054.033  041 352 062     3253          LXI     H,FIO.DIR+DIR.EXT
054.036  315 252 030     3254          CALL    $MOVE           SET DEFAULT EXTENSION
054.041  341             3255          POP     H
054.042  315 156 057     3256          CALL    $SOB            SKIP BLANKS
054.045  006 000         3257          MVI     B,0
054.047  376 077         3258          CPI     '?'
054.051  312 100 054     3259          JE      CAD1            IS '?'
054.054  376 052         3260          CPI     '*'
054.056  312 100 054     3261          JE      CAD1            IS '*'
054.061  376 056         3262          CPI     '.'
054.063  312 100 054     3263          JE      CAD1            IS '.'
054.066  376 101         3264          CPI     'A'
054.070  332 240 054     3265          JC      CAD4            NOT NAME
054.073  376 133         3266          CPI     'Z'+1
054.075  322 240 054     3267          JNC     CAD4            NOT NAME
                         3268
                         3269  *       HAVE ALPHA STRING, CRACK IT
                         3270
054.100  315 260 054     3271  CAD1    CALL    DNT             DECODE NEXT TOKEN
054.103  332 253 054     3272          JC      CAD5            ERROR
054.106  376 072         3273          CPI     ':'
054.110  302 143 054     3274          JNE     CAD2            NOT DEVICE
                         3275
                         3276  *       HAVE EXPLICIT DEVICE
                         3277
054.113  043             3278          INX     H               SKIP ':'
054.114  076 003         3279          MVI     A,3
054.116  271             3280          CMP     C
054.117  332 253 054     3281          JC      CAD5            TOO MANY CHARACTERS
054.122  001 003 000     3282          LXI     B,3
054.125  345             3283          PUSH    H               SAVE (HL)
054.126  041 337 062     3284          LXI     H,FIO.DEV
054.131  315 252 030     3285          CALL    $MOVE           SET EXPLICIT DEVICE
054.134  341             3286          POP     H
054.135  315 260 054     3287          CALL    DNT             DECODE NEXT TOKEN
054.140  332 253 054     3288          JC      CAD5            ERROR
                         3289
                         3290  *       DECODE NAME
                         3291
054.143  001 010 000     3292  CAD2    LXI     B,8             (BC) = COUNT
054.146  345             3293          PUSH    H               SAVE TEXT ADDR
                         3294
                         3295  *       SEE IF NAME IS  UNSPECIFIED
                         3296
054.147  041 342 062     3297          LXI     H,FIO.DIR+DIR.NAM
054.152  345             3298          PUSH    H               SAVE ADDRESS OF DIR.NAM
054.153  315 252 030     3299          CALL    $MOVE           MOVE IN NAME
```

```
054.156  341          3300          POP     H               (HL) = $PIO.DIR+DIR.NAM
054.157  176          3301          MOV     A,M
054.160  247          3302          ANA     A
054.161  302 177 054  3303          JNZ     CAD2.6          IS SPECIFIED
054.164  072 257 054  3304          LDA     CADA            (A) = FILL CHARACTER
054.167  016 010      3305          MVI     C,8             (C) = COUNT
054.171  167          3306  CAD2.4  MOV     M,A
054.172  043          3307          INX     H
054.173  015          3308          DCR     C
054.174  302 171 054  3309          JNZ     CAD2.4
054.177  341          3310  CAD2.6  POP     H
054.200  176          3311          MOV     A,M             (A) = DELIMITER
054.201  376 056      3312          CPI     '.'
054.203  302 236 054  3313          JNE     CAD3            NOT EXTENSION
                      3314
                      3315  *       HAVE EXPLICIT EXTENSION
                      3316
054.206  043          3317          INX     H
054.207  315 260 054  3318          CALL    DNT
054.212  332 253 054  3319          JC      CAD5            ERROR
054.215  076 003      3320          MVI     A,3
054.217  271          3321          CMP     C
054.220  332 253 054  3322          JC      CAD5            TOO LONG
054.223  001 003 000  3323          LXI     B,3
054.226  345          3324          PUSH    H               SAVE TEXT POINTER
054.227  041 352 062  3325          LXI     H,PIO.DIR+DIR.EXT
054.232  315 252 030  3326          CALL    $MOVE           MOVE EXTENSION
054.235  341          3327          POP     H
                      3328
                      3329  *       DONE WITH NAME. MUST HAVE LEGIT DELIMITER
                      3330
054.236  006 001      3331  CAD3    MVI     B,1             (B) = NAME PRESENT FLAG
                      3332
                      3333  *       END OF NAME. EXIT
                      3334  *       (B) = 0 IF NULL, (B) <> 0 IF NON-NULL
                      3335
054.240  315 156 057  3336  CAD4    CALL    $SOB            SKIP BLANKS
054.243  176          3337          MOV     A,M             (A) = NEXT CHARACTER
054.244  315 371 056  3338          CALL    $CFD            CHECK FILE NAME DELIMITER
054.247  330          3339          RC                      ERROR
054.250  170          3340          MOV     A,B
054.251  247          3341          ANA     A               SET 'Z' IF NULL
054.252  311          3342          RET
                      3343
                      3344  *       ERROR
                      3345
054.253  076 007      3346  CAD5    MVI     A,EC.IFN        ILLEGAL FILE NAME
054.255  067          3347          STC
054.256  311          3348          RET
                      3349
054.257  000          3350  CADA    DB      0               FILL CHARACTER FOR OMITTED NAME FIELD
```

```
                                3352  **      DNT - DECODE NEXT TOKEN.
                                3353  *
                                3354  *       DNT COPIES THE NEXT ALPHANUMERIC FIELD INTO A ZERO-FILLED WORK AREA.
                                3355  *
                                3356  *       ENTRY   (HL) = TEXT POINTER
                                3357  *       EXIT    'C' SET IF ERROR
                                3358  *               'C' CLEAR IF OK
                                3359  *               (A) = DELIMTER CHARACTER
                                3360  *               (HL) UPDATED TO DELIMITER CHARACTER
                                3361  *               (DNTA) = STRING
                                3362  *               (C) = LENGTH
                                3363  *               (DE) = #DNTA
                                3364  *       USES    ALL
                                3365
                                3366
054.260  021 372 054           3367  DNT     LXI     D,DNTA
054.263  016 011               3368          MVI     C,9             (C) = SIZE OF DNTA
054.265  101                   3369          MOV     B,C             (B) = MAX ALLOWED +1
054.266  257                   3370          XRA     A
054.267  022                   3371  DNT1    STAX    D               ZERO BUFFER
054.270  023                   3372          INX     D
054.271  015                   3373          DCR     C
054.272  302 267 054           3374          JNZ     DNT1
054.275  021 372 054           3375          LXI     D,DNTA
                                3376
                                3377  *       COPY CHARACTERS
                                3378
054.300  176                   3379  DNT2    MOV     A,M
054.301  376 077               3380          CPI     '?'
054.303  076 200               3381          MVI     A,200Q
054.305  312 342 054           3382          JE      DNT3            IS MATCHONE
054.310  176                   3383          MOV     A,M
054.311  376 052               3384          CPI     '*'
054.313  312 354 054           3385          JE      DNT5            IS WILDCARD
054.316  376 060               3386          CPI     '0'
054.320  332 365 054           3387          JC      DNT4            NOT ALPHANUMERIC
054.323  376 072               3388          CPI     '9'+1
054.325  332 342 054           3389          JC      DNT3            NUMERIC
054.330  376 101               3390          CPI     'A'
054.332  332 365 054           3391          JC      DNT4            DELIMITER
054.335  376 133               3392          CPI     'Z'+1
054.337  322 365 054           3393          JNC     DNT4            DELIMITER
                                3394
                                3395  *       HAVE GOOD CHARACTER
                                3396
054.342  022                   3397  DNT3    STAX    D               STORE CHAR
054.343  023                   3398          INX     D
054.344  043                   3399          INX     H
054.345  014                   3400          INR     C               COUNT
054.346  005                   3401          DCR     B               LIMIT DECREMENT
054.347  302 300 054           3402          JNZ     DNT2            NOT OVERFLOW
                                3403
                                3404  *       OVERFLOW
                                3405
054.352  067                   3406          STC                     FLAG ERR
054.353  311                   3407          RET
```

```
                        3408
                        3409  *       IS '*' WILDCARD
                        3410
054.354  076 200       3411  DNT5    MVI     A,200Q
054.356  022           3412          STAX    D
054.357  023           3413          INX     D
054.360  005           3414          DCR     B
054.361  302 354 054   3415          JNZ     DNT5            FILL WITH MATCH ONE
054.364  043           3416          INX     H               SKIP '*'
                        3417
                        3418  *       END OF STRING
                        3419
054.365  247           3420  DNT4    ANA     A               CLEAR 'C'
054.366  021 372 054   3421          LXI     D,DNTA          SET POINTER
054.371  311           3422          RET
                        3423
054.372               3424  DNTA    DS      9               WORK AREA


                        3426  **      EBM - EXPAND BUFFER TO MAXIMUM.
                        3427  *
                        3428  *       EBM IS CALLED TO EXPAND THE BUFFER 'BUF' TO THE MAXIMUM SIZE,
                        3429  *       WHICH DOES NOT REQUIRE THE OVERLAYING OF THE SYSTEM.
                        3430  *
                        3431  *       ENTRY   NONE
                        3432  *       EXIT    (BUFSIZ) = BUFFER SIZE (MULTIPLE OF 256)
                        3433  *       USES    ALL
                        3434
                        3435
055.003  052 320 040   3436  EBM     LHLD    S.SYSM
055.006  345           3437          PUSH    H
055.007  052 350 040   3438          LHLD    S.OFWA
055.012  021 006 000   3439          LXI     D,OVL0*OVL.ENS+OVL.FLB
055.015  031           3440          DAD     D               (HL) = ADDR. OF OVL0 OVL.FLB ENTRY
055.016  076 002       3441          MVI     A,OVL.RES
055.020  246           3442          ANA     M
055.021  021 010 000   3443          LXI     D,OVL.ENS
055.024  031           3444          DAD     D               (HL) = ADDR. OF OVL1 OVL.FLB ENTRY
000.000               3445          ERRNZ   OVL1-OVL0-1
055.025  246           3446          ANA     M
055.026  302 043 055   3447          JNZ     EBM1            OVL0 AND OVL1 ARE PERM. RESIDENT
055.031  052 324 040   3448          LHLD    S.OMAX
055.034  315 224 030   3449          CALL    $CHL
055.037  353           3450          XCHG
055.040  341           3451          POP     H
055.041  031           3452          DAD     D               (HL) = NEW ADDRESS SOUGHT
055.042  345           3453          PUSH    H
                        3454
055.043  341           3455  EBM1    POP     H
055.044  021 372 377   3456          LXI     D,-6
055.047  031           3457          DAD     D               (HL) = NEW ADDRESS SOUGHT
055.050  377 052       3458          DB      SYSCALL,.SETTP
055.052  332 064 052   3459          JC      IERR1           INTERNAL ERROR 1
055.055  052 322 040   3460          LHLD    S.USRM
```

```
000.001                 3461            IF      .PIP.
                        3462            XCHG
                        3463            LHLD    BUFPTR
                        3464            CALL    $CHL            (HL) = - BUFFER FWA
                        3465            DAD     D
                        3466            MVI     L,0
                        3467            SHLD    BUFSIZ
                        3468            MVI     A,BUFMINL/256-1
                        3469            CMP     H
                        3470            RC                      IF OK
                        3471            MVI     A,EC.NEM
                        3472            JMP     ERROR           NOT ENOUGH MEMORY
                        3473
                        3474            ELSE
                        3475
055.060  174           3476            MOV     A,H             (A) = LIMIT/256
055.061  062 133 062    3477            STA     OBUFLIM         SET LIMIT
055.064  311           3478            RET
                        3479            ENDIF


                        3481  **        CDA - CONVERT DIRECTORY FORMAT TO ASCII.
                        3482  *
                        3483  *         CDA COPIES A DIRECTORY ENTRY FROM PIO.XXX TO A TARGET FIELD.
                        3484  *         THE DEVICE SPECIFICATION (IN PIO.DEV AND PIO.UNI) IS ALSO ENCODED.
                        3485  *         THE TARGET FIELD IS LEFT IN THE FORM:
                        3486  *
                        3487  *         DEV:NAME.XXX <00>
                        3488  *
                        3489  *         ENTRY   (HL) = FWA NAME FIELD
                        3490  *         EXIT    (A) = 0, HAVE WILDCARD
                        3491  *                     = 1, NO WILDCARDS USED
                        3492  *                 'C' CLEAR
                        3493  *         USES    ALL
                        3494
                        3495
055.065  001 000 003    3496  CDA       LXI     B,3*256         (B) = CHARACTER COUNT, (C) = WILDCARD FLAG
055.070  021 337 062    3497            LXI     D,PIO.DEV
055.073  315 131 055    3498            CALL    CDA5            COPY IT
055.076  066 072        3499            MVI     M,':'
055.100  043           3500            INX     H
055.101  006 010        3501            MVI     B,8
055.103  021 342 062    3502            LXI     D,PIO.DIR+DIR.NAM
055.106  315 131 055    3503            CALL    CDA5            COPY IT
055.111  066 056        3504            MVI     M,'.'
055.113  043           3505            INX     H
055.114  006 003        3506            MVI     B,3
000.000                 3507            ERRNZ   DIR.EXT-DIR.NAM-8
055.116  315 131 055    3508            CALL    CDA5            COPY IT
055.121  066 000        3509            MVI     M,0             FLAG END OF NAME
055.123  171           3510            MOV     A,C             (A) (BIT 7) = 1 IF WILDCARDS
055.124  007           3511            RLC
055.125  057           3512            CMA
055.126  346 001        3513            ANI     1               =0 IF WILDCARD
```

```
055.130  311              3514           RET

                          3516  **       CDA5 - CONVERT DIRECTORY FIELD TO ASCII.
                          3517  *
                          3518  *        ZEROS ARE IGNORED, 200Q WILDCARDS ARE MAPPED TO '?'
                          3519  *
                          3520  *        ENTRY   (DE) = FROM
                          3521  *                (HL) = TO
                          3522  *                (B) = COUNT
                          3523  *                (C) = ORA ACCUMULATOR
                          3524  *        EXIT    (DE) ADVANCED
                          3525  *                (HL) = (HL)+(B)
                          3526  *                (C) = (C) .OR. (FROM CHARACTERS PROCESSED)
                          3527  *        USES    ALL
                          3528
                          3529
055.131  032              3530  CDA5     LDAX    D               (A) = CHARACTER
055.132  261              3531           ORA     C
055.133  117              3532           MOV     C,A
055.134  032              3533           LDAX    D
055.135  023              3534           INX     D
055.136  247              3535           ANA     A
055.137  312 151 055      3536           JZ      CDA7            IS 00
055.142  362 147 055      3537           JP      CDA6            NOT 200Q
055.145  076 077          3538           MVI     A,'?'
055.147  167              3539  CDA6     MOV     M,A
055.150  043              3540           INX     H               INCREMENT TO
055.151  005              3541  CDA7     DCR     B
055.152  302 131 055      3542           JNZ     CDA5            IF MORE TO GO
055.155  311              3543           RET


                          3545  **       EWS - EXPAND WILDCARD SPECIFICATION.
                          3546  *
                          3547  *        DWS ENTERS THE FILE NAME IN FIO.XXX INTO THE MANAGED TABLE
                          3548  *        NAMTAB. IF THE FILE NAME CONTAINS WILDCARDS, THE DIRECTORY
                          3549  *        IS READ FOR ELIGIBLE FILES.
                          3550  *
                          3551  *        ENTRY   FIO.XXX = FILE NAME
                          3552  *        EXIT    'C' CLEAR IF OK
                          3553  *                'C' SET IF ERROR
                          3554  *        USES    ALL
                          3555
                          3556
055.156  315 027 053      3557  EWS      CALL    AEN             TRY TO ENTER IT
055.161  320              3558           RNC                     NO WILDCARDS, AM DONE
                          3559
                          3560  *        IS WILDCARD. LOOK UP DEVICE TYPE
                          3561
055.162  052 302 062      3562           LHLD    NAMTLEN
055.165  021 111 063      3563           LXI     D,NAMTAB-FB.NAML
055.170  031              3564           DAD     D               (HL) = ADDRESS OF LAST ENTRY
055.171  315 005 054      3565           CALL    CAD             CONVERT ASCII NAME TO DIRECTORY FORMAT
```

