```
                    2
000.001             3  .SYS.   EQU     1               ASSEMBLE SYSTEM USER'S CODE
                    4          LON     I               SHOW IF-SKIPPED LINES
                    5
                    6  ***     PATCH - PATCH SYSTEM AND USER FILES.
                    7  *
                    8  *       PATCH IS USED TO PATCH BINARY (ABS AND PIC) FILES ON THE
                    9  *       SYSTEM. PATCH RUNS IN TWO MODES, *USER* AND *SYSTEM*.
                   10  *
                   11  *       IF THE FILE TO BE PATCHED HAS A PATCH HISTORY TABLE ON THE
                   12  *       END OF IT, PATCH RUNS IN SYSTEM MODE. OTHERWISE, PATCH RUNS
                   13  *       IN USER MODE.
                   14  *
                   15  *       SYSTEM MODE:
                   16  *       ------------
                   17  *
                   18  *       IN SYSTEM MODE, PATCH WILL VIOLATE SOFTWARE WRITE-PROTECTION ON
                   19  *       THE FILES. FOR THIS REASON, PATCH IS VERY PARTICULAR ABOUT ENTERING
                   20  *       PATCHES.
                   21  *
                   22  *       FIRST, A PATCH SERIES CODE MUST BE GIVEN. THIS CODE REPRESENTS A
                   23  *       PATCH SERIES BYTE, AND A CRC-16 OF THAT BYTE.
                   24  *       NEXT, A PATCH PREREQUESITE CODE IS REQUIRED. THIS CODE REPRESENTS
                   25  *       A 6 BYTE FIELD, WITH A  BIT SET FOR EVERY PREREQUISITE PATCH NEEDED.
                   26  *       THE PSC IS FOLLOWED BY A CRC-16 OF THE PREREQUESITE BYTES PROCEDED
                   27  *       BY THE PATCH SERIES BYTE.
                   28  *
                   29  *       AFTER THE PATCHES ARE ENTERED, A PATCH CHECK CODE MUST BE ENTERED.
                   30  *       THIS PRODUCES A 2 BYTE CRC-16 FOR ALL OF THE ABOVE ENTERED VALUES,
                   31  *       AND ALSO ENCLUDES A CTC-16 FOR THE PATCH CHECK CODE ITSELF.
                   32  *
                   33  *       IF ALL OF THESE CODES ARE ALL OK, THEN THE PATCH IS MADE.
                   34  *
                   35  *       USER MODE:
                   36  *       ----------
                   37  *
                   38  *       IN USER MODE, THE USER SPECIFIES THE PROGRAM NAME, AND THE PATCHES.
                   39  *       WHEN HE CTL-D'S, THE PATCHES ARE ENTERED. A PROGRAM MUST BE WRITE
                   40  *       ACCESSABLE FOR PATCH TO WORK IN USER MODE.
                   41
                   42
                   43  ****    ASSEMBLY CONSTANTS
                   44
000.000            45  CN.FIL  EQU     0               PATCH FILE CHANNEL NUMBER
                   46
                   47  ****
                   48
000.000            49          XTEXT   ECDEF
```

```
                        51X **      ERROR CODE DEFINITIONS.
                        52X
000.000                 53X         ORG     0
000.000                 54X         DS      1       NO ERROR #0
000.001                 55X EC.EOF  DS      1       END OF FILE
000.002                 56X EC.EOM  DS      1       END OF MEDIA
000.003                 57X EC.ILC  DS      1       ILLEGAL SYSCALL CODE
000.004                 58X EC.CNA  DS      1       CHANNEL NOT AVAILABLE
000.005                 59X EC.DNS  DS      1       DEVICE NOT SUITABLE
000.006                 60X EC.IDN  DS      1       ILLEGAL DEVICE NAME
000.007                 61X EC.IFN  DS      1       ILLEGAL FILE NAME
000.010                 62X EC.NRD  DS      1       NO ROOM FOR DEVICE DRIVER
000.011                 63X EC.FNO  DS      1       CHANNEL NOT OPEN
000.012                 64X EC.ILR  DS      1       ILLEGAL REQUEST
000.013                 65X EC.FUC  DS      1       FILE USAGE CONFLICT
000.014                 66X EC.FNF  DS      1       FILE NAME NOT FOUND
000.015                 67X EC.UND  DS      1       UNKNOWN DEVICE
000.016                 68X EC.ICN  DS      1       ILLEGAL CHANNEL NUMBER
000.017                 69X EC.DIF  DS      1       DIRECTORY FULL
000.020                 70X EC.IFC  DS      1       ILLEGAL FILE CONTENTS
000.021                 71X EC.NEM  DS      1       NOT ENOUGH MEMORY
000.022                 72X EC.RF   DS      1       READ FAILURE
000.023                 73X EC.WF   DS      1       WRITE FAILURE
000.024                 74X EC.WPV  DS      1       WRITE PROTECTION VIOLATION
000.025                 75X EC.WP   DS      1       DISK WRITE PROTECTED
000.026                 76X EC.FAP  DS      1       FILE ALREADY PRESENT
000.027                 77X EC.DDA  DS      1       DEVICE DRIVER ABORT
000.030                 78X EC.FL   DS      1       FILE LOCKED
000.031                 79X EC.FAO  DS      1       FILE ALREADY OPEN
000.032                 80X EC.IS   DS      1       ILLEGAL SWITCH
000.033                 81X EC.UUN  DS      1       UNKNOWN UNIT NUMBER
000.034                 82X EC.FNR  DS      1       FILE NAME REQUIRED
000.035                 83X EC.DIW  DS      1       DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036                 84X EC.UNA  DS      1       UNIT NOT AVAILABLE
000.037                 85X EC.ILV  DS      1       ILLEGAL VALUE
000.040                 86X EC.ILO  DS      1       ILLEGAL OPTION
000.041                 87X EC.VPM  DS      1       VOLUME PRESENTLY MOUNTED ON DEVICE
000.042                 88X EC.NVM  DS      1       NO VOLUME PRESENTLY MOUNTED
000.043                 89X EC.FOD  DS      1       FILE OPEN ON DEVICE
000.044                 90X EC.NPM  DS      1       NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045                 91X EC.DNI  DS      1       DISK NOT INITIALIZED
000.046                 92X EC.DNR  DS      1       DISK IS NOT READABLE
000.047                 93X EC.DSC  DS      1       DISK STRUCTURE IS CORRUPT
000.050                 94X EC.NCV  DS      1       NOT CORRECT VERSION OF HDOS
000.051                 95X EC.NOS  DS      1       NO OPERATING SYSTEM MOUNTED
000.052                 96X EC.IOI  DS      1       ILLEGAL OVERLAY INDEX
000.053                 97X EC.OTL  DS      1       OVERLAY TO LARGE
000.054                 98          XTEXT   DIRDEF
```

```
                        100X **      DIRECTORY ENTRY FORMAT.
                        101X
000.000                 102X         ORG     0
                        103X
                        104X
000.377                 105X DF.EMP  EQU     377Q            FLAGS ENTRY EMPTY
000.376                 106X DF.CLR  EQU     376Q            FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
                        107X
000.000                 108X DIR.NAM DS      8               NAME
000.010                 109X DIR.EXT DS      3               EXTENSION
000.013                 110X DIR.PRO DS      1               PROJECT
000.014                 111X DIR.VER DS      1               VERSION
000.015                 112X DIRIDL  EQU     *               FILE IDENTIFICATION LENGTH
                        113X
000.015                 114X DIR.CLU DS      1               CLUSTER FACTOR
000.016                 115X DIR.FLG DS      1               FLAGS
000.017                 116X         DS      1               RESERVED
000.020                 117X DIR.FGN DS      1               FIRST GROUP NUMBER
000.021                 118X DIR.LGN DS      1               LAST GROUP NUMBER
000.022                 119X DIR.LSI DS      1               LAST SECTOR INDEX (IN LAST GROUP)
000.023                 120X DIR.CRD DS      2               CREATION DATE
000.025                 121X DIR.ALD DS      2               LAST ALTERATION DATE
                        122X
000.027                 123X DIRELEN EQU     *               DIRECTORY ENTRY LENGTH
000.027                 124          XTEXT   IOCDEF


                        126X **      I/O CHANNEL DEFINITIONS.
                        127X
000.000                 128X         ORG     0
                        129X
000.000                 130X IOC.LNK DS      2               ADDRESS OF NEXT CHANNEL, =0 IF LAST
000.002                 131X IOC.DDA DS      2               THREAD JUMP TO DEVICE DRIVER (VIA DEV TABLE)
                        132X
000.004                 133X IOC.FLG DS      1               FILE TYPE FLAGS
000.001                 134X FT.DD   EQU     00000001B       =1 IF DIRECTORY DEVICE
000.002                 135X FT.OR   EQU     00000010B       =1 IF OPEN FOR READ
000.004                 136X FT.OW   EQU     00000100B       =1 IF OPEN FOR WRITE
000.010                 137X FT.OU   EQU     00001000B       =1 IF OPEN FOR UPDATE
000.020                 138X FT.OC   EQU     00010000B       =1 IF OPEN FOR CHARACTER MODE   /80.02.GC/
000.003                 139X IOC.SQL EQU     *-IOC.DDA       LENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
                        140X
000.005                 141X IOC.GRT DS      2               ADDRESS OF GROUP RESERVATION TABLE
000.007                 142X IOC.SPG DS      1               SECTORS PER GROUP, THIS DEVICE
000.010                 143X IOC.CGN DS      1               CURRENT GROUP NUMBER
000.011                 144X IOC.CSI DS      1               CURRENT SECTOR INDEX (IN CURRENT GROUP)
000.012                 145X IOC.LGN DS      1               LAST GROUP NUMBER
000.013                 146X IOC.LSI DS      1               LAST SECTOR INDEX (IN LAST GROUP)
000.010                 147X IOC.DRL EQU     *-IOC.FLG       LENGTH OF INFO NORMALLY COPIED BACK TO
                        148X *                               THE CHANNEL TABLE
000.014                 149X IOC.DTA DS      2               DEVICE TABLE ADDRESS FOR THIS DEVICE
000.016                 150X IOC.DES DS      2               SECTOR NUMBER OF DIRECTORY ENTRY
000.020                 151X IOC.DEV DS      2               DEVICE CODE
000.022                 152X IOC.UNI DS      1               UNIT NUMBER (0-9)
000.021                 153X IOC.DIL EQU     *-IOC.DDA       LENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)
```

```
                    154X
000.023             155X IOC.DIR DS      DIRELEN         DIRECTORY ENTRY
                    156X
000.052             157X IOCELEN EQU     *               IOC ENTRY LENGTH
                    158X
000.001             159X IOCCTD  EQU     1               INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)
000.052             160          XTEXT   DEVDEF


                    162X **      DEVICE TABLE ENTRYS.
                    163X
000.000             164X         ORG     0
                    165X
000.000             166X DEV.NAM DS      2               DEVICE NAME
000.000             167X DV.EL   EQU     00000000B       END OF DEVICE LIST FLAG
000.001             168X DV.NU   EQU     00000001B       DEVICE ENTRY NOT IN USE
                    169X
000.002             170X DEV.RES DS      1               DRIVER RESIDENSE CODE
000.001             171X DR.IM   EQU     00000001B       DRIVER IN MEMORY
000.002             172X DR.PR   EQU     00000010B       DRIVER PERMINANTLY RESIDENT
                    173X
000.003             174X DEV.JMP DS      1               JMP TO PROCESSOR
000.004             175X DEV.DDA DS      2               DRIVER ADDRESS
000.006             176X DEV.FLG DS      1               FLAG BYTE
000.001             177X DT.DD   EQU     00000001B       DIRECTORY DEVICE
000.002             178X DT.CR   EQU     00000010B       CAPABLE OF READ OPERATION
000.004             179X DT.CW   EQU     00000100B       CAPABLE OF WRITE OPERATION
000.010             180X DT.RN   EQU     00001000B       Capable of random access        /80.02.sc/
000.020             181X DT.CH   EQU     00010000B       Capable of Character mode       /80.02.sc/
                    182X
000.007             183X DEV.MUM DS      1               MOUNTED UNIT MASK
000.010             184X DEV.MNU DS      1               MAXIMUM NUMBER OF UNITS
000.011             185X DEV.UNT DS      2               ADDRESS OF UNIT SPECIFIC DATA TABLE
                    186X
000.013             187X DEV.DVL DS      2               DRIVER BYTE LENGTH
000.015             188X DEV.DVG DS      1               DRIVER ROUTINE GROUP ADDRESS
                    189X
000.016             190X DEVELEN EQU     *               DEVICE TABLE ENTRY LENGTH


                    192X **      UNIT SPECIFIC DEVICE DATA TABLE ENTRIES
                    193X
000.000             194X         ORG     0
                    195X
000.000             196X UNT.FLG DS      1               UNIT SPECIFIC  *DEV.FLG*
000.001             197X UNT.SPG DS      1               Sectors Per Group               /80.04.GC/
000.002             198X UNT.GRT DS      2               ADDRESS OF GROUP RESERVATION TABLE (IF DT.DD)
000.004             199X UNT.GTS DS      2               GRT SECTOR NUMBER
000.006             200X UNT.DIS DS      2               DIRECTORY FIRST SECTOR NUMBER
                    201X
000.010             202X UNT.SIZ EQU     *               SIZE OF UNIT SPECIFIC DATA TABLE PER UNIT
000.010             203          XTEXT   HOSDEF
```

```
                        205X **      HOSDEF - DEFINE HOS PARAMETER.