```
055.174  330          3566          RC                          ERROR
055.175  052 302 062  3567          LHLD    NAMTLEN
055.200  021 357 377  3568          LXI     D,-FB.NAML
055.203  031          3569          DAD     D
055.204  042 302 062  3570          SHLD    NAMTLEN             REMOVE WILDCARD FROM TABLE
055.207  315 252 060  3571          CALL    $MOVEL
055.212  003 000 337  3572          DW      3,PIO.DEV,DIRNAM            SET DIRECTORY NAME IN XXX:DIRECT.SYS
055.220  315 252 060  3573          CALL    $MOVEL
055.223  013 000 342  3574          DW      8+3,PIO.DIR+DIR.NAM,EWSC    SAVE WILDCARD PATTERN
055.231  001 020 056  3575          LXI     B,EWSB
055.234  041 224 062  3576          LXI     H,DIRNAM
055.237  377 053      3577          DB      SYSCALL,.DECODE GET INFORMATION ABOUT DEVICE
055.241  330          3578          RC                          ERROR
055.242  072 020 056  3579          LDA     EWSB                SEE IF A DIRECTORY DEVICE
055.245  346 001      3580          ANI     DT.DD
055.247  076 005      3581          MVI     A,EC.DNS                    ASSUME DEVICE NOT SUITABLE
055.251  067          3582          STC
055.252  310          3583          RZ                          ERROR
                      3584
                      3585  *       IS DIRECTORY DEVICE, OPEN DIRECTORY
                      3586
055.253  041 224 062  3587          LXI     H,DIRNAM
055.256  076 002      3588          MVI     A,CN.DIR
055.260  377 042      3589          DB      SYSCALL,.OPENR
055.262  076 200      3590          MVI     A,PEC.DF
055.264  330          3591          RC                          DEVICE FORMAT FAILURE
                      3592
                      3593  *       READ DIRECTORY ENTRYS FOR MATCH
                      3594
055.265  315 071 056  3595  EWS1    CALL    GDWP                DE = DIRECTORY WORKSPACE PTR    /79.11.GC/
055.270  001 000 002  3596          LXI     B,512
055.273  076 002      3597          MVI     A,CN.DIR
055.275  325          3598          PUSH    D                   SAVE ADDRESS
055.276  377 004      3599          DB      SYSCALL,.READ       READ BLOCK
055.300  341          3600          POP     H                   (HL) = DIRECTORY ADDRESS
055.301  332 005 056  3601          JC      EWS7                ALL DONE
                      3602
                      3603  *       LOOK AT DIRECTORY BLOCK FOR MATCHES
                      3604
055.304  345          3605          PUSH    H                                                   /79.11.GC/
055.305  315 077 056  3606          CALL    GDWP.                                               /79.11.GC/
055.310  315 307 057  3607          CALL    $INDLB                                              /79.11.GC/
055.313  373 001      3608          DW      DIS.ENL             A  = DIRECTORY ENTRY LENGTH     /79.11.GC/
055.315  341          3609          POP     H                                                   /79.11.GC/
                      3610
055.316  117          3611          MOV     C,A                 (C) = LENGTH
                      3612
                      3613  *       CHECK NEXT ENTRY
                      3614
055.317  176          3615  EWS3    MOV     A,M                 (A) = 1ST CHAR THIS ENTRY
055.320  247          3616          ANA     A
055.321  312 265 055  3617          JZ      EWS1                END OF BLOCK
000.000               3618          ERRNZ   DF.EMP-377Q
055.324  074          3619          INR     A
055.325  312 377 055  3620          JZ      EWS6                ENTRY EMPTY
000.000               3621          ERRNZ   DF.CLR-376Q
```

```
055.330  074            3622          INR     A
055.331  312 005 056     3623          JZ      EWS7            END OF LIST
055.334  315 200 053     3624          CALL    CFE             CHECK FOR FILE ELIGIBILITY
055.337  302 377 055     3625          JNZ     EWS6            NOT TO PROCESS
055.342  345             3626          PUSH    H
055.343  021 056 056     3627          LXI     D,EWSC
055.346  006 013         3628          MVI     B,8+3
055.350  315 253 053     3629          CALL    CWM             CHECK WILDCARD MATCH
055.353  302 376 055     3630          JNZ     EWS4            NO MATCH
                         3631
                         3632  *       HAVE MATCH, ADD TO LSIT
                         3633
055.356  321             3634          POP     D               (DE) = FROM
055.357  325             3635          PUSH    D
055.360  305             3636          PUSH    B               SAVE (C)
055.361  001 013 000     3637          LXI     B,8+3
055.364  041 342 062     3638          LXI     H,PIO.DIR+DIR.NAM
055.367  315 252 030     3639          CALL    $MOVE
055.372  315 027 053     3640          CALL    AEN             ADD TO TABLE
055.375  301             3641          POP     B               RESTORE (C)
                         3642
                         3643  *       LOOKUP NEXT ENTRY
                         3644
055.376  341             3645  EWS4    POP     H
055.377  006 000         3646  EWS6    MVI     B,0
056.001  011             3647          DAD     B               POINT TO NEXT
056.002  303 317 055     3648          JMP     EWS3
                         3649
                         3650  *       ALL DONE, CLOSE DIRECTORY FILE
                         3651
056.005  076 002         3652  EWS7    MVI     A,CN.DIR
056.007  377 046         3653          DB      SYSCALL,.CLOSE
056.011  311             3654          RET
                         3655
056.012  123 131 060     3656  EWSA    DB      'SY0',200Q,200Q,200Q
                         3657
056.020                  3658  EWSB    DS      30
                         3659
056.056                  3660  EWSC    DS      8+3             WILDCARD PATTERN FOR DIRECTORY SEARCH


                         3662  **      GDWP    -  GET DIRECTORY WORKSPACE POINTER              /79.11.GC/
                         3663  *
                         3664  *       GDWP GETS THE DIRECTORY WORKSPACE POINTER
                         3665  *
                         3666  *       ENTRY:  NONE
                         3667  *
                         3668  *       EXIT:   DE      = DIRECTORY WORKSPACE POINTER
                         3669  *
                         3670  *       USES:   DE
                         3671  *
                         3672
056.071  353             3673  GDWP    XCHG
056.072  315 077 056     3674          CALL    GDWP.           HL = DIRECTORY WORKSPACE POINTER
```

```
056.075  353            3675          XCHG
056.076  311            3676          RET
                        3677
056.077  052 120 041    3678  GDWP.   LHLD    S.SCR           HL = SYSTEM SCRATCH
056.102  311            3679          RET
```

```
                        3681  **      INA - INCREASE NAMTAB ALLOCATION.
                        3682  *
                        3683  *       INA IS CALLED TO INCREASE THE NAMTAB ALLOCATION. THE
                        3684  *       BUFFER AREA IS MOVED UP TO MAKE ROOM.
                        3685  *
                        3686  *       ENTRY   NONE
                        3687  *       EXIT    NONE
                        3688  *       USES    A,F,H,L
                        3689
056.103  041 305 062    3690  INA     LXI     H,NAMTMAX+1
056.106  064            3691          INR     M               INCREMENT LENGTH
056.107  041 244 062    3692          LXI     H,BUFPTR+1
056.112  064            3693          INR     M               MOVE BUFFER
056.113  052 245 062    3694          LHLD    BUFSIZ
056.116  174            3695          MOV     A,H
056.117  265            3696          ORA     L
056.120  076 021        3697          MVI     A,EC.NEM        FLAG OUT OF MEMORY IF BUFFER NOT EMPTY
056.122  302 275 052    3698          JNZ     ERROR
056.125  305            3699          PUSH    B
056.126  325            3700          PUSH    D
056.127  315 256 056    3701          CALL    SBE             NOTIFY SYSTEM
056.132  321            3702          POP     D
056.133  301            3703          POP     B
056.134  311            3704          RET
```

```
                        3706  **      LSN - LOCATE SOURCE NAME
                        3707  *
                        3708  *       LSN SCANS THE COMMAND LINE FOR THE FIRST SOURCE FILE NAME.
                        3709  *
                        3710  *       ENTRY   NONE
                        3711  *       EXIT    (HL) = 1ST FILE NAME FWA
                        3712  *       USES    A,F,H,L
                        3713
056.135  041 012 063    3714  LSN     LXI     H,LINE
056.140  176            3715  LSN1    MOV     A,M
056.141  043            3716          INX     H
056.142  376 075        3717          CPI     '='
056.144  310            3718          RE                      GOT IT
056.145  247            3719          ANA     A
056.146  302 140 056    3720          JNZ     LSN1            MORE LINE
056.151  041 012 063    3721          LXI     H,LINE  IS NO =
056.154  311            3722          RET
```

```
                                3724  **      MWN - MERGE WILDCARD NAMES.
                                3725  *
                                3726  *       MWN MERGES A COMPLETELY SPECIFIED FILENAME WITH A WILDCARDED COMPLETELY
                                3727  *       SPECIFIED FILE NAME.
                                3728  *
                                3729  *       BOTH FILE NAMES SHOULD HAVE THE SAME DEVICE SPECIFICATION.
                                3730  *
                                3731  *       FILE NAME FORMAT:
                                3732  *
                                3733  *       DEV:NAMEXXXX.EXT 00
                                3734  *
                                3735  *       ENTRY   (BC) = ADDRESS OF WILDCARDED ASCII NAME
                                3736  *               (DE) = ADDRESS OF NON-WC ASCII NAME
                                3737  *               (HL) = ADDRESS FOR RESULTANT ASCII NAME
                                3738  *       EXIT    NONE
                                3739  *       USES    ALL
                                3740
                                3741
056.155  345                   3742  MWN     PUSH    H               SAVE TARGET ADDRESS
056.156  305                   3743          PUSH    B               SAVE WC PATTERN
056.157  353                   3744          XCHG                    (HL) = MASTER NAME
056.160  315 005 054           3745          CALL    CAD             CONVERT TO DIRECTORY FORMAT
056.163  315 252 060           3746          CALL    $MOVEL
056.166  013 000 342           3747          DW      8+3,PIO.DIR,MWNA        (MWNA) = DECODED MASTER
056.174  341                   3748          POP     H               (HL) = WC PATTERN
056.175  315 005 054           3749          CALL    CAD             (PIO.DIR) = WC PATTERN
056.200  021 316 062           3750          LXI     D,MWNA          (DE) = MASTER PATTERN
056.203  041 342 062           3751          LXI     H,PIO.DIR       (DE) = WC PATTERN ADDRESS
056.206  016 013               3752          MVI     C,8+3           MERGE NAME AND EXTENSION
                                3753
                                3754  *       MERGE NAMES
                                3755
056.210  176                   3756  MWN1    MOV     A,M             (A) = WC PATTERN
056.211  247                   3757          ANA     A
056.212  362 216 056           3758          JP      MWN2            USE THIS
056.215  032                   3759          LDAX    D               IS MATCH CHARACTER, USE MASTER INSTEAD
056.216  167                   3760  MWN2    MOV     M,A             STORE CHARACTER
056.217  023                   3761          INX     D
056.220  043                   3762          INX     H
056.221  015                   3763          DCR     C
056.222  302 210 056           3764          JNZ     MWN1            MERGE TILL DONE
056.225  341                   3765          POP     H               (HL) = TARGET ADDRESS
056.226  303 065 055           3766          JMP     CDA             CONVERT DIRECTORY FORMAT TO ASCII


                                3768  **      REN - REMOVE ENTRY FROM *NAMTAB*
                                3769  *
                                3770  *       REN REMOVES THE FIRST 'FB.NAML' BYTES FROM NAMTAB.
                                3771  *
                                3772  *       THE AMOUNT (FB.NAML) IS REMOVED FROM THE SIZE OF THE TABLE. THE
                                3773  *       TABLE IS NOT CHECKED FOR UNDERFLOW, THE CALLER MUST GUARANTEE THE
                                3774  *       PRESENSE OF AT LEAST FB.NAML BYTES IN NAMTAB.
                                3775  *
                                3776  *       ENTRY   NONE
```

```
                                3777  *       EXIT    NONE
                                3778  *       USES    ALL
                                3779
                                3780
056.231  052 302 062           3781  REN     LHLD    NAMTLEN
056.234  021 357 377           3782          LXI     D,-FB.NAML
056.237  031                   3783          DAD     D               REMOVE COUNT FROM LEN
056.240  042 302 062           3784          SHLD    NAMTLEN
056.243  104                   3785          MOV     B,H
056.244  115                   3786          MOV     C,L             (BC) = REMAINING LENGTH
056.245  021 153 063           3787          LXI     D,NAMTAB+FB.NAML        (DE) = START OF 2ND ENTRY
056.250  041 132 063           3788          LXI     H,NAMTAB
056.253  303 252 030           3789          JMP     $MOVE           MOVE DOWN AND RETURN


                                3791  **      SBE - SET BUFFER EMPTY.
                                3792  *
                                3793  *       THE SYSTEM IS NOTIFIED.
                                3794  *
                                3795  *       ENTRY   NONE
                                3796  *       EXIT    NONE
                                3797  *       USES    ALL
                                3798
                                3799
056.256  041 000 000           3800  SBE     LXI     H,0
056.261  042 245 062           3801          SHLD    BUFSIZ
056.264  052 243 062           3802          LHLD    BUFPTR          (HL) = BUFFER FWA (AND LWA!)
056.267  043                   3803          INX     H
056.270  043                   3804          INX     H
056.271  377 052               3805          DB      SYSCALL,.SETTP
056.273  320                   3806          RNC                     OK
056.274  303 275 052           3807          JMP     ERROR           NOT ENOUGH ROOM


                                3809  **      SDD - SET DEFAULT DEFAULT.
                                3810  *
                                3811  *       SDD IS CALLED TO SETUP THE CURRENT DEFAULT DEVICE
                                3812  *       AND EXTENSION TO 'SY0' AND <NULL>, RESPECTIVELY.
                                3813  *
                                3814  *       ENTRY   NONE
                                3815  *       EXIT    NONE
                                3816  *       USES    NONE
                                3817
                                3818
056.277  315 054 031           3819  SDD     CALL    $SAVALL
056.302  315 252 060           3820          CALL    $MOVEL
056.305  006 000 316           3821          DW      6,SDDA,DEFALT   SET DEFAULT DEFAULT
056.313  303 047 031           3822          JMP     $RSTALL         RESTORE AND RETURN
                                3823
056.316  123 131 060           3824  SDDA    DB      'SY0',0,0,0     DEFAULT DEFAULT VALUES
```

```
                                3826  **      SFS - SKIP FILE SEPERATOR.
                                3827  *
                                3828  *       SFS IS CALLED TO SKIP OVER THE CHARACTERS SEPERATING ONE
                                3829  *       FILE NAME FROM ANOTHER ON THE LINE. THE FILES MAY BE SEPERATED
                                3830  *       BY BLANKS OR A COMMA ALONE, OR BY BLANKS WITH A COMMA. THE
                                3831  *       SYNTAX IS
                                3832  *
                                3833  *       <BLANKS> <,> <BLANKS>
                                3834  *
                                3835  *       ONE, TWO OR ALL THREE FIELDS MAY BE PRESENT.
                                3836  *
                                3837  *       ENTRY   (HL) = POINT TO START OF SEP FIELD
                                3838  *       EXIT    (HL) ADVANCED PAST SEPERATOR FIELD
                                3839  *       USES    A,F,H,L
                                3840
                                3841
056.324  315 156 057           3842  SFS     CALL    $SOB            SKIP BLANKS
056.327  176                   3843          MOV     A,M
056.330  376 054               3844          CPI     ','
056.332  302 336 056           3845          JNE     SFS1            NOT ,
056.335  043                   3846          INX     H               SKIP ,
056.336  303 156 057           3847  SFS1    JMP     $SOB            GET ANY MORE BLANKS AND EXIT


                                3849  **      SND - SET NEW DEFAULTS.
                                3850  *
                                3851  *       SND IS CALLED TO SET A NEW DEFAULT DEVICE AND EXTENSION
                                3852  *       IN THE 'DEFALT' AREA.
                                3853  *
                                3854  *       ENTRY   PIO.DEV = DEVICE CODE
                                3855  *               PIO.UNI = UNIT #
                                3856  *               PIO.DIR+DIR.EXT = EXTENSION
                                3857  *       EXIT    NONE
                                3858  *       USES    NONE
                                3859
                                3860
056.341  315 054 031           3861  SND     CALL    $SAVALL         SAVE REGS
000.000                         3862          ERRNZ   PIO.UNI-PIO.DEV-2
056.344  315 252 060           3863          CALL    $MOVEL
056.347  003 000               3864          DW      3
056.351  337 062               3865          DW      PIO.DEV
056.353  310 062               3866          DW      DEFALT
056.355  315 252 060           3867          CALL    $MOVEL
056.360  003 000               3868          DW      3
056.362  352 062               3869          DW      PIO.DIR+DIR.EXT
056.364  313 062               3870          DW      DEFALT+3
056.366  303 047 031           3871          JMP     $RSTALL         RETURN
```

```
056.371                 3874            XTEXT   CFD


                        3876X **        $CFD - CHECK FILE DELIMITER.
                        3877X *
                        3878X *         $CFD CHECKS AN ASCII CHARACTER TO SEE IF IT IS A LEGAL FILE
                        3879X *         NAME DELIMITER. LEGAL DELIMITERS ARE
                        3880X *
                        3881X *         , = / <BLANK>  <00>
                        3882X *
                        3883X *         ENTRY   (A) = CHARACTER
                        3884X *         EXIT    'C' CLEAR IF OK
                        3885X *                 'C' SET IF ERROR
                        3886X *                  (A) = ERROR CODE
                        3887X *         USES    A,F
                        3888X
                        3889X
056.371  247           3890X $CFD       ANA     A
056.372  310           3891X            RZ                      IS 00
056.373  376 054       3892X            CPI     ','
056.375  310           3893X            RE                      IS ,
056.376  376 075       3894X            CPI     '='
057.000  310           3895X            RE                      IS =
057.001  376 057       3896X            CPI     '/'
057.003  310           3897X            RE                      IS /
057.004  376 040       3898X            CPI     ' '
057.006  310           3899X            RE                      IS ' '
057.007  076 007       3900X            MVI     A,EC.IFN        ILLEGAL FILE NAME
057.011  067           3901X            STC
057.012  311           3902X            RET
057.013                 3903            XTEXT   TYPCC