                        206X *
                        207X
                        208X
000.040                 209X VERS    EQU     2*16+0          VERSION 2.0
                        210X
000.377                 211X SYSCALL EQU     377Q            SYSCALL INSTRUCTION
                        212X
                        213X
000.000                 214X         ORG     0
                        215X
                        216X *       RESIDENT FUNCTIONS
                        217X
000.000                 218X .EXIT   DS      1               EXIT (MUST BE FIRST)
000.001                 219X .SCIN   DS      1               SCIN
000.002                 220X .SCOUT  DS      1               SCOUT
000.003                 221X .PRINT  DS      1               PRINT
000.004                 222X .READ   DS      1               READ
000.005                 223X .WRITE  DS      1               WRITE
000.006                 224X .CONSL  DS      1               SET/CLEAR CONSOLE OPTIONS
000.007                 225X .CLRCO  DS      1               CLEAR CONSOLE BUFFER
000.010                 226X .LOADO  DS      1               LOAD AN OVERLAY
000.011                 227X .VERS   DS      1               RETURN HDOS VERSION NUMBER
000.012                 228X .SYSRES DS      1               PRECEDING FUNCTIONS ARE RESIDENT
                        229X
                        230X
                        231X *       *HDOSOVL0.SYS*  FUNCTIONS
                        232X
000.040                 233X         ORG     40A
                        234X
000.040                 235X .LINK   DS      1               LINK  (MUST BE FIRST)
000.041                 236X .CTLC   DS      1               CTL-C
000.042                 237X .OPENR  DS      1               OPENR
000.043                 238X .OPENW  DS      1               OPENW
000.044                 239X .OPENU  DS      1               OPENU
000.045                 240X .OPENC  DS      1               OPENC
000.046                 241X .CLOSE  DS      1               CLOSE
000.047                 242X .POSIT  DS      1               POSITION
000.050                 243X .DELET  DS      1               DELETE
000.051                 244X .RENAM  DS      1               RENAME
000.052                 245X .SETTP  DS      1               SETTOP
000.053                 246X .DECODE DS      1               NAME DECODE
000.054                 247X .NAME   DS      1               GET FILE NAME FROM CHANNEL
000.055                 248X .CLEAR  DS      1               CLEAR CHAN
000.056                 249X .CLEARA DS      1               CLEAR ALL CHANS
000.057                 250X .ERROR  DS      1               LOOKUP ERROR
000.060                 251X .CHFLG  DS      1               CHANGE FLAGS
000.061                 252X .DISMT  DS      1               FLAG SYSTEM DISK DISMOUNTED
000.062                 253X .LOADD  DS      1               LOAD DEVICE DRIVER
000.063                 254X .OPEN   DS      1               Parametrized Open
                        255X
                        256X
                        257X *       *HDOSOVL1.SYS*  FUNCTIONS
                        258X
000.200                 259X         ORG     200Q
                        260X
```

```
000.200  261X .MOUNT  DS     1          MOUNT  (MUST BE FIRST)
000.201  262X .DMOUN  DS     1          DISMOUNT
000.202  263X .MONMS  DS     1          MOUNT/NO MESSAGE
000.203  264X .DMNMS  DS     1          DISMOUNT/NO MESSAGE
000.204  265X .RESET  DS     1          RESET = DISMOUNT/MOUNT OF UNIT
000.205  266X .CLEAN  DS     1          Clean device
000.206  267X .DAD    DS     1          Dismount All Disks              /80.08.sc/
000.207  268         XTEXT  HOSEQU

         270X **      HDOS SYSTEM EQUIVALENCES.
         271X *
         272X
024.000  273X S.GRT0  EQU    24000A     SYSTEM AREA FOR  GRT0
025.000  274X S.GRT1  EQU    25000A     SYSTEM AREA FOR  GRT1
026.000  275X S.GRT2  EQU    26000A     SYSTEM AREA FOR  GRT2
         276X
030.000  277X ROMBOOT EQU    30000A     ROM BOOT ENTRY
         278X
040.100  279X         ORG    40100A     FREE SPACE FROM PAM-8
         280X
040.100  281X         DS     8          JUMP TO SYSTEM EXIT
040.110  282X D.CON   DS     16         DISK CONSTANTS
040.130  283X SYDD    EQU    *          SYSTEM DISK ENTRY POINT
040.130  284X D.VEC   DS     24*3       SYSTEM ROM ENTRY VECTORS
040.240  285X D.RAM   DS     31         SYSTEM ROM WORK AREA
040.277  286X S.VAL   DS     36         SYSTEM VALUES
040.343  287X S.INT   DS     115        SYSTEM INTERNAL WORK AREAS
041.126  288X         DS     16
041.146  289X S.SOVR  DS     2          STACK OVERFLOW WARNING
041.150  290X         DS     42200A-*   SYSTEM STACK
001.032  291X STACKL  EQU    *-S.SOVR   STACK SIZE
         292X
042.200  293X STACK   EQU    *          LWA+1 SYSTEM STACK
042.200  294X USERFWA EQU    *          USER FWA
042.200  295         XTEXT  ESVAL

         297X **      S.VAL - SYSTEM VALUE DEFINTIONS.
         298X *
         299X *       THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.
         300X *
         301X *       THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.
         302X
         303X
040.277  304X         ORG    S.VAL
         305X
040.277  306X S.DATE  DS     9          SYSTEM DATE (IN ASCII)
040.310  307X S.DATC  DS     2          CODED DATE
040.312  308X S.TIME  DS     4          TIME FROM MIDNIGHT (IN TICS)
040.316  309X S.HIMEM DS     2          HARDWARE HIGH MEMORY ADRESS+1
         310X
```

```
040.320                 311X S.SYSM  DS      2               FWA RESIDENT SYSTEM
                        312X
040.322                 313X S.USRM  DS      2               LWA USER MEMORY
                        314X
040.324                 315X S.OMAX  DS      2               MAX OVERLAY SIZE FOR SYSTEM
                        316X
                        317X
                        318X **      THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL
                        319X
000.200                 320X CSL.ECH EQU     10000000B       SUPPRESS ECHO
000.004                 321X CSL.RAW EQU     00000100B       Raw Mode I/O                    /80.09.sc/
000.002                 322X CSL.WRP EQU     00000010B       WRAP LINES AT WIDTH
000.001                 323X CSL.CHR EQU     00000001B       OPERATE IN CHARACTER MODE
                        324X
000.000                 325X I.CSLMD EQU     0               S.CSLMD IS FIRST BYTE
040.326                 326X S.CSLMD DS      1               CONSOLE MODE
                        327X
000.200                 328X CTP.BKS EQU     10000000B       TERMINAL PROCESSES BACKSPACES
000.100                 329X CTP.FF  EQU     01000000B       Terminal Processes Form-Feed    /80.09.sc/
000.040                 330X CTP.MLI EQU     00100000B       MAP LOWER CASE TO UPPER ON INPUT
000.020                 331X CTP.MLO EQU     00010000B       MAP LOWER CASE TO UPPER ON OUTPUT
000.010                 332X CTP.2SB EQU     00001000B       TERMINAL NEEDS TWO STOP BITS
000.002                 333X CTP.BKM EQU     00000010B       MAP BKSP (UPON INPUT) TO RUBOUT
000.001                 334X CTP.TAB EQU     00000001B       TERMINAL SUPPORTS TAB CHARACTERS
                        335X
000.001                 336X I.CONTY EQU     1               S.CONTY IS 2ND BYTE
000.000                 337X         ERRNZ   *-S.CSLMD-I.CONTY
040.327                 338X S.CONTY DS      1               CONSOLE TYPE FLAGS
000.002                 339X I.CUSOR EQU     2               S.CUSOR IS 3RD BYTE
000.000                 340X         ERRNZ   *-S.CSLMD-I.CUSOR
040.330                 341X S.CUSOR DS      1               CURRENT CURSOR POSITION
000.003                 342X I.CONWI EQU     3               S.CONWI IS 4TH BYTE
000.000                 343X         ERRNZ   *-S.CSLMD-I.CONWI
040.331                 344X S.CONWI DS      1               CONSOLE WIDTH
                        345X
000.001                 346X CO.FLG  EQU     00000001B       CTL-O FLAG
000.200                 347X CS.FLG  EQU     10000000B       CTL-S FLAG
                        348X
000.004                 349X I.CONFL EQU     4               S.CONFL IS 5TH BYTE
000.000                 350X         ERRNZ   *-S.CSLMD-I.CONFL
040.332                 351X S.CONFL DS      1               CONSOLE FLAGS
                        352X
040.333                 353X S.CAADR DS      2               ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040.335                 354X S.CCTAB DS      6               ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
040.343                 355          XTEXT   ESINT


                        357X **      S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
                        358X *
                        359X *       THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
                        360X *       MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
                        361X
                        362X
040.343                 363X         ORG     S.INT
```

```
                    364X
                    365X **     CONSOLE STATUS FLAGS
                    366X
040.343             367X S.CDB  DS    1             CONSOLE DESCRIPTOR BYTE
000.000             368X CDB.H85 EQU  00000000B
000.001             369X CDB.H84 EQU  00000001B     =0 IF H8-5, =1 IF H8-4
040.344             370X S.BAUD DS    2             [0-14]  H8-4 BAUD RATE, =0 IF H8-5
                    371X *                          [15]    =1 IF BAUD RATE => 2 STOP BITS
                    372X
                    373X **     TABLE ADDRESS WORDS
                    374X
040.346             375X S.DLINK DS   2             ADDRESS OF DATA IN HDOS CODE
040.350             376X S.OFWA DS    2             FWA  OVERLAY  TABLE
040.352             377X S.CFWA DS    2             FWA  CHANNEL  TABLE
040.354             378X S.DFWA DS    2             FWA  DEVICE   TABLE
040.356             379X S.RFWA DS    2             FWA  RESIDENT HDOS CODE
                    380X
                    381X **     DEVICE DRIVER DELAYED LOAD FLAGS
                    382X
040.360             383X S.DDLDA DS   2             DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
040.362             384X S.DDLEN DS   2             CODE LENGTH IN BYTES
040.364             385X S.DDGRP DS   1             GROUP NUMBER FOR DRIVER
040.365             386X         DS   1             HOLD PLACE
                    387X *S.DDSEC      DS   2                SECTOR NUMBER FOR DRIVER ( * OBSOLETE ! * )
040.366             388X S.DDDTA DS   2             DEVICE'S ADDRESS IN DEVLST +DEV.RES
040.370             389X S.DDOPC DS   1             OPEN OPCODE PENDEDING
                    390X
                    391X **     OVERLAY MANAGEMENT FLAGS
                    392X
000.001             393X OVL.IN  EQU  00000001B     IN MEMORY
000.002             394X OVL.RES EQU  00000010B     PERMINANTLY RESIDENT
000.014             395X OVL.NUM EQU  00001100B     OVERLAY NUMBER MASK
000.200             396X OVL.UCS EQU  10000000B     USER CODE SWAPPED FOR OVERLAY
                    397X
040.371             398X S.OVLFL DS   1             OVERLAY FLAG
040.372             399X S.UCSF  DS   2             FWA SWAPPED USER CODE
040.374             400X S.UCSL  DS   2             LENGTH SWAPPED USER CODE
040.376             401X S.OVLS  DS   2             SIZE OF OVERLAY CODE
041.000             402X S.OVLE  DS   2             ENTRY POINT OF OVERLAY CODE
                    403X
041.002             404X S.SSN   DS   2             SWAP AREA SECTOR NUMBER
041.004             405X S.OSN   DS   2             OVERLAY SECTOR NUMBER
                    406X
                    407X *      SYSCALL PROCESSING WORK AREAS
                    408X
041.006             409X S.CACC  DS   1             (ACC) UPON SYSCALL
041.007             410X S.CODE  DS   1             SYSCALL INDEX IN PROGRESS
                    411X
                    412X *      JUMPS TO ROUTINES IN RESIDENT HDOS CODE
                    413X
041.010             414X S.JUMPS DS   0             START OF DUMP VECTORS
041.010             415X S.SDD   DS   3             JUMP TO STAND-IN DEVICE DRIVER
041.013             416X S.FASER DS   3             JUMP TO FATSERR (FATAL SYSTEM ERROR)
041.016             417X S.DIREA DS   3             JUMP TO DIREAD (DISK FILE READ)
041.021             418X S.FCI   DS   3             JUMP TO FCI (FETCH CHANNEL INFO)
041.024             419X S.SCI   DS   3             JUMP TO SCI (STORE CHANNEL INFO)
```

```
041.027                 420X S.GUP    DS      3               JUMP TO GUP (GET UNIT POINTER)
                        421X
041.032                 422X S.MOUNT  DS      1               <>0 IF THE SYSTEM DISK IS MOUNTED
041.033                 423X S.DCS    DS      1               DEFAULT CLUSTER SIZE-1
                        424X
041.034                 425X S.BOOTF  DS      1               BOOT FLAGS
000.001                 426X BOOT.P   EQU     00000001B       EXECUTE PROLOGUE UPON BOOTUP
                        427X
                        428X *        STACK VALUE SAVED FOR OVERLAY SYSCALLS
                        429X
041.035                 430X S.OVSTK  DS      2               VALUE OF SP UPON SYSCALLS USING OVERLAY
                        431X
041.037                 432X          DS      1               RESERVED


                        434X **       ACTIVE I/O AREA.
                        435X *
                        436X *        THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION
                        437X *        CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM
                        438X *        THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.
                        439X *
                        440X *        NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY
                        441X *        FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE
                        442X *        8080 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY
                        443X *        COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND
                        444X *        BACKDATED AFTER PROCESSING.
                        445X
041.040                 446X AIO.VEC  DS      3               JUMP INSTRUCTION
041.041                 447X AIO.DDA  EQU     *-2             DEVICE DRIVER ADDRESS
041.043                 448X AIO.FLG  DS      1               FLAG BYTE
041.044                 449X AIO.GRT  DS      2               ADDRESS OF GROUP RESERV TABLE
041.046                 450X AIO.SPG  DS      1               SECTORS PER GROUP
041.047                 451X AIO.CGN  DS      1               CURRENT GROUP NUMBER
041.050                 452X AIO.CSI  DS      1               CURRENT SECTOR INDEX
041.051                 453X AIO.LGN  DS      1               LAST GROUP NUMBER
041.052                 454X AIO.LSI  DS      1               LAST SECTOR INDEX
041.053                 455X AIO.DTA  DS      2               DEVICE TABLE ADDRESS
041.055                 456X AIO.DES  DS      2               DIRECTORY SECTOR
041.057                 457X AIO.DEV  DS      2               DEVICE CODE
041.061                 458X AIO.UNI  DS      1               UNIT NUMBER (0-9)
                        459X
041.062                 460X AIO.DIR  DS      DIRELEN         DIRECTORY ENTRY
                        461X
041.111                 462X AIO.CNT  DS      1               SECTOR COUNT
041.112                 463X AIO.EOM  DS      1               END OF MEDIA FLAG
041.113                 464X AIO.EOF  DS      1               END OF FILE FLAG
041.114                 465X AIO.TFP  DS      2               TEMP FILE POINTERS
041.116                 466X AIO.CHA  DS      2               ADDRESS OF CHANNEL BLOCK (IOC.DDA)
```

```
041.120                 468X S.BDA  DS      1               Boot Device Address (Setup by ROM) /80.09.sc/
041.121                 469X S.SCR  DS      2               SYSTEM SCRATCH AREA ADDRESS
041.123                 470         XTEXT   ASCII

                        472X **     ASCII CHARACTER EQUIVALENCES.
                        473X
000.015                 474X CR     EQU     13              CARRIAGE RETURN
000.012                 475X LF     EQU     10              LINE FEED
000.200                 476X NULL   EQU     200Q            PAD CHARACTER
000.000                 477X NUL2   EQU     0
000.007                 478X BELL   EQU     7               BELL CHARACTER
000.177                 479X RUBOUT EQU     177Q
000.010                 480X BKSP   EQU     10Q             CTL-H
000.026                 481X C.SYN  EQU     26Q             SYNC
000.002                 482X C.STX  EQU     2               STX
000.047                 483X QUOTE  EQU     47Q
000.011                 484X TAB    EQU     11Q
000.033                 485X ESC    EQU     33Q
000.012                 486X NL     EQU     12Q             NEW LINE (HDOS SYSTEMS)
000.212                 487X ENL    EQU     NL+200Q         NL + END-OF-LINE-FLAG
000.014                 488X FF     EQU     14Q             FORM FEED
000.001                 489X CTLA   EQU     01Q             CTL-A
000.002                 490X CTLB   EQU     02Q             CTL-B
000.003                 491X CTLC   EQU     03Q             CTL-C
000.004                 492X CTLD   EQU     04Q             CTL-D
000.017                 493X CTLO   EQU     17Q             CTL-O
000.020                 494X CTLP   EQU     20Q             CTL-P
000.021                 495X CTLQ   EQU     21Q             CTL-Q
000.023                 496X CTLS   EQU     23Q             CTL-S
000.032                 497X CTLZ   EQU     32Q             CTL-Z
041.123                 498         XTEXT   PICDEF

                        500X **     PIC FORMAT EQUIVALENCES.
                        501X
000.000                 502X        ORG     0
                        503X
000.000                 504X PIC.ID DS      1               377Q = BINARY FILE FLAG
000.001                 505X        DS      1               FILE TYPE (FT.PIC)
000.002                 506X PIC.LEN DS     2               LENGTH OF ENTIRE RECORD
000.004                 507X PIC.PTR DS     2               INDEX OF START OF PIC TABLE
                        508X
000.006                 509X PIC.COD DS     0               CODE STARTS HERE
000.006                 510         XTEXT   ABSDEF
```

```
                        512X **     ABS FORMAT EQUIVALENCES.
                        513X
000.000                 514X        ORG     0
                        515X
000.000                 516X ABS.ID DS      1               377Q = BINARY FILE FLAG
000.001                 517X        DS      1               FILE TYPE (FT.ABS)
000.002                 518X ABS.LDA DS     2               LOAD ADDRESS
000.004                 519X ABS.LEN DS     2               LENGTH OF ENTIRE RECORD
000.006                 520X ABS.ENT DS     2               ENTRY POINT
                        521X
000.010                 522X ABS.COD DS     0               CODE STARTS HERE
000.010                 523  XTEXT  FILDEF


                        525X **     FILDEF - FILE TYPE DEFINITIONS.
                        526X *
                        527X *      DB      377Q,FT.XXX
                        528X
                        529X
000.000                 530X FT.ABS EQU     0               ABSOLUTE BINARY
000.001                 531X FT.PIC EQU     1               POSITION INDEPENDANT CODE
000.002                 532X FT.REL EQU     2               RELOCATABLE CODE
000.003                 533X FT.BAC EQU     3               COMPILED BASIC CODE
                        534
042.170                 535         ORG     USERFWA-ABS.COD
042.170  377 000       536         DB      377Q,FT.ABS
042.172  200 042       537         DW      USERFWA         LOAD ADDR
042.174  171 011       538         DW      MEML-USERFWA    LOAD SIZE
042.176  200 042       539         DW      START           ENTRY
```

```
                                542  ***     PATCH - MAIN ROUTINE.
                                543  *
                                544  *       ENTERED HERE FROM *PRS*
                                545
                                546
042.200                         547  PATCH   EQU     *
042.200                         548  START   EQU     *               PROGRAM ENTERS HERE
042.200                         549  RESTART EQU     *
                                550
042.200  061 200 042          551          LXI     SP,STACK        CLEAN STACK
042.203  315 372 042          552          CALL    PRS             PRESET
042.206  315 214 042          553          CALL    PATCH1          DO THE PATCH
042.211  303 200 042          554          JMP     RESTART         DO ANOTHER
                                555
042.214  315 144 043          556  PATCH1  CALL    OFP             OPEN FILE FOR PATCH
042.217  322 225 042          557          JNC     PATCH2          GOT NAME
042.222  257                  558          XRA     A
042.223  377 000              559          DB      SYSCALL,.EXIT   CTL-D, EXIT
                                560
042.225                         561  PATCH2  EQU     *
000.001                         562          IF      .SYS.           SYSTEM OPTIONS
                                563          LDA     PHTSWI
                                564          ANA     A
                                565          JNZ     APF             ADD PHT TO FILE
                                566          ENDIF
042.225  072 261 053          567          LDA     PHIST
042.230  247                  568          ANA     A
042.231  312 244 042          569          JZ      PATCH3          USER MODE, NO PHT
042.234  315 211 044          570          CALL    GPI             GET PATCH ID
042.237  330                  571          RC                      ERROR
042.240  315 315 044          572          CALL    GPC             GET PREREQ CODE
042.243  330                  573          RC                      CTL-D
                                574
                                575  *       GET NEXT ADDRESS
                                576
042.244  315 215 045          577  PATCH3  CALL    GPA             GET PATCH ADDRESS
042.247  332 260 042          578          JC      PATCH5          GOT CTL-D, ENTER PATCH
042.252  315 021 046          579  PATCH4  CALL    APP             ACCEPT PROGRAM PATCHES
042.255  303 244 042          580          JMP     PATCH3          CTL-D
                                581
                                582  *       ALL DONE. GET CHECKSUM, IF NEEDED
                                583
042.260  072 261 053          584  PATCH5  LDA     PHIST
042.263  247                  585          ANA     A
042.264  304 212 046          586          CNZ     CPC             CHECK PATCH CRC, IF SYSTEM MODE
042.267  330                  587          RC                      ERROR
                                588
                                589  *       OFF CTL-C'S, START MODIFING FILE
                                590
042.270  041 000 000          591          LXI     H,0
042.273  076 003              592          MVI     A,CTLC
042.275  377 041              593          DB      SYSCALL,.CTLC   DISABLE CTL-C'S
042.277  315 101 047          594          CALL    EPF             ENTER PATCHES IN FILE
042.302  315 023 050          595          CALL    FVB             FLUSH VIEW BUFFER
042.305  315 025 051          596          CALL    UBH             UPDATE BINARY HEADER
042.310  072 261 053          597          LDA     PHIST
```

```
042.313  247              598          ANA     A
042.314  304 141 051      599          CNZ     WPH             WRITE PATCH HISTORY, IF ANY
042.317  076 000          600          MVI     A,CN.FIL
042.321  377 046          601          DB      SYSCALL,,CLOSE  CLOSE PROGRAM FILE
042.323  311              602          RET     RESTART         EXIT TO EXEC LOOP


                          604  ***     ERROR - FATAL SYSTEM ERROR OCCOURED.
                          605  *
                          606  *       EXIT TO RESTART
                          607  *
                          608  *       ENTRY   (A) = ERROR CODE
                          609
                          610
042.324  315 332 042      611  ERROR   CALL    ERROR.
042.327  303 200 042      612          JMP     RESTART
                          613
042.332  365              614  ERROR.  PUSH    PSW             SAVE CODE
042.333  315 301 052      615          CALL    $CCO            CLEAR CTL-O
042.336  315 136 031      616          CALL    $TYPTX
042.341  012 007 105      617          DB      NL,BELL,'ERROR -',' '+200Q
042.353  361              618          POP     PSW
042.354  046 012          619          MVI     H,NL
042.356  377 057          620          DB      SYSCALL,,ERROR
042.360  311              621          RET


                          623  ***     CCHIT - ENTERED WHEN CTL-C HIT.
                          624  *
                          625
                          626
042.361  373              627  CCHIT   EI
042.362  315 136 031      628          CALL    $TYPTX
042.365  136 303          629          DB      '^','C'+200Q
042.367  303 200 042      630          JMP     RESTART
```

```
                                633  ***     PRS - PRESET PROGRAM.
                                634  *
                                635  *       PRS PRESETS THE PROGRAM, BY
                                636  *
                                637  *       1) CLEARING ALL CHANNELS (INCLUDING THE OVERLAY CHANNEL)
                                638  *       2) PRINTING A PROGRAM TITLE
                                639  *       3) SETING UP THE MEMORY USAGE
                                640  *       4) SETING UP THE DATA TABLES.
                                641  *       5) SETING UP THE CTL-C ADDRESS
                                642  *
                                643  *       ENTRY   NONE
                                644  *       EXIT    NONE
                                645  *       USES    ALL
                                646
                                647
042,372                         648  PRS     EQU     *
                                649
                                650  *       VERIFY THE VERSION OF HDOS                              /79.12.GC/
                                651
042,372  377 011                652          DB      SYSCALL,.VERS                                   /79.12.GC/
042,374  332 130 043            653          JC      PRSERR1                 PROBABLY NO .VERS SYSCALL /79.12.GC/
042,377  376 040                654          CPI     VERS                                            /79.12.GC/
043,001  302 130 043            655          JNZ     PRSERR1                 NOT CORRECT VERSION     /79.12.GC/
                                656
043,004  377 056                657          DB      SYSCALL,.CLEARA         CLEAR ALL CHANNELS BUT OVERLAY
043,006  076 377                658          MVI     A,377Q
043,010  377 055                659          DB      SYSCALL,.CLEAR          CLEAR OVERLAY CHANNEL
                                660
                                661  *       SETUP BUFFERS, POINTERS, AND VALUES
                                662
043,012  257                    663          XRA     A
043,013  062 236 053            664          STA     SIVBV           SECTOR IN VIEW BUFFER NOT VALID
043,016  062 261 053            665          STA     PHIST           NO PHT
043,021  041 000 060            666          LXI     H,PATLIST
043,024  042 246 053            667          SHLD    PLPTR           CLEAR PATCH LIST
                                668
                                669  *       ANNOUNCE SELF
                                670
043,027  315 136 031            671          CALL    $TYPTX
043,032  012 120 101            672          DB      NL,'PATCH Issue #50.06.00.    ',NL,ENL          /79.12.GC/
                                673
                                674  *       COMPUTE FINAL PATCH ADDRESS, REQUEST FROM HDOS
                                675
043,067  052 320 040            676          LHLD    S.SYSM
043,072  353                    677          XCHG
043,073  052 324 040            678          LHLD    S.OMAX
043,076  173                    679          MOV     A,E
043,077  225                    680          SUB     L
043,100  157                    681          MOV     L,A
043,101  172                    682          MOV     A,D
043,102  234                    683          SBB     H
043,103  147                    684          MOV     H,A             (HL) = MAX WITH OVERLAY STILL RESIDENT
043,104  053                    685          DCX     H
043,105  053                    686          DCX     H
043,106  053                    687          DCX     H
043,107  053                    688          DCX     H               4 BYTES OF SLOP
```

```
043.110  042 250 053   689          SHLD    PLMAX           SET MAX
043.113  377 052       690          DB      SYSCALL,.SETTP  REQUEST MAX
043.115  332 324 042   691          JC      ERROR           PROBLEMS, WILL PROBABLY LOOP GENERATING THIS MESSAGE
                       692
                       693  *       SETUP CTL-C
                       694
043.120  041 361 042   695          LXI     H,CCHIT
043.123  076 003       696          MVI     A,CTLC
043.125  377 041       697          DB      SYSCALL,.CTLC   SETUP CTL-C PROCESSING
043.127  311           698          RET                     EXIT
                       699
043.130  076 050       700  PRSERR1 MVI     A,EC.NCV        NOT CORRECT VERS. OF HDOS       /79.12.GC/
                       701
043.132  315 332 042   702  PRSERR  CALL    ERROR.                                          /79.12.GC/
043.135  076 001       703          MVI     A,1                                             /79.12.GC/
043.137  377 000       704          DB      SYSCALL,.EXIT                                   /79.12.GC/
                       705
043.141  303 141 043   706          JMP     *               SHOULD NEVER HAPPEN             /79.12.GC/
```

```
                                    709  ***     OFP - OPEN FILE FOR PATCHES.