                        3905X **        $TYPCC - TYPE A CHARACTER STRING BY COUNT.
                        3906X *
                        3907X *         $TYPCC TYPES A STRING OF CHARACTERS. THE CALLER SUPPLIES
                        3908X *         THE CHARACTER ADDRESS AND COUNT.
                        3909X *
                        3910X *         ENTRY   (HL) = ADDRESS
                        3911X *                 (A) = COUNT
                        3912X *         EXIT    (HL) = LAST CHARACTER ADDRESS+1
                        3913X *         USES    A,F,H,L
                        3914X
                        3915X
057.013                 3916X $TYPCC    EQU     *
057.013  247            3917X           ANA     A
057.014  310            3918X           RZ                      NOTHING TO TYPE
057.015  365            3919X           PUSH    PSW             SAVE COUNT
057.016  176            3920X           MOV     A,M             (A) = CHARACTER
057.017  043            3921X           INX     H
057.020  377 002        3922X           DB      SYSCALL,.SCOUT
057.022  361            3923X           POP     PSW
```

```
057.023  075          3924X          DCR     A
057.024  303 013 057  3925X          JMP     $TYPCC
057.027               3926           XTEXT   WER


                      3928X **       $WER - WRITE ENABLE RAM.
                      3929X *
                      3930X *        $WER IS CALLED TO ENABLE WRITTING TO THE H17 CONTROLLER'S
                      3931X *        RAM AREA.
                      3932X *
                      3933X *        ENTRY   NONE
                      3934X *        EXIT    NONE
                      3935X *        USES    NONE
                      3936X
                      3937X
031.241               3938X $WER     EQU     31241A          IN H17 ROM


                      3940X **       $WDR - WRITE DISABLE RAM.
                      3941X *
                      3942X *        $WDR IS CALLED TO DISABLE WRITTING TO THE H17 CONTROLLER'S
                      3943X *        RAM AREA.
                      3944X *
                      3945X *        ENTRY   NONE
                      3946X *        EXIT    NONE
                      3947X *        USES    NONE
                      3948X
                      3949X
031.222               3950X $WDR     EQU     31222A          IN H17 ROM
057.027               3951           XTEXT   ZERO


                      3953X **       $ZERO - ZERO MEMORY
                      3954X *
                      3955X *        $ZERO ZEROS A BLOCK OF MEMORY.
                      3956X *
                      3957X *        ENTRY   (HL) = ADDRESS
                      3958X *                (B) = COUNT
                      3959X *        EXIT    (A) = 0
                      3960X *        USES    A,B,F,H,L
                      3961X
                      3962X
031.212               3963X $ZERO    EQU     31212A          IN H17 ROM
057.027               3964           XTEXT   MU86
```

```
                                3966X **      $MU86 - MULTIPLY 8X16 UNSIGNED.
                                3967X *
                                3968X *       $MU86 MULTIPLIES A 16 BIT VALUE BY A 8
                                3969X *       BIT VALUE.
                                3970X *
                                3971X *       ENTRY   (A) = MULTIPLIER
                                3972X *               (DE) = MULTIPLICAND
                                3973X *       EXIT    (HL) = RESULT
                                3974X *               'Z' SET IF NOT OVERFLOW
                                3975X *       USES    A,F,H,L
                                3976X
                                3977X
          031.007               3978X $MU86   EQU     31007A          IN H17 ROM
          057,027               3979          XTEXT   CCO


                                3981X **      $CCO - CLEAR CONTROL-O
                                3982X *
                                3983X *       $CCO IS CALLED TO CLEAR THE EFFECT OF THE CTL-O CHARACTER.
                                3984X *
                                3985X *       ENTRY   NONE
                                3986X *       EXIT    NONE
                                3987X *       USES    NONE
                                3988X
                                3989X
          057,027 315 054 031   3990X $CCO    CALL    $SAVALL         SAVE REGISTERS
          057.032 076 004       3991X         MVI     A,I.CONFL
          057,034 001 001 000   3992X         LXI     B,CO.FLG        CLEAR CO.FLG
          057.037 377 006       3993X         DB      SYSCALL,.CONSL
          057,041 303 047 031   3994X         JMP     $RSTALL         RESTORE REGISTERS AND RETURN
          057.044               3995          XTEXT   GNL


                                3997X **      $GNL - GUARANTEE NEW LINE.
                                3998X *
                                3999X *       $GNL GUARANTEES THE START OF A NEW LINE BY ISSUING A CRLF
                                4000X *       IF THE CURSOR IS NOT AT COLUMN 1..
                                4001X *
                                4002X *       ENTRY   NONE
                                4003X *       EXIT    NONE
                                4004X *       USES    ALL
                                4005X
                                4006X
          057,044 076 002       4007X $GNL    MVI     A,I.CUSOR
          057.046 001 000 000   4008X         LXI     B,0
          057,051 377 006       4009X         DB      SYSCALL,.CONSL  READ CURSOR
          057.053 075           4010X         DCR     A
          057,054 310           4011X         RZ                      AT COLUMN 1
          057.055 303 225 057   4012X         JMP     $CRLF           NEW LINE
          057,060               4013          XTEXT   MLU
```

```
                                4015X **      MLU - MAP LOWER CASE LINE TO UPPER CASE.
                                4016X *
                                4017X *       MLU MAPS THE LOWER CASE ALPHABETICS IN A LINE TO UPPER CASE.
                                4018X *
                                4019X *       ENTRY   (HL) = LINE FWA
                                4020X *       EXIT    NONE
                                4021X *       USES    NONE
                                4022X
                                4023X
057.060  365                   4024X $MLU    PUSH    PSW             SAVE (PSW)
057.061  345                   4025X         PUSH    H               SAVE FWA
057.062  053                   4026X         DCX     H               ANTICIPATE INX H
057.063  043                   4027X $MLU1   INX     H
057.064  176                   4028X         MOV     A,M             (A)= CHARACTER
057.065  315 100 057           4029X         CALL    $MCU            MAP CHAR TO UPPER
057.070  167                   4030X         MOV     M,A
057.071  247                   4031X         ANA     A
057.072  302 063 057           4032X         JNZ     $MLU1           MORE TO GO
057.075  341                   4033X         POP     H               RESTORE (HL)
057.076  361                   4034X         POP     PSW             RESTORE (PSW)
057.077  311                   4035X         RET
057.100                        4036          XTEXT   MCU


                                4038X **      MCU - MAP LOWER CASE TO UPPER CASE.
                                4039X *
                                4040X *       MCU MAPS A LOWER CASE ALPHABETIC TO UPPER
                                4041X *       CASE.
                                4042X *
                                4043X *       ENTRY   (A) = CHARACTER
                                4044X *       EXIT    (A) = CHARACTER RESULT
                                4045X *       USES    A,F
                                4046X
                                4047X
057.100  376 141               4048X $MCU    CPI     'a'
057.102  330                   4049X         RC                      NOT LOWER CASE
057.103  376 173               4050X         CPI     'z'+1
057.105  320                   4051X         RNC                     NOT LOWER CASE
057.106  326 040               4052X         SUI     'a'-'A'
057.110  311                   4053X         RET
057.111                        4054          XTEXT   RTL


                                4056X **      $RTL - READ TEXT LINE.
                                4057X *
                                4058X *       $RTL READS A LINE FROM THE TERMINAL.
                                4059X *
                                4060X *       CHARACTER ARE ACCEPTED FROM THE TERMINAL, RUBOUT AND BACKSPACE
                                4061X *       CHARACTERS ARE PROCESSED. WHEN A CARRIAGE RETURN IS ENTERED,
                                4062X *       $RTL RETURNS.
                                4063X *
                                4064X *       ENTRY   (HL) = BUFFER FWA
```

```
                                4065X *         EXIT    'C' CLEAR IF OK
                                4066X *                  DATA IN BUFFER
                                4067X *                  (A) = TEXT LENGTH
                                4068X *                  'C' SET IF CTL-D STRUCK
                                4069X *         USES    A,F
                                4070X
                                4071X
057,111  315 120 057           4072X $RTL.     CALL    $RTL            $RTL IN UPPER CASE
057,114  330                   4073X           RC                      CTL-D
057,115  303 060 057           4074X           JMP     $MLU            MAP LINE TO UPPER CASE
                                4075X
057,120                        4076X $RTL      EQU     *
057,120  345                   4077X           PUSH    H               SAVE FWA
057,121  315 275 060           4078X $RTL1     CALL    $RCHAR
057,124  376 004               4079X           CPI     CTLD
057,126  312 153 057           4080X           JE      $RTL2           CTL-D STRUCK
057,131  167                   4081X           MOV     M,A
057,132  043                   4082X           INX     H
057,133  376 012               4083X           CPI     NL
057,135  302 121 057           4084X           JNE     $RTL1
057,140  053                   4085X           DCX     H
057,141  066 000               4086X           MVI     M,0
057,143  043                   4087X           INX     H
                                4088X
                                4089X *         ALL DONE. COMPUTE LENGTH
                                4090X
057,144  353                   4091X           XCHG                    (DE) = LWA+1
057,145  343                   4092X           XTHL                    (HL) = FWA
057,146  173                   4093X           MOV     A,E
057,147  225                   4094X           SUB     L               (A) = LENGTH
057,150  247                   4095X           ANA     A               CLEAR CARRY
057,151  321                   4096X           POP     D               RESTORE (DE)
057,152  311                   4097X           RET
                                4098X
                                4099X *         CTL-D STRUCK
                                4100X
057,153  341                   4101X $RTL2     POP     H               (HL) = FWA
057,154  067                   4102X           STC
057,155  311                   4103X           RET
057,156                        4104            XTEXT   MOVE


                                4106X **        $MOVE - MOVE DATA
                                4107X *
                                4108X *         $MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                                4109X *         IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                                4110X *         FIRST TO LAST.
                                4111X *
                                4112X *         IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                                4113X *         LAST TO FIRST.
                                4114X *
                                4115X *         THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                                4116X *
                                4117X *         ENTRY   (BC) = COUNT
```

```
                        4118X *             (DE) = FROM
                        4119X *             (HL) = TO
                        4120X *       EXIT  MOVED
                        4121X *             (DE) = ADDRESS OF NEXT FROM BYTE
                        4122X *             (HL) = ADDRESS OF NEXT *TO* BYTE
                        4123X *             'C' CLEAR
                        4124X *       USES  ALL
                        4125X
                        4126X
030.252                 4127X $MOVE   EQU   30252A          IN H17 ROM
057.156                 4128          XTEXT CHL


                        4130X **      $CHL - COMPLEMENT (HL).
                        4131X *
                        4132X *       (HL) = -(HL)          TWO'S COMPLEMENT
                        4133X *
                        4134X *       ENTRY NONE
                        4135X *       EXIT  NONE
                        4136X *       USES  A,F,H,L
                        4137X
                        4138X
030.224                 4139X $CHL    EQU   30224A          IN H17 ROM
057.156                 4140          XTEXT SOB


                        4142X **      $SOB - SKIP OVER BLANKS.
                        4143X *
                        4144X *       $SOB IS CALLED TO SKIP AN ARBITRARILY LONG STRING OF BLANKS AND TABS.
                        4145X *
                        4146X *       ENTRY (HL) = FWA OF (POSSIBLE) BLANK STRING
                        4147X *       EXIT  (HL) = LWA+1 OF BLANK STRING (UNCHANGED IF NO BLANKS)
                        4148X *             (A) = FIRST NON-BLANK, NON-TAB CHARACTER EEN
                        4149X *       USES  A,F,H,L
                        4150X
                        4151X
057.156  053           4152X $SOB    DCX   H               PRE-DECREMENT
057.157  043           4153X $SOB1   INX   H
057.160  176           4154X         MOV   A,M
057.161  376 040       4155X         CPI   ' '
057.163  312 157 057   4156X         JE    $SOB1           GOT BLANK
057.166  376 011       4157X         CPI   TAB
057.170  312 157 057   4158X         JE    $SOB1           GOT TAB
057.173  311           4159X         RET
057.174               4160          XTEXT TBLS
```

```
                                4162X **      $TBLS - TABLE SEARCH
                                4163X *
                                4164X *       TABLE FORMAT
                                4165X *
                                4166X *       DB      KEY1,VAL1,
                                4167X *       .       .
                                4168X *       .       .
                                4169X *       DB      KEYN,VALN
                                4170X *       DB      0
                                4171X *
                                4172X *       ENTRY   (A) = PATTERN
                                4173X *               (H,L) = TABLE FWA
                                4174X *       EXIT    (A) = PATTERN IF FOUND
                                4175X *               'Z' SET IF FOUND
                                4176X *               'Z' CLEAR IF NOT FOUND OR PATTERN=0          /78.10.GC/
                                4177X *       USES    A,F,H,L
                                4178X
                                4179X
057.174  305                   4180X $TBLS   PUSH    B
057.175  376 000               4181X         CPI     0                                              /78.10.GC/
057.177  312 221 057           4182X         JZ      TBL2                                           /78.10.GC/
057.202  107                   4183X         MOV     B,A
057.203  176                   4184X TBL1    MOV     A,M             (A) = CHARACTER
057.204  043                   4185X         INX     H
057.205  270                   4186X         CMP     B
057.206  312 223 057           4187X         JZ      TBL3            IF MATCH
057.211  247                   4188X         ANA     A
057.212  043                   4189X         INX     H               SKIP PAST
057.213  302 203 057           4190X         JNZ     TBL1            IF NOT END OF TABLE
057.216  053                   4191X         DCX     H
057.217  053                   4192X         DCX     H
057.220  257                   4193X         XRA     A               SET TO ZERO FOR OLD USERS      /78.10.GC/
057.221  376 001               4194X TBL2    CPI     1               CLEAR ZERO                     /78.10.GC/
                                4195X
                                4196X *       DONE
                                4197X
057.223  301                   4198X TBL3    POP     B
057.224  311                   4199X         RET
057.225                        4200          XTEXT   DADA


                                4202X **      $DADA - PERFORM (H,L) = (H,L) + (0,A)
                                4203X *
                                4204X *       ENTRY   (H,L) = BEFORE VALUE
                                4205X *               (A) = BEFORE VALUE
                                4206X *       EXIT    (H,L) = (H,L) + (0,A)
                                4207X *               'C' SET IF OVERFLOW
                                4208X *       USES    F,H,L
                                4209X
                                4210X
030.072                         4211X $DADA   EQU     30072A          IN H17 ROM
057.225                         4212          XTEXT   TJMP
```

```
                        4214X **      $TJMP - TABLE JUMP.
                        4215X *
                        4216X *       USAGE
                        4217X *
                        4218X *       CALL      $TJMP           (A) = INDEX
                        4219X *       DW        ADDR1
                        4220X *       .           .
                        4221X *       .           .
                        4222X *       .           .
                        4223X *       DW        ADDRN
                        4224X *
                        4225X *       ENTRY     (A) = INDEX
                        4226X *       EXIT      TO PROCESSOR
                        4227X *                 (A) = INDEX*2
                        4228X *       USES      NONE.
                        4229X
                        4230X
031.061                 4231X $TJMP   EQU       31061A          IN H17 ROM, (A) = INDEX*2
                        4232X
031.062                 4233X $TJMP.  EQU       31062A          IN H17 ROM
057.225                 4234          XTEXT     CRLF



                        4236X **      $CRLF - TYPE CARRIAGE RETURN/ LINE FEED
                        4237X *
                        4238X *       $CRLF IS USED TO GENERATE PADDED CRLF'S.
                        4239X *
                        4240X *       ENTRY     NONE
                        4241X *       EXIT      (A) = 0
                        4242X *       USES      A,F
                        4243X
                        4244X
057.225  076 012       4245X $CRLF   MVI       A,NL
057.227  377 002       4246X         DB        SYSCALL,.SCOUT
057.231  257          4247X         XRA       A
057.232  311          4248X         RET
057.233                4249          XTEXT     TYPCH



                        4251X **      $TYPCH - TYPE SINGLE CHARACTER.
                        4252X *
                        4253X *       ENTRY     (RET) = CHARACTER
                        4254X *       EXIT      TO (RET)+1
                        4255X *                 (A) = CHARACTER TYPED
                        4256X
                        4257X
057.233  343          4258X $TYPCH  XTHL                      (HL) = RETURN ADDRESS
057.234  176          4259X         MOV       A,M             (A) = CHARACTER
057.235  043          4260X         INX       H
057.236  343          4261X         XTHL                      RESTORE ADVANCED EXIT ADDRESS
                        4262X
                        4263X **      $TYPC. - TYPE SINGLE CHARACTER.
```

```
                                4264X *
                                4265X *         ENTRY   (A) = CHARACTER
                                4266X *         EXIT    TO (RET)
                                4267X
057.237  377 002               4268X $TYPC.    DB      SYSCALL,.SCOUT
057.241  311                   4269X           RET
000.001                         4270  $CMP$     EQU     1
057.242                         4271            XTEXT   TYPLN


                                4273X **        $TYPLN - TYPE LINE.
                                4274X *
                                4275X *         $TYPLN IS CALLED TO TYPE A LINE OF TEXT. ZERO BYTES ARE
                                4276X *         TAKEN AS CRLF (WITH THE PROPER PADDING)
                                4277X *
                                4278X *         CALL    $TYPLN
                                4279X *         DB      N                       BYTE COUNT OF FOLLOWING MESSAGE
                                4280X *         DB      'N-CHARACTER MESSAGE'
                                4281X *
                                4282X *         ENTRY   (RET) = TEXT COUNT
                                4283X *                 (RET)+1 - (RET)+N = TEXT
                                4284X *         EXIT    TO (RET)+N+1
                                4285X *         USES    A,F
                                4286X *
                                4287X
                                4288X
057.242  343                   4289X $TYPLN.   XTHL                            (H,L) = COUNT ADDRESS
057.243  176                   4290X           MOV     A,M                     (A) = COUNT
057.244  043                   4291X           INX     H                       (H,L) = TEXT ADDRESS
057.245  345                   4292X           PUSH    H                       SAVE TEXT FWA
057.246  315 072 030           4293X           CALL    $DADA                   CALCULATE RETURN ADDRESS
057.251  343                   4294X           XTHL                            (HL) = TEXT ADDRE
057.252  315 260 057           4295X           CALL    $TYPL.                  OUTPUT LINE
057.255  341                   4296X           POP     H                       (HL) = RETURN ADDRESS
057.256  343                   4297X           XTHL                            RESTORE (HL), SET RETURN ADDRESS
057.257  311                   4298X           RET
                                4299X
                                4300X **        $TYPL. - TYPE LINE.
                                4301X *
                                4302X *         ENTRY   (HL) = ADDRESS
                                4303X *                 (A) = COUNT
                                4304X *         EXIT    NONE
                                4305X *         USES    A,F,H,L
                                4306X
057.260                         4307X $TYPL.    EQU     *
057.260  247                   4308X           ANA     A
057.261  310                   4309X           RZ                              NOTHING TO TYPE
057.262  365                   4310X           PUSH    PSW                     SAVE COUNT
057.263  176                   4311X           MOV     A,M                     (A) = CHARACTER
057.264  043                   4312X           INX     H
057.265  247                   4313X           ANA     A
000.001                         4314X           IF      $CMP$                   IF HAVE COMPRESSED SPACES
                                4315X           JM      TPL2                    IS COMPRESSED SPACE
                                4316X           ENDIF
```

```
057.266  314 225 057  4317X          CZ      $CRLF
057.271  315 237 057  4318X          CALL    $TYPC.          TYPE CHARACTER
057.274  361         4319X TPL1     POP     PSW
057.275  075         4320X          DCR     A
057.276  302 260 057  4321X          JNZ     $TYPL.
057.301  311         4322X          RET
000.001              4323X          IF      $CMP$           IF COMPRESSED TEXT
                     4324X
                     4325X *        HAVE COMPRESSED SPACE.
                     4326X
                     4327X TPL2     DCR     A
                     4328X          CP      $TYPCH          TYPE 00 IF CHARACTER WAS 200Q
                     4329X          DB      0
                     4330X          ANA     A               SET CODES
                     4331X TPL3     JP      TPL1            ALL EXPANDED
                     4332X          PUSH    PSW             SAVE COUNT
                     4333X          CALL    $TYPCH
                     4334X          DB      ' '
                     4335X          POP     PSW
                     4336X          DCR     A
                     4337X          JMP     TPL3
                     4338X          ENDIF
057.302              4339           XTEXT   TYPT2


                     4341X **       $TYPTX - TYPE TEXT.
                     4342X *
                     4343X *        $TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
                     4344X *
                     4345X *        IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED,
                     4346X *        A BYTE WITH THE 200Q BIT SET IS THE LAST BYTE IN THE MESSAGE.
                     4347X *
                     4348X *        ENTRY   (RET) = TEXT
                     4349X *        EXIT    TO (RET+LENGTH)
                     4350X *        USES    A,F
                     4351X
                     4352X
031.136              4353X $TYPTX   EQU     31136A          IN H17 ROM
                     4354X
031.144              4355X $TYPTX.  EQU     31144A          IN H17 ROM
057.302              4356           XTEXT   COMP


                     4358X **       $COMP - COMPARE TWO CHARACTER STRINGS.
                     4359X *
                     4360X *        $COMP COMPARES TWO BYTE STRINGS.
                     4361X *
                     4362X *        ENTRY   (C) = COMPARE COUNT
                     4363X *                (DE) = FWA OF STRING #1
                     4364X *                (HL) = FWA OF STRING #2
                     4365X *        EXIT    'Z' CLEAR, IS MIS-MATCH
                     4366X *                 (C) = LENGTH REMAINING
```

```
                        4367X *                 (DE) = ADDRESS OF MISMATCH IN STRING#1
                        4368X *                 (HL) = ADDRESS OF MISMATCH IN STRING #2
                        4369X *                 'C' SET, HAVE MATCH
                        4370X *                 (C) = 0
                        4371X *                 (DE) = (DE) + (OC)
                        4372X *                 (HL) = (HL) + (OC)
                        4373X *         USES    A,F,C,D,E,H,L
                        4374X
                        4375X
030,060                 4376X $COMP     EQU     30060A          IN H17 ROM
057,302                 4377            XTEXT   SAVALL


                        4379X **        $RSTALL - RESTORE ALL REGISTERS.
                        4380X *
                        4381X *         $RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND
                        4382X *         RETURNS TO THE PREVIOUS CALLER.
                        4383X *
                        4384X *         ENTRY   (SP) = PSW
                        4385X *                 (SP+2) = BC
                        4386X *                 (SP+4) = DE
                        4387X *                 (SP+6) = HL
                        4388X *                 (SP+8) = RET
                        4389X *         EXIT    TO *RET*, REGISTERS RESTORED
                        4390X *         USES    ALL
                        4391X
                        4392X
031.047                 4393X $RSTALL EQU       31047A          IN H17 ROM


                        4395X **        $SAVALL - SAVE ALL REGISTERS ON STACK.
                        4396X *
                        4397X *         $SAVALL SAVES ALL THE REGISTERS ON THE STACK.
                        4398X *
                        4399X *         ENTRY   NONE
                        4400X *         EXIT    (SP) = PSW
                        4401X *                 (SP+2) = BC
                        4402X *                 (SP+4) = DE
                        4403X *                 (SP+6) = HL
                        4404X *         USES    H,L
                        4405X
                        4406X
031,054                 4407X $SAVALL EQU       31054A          IN H17 ROM
057,302                 4408            XTEXT   CDEHL
```

```
                        4410X **      $CDEHL - COMPARE (DE) TO (HL)
                        4411X *
                        4412X *       $CDEHL COMPARES (DE) TO (HL) FOR EQUALITY.
                        4413X *
                        4414X *       ENTRY   NONE
                        4415X *       EXIT    'Z' SET IF (DE) = (HL)
                        4416X *       USES    A,F
                        4417X
                        4418X
030.216                 4419X $CDEHL  EQU     30216A          IN H17 ROM
057.302                 4420          XTEXT   UDD


                        4422X **      $UDD - UNPACK DECIMAL DIGITS.
                        4423X *
                        4424X *       UDD CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
                        4425X *       DECIMAL DIGITS. THE RESULT IS ZERO FILLED.
                        4426X *
                        4427X *       ENTRY   (B,C) = ADDRESS VALUE
                        4428X *               (A) = DIGIT COUNT
                        4429X *               (H,L) = MEMORY ADDRESS
                        4430X *       EXIT    (HL) = (HL) + (A)
                        4431X *       USES    ALL
                        4432X
                        4433X
031.157                 4434X $UDD    EQU     31157A          IN H17 ROM
057.302                 4435          XTEXT   DU66


                        4437X **      $DU66 - UNSIGNED 16 / 16 DIVIDE.
                        4438X *
                        4439X *       (HL) = (BC)/(DE)
                        4440X *
                        4441X *       ENTRY   (BC), (DE) PRESET
                        4442X *       EXIT    (HL) = RESULT
                        4443X *               (DE) = REMAINDER
                        4444X *       USES    ALL
                        4445X
                        4446X
030.106                 4447X $DU66   EQU     30106A          IN H17 ROM
057.302                 4448          XTEXT   DADA2


                        4450X **      $DADA, - ADD (0,A) TO (H,L)
                        4451X *
                        4452X *       ENTRY   NONE
                        4453X *       EXIT    (HL) = (HL) + (0A)
                        4454X *       USES    A,F,H,L
                        4455X
                        4456X
```

```
030.101              4457X $DADA.  EQU     30101A          IN H17 ROM
057.302              4458          XTEXT   HLIHL

                     4460X **      $HLIHL - LOAD HL INDIRECT THROUGH HL.
                     4461X *
                     4462X *       (HL) = ((HL))
                     4463X *
                     4464X *       ENTRY   NONE
                     4465X *       EXIT    NONE
                     4466X *       USES    A,H,L
                     4467X
030.211              4468X $HLIHL  EQU     30211A          IN H17 ROM
057.302              4469          XTEXT   ILDEHL

                     4471X **      ILDEHL  -  INDEXED LOAD OF DE FROM HL
                     4472X *
                     4473X *       'DE' GET THE FULL WORD VALUE POINTED TO BY 'HL', AND 'HL' IS
                     4474X *       INCREMENTED BY TWO.
                     4475X *
                     4476X *       ENTRY:  HL      = ADDRESS OF FULL WORD VALUE
                     4477X *
                     4478X *       EXIT:   DE      = (HL)
                     4479X *               HL      = HL + 2
                     4480X *
                     4481X *       USES:   DE
                     4482X *
                     4483X
057.302  136        4484X ILDEHL  MOV     E,M
057.303  043        4485X         INX     H
057.304  126        4486X         MOV     D,M
057.305  043        4487X         INX     H
057.306  311        4488X         RET
057.307              4489          XTEXT   INDL

                     4491X **      $INDL - INDEXED LOAD.
                     4492X *
                     4493X *       $INDL LOADS DE WITH THE TWO BYTES AT (HL)+DISPLACMENT
                     4494X *
                     4495X *       THIS ACTS AS AN INDEXED FULL WORD LOAD.
                     4496X *
                     4497X *       (DE) = ( (HL) + DSPLACEMENT )
                     4498X *
                     4499X *       ENTRY   ((RET)) = DISPLACMENT (FULL WORD)
                     4500X *               (HL) = TABLE ADDRESS
                     4501X *       EXIT    TO (RET+2)
                     4502X *       USES    A,F,D,E
                     4503X
```

```
                        4504X
030.234                 4505X $INDL   EQU     30234A          IN H17 ROM
057.307                 4506          XTEXT   INDXX


                        4508X **      $INDLB  -  INDEXED LOAD BYTE
                        4509X *
                        4510X *       BYTE INDEXED LOAD PRIMITIVE
                        4511X *
                        4512X *       ENTRY:  HL      = BASE ADDRESS
                        4513X *               (RET)   = FULL WORD RELOCATION
                        4514X *
                        4515X *       EXIT:   A       = ( HL + (RET) )
                        4516X *
                        4517X *       USES:   A
                        4518X *
                        4519X
057.307  353           4520X $INDLB  XCHG                    DE = BASE
057.310  343           4521X         XTHL                    SAVE  .DE.
057.311  325           4522X         PUSH    D               SAVE  BASE
057.312  305           4523X         PUSH    B               SAVE  .BC.
                        4524X
057.313  116           4525X         MOV     C,M
057.314  043           4526X         INX     H
057.315  106           4527X         MOV     B,M             BC = OFFSET
057.316  043           4528X         INX     H               HL =  .RET.
                        4529X
057.317  353           4530X         XCHG                    HL = BASE
057.320  011           4531X         DAD     B               HL = BASE + OFFSET
057.321  176           4532X         MOV     A,M             A  = ( BASE + OFFSET )
057.322  353           4533X         XCHG                    HL =  .RET.
                        4534X
057.323  301           4535X         POP     B               RESTORE  .BC.
057.324  321           4536X         POP     D               RESTORE  BASE
057.325  343           4537X         XTHL                    HL = .DE. ; (SP) = .RET.
057.326  353           4538X         XCHG                    DE = .DE. ; HL = BASE
057.327  311           4539X         RET


                        4541X **      $INDS   -  INDEXED STORE
                        4542X *
                        4543X *       INDEXED STORE PRIMITIVE.
                        4544X *
                        4545X *       ENTRY:  HL = BASE ADDRESS
                        4546X *               DE = VALUE TO STORE
                        4547X *
                        4548X *       EXIT:   ( HL + (RET) ) = DE
                        4549X *
                        4550X *       USES:   NONE
                        4551X *
                        4552X
057.330  315 306 060   4553X $INDS   CALL    XCHGBC
```

```
057.333  343            4554X          XTHL                    SAVE  .BC.
057.334  325            4555X          PUSH    D
057.335  315 302 057    4556X          CALL    ILDEHL          DE = OFFSET
057.340  315 306 060    4557X          CALL    XCHGBC          BC = .RET.
057.343  353            4558X          XCHG                    DE = BASE ; HL = OFFSET
057.344  031            4559X          DAD     D               HL = BASE + OFFSET
057.345  353            4560X          XCHG
057.346  343            4561X          XTHL                    SAVE  BASE
057.347  353            4562X          XCHG                    DE = VALUE
057.350  315 005 060    4563X          CALL    ISDEHL
057.353  341            4564X          POP     H               HL = BASE
057.354  315 306 060    4565X          CALL    XCHGBC
057.357  343            4566X          XTHL                    RESTORE  .BC.
057.360  315 306 060    4567X          CALL    XCHGBC
057.363  311            4568X          RET


                        4570X **       $INDSB  -  INDEXED BYTE STORE
                        4571X *
                        4572X *        INDEXED BYTE STORE.
                        4573X *
                        4574X *        ENTRY:  A       = VALUE TO STORE
                        4575X *                HL      = BASE ADDRESS
                        4576X *                (RET)   = OFFSET
                        4577X *
                        4578X *        EXIT:   NONE
                        4579X *
                        4580X *        USES:   PSW
                        4581X *
                        4582X
057.364  353            4583X $INDSB   XCHG                    DE = BASE
057.365  343            4584X          XTHL                    SAVE  .DE.
057.366  325            4585X          PUSH    D               SAVE  BASE
057.367  305            4586X          PUSH    B               SAVE  .BC.
                        4587X
057.370  116            4588X          MOV     C,M
057.371  043            4589X          INX     H
057.372  106            4590X          MOV     B,M             BC = OFFSET
057.373  043            4591X          INX     H               HL =  .RET.
                        4592X
057.374  353            4593X          XCHG                    HL = BASE
057.375  011            4594X          DAD     B               HL = BASE + OFFSET
057.376  167            4595X          MOV     M,A             ( BASE + OFFSET ) = A
057.377  353            4596X          XCHG
                        4597X
060.000  301            4598X          POP     B               RESTORE  .BC.
060.001  321            4599X          POP     D               RESTORE  BASE
060.002  343            4600X          XTHL                    HL = .DE. ; (SP) = .RET.
060.003  353            4601X          XCHG                    DE = .DE. ; HL = BASE
060.004  311            4602X          RET
060.005                 4603           XTEXT   ISDEHL
```

```
                                4605X **      ISDEHL  -  INDEXED STORE OF DE AT HL
                                4606X *
                                4607X *       STORE 'DE' AT THE ADDRESS POINTED TO BY 'HL', AND INCREMENT 'HL'
                                4608X *       BY 2.
                                4609X *
                                4610X *       ENTRY:  DE      = VALUE
                                4611X *               HL      = ADDRESS OF VALUE
                                4612X *
                                4613X *       EXIT:   (HL)    = DE
                                4614X *               HL      = HL + 2
                                4615X *
                                4616X *       USES:   HL
                                4617X *
                                4618X
060.005  163                    4619X ISDEHL  MOV     M,E
060.006  043                    4620X         INX     H
060.007  162                    4621X         MOV     M,D
060.010  043                    4622X         INX     H
060.011  311                    4623X         RET
060.012                         4624          XTEXT   DAD


                                4626X **      $DAD - DECODE AUGUSTAN DATE.
                                4627X *
                                4628X *       $DAD DECODES A 15 BIT DATE CODE OF THE FORMAT:
                                4629X *
                                4630X *       ---------------------------------------
                                4631X *       I  0  I  6 BITS  I  4 BITS  I  5 BITS  I
                                4632X *       ---------------------------------------
                                4633X *              YEAR-70       MON       DAY
                                4634X *               1-63        1-12      1-31
                                4635X *
                                4636X *       TO THE FORM:
                                4637X *
                                4638X *       DD-MMM-YY
                                4639X *
                                4640X *       ENTRY   (DE) = 15 BIT VALUE
                                4641X *               (HL) = ADDRESS FOR DECODE
                                4642X *       EXIT    'C' CLEAR IF OK
                                4643X *                (DE) = (DE)+9
                                4644X *               'C' SET IF ERROR
                                4645X *       USES    ALL
                                4646X
                                4647X
060.012  102                    4648X $DAD    MOV     B,D
060.013  113                    4649X         MOV     C,E
060.014  021 040 000            4650X         LXI     D,32
060.017  345                    4651X         PUSH    H               SAVE ADDRESS
060.020  315 106 030            4652X         CALL    $DU66           (DE) = DAY, (HL) = YEAR & MONTH
060.023  343                    4653X         XTHL                    (HL) = ADDRESS
060.024  102                    4654X         MOV     B,D
060.025  113                    4655X         MOV     C,E
060.026  173                    4656X         MOV     A,E
060.027  247                    4657X         ANA     A
```