                                    710  *
                                    711  *       OFP PROMPTS THE USER FOR A FILE NAME TO PATCH, AND
                                    712  *
                                    713  *       1) DETERMINES IF THE DEVICE IS SUITABLE (MUST BE H17)
                                    714  *       2) DETERMINES IF THE FILE EXITS
                                    715  *       3) DETERMINES ITS TYPE (ABS, PIC)
                                    716  *       4) SEES IF THERE IS A PATCH HISTORY TABLE (PHT) ON IT
                                    717  *       5) DETERMINES THE PROGRAMS FWA AND LWA.
                                    718  *
                                    719  *       ENTRY   NONE
                                    720  *       EXIT    'C' CLEAR IF OK
                                    721  *                FILE DESCRIPTOR VALUES SETUP.
                                    722  *               'C' SET IF CTL-D
                                    723  *       USES    ALL
                                    724
                                    725
043.144  315 136 031               726  OFP     CALL    $TYPTX
043.147  106 151 154               727          DB      'File Name?',' '+200Q
043.162  041 071 054               728          LXI     H,LINE
043.165  315 027 053               729          CALL    $RTL.           READ LINE IN UPPER CASE
043.170  330                      730          RC                      CTL-D
                                    731
                                    732  *       SEE IF ANY SWITCHES SPECIFIED
                                    733
043.171  315 137 044               734          CALL    DCS             DECODE COMMAND SWITCHES
                                    735
                                    736  *       GOT FILE NAME.
                                    737
043.174  001 077 044               738          LXI     B,OFPA          LOOK UP DEVICE TYPE
043.177  021 240 053               739          LXI     D,DEFALT
043.202  377 053                   740          DB      SYSCALL,.DECODE
043.204  332 324 042               741          JC      ERROR           ERROR
043.207  072 077 044               742          LDA     OFPA+0          (A) = DEVICE TYPE
043.212  057                      743          CMA
043.213  346 005                   744          ANI     DT,DD+DT,CW     MUST BE DIRECTORY DEVICE, CAPABLE OF WRITE
043.215  076 005                   745          MVI     A,EC.DNS
043.217  302 324 042               746          JNZ     ERROR           ERROR
043.222  021 240 053               747          LXI     D,DEFALT
043.225  041 071 054               748          LXI     H,LINE
043.230  076 000                   749          MVI     A,CN.FIL
043.232  377 042                   750          DB      SYSCALL,.OPENR  OPEN FILE FOR READ
043.234  332 324 042               751          JC      ERROR           ERROR
                                    752
                                    753  *       DETERMINE PROGRAM TYPE
                                    754
043.237  001 000 000               755          LXI     B,0
043.242  315 235 050               756          CALL    PPF             POSITION PROGRAM FILE TO BEGINNING
043.245  001 000 001               757          LXI     B,256
043.250  021 000 056               758          LXI     D,BUFFER
043.253  076 000                   759          MVI     A,CN.FIL
043.255  377 004                   760          DB      SYSCALL,.READ   READ FIRST BLOCK
043.257  332 324 042               761          JC      ERROR           ERROR
043.262  052 000 056               762          LHLD    BUFFER
043.265  054                      763          INR     L
043.266  076 020                   764          MVI     A,EC.IFC
```

```
043.270  302 324 042   765          JNZ     ERROR           NOT BINARY FILE
043.273  174           766          MOV     A,H
043.274  062 252 053   767          STA     FILTYP          SET FILE TYPE
043.277  376 000       768          CPI     FT.ABS          SEE IF ABS
043.301  312 332 043   769          JE      OFP1            IS ABS
043.304  376 001       770          CPI     FT.PIC
043.306  076 020       771          MVI     A,EC.IFC        ASSUME ILLEGAL FILE CONTENT
043.310  302 324 042   772          JNE     ERROR           ERROR
                       773
                       774  *       IS PIC FILE
                       775
043.313  041 000 000   776          LXI     H,0
043.316  042 230 053   777          SHLD    SKEW            ADDRESS N IS BYTE N
043.321  042 262 053   778          SHLD    PGMFWA          PROGRAM STARTS AT 0
043.324  052 002 056   779          LHLD    BUFFER+PIC.LEN  (HL) = PROGRAM LENGTH
043.327  303 355 043   780          JMP     OFP1.5
                       781
                       782  *       IS ABS FILE
                       783
043.332  052 002 056   784  OFP1    LHLD    BUFFER+ABS.LDA
043.335  042 262 053   785          SHLD    PGMFWA          STORE PROGRAM'S FWA
043.340  315 224 030   786          CALL    $CHL            (HL) = -USERFWA
043.343  021 010 000   787          LXI     D,ABS.COD
043.346  031           788          DAD     D               (HL) = ABS.COD-USERFWA
043.347  042 230 053   789          SHLD    SKEW            PGM ADDRESS-USERFWA+ABS.COD = FILE ADDRESS
043.352  052 004 056   790          LHLD    BUFFER+ABS.LEN  (HL) = PROGRAM LENGTH
                       791
043.355  042 266 053   792  OFP1.5  SHLD    FILSIZ          SET FILE SIZE
043.360  353           793          XCHG                    (DE) = LENGTH
043.361  052 262 053   794          LHLD    PGMFWA
043.364  031           795          DAD     D
043.365  053           796          DCX     H               (HL) = PGM LWA
043.366  042 264 053   797          SHLD    PGMLWA
043.371  353           798          XCHG
043.372  052 230 053   799          LHLD    SKEW
043.375  031           800          DAD     D               (HL) = FILE INDEX OF PROGRAM LWA
043.376  044           801          INR     H               (H) = SECTOR NUMBER OF LAST PGM SECTOR+1
043.377  114           802          MOV     C,H
044.000  006 000       803          MVI     B,0             (BC) = SECTOR NUMBER WHERE PHT MIGHT BE
044.002  315 235 050   804          CALL    PPF             POSITION BEFORE PHT
044.005  332 051 044   805          JC      OFP2            MISSING
                       806
044.010  001 000 001   807          LXI     B,256
044.013  021 000 057   808          LXI     D,PHT
044.016  076 000       809          MVI     A,CN.FIL
044.020  377 004       810          DB      SYSCALL,.READ   TRY TO READ SUPPOSED PHT
044.022  315 324 047   811          CALL    AEE             ALLOW END OF FILE ERROR
044.025  332 051 044   812          JC      OFP2            NO PHT
044.030  021 175 053   813          LXI     D,PHTFORM+PHT.HDR
044.033  041 000 057   814          LXI     H,PHT
044.036  016 011       815          MVI     C,PHTHDRL
044.040  315 060 030   816          CALL    $COMP           SEE IF IN PROPER FORMAT FOR PHT
044.043  076 001       817          MVI     A,1
044.045  062 261 053   818          STA     PHIST           ASSUME HAVE PHT
044.050  310           819          RE                      ALL OK, HAVE PHT
                       820
```

```
                                821  *       DONT HAVE A PHT. AM IN USER MODE
                                822
044.051  257                    823  OFP2    XRA     A
044.052  062 261 053            824          STA     PHIST
044.055  076 000                825          MVI     A,CN.FIL
044.057  377 046                826          DB      SYSCALL,.CLOSE  CLOSE FILE WHICH IS OPEN FOR READ
044.061  076 000                827          MVI     A,CN.FIL
044.063  021 240 053            828          LXI     D,DEFALT
044.066  041 071 054            829          LXI     H,LINE
044.071  377 044                830          DB      SYSCALL,.OPENU  OPEN USER FILE FOR UPDATE!
044.073  332 324 042            831          JC      ERROR
044.076  311                    832          RET
                                833
                                834
044.077                         835  OFPA    DS      32              .DECODE BUFFER


                                837  **      DCS - DECODE COMMAND SWITCHES,
                                838  *
                                839  *       DCS DECODES AND REMOVES SWITCHES FROM THE COMMAND LINE
                                840  *       IN *LINE*
                                841  *
                                842  *       ENTRY   (HL) = #LINE
                                843  *       EXIT    'C' SET IF ERROR
                                844  *                MESSAGE ALREADY GIVEN
                                845  *               'C' CLEAR IF OK
                                846  *       USES    ALL
                                847
                                848
044.137  257                    849  DCS     XRA     A
044.140  062 232 053            850          STA     SDISP           PRESET SWITCHES
000.001                         851          IF      .SYS.
                                852          STA     CHECKC          CLEAR CHECK FLAG
                                853          STA     PHTSWI          CLEAR PHT ADD SWITCH
                                854          ENDIF
044.143  021 160 044            855          LXI     D,DCSA
044.146  315 064 052            856          CALL    $DRS            DECODE AND REMOVE SWITCHES
044.151  332 324 042            857          JC      ERROR           SWITCH ERROR
044.154  041 071 054            858          LXI     H,LINE
044.157  311                    859          RET
                                860
                                861
044.160                         862  DCSA    DS      0               FWA SWITCH TABLE
044.160  104 111 123            863          DB      'DISP','L'+200Q,'A'+200Q,'C'+200Q,'E'+200Q,200Q
044.171  174 044                864          DW      SW.DISP         DISPLACE SWITCH
                                865
000.001                         866          IF      .SYS.
                                867          DB      'PHT',200Q
                                868          DW      SW.PHT
                                869
                                870          DB      'CHECK',200Q
                                871          DW      SW.CHK
                                872          ENDIF
                                873
```

```
044.173  000            874          DB     0              END OF TABLE


                         876  **      SW.DISP - PROCESS /DISP:NN SWITCH
                         877
044.174  315 235 051    878  SW.DISP CALL   $DNS.          DECODE NUMERIC SWITCH
044.177  076 032        879          MVI    A,EC.IS
044.201  332 324 042    880          JC     ERROR          ILLEGAL SWITCH
044.204  173            881          MOV    A,E
044.205  062 232 053    882          STA    SDISP          SET SECTOR DISPLACEMENT
044.210  311            883          RET


                         885  **      SW.PHT -  /PHT SWITCH
                         886  *
                         887  *       ADD PHT TO FILE
                         888
000.001                  889          IF     .SYS.
                         890  SW.PHT  MVI    A,1
                         891          STA    PHTSWI
                         892          RET
                         893  SW.CHK  SPACE  3,10
                         894  **      SW.CHK - /CHECK SWITCH
                         895  *
                         896  *       SHOW CHECKSUM VALUES
                         897
                         898  SW.CHK  MVI    A,1
                         899          STA    CHECKC
                         900          RET
                         901          ENDIF
```

```
                                    904  ***     GPI - GET PATCH ID.
                                    905  *
                                    906  *       GPI PROMPTS THE USER WITH
                                    907  *
                                    908  *       PATCH ID?
                                    909  *
                                    910  *       AND READS IN A PATCH ID SEQUENCE.
                                    911  *
                                    912  *       THE ID IS A 6 CHARACTER SEQUENCE, REPRESENTING THE ID NUMBER
                                    913  *       (1 TO 256) AND A 2 BYTE CRC-16 FOR IT.
                                    914  *
                                    915  *       SEE 'RES' FOR A DESCRIPTION OF THE ENCODING TECHNIQUE.
                                    916  *
                                    917  *       ENTRY   NONE
                                    918  *       EXIT    'C' CLEAR IF OK
                                    919  *                PATCHID = ID BYTE
                                    920  *               'C' SET IF CTL-D
                                    921  *       USES    ALL
                                    922
                                    923
044.211  315 136 031               924  GPI     CALL    $TYPTX
044.214  120 141 164               925          DB      'Patch ID?',' '+200Q
044.226  006 001                   926          MVI     B,1             1 BYTE REPLY
044.230  315 253 050               927          CALL    RES             READ  ENCODED STRING
044.233  330                       928          RC                      CTL-D
044.234  312 305 044               929          JE      GPI2            IS OK
                                    930
                                    931  *       BAD PATCH ID
                                    932
044.237  315 136 031               933          CALL    $TYPTX
044.242  007 111 156               934          DB      BELL,'Invalid Patch ID. Try Again...',ENL
044.302  303 211 044               935          JMP     GPI
                                    936
                                    937  *       GOT GOOD ID
                                    938
044.305  072 071 054               939  GPI2    LDA     LINE            (A) = SEQUENCE BYTE
044.310  062 260 053               940          STA     PATCHID         SET PATCH ID CODE
044.313  247                       941          ANA     A               CLEAR CARRY
044.314  311                       942          RET
```

```
                                945  ***     GPC - GET PREREQ CODE.