```
060.030  312 130 060  4658X          JZ      DAD1            BAD VALUE
060.033  076 002      4659X          MVI     A,2
060.035  315 157 031  4660X          CALL    $UDD            UNPACK DAY
060.040  066 055      4661X          MVI     M,'-'
060.042  043          4662X          INX     H
060.043  301          4663X          POP     B               (BC) = YEAR & MONTH
060.044  021 020 000  4664X          LXI     D,16
060.047  345          4665X          PUSH    H               SAVE ADDRESS
060.050  315 106 030  4666X          CALL    $DU66
060.053  343          4667X          XTHL                    (HL) = ADDRESS, ((SP)) = YEAR
060.054  173          4668X          MOV     A,E
060.055  207          4669X          ADD     A
060.056  203          4670X          ADD     E               (A) = 3*MONTH
060.057  312 130 060  4671X          JZ      DAD1            BAD VALUE
060.062  376 047      4672X          CPI     13*3
060.064  322 130 060  4673X          JNC     DAD1            TOO LARGE
060.067  353          4674X          XCHG                    (DE) = ADDRESS
060.070  041 130 060  4675X          LXI     H,DADB-3
060.073  315 101 030  4676X          CALL    $DADA.          (HL) = ADDRESS OF MONTH
060.076  001 003 000  4677X          LXI     B,3
060.101  353          4678X          XCHG                    (HL) = BUFFER ADDR, (DE) = ADDR IN 'DADB'
060.102  315 252 030  4679X          CALL    $MOVE           MOVE MONTH IN
060.105  066 055      4680X          MVI     M,'-'
060.107  043          4681X          INX     H
060.110  301          4682X          POP     B               (BC) = YEAR
060.111  171          4683X          MOV     A,C
060.112  306 106      4684X          ADI     70
060.114  376 144      4685X          CPI     100
060.116  077          4686X          CMC
060.117  330          4687X          RC                      TOO LARGE
060.120  117          4688X          MOV     C,A             (BC) = YEAR
060.121  076 002      4689X          MVI     A,2
060.123  315 157 031  4690X          CALL    $UDD            UNPACK YEAR
060.126  247          4691X          ANA     A
060.127  311          4692X          RET
                      4693X
                      4694X *        ILLEGAL FORMAT. (NOT ALL ILLEGALS EXIT HERE!)
                      4695X
060.130  341          4696X DAD1     POP     H               RESTORE STACK
060.131  067          4697X          STC                     FLAG ERROR
060.132  311          4698X          RET
                      4699X
060.133  112 141 156  4700X DADB     DB      'JanFebMarAprMayJunJulAugSepOctNovDec'
060.177               4701           XTEXT   UDDN


                      4703X **       $UDDN - UNPACK DECIMAL DIGITS.
                      4704X *
                      4705X *        UDDN CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
                      4706X *        DECIMAL DIGITS. THE RESULT IS NULL FILLED TO THE LEFT.
                      4707X *
                      4708X *        ENTRY   (B,C) = ADDRESS VALUE
                      4709X *                (A) = DIGIT COUNT
                      4710X *                (H,L) = MEMORY ADDRESS
```

```
                        4711X *       EXIT    (HL) = (HL) + (A)
                        4712X *       USES    ALL
                        4713X
                        4714X
060.177                 4715X $UDDN   EQU     *
060.177  315 072 030    4716X         CALL    $DADA
060.202  345            4717X         PUSH    H               SAVE FINAL (H,L) VALUE
                        4718X
060.203  365            4719X UDDN1   PUSH    PSW
060.204  345            4720X         PUSH    H
060.205  021 012 000    4721X         LXI     D,10
060.210  315 106 030    4722X         CALL    $DU66           (H,L) = VALUE/10
060.213  104            4723X         MOV     B,H
060.214  115            4724X         MOV     C,L             (BC) = QUOTIENT
060.215  341            4725X         POP     H
060.216  076 060        4726X         MVI     A,'0'
060.220  203            4727X         ADD     E               ADD REMAINDER
060.221  053            4728X         DCX     H
060.222  167            4729X         MOV     M,A             STORE DIGIT
060.223  170            4730X         MOV     A,B
060.224  261            4731X         ORA     C
060.225  312 237 060    4732X         JZ      UDDN2           ALL ZEROS
060.230  361            4733X         POP     PSW
060.231  075            4734X         DCR     A
060.232  302 203 060    4735X         JNZ     UDDN1           IF MORE TO GO
                        4736X
                        4737X *       ALL DONE. EXIT
                        4738X
060.235  341            4739X UDDN1.5 POP     H               RESTORE H
060.236  311            4740X         RET                       RETURN
                        4741X
                        4742X *       DIGITS LEADING THIS ONE ARE ZERO. STORE NULLS INSTEAD.
                        4743X
060.237  361            4744X UDDN2   POP     PSW
060.240  075            4745X UDDN3   DCR     A
060.241  312 235 060    4746X         JE      UDDN1.5         ALL DONE
060.244  053            4747X         DCX     H
060.245  066 000        4748X         MVI     M,0
060.247  303 240 060    4749X         JMP     UDDN3
060.252                 4750          XTEXT   MOVEL
```

```
                        4752X **      $MOVEL - MOVE DATA
                        4753X *
                        4754X *       $MOVEL MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                        4755X *       IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                        4756X *       FIRST TO LAST.
                        4757X *
                        4758X *       IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                        4759X *       LAST TO FIRST.
                        4760X *
                        4761X *       THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                        4762X *
                        4763X *       CALL    $MOVEL
```

```
                        4764X *         DW      COUNT
                        4765X *         DW      FROM
                        4766X *         DW      TO
                        4767X *
                        4768X *         ENTRY   ((SP)) = RET
                        4769X *                 (RET+0) = COUNT (WORD VALUE)
                        4770X *                 (RET+2) = FROM
                        4771X *                 (RET+4) = TO
                        4772X *         EXIT    TO (RET+6)
                        4773X *                 (DE) = ADDRESS OF NEXT FROM BYTE
                        4774X *                 (HL) = ADDRESS OF NEXT *TO* BYTE
                        4775X *                 'C' CLEAR
                        4776X *         USES    ALL
                        4777X
                        4778X
060.252  341           4779X $MOVEL  POP     H               (HL) = RET
060.253  116           4780X         MOV     C,M
060.254  043           4781X         INX     H
060.255  106           4782X         MOV     B,M             (BC) = COUNT
060.256  043           4783X         INX     H
060.257  136           4784X         MOV     E,M
060.260  043           4785X         INX     H
060.261  126           4786X         MOV     D,M             (DE) = FROM
060.262  043           4787X         INX     H
060.263  325           4788X         PUSH    D               ((SP)) = FROM
060.264  136           4789X         MOV     E,M
060.265  043           4790X         INX     H
060.266  126           4791X         MOV     D,M             (DE) = TO
060.267  043           4792X         INX     H
060.270  343           4793X         XTHL                    ((SP)) = RET, (HL) = FROM
060.271  353           4794X         XCHG                    (DE) = FROM , (HL) = TO
060.272  303 252 030   4795X         JMP     $MOVE           MOVE IT
060.275               4796          XTEXT   RCHAR


                        4798X **        $RCHAR - READ SINGLE CHARACTER FROM CONSOLE,
                        4799X *
                        4800X *         ENTRY   NONE
                        4801X *         EXIT    (A) = CHARACTER
                        4802X *         USES    A,F
                        4803X
                        4804X
060.275  377 001       4805X $RCHAR  DB      SYSCALL,,SCIN
060.277  332 275 060   4806X         JC      $RCHAR          NOT READY
060.302  311           4807X         RET
                        4808X
060.303  377 002       4809X $WCHAR  DB      SYSCALL,,SCOUT
060.305  311           4810X         RET
060.306               4811          XTEXT   XCHGBC
```

```
                        4813X **      XCHGBC  -  XCHG BC
                        4814X *
                        4815X *       EXCHANGE THE 'BC' REGISTER PAIR WITH THE 'HL' REGISTER PAIR.
                        4816X *
                        4817X *       ENTRY:  BC      = ORIGINAL BC
                        4818X *               HL      = ORIGINAL HL
                        4819X *
                        4820X *       EXIT:   BC      = ORIGINAL HL
                        4821X *               HL      = ORIGINAL BC
                        4822X *
                        4823X *       USES:   BC,HL
                        4824X *
                        4825X
060.306  365            4826X XCHGBC  PUSH    PSW
060.307  170            4827X         MOV     A,B
060.310  104            4828X         MOV     B,H
060.311  147            4829X         MOV     H,A
060.312  171            4830X         MOV     A,C
060.313  115            4831X         MOV     C,L
060.314  157            4832X         MOV     L,A
060.315  361            4833X         POP     PSW
060.316  311            4834X         RET
060.317                 4835          XTEXT   DRS


                        4837X **      $DRS - DECODE AND REMOVE SWITCHES.
                        4838X *
                        4839X *       $DRS IS CALLED TO DECODE COMMAND SWITCHES FROM A LINE
                        4840X *       OF TEXT.  SWITCHES TAKE THE FORM:
                        4841X *
                        4842X *       /XXXXX
                        4843X *
                        4844X *       AFTER A SWITCH HAS BEEN LOCATED, IT (AND THE PRECEDING '/')
                        4845X *       ARE REPLACED WITH BLANKS.
                        4846X *
                        4847X *       VALID SWITCH DESCRIPTIONS ARE ENCODED INTO A TABLE
                        4848X *       SUPPLIED BY THE CALLER, IN THE FORMAT:
                        4849X *
                        4850X *       DB      'X...X'         REQUIRED SWITCH CHARACTERS
                        4851X *       DB      'C'+200Q,...,'C'+200Q   OPTIONAL CHARACTERS
                        4852X *       DB      200Q            END OF CHARACTERS
                        4853X *       DW      ADDR            PROCESSOR ADDRESS (CALLED WHEN SWITCH DETECTED)
                        4854X *
                        4855X *       DB      'Y...Y'         NEXT SWITCH
                        4856X *       .       .
                        4857X *       .       .
                        4858X *       .       .
                        4859X *
                        4860X *       DB      0               FLAGS END OF TABLE
                        4861X *
                        4862X *       SWITCHES MUST BE FOLLOWED BY A ':', A '/' (ANOTHER SWITCH)
                        4863X *       A ',', OR A 00 BYTE.
                        4864X *
                        4865X *       UPON DETECTION OF A VALID SWITCH, $DRS CALLS THE USER PROCESS
```

```
                                4866X *       ROUTINE. UPON ENTRY,
                                4867X *       (HL) = ADDRESS OF THE FIRST BYTE FOLLOWING THE SWTICH
                                4868X *       'Z' CLEAR IF CHARACTER = '/', ',', OR 00
                                4869X *       'Z' SET IF CHARACTER = ':'
                                4870X *
                                4871X *       THE USER ROUTINE CAN DECODE SWITCH SUB-OPTIONS, IF DESIRED.
                                4872X *       THE USER ROUTINE MAY USE ALL REGISTERS.
                                4873X *
                                4874X *       ENTRY   (DE) = SWITCH TABLE FWA
                                4875X *               (HL) = LINE FWA
                                4876X *       EXIT    'C' CLEAR IF OK
                                4877X *               'C' SET IF ERROR
                                4878X *                (HL) = ADDRESS OF START OF BAD SWITCH
                                4879X *                (A) = ERROR CODE
                                4880X *       USES    ALL
                                4881X
                                4882X
060.317                         4883X $DRS    EQU     *
                                4884X
                                4885X *       LOOK FOR SWITCHES
                                4886X
060.317  176                    4887X $DRS1   MOV     A,M
060.320  247                    4888X         ANA     A
060.321  310                    4889X         RZ                      END OF LINE
060.322  043                    4890X         INX     H
060.323  376 057                4891X         CPI     '/'
060.325  302 317 060            4892X         JNE     $DRS1           NOT A SWITCH
060.330  042 114 061            4893X         SHLD    $DRSB           ($DRSB) = SWITCH FWA (AFTER '/')
                                4894X
                                4895X *       GOT A SWITCH, LOOK FOR A MATCH IN THE CALLER'S TABLE
                                4896X
060.333  325                    4897X         PUSH    D               SAVE TABLE FWA
060.334  052 114 061            4898X $DRS2   LHLD    $DRSB           (HL) = SWITCH FWA
060.337  032                    4899X $DRS3   LDAX    D               (A) = TABLE ENTRY
060.340  346 177                4900X         ANI     177Q
060.342  312 012 061            4901X         JZ      $DRS6           GOT A MATCH
060.345  276                    4902X         CMP     M
060.346  302 356 060            4903X         JNE     $DRS4           NO MATCH
060.351  023                    4904X         INX     D
060.352  043                    4905X         INX     H
060.353  303 337 060            4906X         JMP     $DRS3           SEE IF MORE MATCH
                                4907X
                                4908X *       HAVE MIS-MATCH. SEE IF THE MISSING CHARACTER IS SIGNIFICANT
                                4909X
                                4910X $DRS4   MOV     A,M             (A) = LINE CHARACTER WE COULDNT MATCH
060.356  176
060.357  315 063 061            4911X         CALL    $DRS15          SEE IF OK TERMINATOR
060.362  302 372 060            4912X         JNE     $DRS4.5         NO MATCH ON THIS SWITCH
060.365  032                    4913X         LDAX    D               (A) = NEXT CHARACTER IN SWITCH PATTERN
060.366  247                    4914X         ANA     A
060.367  372 012 061            4915X         JM      $DRS6           HAVE SUFFICIENT MATCH
060.372  315 076 061            4916X $DRS4.5 CALL    $DRS20          SKIP TABLE ENTRY
060.375  032                    4917X         LDAX    D
060.376  247                    4918X         ANA     A
060.377  302 334 060            4919X         JNZ     $DRS2           MORE SWITCHES IN TABLE TO CHECK
                                4920X
                                4921X *       BAD SWITCH
```

```
                                4922X
061.002  321                   4923X $DRS5   POP     D               RESTORE STACK
061.003  052 114 061           4924X         LHLD    $DRSB           POINT TO BAD SWITCH
061.006  067                   4925X         STC
061.007  076 032               4926X         MVI     A,EC.IS         ILLEGAL SWITCH
061.011  311                   4927X         RET
                                4928X
                                4929X *       HAVE SWITCH. CHECK IT'S FOLLOWING CHARACTER
                                4930X
061.012  315 156 057           4931X $DRS6   CALL    $SOB            SKIP OVER BLANKS
061.015  176                   4932X         MOV     A,M
061.016  315 063 061           4933X         CALL    $DRS15          CHECK CHARACTER
061.021  302 002 061           4934X         JNE     $DRS5           IN ERROR
061.024  315 076 061           4935X         CALL    $DRS20          GET PROCESSOR ADDRESS
061.027  021 041 061           4936X         LXI     D,$DRS7
061.032  345                   4937X         PUSH    H               SAVE (HL)
061.033  325                   4938X         PUSH    D               SET RETURN ADDRESS FOR TABLE CODE
061.034  305                   4939X         PUSH    B               SAVE PROCESSOR ADDRESS
061.035  176                   4940X         MOV     A,M             (A) = NEXT CHARACTER
061.036  376 072               4941X         CPI     ':'             SET CONDITION CODES
061.040  311                   4942X         RET                     CALL USER PROCESS
                                4943X
                                4944X *       USER PROCESS RETURNS HERE
                                4945X
061.041  321                   4946X $DRS7   POP     D               (DE) = LAST CHARACTER OF SWITCH+1
061.042  052 114 061           4947X         LHLD    $DRSB           (HL) = FIRST CHARACTER OF SWITCH AFTER /
061.045  053                   4948X         DCX     H               (HL) = ADDRESS OF '/'
                                4949X
                                4950X *       REPLACE SWITCH WITH BLANKS
                                4951X
061.046  066 040               4952X $DRS8   MVI     M,' '
061.050  043                   4953X         INX     H
061.051  315 216 030           4954X         CALL    $CDEHL
061.054  302 046 061           4955X         JNE     $DRS8           NOT THERE YET
061.057  321                   4956X         POP     D               (DE) = SWITCH TABLE FWA
061.060  303 317 060           4957X         JMP     $DRS1           LOOK FOR MORE SWITCHES

                                4959X **      $DRS15 - CHECK FOR VALID DELIMITER CHARACTER.
                                4960X *
                                4961X *       $DRS15 CHECKS THE NEXT TEXT CHARACTER TO SEE IF IT IS
                                4962X *
                                4963X *       00, '/', ',', ':'
                                4964X *
                                4965X *       ENTRY   (A) = CHARACTER
                                4966X *       EXIT    'Z' SET IFF CHARACTER IS ONE OF THE ABOVE
                                4967X *       USES    F
                                4968X
061.063  247                   4969X $DRS15  ANA     A
061.064  310                   4970X         RZ                      IS 00
061.065  376 057               4971X         CPI     '/'
061.067  310                   4972X         RE
061.070  376 054               4973X         CPI     ','
061.072  310                   4974X         RE
061.073  376 072               4975X         CPI     ':'
061.075  311                   4976X         RET
```

```
                        4978X **      $DRS20 - GET PROCESSOR ADDRESS.
                        4979X *
                        4980X *       $DRS20 IS CALLED TO GET THE PROCESSOR ADDRESS FIELD OUT OF
                        4981X *       AN ENTRY IN THE SWITCH TABLE. THE CALLER SUPPLIES A POINTER
                        4982X *       TO SOMEWHERE IN THE TEXT PART OF THE SWITCH DESCRIPTION;
                        4983X *       $DRS20 ADVANCES THE POINTER TO THE PROCESSOR ADDRESS.
                        4984X *
                        4985X *       ENTRY   (DE) = POINTER TO TEXT PART OF SWITCH ENTRY
                        4986X *       EXIT    (DE) = POINTER TO 1ST BYTE OF NEXT SWITCH TABLE ENTRY
                        4987X *               (BC) = PROCESSOR ADDRESS FROM TABLE
                        4988X *       USES    A,F,B,C,D,E
                        4989X
                        4990X
061,076  032           4991X $DRS20  LDAX    D
061,077  023           4992X         INX     D
061,100  376 200       4993X         CPI     200Q
061,102  302 076 061   4994X         JNE     $DRS20
061,105  032           4995X         LDAX    D               (A) = LOW BYTE OF PROCESSOR ADDRESS
061,106  117           4996X         MOV     C,A
061,107  023           4997X         INX     D
061,110  032           4998X         LDAX    D
061,111  107           4999X         MOV     B,A             (BC) = PROCESSOR ADDRESS
061,112  023           5000X         INX     D
061,113  311           5001X         RET
                        5002X
061,114  000 000       5003X $DRSB   DW      0               POINTER TO SWITCH BEING PROCESSED
000,001                 5004          IF      .PIP.
                        5005          XTEXT   DTB
                        5006          XTEXT   FOPE
                        5007          XTEXT   FWRIB
                        5008          XTEXT   FCLO
                        5009          XTEXT   FUTIL
                        5010          ELSE
061,116                 5011          XTEXT   BRP


                        5013X **      BRP - BAUD RATE PROMPT
                        5014X *
                        5015X *       Prompt console for baud rate determining spaces at interrupt time
                        5016X *       if current console is 8250.  Should be used before jumping to
                        5017X *       ROMBOOT.
                        5018X *
                        5019X *       ENTRY   S.CDB  = CONSOLE DEFINITION BYTE, describes current console.
                        5020X *       EXIT    NONE
                        5021X *       USES    NONE
                        5022X *
                        5023X
061,116  365           5024X BRP     PUSH    PSW
061,117  377 007       5025X         DB      SYSCALL,.CLRCO  CLEAR ANY TYPE-AHEAD
061,121  315 136 031   5026X         CALL    $TYPTX
061,124  012 111 156   5027X         DB      NL,'Install a Bootable Disk in SY0:.  Hit Return to Reboot:'
061,214  240           5028X         DB      ' '+200Q
061,215  377 001       5029X BRP0    DB      SYSCALL,.SCIN   WAIT FOR A NEWLINE
061,217  376 012       5030X         CPI     NL
```

```
061.221  302 215 061  5031X          JNZ     BRP0
061.224  072 343 040  5032X          LDA     S.CDB
061.227  376 001      5033X          CPI     CDB.H84
061.231  302 303 061  5034X          JNZ     BRP1            IF  NOT 8250
                      5035X
061.234  315 136 031  5036X          CALL    $TYPTX
061.237  012 124 171  5037X          DB      NL,'Type spaces to determine BAUD RATE',ENL
                      5038X.
061.303  076 156      5039X BRP1     MVI     A,AC.DLY
061.305  315 053 000  5040X          CALL    .DLY            WAIT FOR CHARACTER TO BE OUTPUT
061.310  257          5041X          XRA     A
061.311  323 351      5042X          OUT     SC.ACE+UR.IER   CLEAR CONSOLE
061.313  323 373      5043X          OUT     SC.UART+USR
061.315  361          5044X          POP     PSW
061.316  311          5045X          RET
                      5046           ENDIF
```

```
061.317                5049  PATCH   DS      64              PATCH AREA
```

```
000.000                 5052            IF      ONECOPY
                        5053
                        5054
                        5055    **      FDN - FILE DESCRIPTOR NODES.
                        5056    *
                        5057    *       THESE NODES ARE USED TO KEEP TRACK OF FILES WHICH ARE BEING
                        5058    *       HELD IN MEMORY WHILE TRANSFERING.
                        5059
062.017                 5060    FDN     DS      0               START OF TYPICAL NODE
000.000                 5061    FDN.LNK EQU     *-FDN           LINK TO NEXT NODE IN CHAIN
062.017                 5062            DS      1               ALL IN SAME PAGE, JUST KEEP PAGE INDEX
000.001                 5063    FDN.STA EQU     *-FDN           STATUS BYTE
000.020                 5064    ST.CNT  EQU     DIF.CNT         IS CONTIGUOUS
000.002                 5065    ST.OPR  EQU     00000010B       IS BEING READ
000.001                 5066    ST.OPW  EQU     00000001B       OPEN FOR WRITE
062.020                 5067            DS      1               STATUS BYTE
000.002                 5068    FDN.SIZ EQU     *-FDN           TOTAL SIZE OF FILE (IF ST.CNT SET)
062.021                 5069            DS      1               SIZE IN GROUPS
000.003                 5070    FDN.AMR EQU     *-FDN           AMOUNT ALREADY READ
062.022                 5071            DS      2                IN SECTORS
000.005                 5072    FDN.AMW EQU     *-FDN           AMOUNT ALREADY WRITTEN
062.024                 5073            DS      2                IN SECTORS
000.007                 5074    FDN.ADR EQU     *-FDN           ADDRESS IN BUFFER
062.026                 5075            DS      1                ADDRESS/256 (MUST BE EVEN PAGE)
000.010                 5076    FDN.AIM EQU     *-FDN           AMOUNT IN MEMORY
062.027                 5077            DS      1                 IN SECTORS
000.011                 5078    FDNELEN EQU     *-FDN           ENTRY LENGTH
062.017                 5079            ORG     FDN             ORG BACK OVER DEFINITION AREA
                        5080
                        5081
                        5082
                        5083    **      TABLE. A LINK OF 0 IS A NULL LINK.
                        5084    *
                        5085    *       THE ENTIRE GROUP OF NODES MUST RESIDE
                        5086    *       IN THE SAME PAGE
                        5087
062.017                 5088    FDNFWA  EQU     *               START OF NODES
                        5089
062.017  021            5090    FDNFRE  DB      #FDN.1          START OF FREE CHAIN
062.020  000            5091    FDNHEAD DB      0                       ACTIVE LIST NOW EMPTY
                        5092
062.021                 5093    FDN.1   DS      0
062.021  032            5094            DB      #FDN.2          FDN.LNK
062.022  000            5095            DB      0               FDN.STA
062.023  000            5096            DB      0               FDN.SIZ
062.024  000 000        5097            DW      0               FDN.AMR
062.026  000 000        5098            DW      0               FDN.AMW
062.030  000            5099            DB      0               FDN.ADR
062.031  000            5100            DB      0               FDN.AIM
                        5101
062.032                 5102    FDN.2   DS      0
062.032  043            5103            DB      #FDN.3          FDN.LNK
062.033  000            5104            DB      0               FDN.STA
062.034  000            5105            DB      0               FDN.SIZ
062.035  000 000        5106            DW      0               FDN.AMR
062.037  000 000        5107            DW      0               FDN.AMW
```

```
062.041  000          5108           DB      0               FDN.ADR
062.042  000          5109           DB      0               FDN.AIM
                      5110
062.043              5111  FDN.3     DS      0
062.043  054          5112           DB      #FDN.4          FDN.LNK
062.044  000          5113           DB      0               FDN.STA
062.045  000          5114           DB      0               FDN.SIZ
062.046  000 000      5115           DW      0               FDN.AMR
062.050  000 000      5116           DW      0               FDN.AMW
062.052  000          5117           DB      0               FDN.ADR
062.053  000          5118           DB      0               FDN.AIM
                      5119
062.054              5120  FDN.4     DS      0
062.054  065          5121           DB      #FDN.5          FDN.LNK
062.055  000          5122           DB      0               FDN.STA
062.056  000          5123           DB      0               FDN.SIZ
062.057  000 000      5124           DW      0               FDN.AMR
062.061  000 000      5125           DW      0               FDN.AMW
062.063  000          5126           DB      0               FDN.ADR
062.064  000          5127           DB      0               FDN.AIM
                      5128
062.065              5129  FDN.5     DS      0
062.065  076          5130           DB      #FDN.6          FDN.LNK
062.066  000          5131           DB      0               FDN.STA
062.067  000          5132           DB      0               FDN.SIZ
062.070  000 000      5133           DW      0               FDN.AMR
062.072  000 000      5134           DW      0               FDN.AMW
062.074  000          5135           DB      0               FDN.ADR
062.075  000          5136           DB      0               FDN.AIM
                      5137
062.076              5138  FDN.6     DS      0
062.076  107          5139           DB      #FDN.7          FDN.LNK
062.077  000          5140           DB      0               FDN.STA
062.100  000          5141           DB      0               FDN.SIZ
062.101  000 000      5142           DW      0               FDN.AMR
062.103  000 000      5143           DW      0               FDN.AMW
062.105  000          5144           DB      0               FDN.ADR
062.106  000          5145           DB      0               FDN.AIM
                      5146
062.107              5147  FDN.7     DS      0
062.107  120          5148           DB      #FDN.8          FDN.LNK
062.110  000          5149           DB      0               FDN.STA
062.111  000          5150           DB      0               FDN.SIZ
062.112  000 000      5151           DW      0               FDN.AMR
062.114  000 000      5152           DW      0               FDN.AMW
062.116  000          5153           DB      0               FDN.ADR
062.117  000          5154           DB      0               FDN.AIM
                      5155
062.120              5156  FDN.8     DS      0
062.120  000          5157           DB      0               FDN.LNK
062.121  000          5158           DB      0               FDN.STA
062.122  000          5159           DB      0               FDN.SIZ
062.123  000 000      5160           DW      0               FDN.AMR
062.125  000 000      5161           DW      0               FDN.AMW
062.127  000          5162           DB      0               FDN.ADR
062.130  000          5163           DB      0               FDN.AIM
```

```
                                5164
000.010                         5165  FDNCNT  EQU     *-FDN.1/FDNELEN         NUMBER OF NODES
                                5166
000.062                         5167          SET     */256
000.000                         5168          ERRNZ   FDNFWA/256-.    MUST BE ALL IN SAME PAGE
                                5169
062.131  000                   5170  VOLFLAG DB      0               =0 IF READING FROM SOURCE, =377Q IF WRITTING TO DEST
062.132  000                   5171  VOLSER  DB      0               SERIAL NUMBER OF CURRENT DISK
                                5172
062.133  000                   5173  OBUFLIM DB      0               BUFFER LIMIT/256
062.134  000                   5174  OBUFPTR DB      0               NEXT FREE PAGE IN BUFFER/256
                                5175
                                5176
                                5177          ENDIF
                                5178
062.135                         5179          XTEXT   FERROR          APPEARS HERE TO ALLOW FDN. TO BE IN ONE PAGE


                                5181X **       $FERROR - PROCESS FILE ERRORS.
                                5182X *
                                5183X *        $FERROR IS CALLED TO COMPLAIN ABOUT AN ERROR ENCOUNTERED
                                5184X *        WHEN PROCESSING FILES.
                                5185X *
                                5186X *        ENTRY   (A) = ERROR CODE
                                5187X *                (HL) = ADDRESS OF FILE NAME - FB.NAM
                                5188X *        EXIT    TO RESTART
                                5189X *        USES    ALL
                                5190X
                                5191X
062.135  365                    5192X $FERROR PUSH    PSW             SAVE CODE
062.136  315 136 031            5193X         CALL    $TYPTX
062.141  012 007 105            5194X         DB      NL,BELL,'ERROR ON FILE',' '+200Q
062.161  021 012 000            5195X         LXI     D,FB.NAM
062.164  031                    5196X         DAD     D
                                5197X
                                5198X *        PRINT FILE NAME
                                5199X
062.165  176                    5200X $FERR1  MOV     A,M
062.166  043                    5201X         INX     H               ADVANCE MESSAGE
062.167  247                    5202X         ANA     A
062.170  312 201 062            5203X         JZ      $FERR2
062.173  315 303 060            5204X         CALL    $WCHAR
062.176  303 165 062            5205X         JMP     $FERR1
                                5206X
                                5207X *        TYPE ERROR MESSAGE
                                5208X
062.201  315 136 031            5209X $FERR2  CALL    $TYPTX
062.204  040 055 240            5210X         DB      ' -',' '+200Q
062.207  046 012                5211X         MVI     H,NL
062.211  361                    5212X         POP     PSW             (A) = CODE
062.212  377 057                5213X         DB      SYSCALL,.ERROR
062.214  303 200 042            5214X         JMP     RESTART         EXIT
```

```
062.217  000          5217  COMAND  DB      0               COMMAND IN PROGRESS
062.220  000          5218  MODE    DB      0               <>0 IF LINE PASSED ON STACK
062.221  000          5219  JGL     DB      0               /JGL FLAG (<>0 IF /JGL SPECIFIED)
062.222  000          5220  SUPRES  DB      0               /SUP FLAG (<>0 OF /SUP SPECIFIED)
062.223  001          5221  SYSTEM  DB      1               /S FLAG (=0 IF /S SPECIFIED)
                      5222
062.224  130 130 130  5223  DIRNAM  DB      'XXX:DIRECT.SYS',0      DIRECTORY FILE NAME
                      5224
062.243  132 063      5225  BUFPTR  DW      BUFF            POINTER TO START OF BUFFER
062.245  000 000      5226  BUFSIZ  DW      0               BUFFER LENGTH


                      5228  **      FILE BLOCKS
                      5229
000.001               5230          IF      .PIP.
                      5231  DESTFB  DS      0               DESTINATION FILE BLOCK
                      5232          DB      CN.DES          CHANNEL NUMBER
                      5233          DB      0               FLAGS
                      5234          DW      DESTBUF
                      5235          DW      DESTBUF
                      5236          DW      DESTBUF
                      5237          DW      DESTBFE         END OF BLOCK
                      5238          DS      FB.NAML         NAME AREA
                      5239          ELSE
062.247               5240  DESTFB  DS      0               DUMY BUFFER
062.247  310          5241          DB      200             ILLEGAL CHANNEL NUMBER
062.250  000          5242          DB      0               FLAGS
062.251  000 000      5243          DW      0
062.253  000 000      5244          DW      0
062.255  000 000      5245          DW      0
062.257  000 000      5246          DW      0               END OF BLOCK
062.261               5247          DS      FB.NAML         NAME AREA
                      5248          ENDIF


062.302  000 000      5250  NAMTLEN DW      0               NAME TABLE POINTER
062.304  000 000      5251  NAMTMAX DW      0               MAXIMUM SIZE OF NAME TABLE
000.000               5252          IF      ONECOPY
062.306  000 000      5253  NAMTPTR DW      0               POINTER TO ACTIVE ELEMENT IN NAMTAB
                      5254          ENDIF
                      5255
```

```
                                5259  ***     PRS - PRESET PIP PROGRAM.
                                5260  *
                                5261  *       PRS IS CALLED TO PERFORM ONE-TIME-ONLY PRESETTING OF
                                5262  *       THE PROGRAM ENVIRONMENT.
                                5263  *
                                5264  *       THE CODE IS OVERLAID BY BUFFERS AND WORK AREAS WHEN PIP IS RUNNING.
        000.001                 5265          IF      .PIP.
                                5266  *       BE CAREFUL NOT TO USE ANY OF THE BUFFERS AND WORK AREAS BEFORE
                                5267  *       THE AREA *LINE*.
                                5268          ELSE
                                5269  *       DO NOT USE ANY OF THE BUFFERS AND WORK AREAS IN *PRS*
                                5270          ENDIF
                                5271  *
                                5272  *
                                5273  *       ENTRY   NONE
                                5274  *
                                5275  *       EXIT    IF  CORRECT VERSION OF HDOS
                                5276  *                   NONE
                                5277  *                 ELSE
                                5278  *                   EXIT TO HDOS
                                5279  *
                                5280  *       USES    ALL
                                5281  *
                                5282
        062.310                 5283  ENTRY   EQU     *               INITIAL ENTRY POINT
        062.310  377 011       5284  PRS     DB      SYSCALL,.VERS
        062.312  332 361 063   5285          JC      PRS1            ERROR IN GETTING VERSION
        062.315  376 026       5286          CPI     VERS
        062.317  302 361 063   5287          JNZ     PRS1            NOT CORRECT VERSION OF HDOS
        062.322  041 132 063   5288          LXI     H,RMEML         (HL) = RUN-TIME HIGH MEMORY
        062.325  377 052       5289          DB      SYSCALL,.SETTP  SET HI MEMORY
        062.327  332 364 063   5290          JC      PRS2            IF  ERROR
        062.332  041 342 042   5291          LXI     H,CCHIT
        062.335  076 003       5292          MVI     A,CTLC
        062.337  377 041       5293          DB      SYSCALL,.CTLC   SET CTL-C PROCESSING
        062.341  076 377       5294          MVI     A,377Q
        062.343  377 046       5295          DB      SYSCALL,.CLOSE  CLOSE OVERLAY CHANNEL
        000.001                 5296          IF      .PIP.
                                5297
                                5298  *       SEE IF COMMAND LINE PASSED ON STACK
                                5299
                                5300          LXI     H,0
                                5301          DAD     SP
                                5302          XCHG
                                5303          MVI     A,#STACK
                                5304          SUB     E
                                5305          MOV     C,A
                                5306          MVI     A,STACK/256
                                5307          SBB     D
                                5308          MOV     B,A             (BC) = BYTES ON STACK
                                5309          ORA     C
                                5310          STA     MODE            SET MODE <>0 IF LINE ON STACK
                                5311          JZ      START           NO LINE
                                5312
                                5313  *       HAVE LCOMMAND ON STACK. COPY INTO LINE BUFFER
                                5314  *       (BC) = COUNT
```