                                946  *
                                947  *       GPC PROMPTS THE USER FOR THE ENTRY OF THE PREREQUESITE CODE.
                                948  *       THIS CODE CONSISTS OF 14 CHARACTERS, ENCODEING 5 BYTES OF
                                949  *       CODE, AND 2 BYTES OF CRC-16 FOR THAT CODE.
000.000                         950          ERRNZ   PATPRQL-5       DOCUMENTATION ASSUMES 5 BYTES OF PREREQ CODE
                                951  *
                                952  *       ENTRY   NONE
                                953  *       EXIT    'C' CLEAR IF OK
                                954  *               'C' SET IF CTL-D
                                955  *       USES    ALL
                                956
                                957
044.315  257                   958  GPC     XRA     A
044.316  062 214 045            959          STA     GPCC            FLAG MISSING NO PREREQUESITES
044.321  315 136 031            960          CALL    $TYPTX
044.324  120 162 145            961          DB      'Prerequesite Code?',' '+200Q
044.347  006 005                962          MVI     B,PATPRQL       SET NUMBER OF BYTES TO GET
044.351  315 253 050            963          CALL    RES             READ ENCODED STRING
044.354  330                    964          RC                      CTL-D
044.355  312 022 045            965          JZ      GPC1            CODE OK
                                966
                                967  *       ERROR IN CODE ENTRY
                                968
                                969
044.360  315 136 031            970          CALL    $TYPTX
044.363  007 111 154            971          DB      BELL,'Illegal Code. Try Again...',ENL
045.017  303 315 044            972          JMP     GPC             TRY AGAIN
                                973
                                974  *       GOT COOD CODE, COPY INTO TABLE, SEE IF ANY MISSING
                                975
045.022  021 071 054            976  GPC1    LXI     D,LINE
045.025  001 005 000            977          LXI     B,PATPRQL
045.030  041 253 053            978          LXI     H,PATPRQ
045.033  315 252 030            979          CALL    $MOVE           MOVE 5 BYTES INTO PATPRQ TABLE
                                980
                                981  *       SEE IF ANY PREREQS MISSING
                                982
045.036  041 253 053            983          LXI     H,PATPRQ
045.041  021 023 057            984          LXI     D,PHT+PHT.HIS
045.044  006 005                985          MVI     B,PATPRQL
045.046  032                    986  GPC3    LDAX    D               (A) = HISTORY BYTE
045.047  057                    987          CMA
045.050  246                    988          ANA     M               NON-ZERO IF ONE MISSING
045.051  312 136 045            989          JZ      GPC5            NOT MISSING IN THIS BYTE
                                990
                                991  *       MISSING A PATCH.
                                992  *       (A) = BIT INDEX OF MISSING PATCH
                                993  *       (B) = 5-BYTE INDEX OF MISSING PATCH BIT
                                994
045.054  365                   995          PUSH    PSW             SAVE BIT
045.055  076 005               996          MVI     A,5
045.057  220                   997          SUB     B               (A) = BYTE INDEX
045.060  207                   998          ADD     A
045.061  207                   999          ADD     A
045.062  207                  1000          ADD     A               (A) = 8*BYTE INDEX
```

```
045.063  117           1001          MOV     C,A             (C) = BIT INDEX OF LOW/ORDER BIT
045.064  361           1002          POP     PSW             (A) = BIT INDEX
045.065  014           1003  GPC4    INR     C
045.066  247           1004          ANA     A
045.067  312 136 045   1005          JZ      GPC5            NO MORE MISSING PATCHES
045.072  037           1006          RAR
045.073  322 065 045   1007          JNC     GPC4            THIS ONE NOT MISSING
                       1008
                       1009  *       (C) = # OF MISSING PATCH. REPORT IT
                       1010
045.076  365           1011          PUSH    PSW
045.077  305           1012          PUSH    B
045.100  325           1013          PUSH    D
045.101  345           1014          PUSH    H               SAVE ALL REGS
045.102  006 000       1015          MVI     B,0             (BC) = PATCH NUMBER +1
045.104  013           1016          DCX     B               CORRECT
045.105  041 176 045   1017          LXI     H,GPCB
045.110  076 003       1018          MVI     A,3
045.112  315 074 053   1019          CALL    $UDDN           UNPACK PATCH NUMBER
045.115  041 154 045   1020          LXI     H,GPCA
045.120  377 003       1021          DB      SYSCALL,.PRINT  FINK ON HIM
045.122  076 001       1022          MVI     A,1
045.124  062 214 045   1023          STA     GPCC            FLAG ONE MISSING
045.127  341           1024          POP     H
045.130  321           1025          POP     D
045.131  301           1026          POP     B
045.132  361           1027          POP     PSW             RESTORE REGS
045.133  303 065 045   1028          JMP     GPC4            TRY SOME MORE
                       1029
                       1030  *       NONE MISSING IN THIS BYTE. TRY NEXT
                       1031
045.136  023           1032  GPC5    INX     D
045.137  043           1033          INX     H
045.140  005           1034          DCR     B
045.141  302 046 045   1035          JNZ     GPC3            MORE TO TRY
                       1036
                       1037  *       DONE WITH CHECKS. SEE IF ANY MISSING
                       1038
045.144  072 214 045   1039          LDA     GPCC
045.147  247           1040          ANA     A
045.150  302 200 042   1041          JNZ     RESTART         IF ANY MISSING
045.153  311           1042          RET                     ALL OK
                       1043
045.154  007 122 145   1044  GPCA    DB      BELL,'Required Patch # '
045.176  060 060 060   1045  GPCB    DB      '000 missing.',BELL,ENL
                       1046
045.214  000           1047  GPCC    DB      00              <>0 IF PATCH MISSING
```

```
                                1050  ***     GPA - GET PATCH ADDRESS.
                                1051  *
                                1052  *       GPA GETS THE ADDRESS FOR THE NEXT ROUND OF PATCHES.
                                1053  *
                                1054  *       THE PATCH ADDRESS IS ENTERED IN THE PATCH LIST.
                                1055  *
                                1056  *       ENTRY   NONE
                                1057  *       EXIT    'C' CLEAR IF OK
                                1058  *                PATADR=ADDRESS
                                1059  *                'C' SET IF CTL-D
                                1060  *       USES    ALL
                                1061
                                1062
045.215  315 136 031            1063  GPA     CALL    $TYPTX
045.220  012 101 144            1064          DB      NL,'Address?',' '+200Q
045.232  041 077 054            1065          LXI     H,LINE+6
045.235  315 027 053            1066          CALL    $RTL.           READ LINE IN UPPER CASE
045.240  330                    1067          RC                      CTL-D
045.241  041 071 054            1068          LXI     H,LINE
045.244  076 006                1069          MVI     A,6
045.246  066 060                1070  GPA1    MVI     M,'0'           PUT IN 6 PRECEDING ZEROS
045.250  043                    1071          INX     H
045.251  075                    1072          DCR     A
045.252  302 246 045            1073          JNZ     GPA1
045.255  053                    1074          DCX     H
                                1075
                                1076  *       FIND LAST CHARACTER IN ENTERED LINE
                                1077
045.256  043                    1078  GPA2    INX     H
045.257  176                    1079          MOV     A,M
045.260  247                    1080          ANA     A
045.261  302 256 045            1081          JNZ     GPA2            FIND LAST CHARACTER
045.264  021 372 377            1082          LXI     D,-6
045.267  031                    1083          DAD     D               (HL) = ADDRESS OF LAST 6 DIGITS
045.270  315 373 047            1084          CALL    DOB             DECODE OCTAL BYTE
045.273  332 363 045            1085          JC      GPA3            ERROR
045.276  365                    1086          PUSH    PSW             SAVE VALUE
045.277  315 373 047            1087          CALL    DOB
045.302  332 363 045            1088          JC      GPA3            ERROR
045.305  321                    1089          POP     D               (D) = HIGH ORDER BYTE
045.306  137                    1090          MOV     E,A             (DE) = ADDRESS
                                1091
                                1092  *       SEE IF ADDRESS IS BEFORE FILE
                                1093
045.307  052 230 053            1094          LHLD    SKEW
045.312  315 224 030            1095          CALL    $CHL            (HL) = SUBTRACT FACTOR
045.315  173                    1096          MOV     A,E
045.316  225                    1097          SUB     L
045.317  172                    1098          MOV     A,D
045.320  234                    1099          SBB     H               'C' SET IF ADDRESS TOO LOW
045.321  353                    1100          XCHG                    (HL) = ADDRESS
045.322  042 233 053            1101          SHLD    CPA             SET CURRENT PATCH ADDRESS
045.325  320                    1102          RNC                     IS OK, EXIT
                                1103
                                1104  *       PATCH ADDRESS TOO LOW
                                1105
```

```
045.326  315 136 031  1106          CALL    $TYPTX
045.331  007 120 141  1107          DB      BELL,'Patch Address Too Low',ENL
045.360  303 215 045  1108          JMP     GPA             TRY AGAIN
                      1109
                      1110  *       ERROR IN INPUT
                      1111
045.363  315 136 031  1112  GPA3    CALL    $TYPTX
045.366  007 111 154  1113          DB      BELL,'Illegal Address Value.',ENL
046.016  303 215 045  1114          JMP     GPA             TRY AGAIN
```

```
                                1117  ***     APP - ACCEPT PROGRAM PATCHES.
                                1118  *
                                1119  *       APP READS THE PROGRAM PATCHES FROM THE CONSOLE,
                                1120  *       STARTING AT THE CURRENT PATCH ADDRESS, AND INCREMENTING BY
                                1121  *       ONE BYTE EACH TIME, APP PROMPTS
                                1122  *
                                1123  *       AAAAAA = VVV/
                                1124  *
                                1125  *       WHERE 'AAAAAA' = ADDRESS, 'VVV' = OLD VALUE, THE USER MAY
                                1126  *       THEN TYPE
                                1127  *
                                1128  *       NNN             NEW VALUE
                                1129  *       CR              LEAVE THE SAME
                                1130  *       CTL-D           START NEW ADDRESS
                                1131  *
                                1132  *       THE VALUES RECEIVED ARE ENTERED IN THE PATCH LIST.
                                1133  *
                                1134  *       ENTRY   NONE
                                1135  *       EXIT    WHEN CTL-D HIT
                                1136  *       USES    ALL
                                1137
                                1138
046.021                         1139  APP     EQU     *
046.021  052 233 053            1140          LHLD    CPA             (HL) = DESIRED BYTE ADDRESS
046.024  315 111 050            1141          CALL    LBF             LOCATE BYTE IN TILE
046.027  353                    1142          XCHG                    (DE) = ADDRESS IN VIEWBFR
                                1143
                                1144  *       GOT CURRENT VALUE, DISPLAY IT
                                1145
046.030  052 233 053            1146  APP4    LHLD    CPA
046.033  174                    1147          MOV     A,H
046.034  315 147 053            1148          CALL    $TOD            TYPE OCTAL DIGITS
046.037  175                    1149          MOV     A,L
046.040  315 147 053            1150          CALL    $TOD
046.043  315 136 031            1151          CALL    $TYPTX
046.046  040 075 240            1152          DB      ' =',' '+200Q
046.051  032                    1153          LDAX    D               (A) = OLD VALUE
046.052  315 147 053            1154          CALL    $TOD
046.055  315 136 031            1155          CALL    $TYPTX
046.060  257                    1156          DB      '/'+200Q
                                1157
                                1158  *       ACCEPT NEW VALUE
                                1159
046.061  325                    1160          PUSH    D               SAVE ADDRESS OF CURRENT VALUE
046.062  041 074 054            1161          LXI     H,LINE+3
046.065  315 027 053            1162          CALL    $RTL.           READ IN UPPER CASE
046.070  321                    1163          POP     D
046.071  330                    1164          RC                      CTL-D
046.072  072 074 054            1165          LDA     LINE+3
046.075  247                    1166          ANA     A
046.076  032                    1167          LDAX    D               (A) = CURRENT VALUE
046.077  312 132 046            1168          JZ      APP6            LEFT TO CURRENT VALUE
                                1169
                                1170  *       GOT THE NEW VALUE
                                1171
046.102  076 060                1172          MVI     A,'0'
```

```
046.104  041 071 054  1173          LXI     H,LINE          PUT 2 LEADING '0'S BEFORE
046.107  167         1174          MOV     M,A
046.110  043         1175          INX     H
046.111  167         1176          MOV     M,A
046.112  043         1177  APP5    INX     H
046.113  176         1178          MOV     A,M
046.114  247         1179          ANA     A               LOOK FOR END OF VALUE
046.115  302 112 046 1180          JNZ     APP5            NOT AT END OF LINE
046.120  021 375 377 1181          LXI     D,-3
046.123  031         1182          DAD     D               (HL) = FWA OF 3 DIGIT VALUE
046.124  315 373 047 1183          CALL    DOB             DECODE OCTAL BYTES
046.127  332 165 046 1184          JC      APP7            ERROR
                     1185
                     1186  *       ADD VALUE TO PATCH LIST
                     1187  *
                     1188  *       (A) = NEW VALUE
                     1189
046.132  365         1190  APPA    PUSH    PSW             SAVE VALULE
046.133  072 233 053 1191          LDA     CPA
046.136  315 170 047 1192          CALL    ABL             INSERT ADDRESS IN LIST
046.141  072 234 053 1193          LDA     CPA+1
046.144  315 170 047 1194          CALL    ABL
046.147  361         1195          POP     PSW
046.150  315 170 047 1196          CALL    ABL             ADD BYTE TO LIST
046.153  052 233 053 1197          LHLD    CPA
046.156  043         1198          INX     H
046.157  042 233 053 1199          SHLD    CPA             INCREMENT ADDRESS
046.162  303 021 046 1200          JMP     APP             GET ANOTHER
                     1201
                     1202  *       BAD CALL
                     1203
046.165  315 136 031 1204  APP7    CALL    $TYPTX
046.170  007 111 154 1205          DB      BELL,'Illegal Value',ENL
046.207  303 021 046 1206          JMP     APP             TRY AGAIN
```

```
                                1210  ***     CPC - CHECK PATCH CRC.
                                1211  *
                                1212  *       CPC COMPUTES A CRC-16 FOR THE ENTIRE PATCH SERIES, ENCLUDING
                                1213  *       THE SERIAL NUMBER, PREREQUESTIE LIST, AND ALL ADDRESSES AND
                                1214  *       VALUES, AND THEN CHECKS THAT CRC AGAINST THE ONE ENTERED
                                1215  *       BY THE USER.
                                1216  *
                                1217  *       ENTRY   NONE
                                1218  *       EXIT    'C' CLEAR IF OK
                                1219  *               'C' SET IF ERROR
                                1220  *       USES    ALL
                                1221
                                1222
046.212  315 136 031            1223  CPC     CALL    $TYPTX
046.215  012 120 141            1224          DB      NL,'Patch Check Code?',' '+200Q
046.240  021 000 060            1225          LXI     D,PATLIST
046.243  052 246 053            1226          LHLD    PLPTR
046.246  175                    1227          MOV     A,L
046.247  223                    1228          SUB     E
046.250  117                    1229          MOV     C,A             (BC) = LENGTH OF VALUES IN PATLIST
046.251  174                    1230          MOV     A,H
046.252  232                    1231          SBB     D
046.253  107                    1232          MOV     B,A
046.254  315 335 047            1233          CALL    CBS             CRC BYTE STRING
046.257  042 077 047            1234          SHLD    CPCA            STORE EXPECTED CRC
000.001                         1235          IF      ,SYS.
                                1236
                                1237  *       TYPE THE EXPECTED CRC, AS A SUBTLE HINT!
                                1238
                                1239          LDA     CHECKC
                                1240          ANA     A
                                1241          JZ      CPC1            DONT SHOW IT, HE'S PLAYING DUMB
                                1242          CALL    $TYPTX
                                1243          DB      'Expecting:',' '+200Q
                                1244          LDA     CPCA
                                1245          CALL    $TOD
                                1246          LDA     CPCA+1
                                1247          CALL    $TOD
                                1248          CALL    $TYPTX
                                1249          DB      ENL
                                1250  CPC1    EQU     *
                                1251          ENDIF
046.262  006 002                1252          MVI     B,2
046.264  315 253 050            1253          CALL    RES             READ ENCRYPTED STRING
046.267  330                    1254          RC                      CTL-D
046.270  312 355 046            1255          JE      CPC2            IS OK
046.273  315 136 031            1256          CALL    $TYPTX
046.276  007 103 150            1257          DB      BELL,'Check Code Entered Incorrectly. Try Again.',ENL
046.352  303 212 046            1258          JMP     CPC             TRY AGAIN
                                1259
                                1260  *       GOT GOOD CHECK CODE. SEE IF MATCHES
                                1261
046.355  052 071 054            1262  CPC2    LHLD    LINE
046.360  353                    1263          XCHG
046.361  052 077 047            1264          LHLD    CPCA
046.364  315 216 030            1265          CALL    $CDEHL          COMPARE
```

```
046.367  310           1266          RE                    EVERYTHING OK
046.370  315 136 031   1267          CALL    $TYPTX
046.373  007 103 150   1268          DB      BELL,'Check Code Does Not Match Patch'
047.033  012 105 162   1269          DB      NL,'Error In Patch Entry. Try Again.',ENL
047.075  067           1270          STC
047.076  311           1271          RET
                       1272
047.077  000 000       1273  CPCA    DW      0               TEMP STORE FOR COMPUTED CRC
```

```
                                1277  ***     EPF - ENTER PATCHES IN FILE.
                                1278  *
                                1279  *       EPF IS CALLED TO ENTER THE PATCHES STORED IN PATLIST INTO
                                1280  *       THE FILE.
                                1281  *
                                1282  *       IF WE ARE IN USER MODE, THE FILE MUST ALREADY BE OPEN FOR UPDATE.
                                1283  *       IF WE ARE IN SYSTEM MODE, EPF WILL KLUDGE THE FILE OPEN.
                                1284  *
                                1285  *       ENTRY   NONE
                                1286  *       EXIT    NONE
                                1287  *       USES    ALL
                                1288
                                1289
047.101                         1290  EPF     EQU     *
047.101  072 261 053            1291          LDA     PHIST
047.104  247                    1292          ANA     A
047.105  304 354 047            1293          CNZ     CFU             CLUDGE FILE TO UPDATE STATUS
                                1294
                                1295  *       PUT IN PATCHES, A BYTE AT A TIME
                                1296
047.110  021 000 060            1297  EPF1    LXI     D,PATLIST
047.113  052 264 053            1298          LHLD    PGMLWA
047.116  104                    1299          MOV     B,H
047.117  115                    1300          MOV     C,L             (BC) = CURRENT LWA
047.120  052 246 053            1301  EPF2    LHLD    PLPTR
047.123  315 216 030            1302          CALL    $CDEHL          SEE IF ALL IN
047.126  310                    1303          RE                      ALL IN
                                1304
                                1305  *       GOT ANOTHER PATCH. INSTALL IT
                                1306
047.127  032                    1307          LDAX    D
047.130  157                    1308          MOV     L,A
047.131  023                    1309          INX     D
047.132  032                    1310          LDAX    D
047.133  147                    1311          MOV     H,A             (HL) = ADDRESS
047.134  023                    1312          INX     D
                                1313
                                1314  *       SEE IF EXTENDING PROGRAM LWA
                                1315
047.135  171                    1316          MOV     A,C
047.136  225                    1317          SUB     L
047.137  170                    1318          MOV     A,B
047.140  234                    1319          SBB     H
047.141  322 151 047            1320          JNC     EPF3            NOT EXTENDING
047.144  104                    1321          MOV     B,H
047.145  115                    1322          MOV     C,L             UPDATE (BC)
047.146  042 264 053            1323          SHLD    PGMLWA          SET NEW LENGTH
047.151  315 111 050            1324  EPF3    CALL    LBF             LOCATE BYTE IN FILE
047.154  032                    1325          LDAX    D
047.155  023                    1326          INX     D
047.156  167                    1327          MOV     M,A             PATCH SECTOR
047.157  076 001                1328          MVI     A,1
047.161  062 237 053            1329          STA     SIVBA           FLAG SECTOR AALTERED
047.164  043                    1330          INX     H
047.165  303 120 047            1331          JMP     EPF2            GET NEXT PATCH
```

```
                        1335  **      APF - ADD PHT TO FILE.
                        1336  *
                        1337  *       APF IS CALLED TO ADD A PHT TO A FILE.
                        1338  *
                        1339  *       THE PROPER PHT SECTOR ADDRESS IS COMPUTED, THE FILE IS KLUDGED OPEN
                        1340  *       FOR UPDATE, AND A BLANK PHT IS ADDED.
                        1341  *
                        1342  *       ENTRY   NONE
                        1343  *       EXIT    TO RESTART
                        1344  *       USES    ALL
                        1345
                        1346
     000.001            1347          IF      .SYS.
                        1348  APF     CALL    CFU             KLUDGE FILE FOR UPDATE
                        1349          LXI     B,PHTL
                        1350          LXI     D,PHTFORM
                        1351          LXI     H,PHT
                        1352          CALL    $MOVE           MOVE IN ARCHTYPE PHT FORM
                        1353          LXI     B,9
                        1354          LXI     D,S.DATE
                        1355          LXI     H,PHT+PHT.DAT
                        1356          CALL    $MOVE           MOVE IN DATE
                        1357          CALL    WPH.            ADD PHT
                        1358          MVI     A,CN.FIL
                        1359          DB      SYSCALL,.CLOSE  CLOSE IT
                        1360          JMP     RESTART         TRY AGAIN
                        1361          ENDIF
```

```
                                1365  **      ABL - ADD BYTE TO LIST.
                                1366  *
                                1367  *       ABL ADDS A SINGLE BYTE TO THE PATCH LIST (PATLIST).
                                1368  *
                                1369  *       ENTRY   (A) = BYTE
                                1370  *       EXIT    NONE
                                1371  *       USES    A,F,D,E,H,L
                                1372
                                1373
047.170  052 246 053           1374  ABL     LHLD    PLPTR
047.173  167                   1375          MOV     M,A             STORE IN LIST
047.174  043                   1376          INX     H
047.175  042 246 053           1377          SHLD    PLPTR           UPDATE COUNT
047.200  353                   1378          XCHG
047.201  052 250 053           1379          LHLD    PLMAX           SEE IF OVERFLOW SOON OR NOW
047.204  045                   1380          DCR     H               ALLOW 256 BYTE WARNING
047.205  173                   1381          MOV     A,E
047.206  225                   1382          SUB     L
047.207  172                   1383          MOV     A,D
047.210  234                   1384          SBB     H
047.211  330                   1385          RC                      PLENTY OF ROOM
047.212  247                   1386          ANA     A
047.213  302 273 047           1387          JNZ     ABL1            CLEAN OUT OF ROOM!
                               1388
                               1389  *       RUNNING OUT OF ROOM, WARN HIM BEFORE ITS TOO LATE!
                               1390
047.216  315 136 031           1391          CALL    $TYPTX
047.221  007 122 165           1392          DB      BELL,'Running low on space. Please finish up!',ENL
047.272  311                   1393          RET
                               1394
                               1395  *       OUT OF ROOM
                               1396
047.273  315 136 031           1397  ABL1    CALL    $TYPTX
047.276  007 117 165           1398          DB      BELL,'Out of RAM Space!',ENL
047.321  303 200 042           1399          JMP     RESTART


                               1401  **      AEE - ALLOW EOF ERROR.
                               1402  *
                               1403  *       AEE IS CALLED IMMEDIATELY AFTER A SYSTEM CALL, TO CHECK
                               1404  *       FOR THE TYPE OF A RETURNED ERROR.  IF THERE IS NO ERROR, OR THE
                               1405  *       ERROR TYPE IS 'EC.EOF', THEN AEE RETURNS TO THE CALLER.
                               1406  *       IF THE ERROR WAS NOT EOF, THEN AEE JUMPS TO *ERROR*
                               1407  *
                               1408  *       ENTRY   'C' SET IF ERROR
                               1409  *                (A) = ERROR CODE
                               1410  *       EXIT    TO *ERROR* IF ERROR <> EC.EOF
                               1411  *               TO CALLER IF ERROR = EC.EOF
                               1412  *               TO CALLER IF NO ERROR
                               1413  *       USES    NONE
                               1414
                               1415
047.324  320                   1416  AEE     RNC                     NO ERROR
047.325  365                   1417          PUSH    PSW             SAVE PSW
```

```
047.326  376 001       1418          CPI     EC.EOF
047.330  302 324 042   1419          JNE     ERROR           NOT RIGHT TYPE OF ERROR
047.333  361           1420          POP     PSW
047.334  311           1421          RET                     RETURN WITH EOF ERROR


                       1423  **      CBS - CRC BYTE STRING.
                       1424  *
                       1425  *       CBS COMPUTES THE CRC OF A STRING OF MEMORY BYTES.
                       1426  *       THE CRC IS COMPUTED BASED ON A PREFIX OF 377377A.
                       1427  *
                       1428  *       * * WARNING * * :THIS ROUTINE IS NOT ESPECIALLY COMPATIBLE
                       1429  *       WITH OTHER CRC GENERATING ROUTINES. ITS CRC SHOULD BE COMPARED
                       1430  *       ONLY WITH OTHER CRC'S GENERATED BY *CBS*, AND
                       1431  *       NO OTHER ROUTINES.
                       1432  *
                       1433  *       ENTRY   (DE) = ADDRESS OF STRING
                       1434  *               (BC) = COUNT
                       1435  *       EXIT    (HL) = NEW CRC VALUE
                       1436  *               (DE) = (DE) + (BC)
                       1437  *       USES    ALL
                       1438
                       1439
047.335  041 377 377   1440  CBS     LXI     H,377377A
047.340  170           1441  CBS1    MOV     A,B
047.341  261           1442          ORA     C
047.342  310           1443          RZ                      ALL DONE
047.343  032           1444          LDAX    D
047.344  315 316 052   1445          CALL    $CRC            ADD TO CRC
047.347  023           1446          INX     D
047.350  013           1447          DCX     B
047.351  303 340 047   1448          JMP     CBS1            DO NEXT ONE


                       1450  **      CFU - KLUDGE FILE TO UPDATE STATUS.