```
                                5315  *       (DE) = FWA
                                5316
                                5317          LXI     H,LINE
                                5318          CALL    $MOVE           COPY
                                5319          MVI     M,0             ENSURE END
                                5320          ELSE    ONECOPY
062.345  315 034 064           5321          CALL    $DOS            DISMOUNT OPERATING SYSTEM
062.350  332 364 063           5322          JC      PRS2            IF ERROR
062.353  315 136 031           5323          CALL    $TYPTX
062.356  012 011 011           5324          DB      NL,TAB,TAB,TAB,'   ','ONECOPY'
062.374  012 011 011           5325          DB      NL,TAB,TAB,TAB,'Version:  ',VERS/16+'0','.',VERS&0FH+'0'
063.015  012 011 011           5326          DB      NL,TAB,TAB,'      ','Issue:  $50.05.00 '
063.050  012 012 011           5327          DB      NL,NL,' ONECOPY is used to copy files for systems with only one'
063.142  012 146 154           5328          DB      NL,'floppy drive. Read the appropriate manual before using.'
063.232  212                   5329          DB      ENL
063.233  315 136 031           5330          CALL    $TYPTX
063.236  012 111 156           5331          DB      NL,'Insert the initial source disk. Hit RETURN when ready:',' '+200Q
063.326  315 077 056           5332          CALL    GDWP.                                          /79.11.GC/
063.331  315 120 057           5333          CALL    $RTL            GET CR
                                5334
                                5335  *       READ NEW DISK'S LABEL
                                5336
063.334  315 347 046           5337          CALL    GETLAB          GET LABEL
063.337  332 275 052           5338          JC      ERROR
063.342  315 326 046           5339          CALL    MND             MOUNT NEW DISK
063.345  332 275 052           5340          JC      ERROR           IF ERROR
063.350  072 000 027           5341          LDA     LABEL+LAB.SER
063.353  062 132 062           5342          STA     VOLSER          SET CURRENT VOLUME NUMBER
                                5343          ENDIF
063.356  303 207 042           5344          JMP     START           START PROGRAM
                                5345
063.361  076 050               5346  PRS1    MVI     A,EC.NCV        NOT CORRECT VERSION
063.363  067                   5347          STC
063.364  046 012               5348  PRS2    MVI     H,NL
063.366  377 057               5349          DB      SYSCALL,.ERROR
063.370  303 337 042           5350          JMP     EXIT
                                5351
000.000                         5352          IF      ONECOPY
063.373                         5353          XTEXT   DTB
```

```
                                5355X **      $DTB - DELETE TRAILING BLANKS.
                                5356X *
                                5357X *       $DTB DELETES THE TRAILING BLANKS FROM A CODED LINE.
                                5358X *
                                5359X *       ENTRY   (HL) = LINE FWA
                                5360X *       EXIT    (A) = LENGTH OF RESULT (ENCLUDING 00 TERMINATOR BYTE)
                                5361X *       USES    A,F
                                5362X
                                5363X
063.373  325                   5364X $DTB    PUSH    D               SAVE (DE)
063.374  124                   5365X         MOV     D,H
063.375  135                   5366X         MOV     E,L             (DE) = FWA
063.376  033                   5367X         DCX     D               (DE) = FWA-1
```

```
063.377  176           5368X $DTB1   MOV     A,M
064.000  043           5369X         INX     H
064.001  247           5370X         ANA     A               FIND END OF LINE
064.002  302 377 063    5371X         JNZ     $DTB1
064.005  053           5372X         DCX     H               (HL) = ADDRESS OF TERMINATING ZERO BYTE
                       5373X
                       5374X *       GOT END OF LINE. DELETE TRAILING BLANKS
                       5375X
064.006  053           5376X $DTB2   DCX     H               BACKUP ONE CHARACTER
064.007  315 216 030    5377X         CALL    $CDEHL
064.012  312 023 064    5378X         JE      $DTB3           GONE PAST FRONT OF LINE, MUST BE ALL BLANKS
064.015  176           5379X         MOV     A,M
064.016  376 040       5380X         CPI     ' '
064.020  312 006 064    5381X         JE      $DTB2           GOT BLANK
                       5382X
                       5383X *       HAVE TRIMED LINE, COMPUTE LENGTH
                       5384X
064.023  043           5385X $DTB3   INX     H
064.024  066 000       5386X         MVI     M,0             TERMINATE LINE
064.026  175           5387X         MOV     A,L
064.027  223           5388X         SUB     E               (A) = LENGTH +1 (FOR 00 BYTE)
064.030  353           5389X         XCHG
064.031  043           5390X         INX     H               (HL) = LINE FWA
064.032  321           5391X         POP     D               RESTORE (DE)
064.033  311           5392X         RET
064.034               5393          XTEXT   DOS


                       5395X **      $DOS - DISMOUNT OPERATING SYSTEM.
                       5396X *
                       5397X *       $DOS DISMOUNTS SY2:, SY1: (IF MOUNTED), AND SY0:.       /79.11.GC/
                       5398X *
                       5399X *       THE USER IS MESSAGED ABOUT THE DISKS, AND THE OPERATING
                       5400X *       SYSTEM IS NOTIFIED.
                       5401X *
                       5402X *
                       5403X *       ENTRY   NONE
                       5404X *
                       5405X *       EXIT    (PSW)   = 'C' CLEAR IF NO ERROR
                       5406X *                         'C' SET   IF    ERROR
                       5407X *                          (A)  = ERROR CODE
                       5408X *
                       5409X *       USES    ALL
                       5410X *
                       5411X
064.034  315 136 031    5412X $DOS    CALL    $TYPTX
064.037  012 007 104    5413X         DB      NL,BELL,'Dismounting All Disks!',NL,ENL
                       5414X
064.071  076 000       5415X         MVI     A,OVL0
064.073  377 010       5416X         DB      SYSCALL,.LOADO
064.075  330           5417X         RC
064.076  076 001       5418X         MVI     A,OVL1
064.100  377 010       5419X         DB      SYSCALL,.LOADO
064.102  330           5420X         RC
```

```
                                5421X
064.103  041 243 064           5422X          LXI     H,DOSC
064.106  315 221 064           5423X          CALL    DOS.
064.111  330                   5424X          RC
064.112  041 236 064           5425X          LXI     H,DOSB
064.115  315 221 064           5426X          CALL    DOS.
064.120  330                   5427X          RC                      FATAL ERROR
064.121  041 231 064           5428X          LXI     H,DOSA
064.124  315 221 064           5429X          CALL    DOS.
064.127  330                   5430X          RC
                                5431X
064.130  315 136 031           5432X          CALL    $TYPTX
064.133  012 122 145           5433X          DB      NL,'Remove the Disk(s), Hit RETURN when ready:',' '+200Q
064.207  315 275 060           5434X DOS1     CALL    $RCHAR          READ CHARACTER
064.212  376 012               5435X          CPI     NL
064.214  302 207 064           5436X          JNE     DOS1
064.217  247                   5437X          ANA     A               CLEAR CARRY
064.220  311                   5438X          RET
                                5439X
                                5440X *        DISMOUNT A DEVICE WITHOUT REGARD TO WHETHER MOUNTED OR NOT
                                5441X
064.221  377 201               5442X DOS.     DB      SYSCALL,.DMOUN
064.223  320                   5443X          RNC
064.224  376 042               5444X          CPI     EC.NVM          NO VOLUME MOUNTED ERROR NOT CONSIDERED FATAL
064.226  310                   5445X          RZ                      NOT FATAL, CARRY NOW CLEAR
064.227  067                   5446X          STC                     FLAG FATAL ERROR
064.230  311                   5447X          RET
                                5448X
064.231  123 131 060           5449X DOSA     DB      'SY0:',0
064.236  123 131 061           5450X DOSB     DB      'SY1:',0
064.243  123 131 062           5451X DOSC     DB      'SY2:',0
                                5452           ENDIF
                                5453
064.250                         5454 MEML      EQU     *               MEMORY LENGTH
```

```
                        5457  **      THE FOLLOWING BUFFERS AND AREAS OVERLAY THE PRS CODE.
                        5458  *
                        5459  *       *PRS* MAY NOT USE ANY CELLS BELOW *LINE*, AT THE
                        5460  *       RISK OF SMASHING ITSELF
                        5461
    062.310             5462          ORG     PRS
                        5463
    062.310             5464  DEFALT  DS      6               DEFAULT BLOCK
                        5465
    062.316             5466  MWNA    DS      FB.NAML         MWN WORK AREA
                        5467
    000.001             5468          IF      .PIP.
                        5469  DESTBUF DS      256             DESTINATION FILE BUFFER (ALSO USED BY *CCW*)
                        5470  DESTBFE EQU     *               END OF BUFFER
                        5471          ENDIF
                        5472
                        5473  **      * * NOTE * *
                        5474  *       DIRWORK  USES THE SYSTEM SCRATCH AREA, LABEL. DIRWORK WILL NOT
                        5475  *       BE PRESERVED DURING A SYSCALL !!
                        5476
    027.000             5477  LABEL   EQU     S.GRT2+256      USE EXTRA GRT TABLE AS BUFFER   /79.12.GC/
                        5478
                        5479  *DIRWORK        EQU     SECSCR          USE SECTOR SCRATCH AREA /79.11.GC/


                        5481  **      PIO.XXX - IMAGE OF SYSTEM AIO.XXX AREA
                        5482  *
                        5483  *       THESE CELLS MIRROR THE SYSTEM AIO.XXX AREA
                        5484
                        5485
    062.337             5486  PIO.DEV DS      2               DEVICE CODE
    062.341             5487  PIO.UNI DS      1               UNIT NUMBER (0-9)
                        5488
    062.342             5489  PIO.DIR DS      DIRELEN         DIRECTORY ENTRY
                        5490
    062.371             5491  $FOPWRK DS      FB.NAML         WORK AREA FOR $FOPE
                        5492
                        5493
    000.001             5494          IF      .PIP.
                        5495          ERRMI   *-MEML          FOLLOWING MUST NOT OVERLAY *PRS*
                        5496          ENDIF
    063.012             5497  LINE    DS      80              COMMAND BUFFER
                        5498
                        5499
    063.132             5500  NAMTAB  DS      0               NAME TABLE
                        5501
                        5502
    002.000             5503  BUFMINL EQU     512             MINIMUM SIZE FOR BUFFER (WHEN IN USE)
    063.132             5504  BUFF    EQU     *               BUFFER AREA STARTS AFTER NAMTAB
                        5505
    063.132             5506  RMEML   EQU     *               INITIAL RUNNING MEMORY LENGTH
                        5507
                        5508
                        5509
    063.132             5510          END
ASSEMBLY COMPLETE
 5510 STATEMENTS
    0 ERRORS DETECTED
 8730 BYTES FREE
```

```
$TYPC.  057237  4268L  4318
$TYPCC  057013  2393   2527  2756  3916E 3925
$TYPCH  057233  4258L
$TYPL.  057260  4295   4307E 4321
$TYPLN. 057242  4289L
$TYPTX  031136   969   1147  1547  2057  2823  2873  2877  2888  2904  4353E 5026  5036
                5193  5209  5323  5330  5412  5432
$TYPTX. 031144  4355E
$UDD    031157  4434E  4660  4690
$UDDN   060177  1546   2494  2500  2510  2702  4715E
$WCHAR  060303  2876   4809L 5204
$WDR    031222  3950E
$WER    031241  2148   3938E
$ZERO   031212  1652   3963E
,       000062  5167S  5168
.ABUSS  040024   808E
.ALARM  002136   781E
.ALEDS  040013   806E  2044  2051  2102
.CHFLG  000060   451L
.CLEAR  000055   448L
.CLEARA 000056   449L   902
.CLOSE  000046   441L  1785  1924  2479  3653  5295
.CLRCO  000007   425L   971  5025
.CONSL  000006   424L  3993  4009
.CRC    002347   789E
.CRCSUM 040027   809E
.CTC    002172   783E
.CTLC   000041   436L  5293
.CTLFLG 040011   805E
.DECODE 000053   446L  2341  3577
.DELET  000050   443L  1881
.DISMT  000061   452L
.DLEDS  040021   807E
.DLY    000053   778E  5040
.DMNMS  000203   463L  2030
.DMOUN  000201   461L  5442
.DOD    003122   792E
.DODA   003356   794E
.DSPMOD 040007   803E
.DSPROT 040006   802E
.DUMP   001374   780E
.ERROR  000057   450L  2894  5213  5349
.EXIT   000000   418L   962  2881
.HORN   002140   782E  2060
.IDENT  000000   777E
.IOWRK  040002   800E
.LINK   000040   435L
.LOAD   001267   779E
.LOADD  000062   453L
.LOADO  000010   426L  5416  5419
.MFLAG  040010   804E  2043
.MONMS  000202   462L  2124
.MOUNT  000200   460L
.NAME   000054   447L
.OPENC  000045   440L  1888
.OPENR  000042   437L  1671  2368  3589
.OPENU  000044   439L  1897
.OPENW  000043   438L  1868
```

```
.PCHL   002264      785E
.PIP.   000001       1E      4     948     986    1034    1148    1159    2166    2316    2375    2388    2515
                    2595    2747    2825    2846    2911    3022    3186    3461    5004    5230    5265    5296    5468
                    5494
.POSIT  000047      442L    1729    1906
.PRINT  000003      421L    2056    2907
.RCK    003260      793E
.READ   000004      422L    1743    2402    3599
.REGI   040005      801E
.REGPTR 040035      812E
.RENAM  000051      444L
.RESET  000204      464L
.RNB    002331      788E
.RNP    002325      787E
.SCIN   000001      419L    2064    2072    4805    5029
.SCOUT  000002      420L    3922    4246    4268    4809
.SETTP  000052      445L    3458    3805    5289
.SRS    002265      786E
.START  040000      799E
.SYSRES 000012      428L
.TICCNT 040033      811E
.TPERR  002205      784E
.TPERRX 040031      810E
.UIVEC  040037      813E
.VERS   000011      427L    5284
.WNB    003024      791E
.WNP    003017      790E
.WRITE  000005      423L    1921
ABS.COD 000010      871L     874
ABS.ENT 000006      869L
ABS.ID  000000      865L
ABS.LDA 000002      867L
ABS.LEN 000004      868L
AC.DLY  000156       66E    5039
ACL     043173      920     1146L
AEN     053027     2614     2942L   3557    3640
AENA    053101     2942     2959    2964L
AIO.CGN 041047      634L
AIO.CHA 041116      649L
AIO.CNT 041111      645L
AIO.CSI 041050      635L
AIO.DDA 041041      630E
AIO.DES 041055      639L
AIO.DEV 041057      640L
AIO.DIR 041062      643L
AIO.DTA 041053      638L
AIO.EOF 041113      647L
AIO.EOM 041112      646L
AIO.FLG 041043      631L
AIO.GRT 041044      632L
AIO.LGN 041051      636L
AIO.LSI 041052      637L
AIO.SPG 041046      633L
AIO.TFF 041114      648L
AIO.UNI 041061      641L
AIO.VEC 041040      629L
BELL    000007      473E    2058    2874    2889    2905    5194    5413
BKSP    000010      475E
```

```
CS.FLG  000200  702E
CSL.CHR 000001  679E
CSL.ECH 000200  677E
CSL.WRP 000002  678E
CTLA    000001  484E
CTLB    000002  485E
CTLC    000003  486E  5292
CTLD    000004  487E  4079
CTLO    000017  488E
CTLP    000020  489E
CTLQ    000021  490E
CTLS    000023  491E
CTLZ    000032  492E
CTP.2SB 000010  687E
CTP.BKM 000002  688E
CTP.BKS 000200  684E
CTP.MLI 000040  685E
CTP.MLO 000020  686E
CTP.TAB 000001  689E
CTS     053236  2813  3106L
CWM     053253  2430  3130L  3138  3629
CWM1    053262  3132  3135L
D.CON   040110  390L
D.DLYHS 040244  510L
D.DLYMO 040243  509L  2036
D.DRVTB 040251  515L  1484
D.DVCTL 040242  507L
D.E.CHK 040267  526L
D.E.HCK 040270  527L
D.E.HSY 040266  525L
D.E.MDS 040265  524L
D.E.TRK 040272  529L
D.E.VOL 040271  528L
D.ERR   040265  523L
D.ERRL  040273  530L
D.HECNT 040261  517L
D.OECNT 040264  519L
D.OPR   040273  534L
D.OPW   040275  535L
D.RAM   040240  393L  502  537
D.RAML  000037  537E
D.SECNT 040262  518L
D.TRKPT 040245  512L
D.TS    040241  505L
D.TT    040240  504L
D.VEC   040130  392L
D.VOLPT 040247  513L
DAD1    060130  4658  4671  4673  4696L
DADB    060133  4675  4700L
DC.ABT  000007  724L
DC.CLO  000006  723L
DC.LOD  000011  726L
DC.MAX  000012  727L
DC.MOU  000010  725L
DC.OPR  000003  720L
DC.OPU  000005  722L
DC.OPW  000004  721L
DC.REA  000000  717L
```

```
DC.RER  000002   719L   2149
DC.WRI  000001   718L
DDF     053271  1486   3160L
DDF.BOL 000011   824E
DDF.BOO 000000   823L
DDF.LAB 000011   825L   2145
DDF.RGT 000012   826L
DDF.USR 000014   827L
DDF1    053276  3163L   3168
DDF1.0  053311  3169L   3176
DDF2    053314  3166    3173L
DDF3    053364  3195    3199L
DDFA    053371  3169    3204L
DDFB    054002  3192    3205L
DEFALT  062310  2987    3173   3821   3866   3870   5464L
DESTERR 052056  1869    1889   1898   1922   1925   2857L
DESTFB  062247   904    1495   1856   1866   1878   1886   1895   2857   3185   3193   5240L
DEV.DDA 000004   238L
DEV.DVG 000016   250L
DEV.DVL 000014   249L
DEV.FLG 000006   239L
DEV.JMP 000003   237L
DEV.MNU 000011   246L
DEV.MUM 000010   245L
DEV.NAM 000000   229L
DEV.RES 000002   233L
DEV.SPG 000007   244L   2350
DEV.UNT 000012   247L   2353
DEVELEN 000017   252E
DF.CLR  000376   177E   3621
DF.EMP  000377   176E   2413   3618
DIF.CNT 000020   202E   2800   5064
DIF.LOC 000100   200E   2798
DIF.SYS 000200   199E   2797   3066
DIF.WP  000040   201E   2799
DIR.ALD 000025   192L
DIR.CLU 000015   185L
DIR.CRD 000023   191L   2710   2721
DIR.EXT 000010   180L   2676   3253   3325   3507   3869
DIR.FGN 000020   188L   1704   2676   2683
DIR.FLG 000016   186L   1693   2721   3063
DIR.LGN 000021   189L
DIR.LSI 000022   190L   2683   2710
DIR.NAM 000000   179L   2426   3021   3297   3502   3507   3574   3638
DIR.PRO 000013   181L
DIR.VER 000014   182L
DIRELEN 000027   194E    294    306    643   5489
DIRIDL  000015   183E
DIRNAM  062224  2329    2331   2339   2366   3572   3576   3587   5223L
DIS.ENL 001373   310L   2456   3608
DIS.ENT 000000   305E
DIS.LNK 001376   312L
DIS.SEC 001374   311L
DM.MR   000000   752E
DM.MW   000001   753E
DM.RR   000002   754E
DM.RW   000003   755E
DNT     054260  3271   3287   3318   3367L
```

DNT1	054267	3371L	3374		
DNT2	054300	3379L	3402		
DNT3	054342	3382	3389	3397L	
DNT4	054365	3387	3391	3393	3420L
DNT5	054354	3385	3411L	3415	
DNTA	054372	3367	3375	3421	3424L
DOS.	064221	5423	5426	5429	5442L
DOS1	064207	5434L	5436		
DOSA	064231	5428	5449L		
DOSB	064236	5425	5450L		
DOSC	064243	5422	5451L		
DR.IM	000001	234E			
DR.PR	000002	235E			
DT.CR	000002	241E			
DT.CW	000004	242E			
DT.DD	000001	240E	2344	3580	
DV.EL	000000	230E			
DV.NU	000001	231E			
EBM	055003	1497	3436L		
EBM1	055043	3447	3455L		
EC.CNA	000004	336L			
EC.DDA	000027	355L			
EC.DIF	000017	347L			
EC.DIW	000035	361L			
EC.DNI	000045	369L			
EC.DNR	000046	370L			
EC.DNS	000005	337L	2345	3200	3581
EC.DSC	000047	371L			
EC.EOF	000001	333L	1749		
EC.EOM	000002	334L			
EC.FAO	000031	357L			
EC.FAP	000026	354L			
EC.FL	000030	356L			
EC.FNF	000014	344L	1883		
EC.FNO	000011	341L			
EC.FNR	000034	360L			
EC.FOD	000043	367L			
EC.FUC	000013	343L			
EC.ICN	000016	346L			
EC.IDN	000006	338L			
EC.IFC	000020	348L			
EC.IFN	000007	339L	3346	3900	
EC.ILC	000003	335L			
EC.ILO	000040	364L			
EC.ILR	000012	342L			
EC.ILV	000037	363L			
EC.IOI	000052	374L			
EC.IS	000032	358L	4926		
EC.NCV	000050	372L	5346		
EC.NEM	000021	349L	3697		
EC.NOS	000051	373L			
EC.NPM	000044	368L			
EC.NRD	000010	340L			
EC.NVM	000042	366L	5444		
EC.OTL	000053	375L			
EC.RF	000022	350L			
EC.UNA	000036	362L			
EC.UND	000015	345L			

```
EC.UUN  000033  359L
EC.VPM  000041  365L
EC.WF   000023  351L
EC.WP   000025  353L
EC.WPV  000024  352L
ENL     000212  482E    1549    2833    2920    2921    2922    2923    2924    2925    2926    2927    5037
                5329    5413
ENTRY   062310  878     5283E
ERROR   052275  931     1068    1103    1487    1492    1510    1884    2031    2078    2125    2327    2342
                2346    2370    2815    2822    2887L   3698    3807    5338    5340
ERROR1  052326  2892    2899L
ERROR2  052331  2900L   2902
ERRORA  052367  2899    2910L
ESC     000033  480E
EWS     055156  2996    3557L
EWS1    055265  3595L   3617
EWS3    055317  3615L   3648
EWS4    055376  3630    3645L
EWS6    055377  3620    3625    3646L
EWS7    056005  3601    3623    3652L
EWSA    056012  3656L
EWSB    056020  3575    3579    3658L
EWSC    056056  3574    3627    3660L
EXIT    042337  888     921     961L    5350
FB.CHA  000000  318L
FB.FLG  000001  319L    904
FB.FWA  000002  320L
FB.LIM  000006  322L
FB.LWA  000010  323L
FB.NAM  000012  324L    325     1495    1856    1866    1878    1886    1895    2851    3185    3193    5195
FB.NAML 000021  325E    1505    1554    1620    1621    2434    2947    2957    2964    3563    3568    3782
                3787    5247    5466    5491
FB.PTR  000004  321L
FBENL   000033  326E
FDN     062017  5060L   5061    5063    5068    5070    5072    5074    5076    5078    5079
FDN.1   062021  1995    5090    5093L   5165
FDN.2   062032  5094    5102L
FDN.3   062043  5103    5111L
FDN.4   062054  5112    5120L
FDN.5   062065  5121    5129L
FDN.6   062076  5130    5138L
FDN.7   062107  5139    5147L
FDN.8   062120  5148    5156L
FDN.ADR 000007  1734    1756    1769    1914    5074E
FDN.AIM 000010  1769    1771    1824    1943    5076E
FDN.AMR 000003  1725    1771    5070E
FDN.AMW 000005  1902    1929    5072E
FDN.LNK 000000  1929    1948    5061E
FDN.SIZ 000002  1714    1875    5068E
FDN.STA 000001  1603    1659    1714    1725    1734    1756    1844    1875    1948    5063E
FDNCNT  000010  2000    5165E
FDNELEN 000011  1650    2001    5078E   5165
FDNFRE  062017  1611    1645    1649    1958    1961    1997    5090L
FDNFWA  062017  5088E   5168
FDNHEAD 062020  1524    1597    1622    1818    1957    1999    5091L
FF      000014  483E
FT.ABS  000000  855E    875
FT.BAC  000003  858E
```

```
LAB.SPG 000007   837L
LAB.SYS 000001   840E
LAB.VER 000011   844L
LAB.VLT 000010   843L
LABEL   027000  2085   2146   5341   5477E
LF      000012   470E
LINE    063012   922   1155   2816   3107   3160   3714   3721   5497L
LIST    046371   941   2301L
LIST1   047002  2302   2307L
LIST1.5 047055  2334   2339L
LIST10  050072  2483   2512L
LIST2   047174  2386   2391L
LIST3   047201  2398L  2411   2464
LIST4   047220  2409L  2463
LIST5   047250  2423L  2447
LIST6   047267  2431L
LIST7   047321  2414   2420   2451L  2474
LIST8   047350  2433   2468L
LIST9   047367  2404   2416   2478L
LSN     056135  2572   2980   3106   3714L
LSN1    056140  3715L  3720
LSTA    050107  2307   2308   2384   2481   2530L  2658   2671
LSTB    050110  2308   2472   2489   2532L
LSTC    050111  2310   2495   2533L  2697   2699
LSTD    050113  2340   2343   2347   2355   2534L
LSTE    050143  2361   2501   2535L  2687   3083
LSTF    050145  2351   2469   2504   2536L
LSTG    050146  2383   2537L  2540
LSTG1   050204  2312   2538L
LSTGL   000051  2387   2540E
LSTH    050217  2512   2542L  2546
LSTH1   050223  2493   2543L
LSTH2   050244  2498   2544L
LSTH3   050261  2508   2545L
LSTHL   000054  2511   2546E
M.FOX   000303   772E
M.PAM8  000021   771E
MAD     046130  1434   1522   1532   2021E
MAD0    046144  2035L  2092
MAD1    046157  2039   2041L
MAD2    046173  2047L  2050
MAD3    046227  2064L  2068
MAD4    046242  2065   2072L  2073
MAD4.5  046300  2087   2094L
MAD5    046314  2104L  2107
MEML    064250   877   5454E
MND     046326  2108   2123L  5339
MNDA    046342  2029   2123   2129L
MODE    062220   886    918   5218L
MOUNT   043217   947   1431E
MOUNTA  043230  1432   1437L
MWN     056155  1857   3742L
MWN1    056210  3756L  3764
MWN2    056216  3758   3760L
MWNA    062316  3747   3750   5466L
NAMERR  052044  1672   1750   2850L
NAMTAB  063132  1606   1653   1855   2421   2442   2957   3563   3787   3788   5500L
NAMTLEN 062302   910   1504   1632   2441   2569   2946   2949   3562   3567   3570   3781   3784
```

5250L

```
PFIB1   051357  1092  2796L
PFIC    051364  2645  2691  2801L
PIO.DEV 062337  2592  3249  3284  3497  3572  3862  3865  5486L
PIO.DIR 062342  2426  3021  3253  3297  3325  3502  3574  3638  3747  3751  3869  5489L
PIO.UNI 062341  3862  5487L
PIP     042200   880E
PIP1    042220   890   898L
PIPA    042354   923   985L
PIPB    042325   937L  938   940   942   944   946
PRS     062310  5284L 5462
PRS1    063361  5285  5287  5346L
PRS2    063364  5290  5322  5348L
QUOTE   000047   478E
REN     056231  1962  3781L
RESTART 042200   884E  894   972   2895  2908  2995  5214
RMEML   063132  5288  5506E
ROMBOOT 030000   385E
RPH     044203  1523  1587E 1786
RPH1    044232  1601  1611L
RPH2    044251  1624L 1630
RPH2.5  044325  1607  1667L
RPH3    045030  1677  1696  1722L
RPH4    045122  1748  1767L
RUBOUT  000177   474E
S.BAUD  040344   553L
S.BOOTF 041034   608L
S.CAADR 040333   708L
S.CACC  041006   592L
S.CCTAB 040335   709L
S.CDB   040343   550L  5032
S.CFWA  040352   560L  1688
S.CODE  041007   593L
S.CONFL 040332   706L
S.CONTY 040327   693L
S.CONWI 040331   699L
S.CSLMD 040326   682L   692   695   698   705  1154
S.CUSOR 040330   696L
S.DATC  040310   664L
S.DATE  040277   663L  2312
S.DCS   041033   606L
S.DDDTA 040366   571L
S.DDGRP 040364   568L
S.DDLDA 040360   566L
S.DDLEN 040362   567L
S.DDOPC 040370   572L
S.DFWA  040354   561L
S.DIREA 041016   600L
S.DLINK 040346   558L
S.FASER 041013   599L
S.FCI   041021   601L
S.GRT0  024000   381E
S.GRT1  025000   382E
S.GRT2  026000   383E  5477
S.GUP   041027   603L  2356
S.HIMEM 040316   666L
S.INT   040343   395L   546
S.JUMPS 041010   597L
S.MOUNT 041032   605L
```

```
S.OFWA  040350   559L   3438
S.OMAX  040324   672L   3448
S.OSN   041004   588L
S.OVLE  041000   585L
S.OVLFL 040371   581L
S.OVLS  040376   584L
S.OVSTK 041035   613L
S.RFWA  040356   562L
S.SCI   041024   602L
S.SCR   041120   651L   3678
S.SDD   041010   598L
S.SOVR  041146   397L    399
S.SSN   041002   587L
S.SYSM  040320   668L   3436
S.TIME  040312   665L
S.UCSF  040372   582L
S.UCSL  040374   583L
S.USRM  040322   670L   3460
S.VAL   040277   394L    661
SBE     056256  3701   3800L
SC.ACE  000350    65E   5042
SC.UART 000372   134E   5043
SDD     056277   898   1489  3819L
SDDA    056316  3821   3824L
SFS     056324  2616   2999  3842L
SFS1    056336  3845   3847L
SND     056341  2990   3861L
ST.CNT  000020  1711   1864  5064E
ST.OPR  000002  1605   1675  1686  1759  1834  1950  5065E
ST.OPW  000001  1846   1848  5066E
STACK   042200   401E    889
STACKL  001032   399E
START   042207   889L   5344
SUPRES  062222   927   1083  2513  5220L
SW.BRE  043115  1010   1097L
SW.BRE1 043132  1099   1104L
SW.JGL  043100  1030   1089L
SW.LIS  043140  1006   1110L
SW.LIS1 043153  1112   1118L
SW.MOU  043166  1018   1129L
SW.SUP  043072  1026   1082L
SW.SYS  043065  1022   1075L  1093
SW.VER  043161  1014   1124L
SWIT1   043047  1063L  1125  1130
SYDD    040130   391E   2150
SYSCALL 000377   411E    902   962   971  1671  1729  1743  1785  1868  1881  1888  1897
                1906   1921  1924  2030  2056  2064  2072  2124  2341  2368  2402  2479  2881
                2894   2907  3458  3577  3589  3599  3653  3805  3922  3993  4009  4246  4268
                4805   4809  5025  5029  5213  5284  5289  5293  5295  5349  5416  5419  5442
SYSTEM  062223   929   1076  3069  5221L
TAB     000011   479E   2537  2537  2537  2537  2537  2537  2650  2656  2703  2727  2831
                4157   5324  5324  5324  5325  5325  5325  5326  5326
TBL1    057203  4184L  4190
TBL2    057221  4182   4194L
TBL3    057223  4187   4198L
TPL1    057274  4319L
UC.2SB  000004    91E
UC.5BW  000000    87E
```

```
UC.6BW  000001      88E
UC.7BW  000002      89E
UC.8BW  000003      90E
UC.BI   000020     110E
UC.CTS  000020     119E
UC.DCS  000001     115E
UC.DDR  000002     116E
UC.DLA  000200      96E
UC.DR   000001     106E
UC.DRL  000010     118E
UC.DSR  000040     120E
UC.DTR  000001      99E
UC.EDA  000001      77E
UC.EPS  000020      93E
UC.FE   000010     109E
UC.IID  000006      84E
UC.IIP  000001      83E
UC.LOO  000020     103E
UC.MSI  000010      80E
UC.OR   000002     107E
UC.OU1  000004     101E
UC.OU2  000010     102E
UC.PE   000004     108E
UC.PEN  000010      92E
UC.RI   000100     121E
UC.RLS  000200     122E
UC.RSI  000004      79E
UC.RTS  000002     100E
UC.SB   000100      95E
UC.SKP  000040      94E
UC.TER  000004     117E
UC.THE  000040     111E
UC.TRE  000002      78E
UC.TSE  000100     112E
UCI.ER  000020     156E
UCI.IE  000002     158E
UCI.IR  000100     154E
UCI.RE  000004     157E
UCI.RO  000040     155E
UCI.TE  000001     159E
UDDN1   060203    4719L     4735
UDDN1.5 060235    4739L     4746
UDDN2   060237    4732      4744L
UDDN3   060240    4745L     4749
UDR     000000     131E
UMI.16X 000002     149E
UMI.1B  000100     139E
UMI.1X  000001     148E
UMI.2B  000300     141E
UMI.64X 000003     150E
UMI.HB  000200     140E
UMI.L5  000000     144E
UMI.L6  000004     145E
UMI.L7  000010     146E
UMI.L8  000014     147E
UMI.PA  000020     143E
UMI.PE  000040     142E
UNT.DIS 000005     261L
```

```
UNT.FLG 000000    258L
UNT.GRT 000001    259L    2359
UNT.GTS 000003    260L
UNT.SIZ 000007    263E
UO.CLK  000001    764E    2042
UO.DDU  000002    763E    2042
UO.HLT  000200    761E    2042
UO.NFR  000100    762E
UR.DLL  000000     72E
UR.DLM  000001     74E
UR.IER  000001     76E    5042
UR.IIR  000002     82E
UR.LCR  000003     86E
UR.LSR  000005    105E
UR.MCR  000004     98E
UR.MSR  000006    114E
UR.RBR  000000     68E
UR.THR  000000     70E
USERFWA 042200    402E     874    876    877
USR     000001    132E    5043
USR.FE  000040    163E
USR.OE  000020    164E
USR.PE  000010    165E
USR.RXR 000002    167E
USR.TXE 000004    166E
USR.TXR 000001    168E
VERS    000026    409E    2832   2832   5286   5325   5325
VERSN   051365     945    2811E
VOLFLAG 062131    1483    1517   1527   2095   5170L
VOLSER  062132    1485    2084   5171L  5342
WPH     045156    1533    1814E  1963
WPH0    045211    1827    1842L
WPH1    045263    1865    1874L
WPH1.5  045307    1882    1885L
WPH2    045325    1847    1895L
WPH3    045357    1870    1890   1911L
WPH4    046032    1837    1943L
XCHGBC  060306    4553    4557   4565   4567   4826L

13804 BYTES FREE
```