                       1451  *
                       1452  *       ********
                       1453  *       *      *
                       1454  *       * NOTE *
                       1455  *       *      *
                       1456  *       ********
                       1457  *
                       1458  *       FOR PROTECTION, SYSTEM FILES (WHICH PROBABLY HAVE THE WRITE-PROTECT)
                       1459  *       BIT SET ARE OPENED ONLY FOR READ ACCESS (THEY CANNOT BE OPENED FOR
                       1460  *       UPDATE ACCESS, IF WRITE PROTECTED)
                       1461  *
                       1462  *       CFU SCREWS AROUND WITH THE CANNEL AND SETS THE APPROPRIATE
                       1463  *       BITS TO ALLOW READ/WRITE (I.E., UPDATE) ACCESS.
                       1464  *
                       1465  *       ENTRY   FILE OPEN ON CN,FIL
                       1466  *       EXIT    NONE
                       1467  *       USES    A,F,D,E,H,L
```

```
                                1468
                                1469
047.354  052 352 040  1470  CFU   LHLD   S.CFWA
000.000               1471        ERRNZ  CN.FIL
047.357  315 211 030  1472        CALL   $HLIHL          (HL) = CHANNEL FWA
047.362  021 004 000  1473        LXI    D,IOC.FLG
047.365  031         1474        DAD    D               (HL) = ADDRESS OF IOC.FLG BYTE
047.366  076 014      1475        MVI    A,FT.OW+FT.OU
047.370  266         1476        ORA    M
047.371  167         1477        MOV    M,A             CHANGE FLAGS TO 'OPEN FOR UPDATE'
047.372  311         1478        RET                    EXIT


                                1480  **    DOB - DECODE OCTAL BYTE.
                                1481  *
                                1482  *     DOB DECODES A THREE DIGIT OCTAL BYTE INTO BINARY.
                                1483  *
                                1484  *     ENTRY  (HL) = ADDRESS OF THREE DIGITS
                                1485  *     EXIT   'C' CLEAR IF OK
                                1486  *             (HL) = (HL)+3
                                1487  *             (A) = VALUE
                                1488  *            'C' SET IF ERROR
                                1489  *     USES   A,F,E,H,L
                                1490
                                1491
047.373  021 003 000  1492  DOB   LXI    D,3             (E) = DIGIT COUNT, (D) = ACCUM
047.376  172         1493  DOB1  MOV    A,D
047.377  207         1494        ADD    A
050.000  207         1495        ADD    A
050.001  207         1496        ADD    A               (A) = ACCUM*8
050.002  127         1497        MOV    D,A
050.003  176         1498        MOV    A,M             (A) = DIGIT
050.004  043         1499        INX    H
050.005  326 060      1500        SUI    '0'
050.007  330         1501        RC                     ERROR
050.010  376 010      1502        CPI    8
050.012  077         1503        CMC
050.013  330         1504        RC                     ERROR
050.014  202         1505        ADD    D               ADD TO ACCUM
050.015  127         1506        MOV    D,A
050.016  035         1507        DCR    E
050.017  302 376 047  1508        JNZ    DOB1            GET ANOTHER DIGIT
050.022  311         1509        RET


                                1511  **    FVB - FLUSH VIEW BUFFER.
                                1512  *
                                1513  *     FVB FLUSHES THE CONTENTS OF THE VIEW BUFFER OUT TO THE DISK FILE,
                                1514  *     IF THEY HAVE BEEN CHANGED (SIVBA <> 0).
                                1515  *
                                1516  *     THE FILE MUST BE ALREADY OPEN FOR UPDATE.
                                1517  *
```

```
                                1518  *       ENTRY   NONE
                                1519  *       EXIT    NONE
                                1520  *       USES    ALL
                                1521
                                1522
050.023  072 237 053            1523  FVB     LDA     SIVBA
050.026  247                    1524          ANA     A
050.027  310                    1525          RZ                      NOT ALTERED
050.030  072 235 053            1526          LDA     SIVB
050.033  117                    1527          MOV     C,A
050.034  006 000                1528          MVI     B,0             (BC) = SECTOR NUMBER
050.036  315 235 050            1529          CALL    PPF             POSITION OVER IT
050.041  315 324 047            1530          CALL    AEE             ALLOW EOF ERROR, ONLY
050.044  322 067 050            1531          JNC     FVB1            GOT THERE
                                1532
                                1533  *       COULDNT POSITION THERE.  MUST BE EXTENDING THE FILE.  WE'LL
                                1534  *       WRITE GARBAGE ENOUGH TO DO THE EXTEND...
                                1535
050.047  101                    1536          MOV     B,C
050.050  016 000                1537          MVI     C,0             (BC) = BYTES NEEDED TO EXTEND
050.052  021 000 004            1538          LXI     D,4000A         (DE) = ADDRESS OF GARBAGE (PROBABLY 0'S)
050.055  076 000                1539          MVI     A,CN.FIL
050.057  377 005                1540          DB      SYSCALL,.WRITE  WRITE IT
050.061  332 324 042            1541          JC      ERROR           ERROR ON WRITE
050.064  303 023 050            1542          JMP     FVB             NOW TRY IT
                                1543
                                1544  *       GOT THERE, WILL WRITE REPLACEMENT SECTOR
                                1545
050.067  001 000 001            1546  FVB1    LXI     B,256
050.072  021 000 055            1547          LXI     D,VIEWBFR
050.075  076 000                1548          MVI     A,CN.FIL
050.077  377 005                1549          DB      SYSCALL,.WRITE  WRITE IT
050.101  332 324 042            1550          JC      ERROR           ERROR
050.104  257                    1551          XRA     A
050.105  062 237 053            1552          STA     SIVBA           CLEAR ALTERED FLAG
050.110  311                    1553          RET                     EXIT


                                1555  **      LBF - LOCATE BYTE IN FILE.
                                1556  *
                                1557  *       LBF LOCATES A BYTEE IN THE USER PROGRAM FILE, AND BRINGS THE
                                1558  *       SECTOR CONTAINING IT INTO VIEWBFR.
                                1559  *
                                1560  *       IF A NEW SECTOR IS NEEDED, THE VIEWBFR IS
                                1561  *       FLUSHED BACK TO THE DISK (IF NECESSARY) FIRST.
                                1562  *
                                1563  *       IF THE REQUIRED BYTE CANNOT BE REACHED (OFF THE END OF THE FILE)
                                1564  *       THEN A SECTOR FULL OF ZEROS IS 'IMAGINED' FOR THE OCCASION.
                                1565  *       WHEN THIS SECTOR IS FLUSHED BACK TO THE DISK (VIA FVB)
                                1566  *       THE DISK FILE WILL BE EXTENDED FOR IT.
                                1567  *
                                1568  *       ENTRY   (HL) = USER PROGRAM ADDRESS FOR BYTE
                                1569  *       EXIT    (HL) = ADDRESS IN VIEWBFR OF VALUE
                                1570  *       USES    A,F,H,L
```

```
                         1571
                         1572
050.111  305            1573  LBF     PUSH    B
050.112  325            1574          PUSH    D               SAVE REGISTERS
050.113  353            1575          XCHG
050.114  052 230 053     1576          LHLD    SKEW            (HL) = ADDRESS TO FILE INDEX FACTOR
050.117  031            1577          DAD     D               (H) = SECTOR, (L) = INDEX IN SECTOR
050.120  353            1578          XCHG                    (DE) = FILE POINTER
050.121  325            1579          PUSH    D               SAVE FOR LATER USE
050.122  052 235 053     1580          LHLD    SIVB            (L) = SECTOR # IN VIEWBFR
000.000                  1581          ERRNZ   SIVBV-SIVB-1    (H) <> 0 IF VIEWBFR VALID
050.125  174            1582          MOV     A,H
050.126  247            1583          ANA     A
050.127  312 150 050     1584          JZ      LBF4            ISNT VALID, SO JUST READ A NEW ONE
050.132  172            1585          MOV     A,D
050.133  275            1586          CMP     L               SEE IF ONE WE WANT
050.134  302 145 050     1587          JNE     LBF1            NOT RIGHT ONE
050.137  341            1588          POP     H
050.140  046 055        1589          MVI     H,VIEWBFR/256   (HL) = ADDRESS
050.142  321            1590          POP     D               RESTORE REGS
050.143  301            1591          POP     B
050.144  311            1592          RET                     EXIT WITH BYTE
                         1593
                         1594  *       WILL HAVE TO REPLACE SECTOR IN BFR, FLUSH IT
                         1595
050.145  315 023 050     1596  LBF1    CALL    FVB             FLUSH VIEW BUFFER
                         1597
                         1598  *       READ IN THE SECTOR CONTAINING THE BYTE.
                         1599
050.150  321            1600  LBF4    POP     D               (D) = SECTOR NEEDED
050.151  325            1601          PUSH    D
050.152  112            1602          MOV     C,D
050.153  006 000        1603          MVI     B,0
050.155  315 235 050     1604          CALL    FPF             POSITION TO IT
050.160  315 324 047     1605          CALL    AEE             ALLOW EOF ERROR, ONLY
050.163  332 206 050     1606          JC      LBF5            IS EOF
050.166  001 000 001     1607          LXI     B,256
050.171  021 000 055     1608          LXI     D,VIEWBFR
050.174  076 000        1609          MVI     A,CN.FIL
050.176  377 004        1610          DB      SYSCALL,.READ           READ IT
050.200  315 324 047     1611          CALL    AEE             ALLOW EOF ERROR, ONLY
050.203  322 216 050     1612          JNC     LBF6            GOT THE SECTOR
                         1613
                         1614  *       SECTOR IS OFF END OF FILE. MAKE ONE UP OF ALL ZEROS.
                         1615
050.206  006 000        1616  LBF5    MVI     B,0
050.210  041 000 055     1617          LXI     H,VIEWBFR
050.213  315 212 031     1618          CALL    $ZERO           ZERO IT
                         1619
                         1620  *       BUFFER IS READ IN. SETUP DESCRIPTOR CELLS.
                         1621
050.216  341            1622  LBF6    POP     H               (H) = SECTOR, (L) = INDEX
050.217  174            1623          MOV     A,H
050.220  062 235 053     1624          STA     SIVB            SET SECTOR IN VIEWBFR
050.223  076 001        1625          MVI     A,1
050.225  062 236 053     1626          STA     SIVBV           SET VIEWBFR VALID
```

```
050.230  046 055      1627          MVI     H,VIEWBFR/256   (HL) = ADDRESS IN VIEWBFR
050.232  321          1628          POP     D               RESTORE REGISTERS
050.233  301          1629          POP     B
050.234  311          1630          RET


                      1632  **      PPF - POSITION PROGRAM FILE.
                      1633  *
                      1634  *       PPF IS CALLED TO POSITION THE PROGRAM FILE IMMEDIATELY BEFORE
                      1635  *       A GIVEN SECTOR.
                      1636  *
                      1637  *       ENTRY   (BC) = SECTOR NUMBER
                      1638  *       EXIT    'C' SET IF ERROR
                      1639  *                       (A) = ERROR CODE
                      1640  *               'C' CLEAR IF OK
                      1641  *       USES    ALL
                      1642
                      1643
050.235  072 232 053  1644  PPF     LDA     SDISP           (A) = SECTORS PROGRAM IS DISPLACED
050.240  201          1645          ADD     C
050.241  117          1646          MOV     C,A
050.242  076 000      1647          MVI     A,0
050.244  210          1648          ADC     B
050.245  107          1649          MOV     B,A
050.246  076 000      1650          MVI     A,CN.FIL
050.250  377 047      1651          DB      SYSCALL,.POSIT
050.252  311          1652          RET


                      1654  **      RES - READ ENCODED STRING.
                      1655  *
                      1656  *       RES READS AN ENCODED CHARACTER STRING.
                      1657  *
                      1658  *       MANY OF THE USER ENTRYS ARE ENCODED, TO MAKE THEM HARD TO UNDERSTAND.
                      1659  *       THIS ENCOURAGES THE USER TO ENTER THEM CAREFULLY, AND DISCOURAGES
                      1660  *       THE USER FROM OMITTING THINGS BECAUSE HE'S LAZY, OR DOSENT THINK
                      1661  *       THAT THEY'RE NEEDED.
                      1662  *
                      1663  *       A N BYTE STRING IS ENCODED AS N*2+4 CHARACTERS.
                      1664  *
                      1665  *       THE STRING IS DECODED BY DECREMENTING THE ASCII CHARACTER BY 1,
                      1666  *        AND XOR'ING IT AGAINST A CHARACTER PATTERN
                      1667  *       (RESA). THIS STRING IS REPEATED EVERY 32 CHARACTERS UNTIL
                      1668  *       THE ENTIRE ENTRY IS XOR'ED, THEN, THE LOW 4 BITS OF EACH
                      1669  *       CHARACTER ARE TAKEN, AND COMPRESSED TO FORM N+2 BYTES. THE 'N'
                      1670  *       BYTES ARE THE VALUE, THE 2 END BYTES ARE A CRC-16 OF THE N BYTES.
                      1671  *       THUS, THE ENTRY CAN BE CHECKED FOR INTERNAL CONSISTANCY.
                      1672  *
                      1673  *       ENTRY   (B) = N (NUMBER OF BYTES)
                      1674  *       EXIT    'C' SET IF CTL-D STRUCK
                      1675  *               'C' CLEAR IF ENTRY MADE
                      1676  *                'Z' SET IF ENTRY VALID
```

```
                                1677  *                (LINE, LINE+N-1 = VALUE BYTES)
                                1678  *                'Z' CLEAR IF ENTRY BAD
                                1679  *       USES     ALL
                                1680
                                1681
050.253                         1682  RES      EQU      *
000.001                         1683           IF       .SYS.
                                1684           LDA      CHECKC
                                1685           ANA      A
                                1686           CNZ      EES             ENCODE ENCRPYTED STRING FOR WIZARDS
                                1687           ENDIF
050.253  041 071 054            1688           LXI      H,LINE
050.256  315 027 053            1689           CALL     $RTL.           READ LINE, MAP TO UPPER CASE
050.261  330                    1690           RC                       CTL-D
050.262  170                    1691           MOV      A,B             (A) = N
050.263  306 002                1692           ADI      2               +2 FOR CRC
050.265  107                    1693           MOV      B,A
050.266  305                    1694           PUSH     B               SAVE N+2
050.267  001 071 054            1695           LXI      B,LINE          (BC) = TARGET FOR DECODED AND COMPRESSED STRING
050.272  353                    1696           XCHG                     (DE) = ENTERED STRING
050.273  041 364 050            1697           LXI      H,RESA          (HL) = XOR STRING
                                1698
                                1699  *       XOR STRING
                                1700
050.276  365                    1701  RES1     PUSH     PSW             SAVE REMAINING COUNT
050.277  032                    1702           LDAX     D
050.300  075                    1703           DCR      A               DECREMENT
050.301  256                    1704           XRA      M
050.302  346 017                1705           ANI      17Q
050.304  207                    1706           ADD      A               MOVE FIRST NIBBLE LEFT
050.305  207                    1707           ADD      A
050.306  207                    1708           ADD      A
050.307  207                    1709           ADD      A
050.310  365                    1710           PUSH     PSW
050.311  023                    1711           INX      D
050.312  043                    1712           INX      H
050.313  032                    1713           LDAX     D
050.314  075                    1714           DCR      A               DECREMENT
050.315  256                    1715           XRA      M
050.316  346 017                1716           ANI      17Q
050.320  343                    1717           XTHL                     (H) = FIRST NIBBLE
050.321  204                    1718           ADD      H               (A) = FULL BYTE VALUE
050.322  341                    1719           POP      H               (HL) = ENTERED LINE POINTER
050.323  002                    1720           STAX     B               STORE IN LINE
050.324  003                    1721           INX      B
050.325  023                    1722           INX      D
050.326  043                    1723           INX      H
050.327  175                    1724           MOV      A,L
050.330  376 024                1725           CPI      #RESAE
050.332  302 340 050            1726           JNE      RES2            STRING NOT USED UP, YET
050.335  041 364 050            1727           LXI      H,RESA          START OVER
050.340  361                    1728  RES2     POP      PSW             (A) = COUNT LEFT
050.341  075                    1729           DCR      A
050.342  302 276 050            1730           JNZ      RES1
                                1731
                                1732  *       CRC VALUE TO SEE IF OK
```

```
                                1733
050.345  021 071 054           1734            LXI     D,LINE
050.350  301                   1735            POP     B               (B) = N+2
050.351  110                   1736            MOV     C,B
050.352  006 000               1737            MVI     B,0             (BC) = N+2
050.354  315 335 047           1738            CALL    CBS             CRC BYTE STRING
050.357  174                   1739            MOV     A,H
050.360  265                   1740            ORA     L               CRC OF STRING AND CRC MUST BE 0
050.361  311                   1741            RET
                                1742
050.362  012 014               1743            DB      NL,FF           SHOW MESSAGE TO SNOOPS
                                1744  **        * WARNING: THIS MESSAGE MUST BE AN EVEN NUMBER OF BYTES * *
050.364  110 105 101           1745  RESA      DB      'HEATH SOFTWARE PATCH UTILITY   '
051.024  014                   1746  RESAE     DB      FF              END ADDRESS+1 OF MESSAGE


                                1748  **        EES - ENCODE ENCODED STRING.
                                1749  *
                                1750  *         IF THE 'C' OPTION IS SELECTED IN '.SYS.' VERSIONS
                                1751  *         OF PATCH, THEN EES IS CALLED BEFORE RES.
                                1752  *
                                1753  *         EES ACCEPTS A VALUE FROM THE CONSOLE, IN THE FORM
                                1754  *         OF OCTAL BYTES, ALL OF WHICH MUST BE 3 DIGITS, BUT
                                1755  *         WHICH MAY BE SEPERATED BY BLANKS.
                                1756  *
                                1757  *         THESE N BYTES WILL BE ENCRYPTED INTO THE 2*N+4 BYTE ALPHA STRING
                                1758  *         RES REQUIRES, AND THAT STRING WILL BE TYPED FOR THE USER.
                                1759  *
                                1760  *         ENTRY   (B) = NUMBER OF BYTES WANTED
                                1761  *         EXIT    NONE
                                1762  *         USES    NONE
                                1763
000.001                         1764            IF      .SYS.
                                1765
                                1766  EES       CALL    $SAVALL         SAVE REGISTERS
                                1767  EES0      CALL    $TYPTX
                                1768            DB      NL,'Enter Value as Octal Bytes:',' '+2000
                                1769            LXI     H,LINE
                                1770            CALL    $RTL.
                                1771            JC      $RSTALL         RESTORE ALL AND EXIT, CTL-D
                                1772            LXI     D,EESA
                                1773            MOV     C,B             (C) = BYTES WANTED
                                1774  EES1      MOV     A,C
                                1775            ANA     A
                                1776            JZ      EES3            ALL DONE INPUTTING, TYPE VALUE
                                1777            PUSH    B
                                1778            PUSH    D
                                1779            CALL    $SOB
                                1780            CALL    DOB             DECODE OCTAL BYTE
                                1781            POP     D
                                1782            POP     B
                                1783            JC      EES0            TRY AGAIN
                                1784            STAX    D               STORE VALUE
                                1785            INX     D               INCR ADDRESS
                                1786            DCR     C
```

```
1787            JMP     EES1
1788
1789  *         GOT VALUES. CRC IT
1790
1791  EES3      LXI     D,EESA
1792            MOV     C,B
1793            PUSH    B               SAVE COUNT
1794            MVI     B,0             (BC) = BYTE COUNT
1795            CALL    CBS             CRC BYTE STRING
1796            XCHG
1797            MOV     M,D
1798            INX     H
1799            MOV     M,E             STORE CRC
1800            POP     B
1801
1802  *         TYPE ENCRYPTED STRING.
1803  *         (B) = # OF BYTES -2
1804
1805            CALL    $TYPTX
1806            DB      'Encoded Entry =',' '+200Q
1807            INR     B
1808            INR     B
1809            LXI     D,EESA
1810            LXI     H,RESA
1811  EES4      LDAX    D               (A) = VALUE
1812            RAR
1813            RAR
1814            RAR
1815            RAR
1816            XRA     M
1817            ANI     17Q
1818            ORI     '@'
1819            INR     A
1820            CALL    $WCHAR          TYPE CHARACTER
1821            LDAX    D
1822            INX     H
1823            XRA     M
1824            ANI     17Q
1825            ORI     '@'
1826            INR     A
1827            CALL    $WCHAR          TYPE 2ND CHARACTER
1828            INX     H
1829            INX     D
1830            MVI     A,$RESAE
1831            CMP     L
1832            JNE     EES5
1833            LXI     H,RESA
1834  EES5      DCR     B
1835            JNZ     EES4            MORE TO GO
1836            CALL    $TYPTX
1837            DB      ENL             CRLF
1838            JMP     $RSTALL         RESTORE AND EXIT
1839
1840  EESA      DS      80              LINE BUFFER
1841
1842            ENDIF
```

```
                                1844  **      UBH - UPDATE BINARY HEADER.
                                1845  *
                                1846  *       UBH UPDATES THE LENGTH FIELD IN THE PROGRAMS BINARY HEADER
                                1847  *       TO REFLECT ANY PATCHES ADDED TO THE END OF THE PROGRAM.
                                1848  *
                                1849  *       ENTRY   NONE
                                1850  *       EXIT    NONE
                                1851  *       USES    ALL
                                1852
                                1853
     051.025  001 000 000       1854  UBH     LXI     B,0
     051.030  315 235 050       1855          CALL    PPF             REWIND FILE
     051.033  332 324 042       1856          JC      ERROR
     051.036  001 000 001       1857          LXI     B,256
     051.041  021 000 056       1858          LXI     D,BUFFER
     051.044  076 000           1859          MVI     A,CN.FIL
     051.046  377 004           1860          DB      SYSCALL,.READ   READ IN HEADER TABLE
     051.050  332 324 042       1861          JC      ERROR
                                1862
                                1863  *       SET NEW PROGRAM LENGTH
                                1864
     051.053  052 264 053       1865          LHLD    PGMLWA
     051.056  353               1866          XCHG
     051.057  052 262 053       1867          LHLD    PGMFWA          (HL) = FWA
     051.062  173               1868          MOV     A,E
     051.063  225               1869          SUB     L
     051.064  157               1870          MOV     L,A
     051.065  172               1871          MOV     A,D
     051.066  234               1872          SBB     H
     051.067  147               1873          MOV     H,A             (HL) = NEW LENGTH-1
     051.070  043               1874          INX     H               (HL) = NEW LENGTH
     051.071  072 252 053       1875          LDA     FILTYP
     051.074  376 000           1876          CPI     FT.ABS
     051.076  312 107 051       1877          JE      UBH1            IS ABSOLUTE BINARY
                                1878
                                1879  *       IS PIC PROGRAM
                                1880
     051.101  042 002 056       1881          SHLD    BUFFER+PIC.LEN
     051.104  303 112 051       1882          JMP     UBH2
                                1883
                                1884  *       IS ABS PROGRAM
                                1885
     051.107  042 004 056       1886  UBH1    SHLD    BUFFER+ABS.LEN
     051.112  001 000 000       1887  UBH2    LXI     B,0
     051.115  315 235 050       1888          CALL    PPF             REWIND AGAIN
     051.120  332 324 042       1889          JC      ERROR
     051.123  001 000 001       1890          LXI     B,256
     051.126  021 000 056       1891          LXI     D,BUFFER
     051.131  076 000           1892          MVI     A,CN.FIL
     051.133  377 005           1893          DB      SYSCALL,.WRITE  REPLACE HEADER
     051.135  320               1894          RNC                     RETURN IF OK
     051.136  303 324 042       1895          JMP     ERROR
```

```
                                1897  **      WPH - WRITE PATCH HISTORY.
                                1898  *
                                1899  *       WPH UPDATES THE PATCH HISTORY TABLE, AND APPENDS IT TO THE END OF
                                1900  *       THE BINARY FILE.
                                1901  *
                                1902  *       ENTRY   NONE
                                1903  *       EXIT    NONE
                                1904  *       USES    ALL
                                1905
                                1906
051.141  072 260 053           1907  WPH     LDA     PATCHID
051.144  107                   1908          MOV     B,A             (B) = PATCH ID
051.145  346 370               1909          ANI     370Q
051.147  037                   1910          RAR
051.150  037                   1911          RAR
051.151  037                   1912          RAR                     (A) = PATCHID/8
051.152  041 023 057           1913          LXI     H,PHT+PHT.HIS
051.155  315 101 030           1914          CALL    $DADA.          (HL) = ADDRESS OF BYTE FOR PATCH CODE
                                1915
                                1916  *       COMPUTE BIT TO SET TO INDICATE THIS PATCH
                                1917
051.160  170                   1918          MOV     A,B
051.161  346 007               1919          ANI     7
051.163  107                   1920          MOV     B,A             (B) = BIT INDEX
051.164  257                   1921          XRA     A
051.165  067                   1922          STC
051.166  027                   1923  WPH1    RAL
051.167  005                   1924          DCR     B
051.170  362 166 051           1925          JP      WPH1
051.173  266                   1926          ORA     M               SET BIT
051.174  167                   1927          MOV     M,A
                                1928
                                1929  **      WPH. - WRITE PHT TO END OF FILE
                                1930  *
                                1931  *       WRITE PHT TO END OF FILE
                                1932
051.175  052 264 053           1933  WPH.    LHLD    PGMLWA
051.200  353                   1934          XCHG
051.201  052 230 053           1935          LHLD    SKEW            COMPUTE LAST PROGRAM SECTOR USED
051.204  031                   1936          DAD     D               (HL) = FILE ADDRESS OF LAST PROGRAM BYTE
051.205  114                   1937          MOV     C,H
051.206  006 000               1938          MVI     B,0             (BC) = SECTOR NUMBER
051.210  003                   1939          INX     B               POINT TO ONE AFTER, FOR PHT
051.211  315 235 050           1940          CALL    PPF             POSITION TO IT
051.214  332 324 042           1941          JC      ERROR           COULDNT GET THERE
051.217  001 000 001           1942          LXI     B,256
051.222  021 000 057           1943          LXI     D,PHT
051.225  076 000               1944          MVI     A,CN.FIL
051.227  377 005               1945          DB      SYSCALL+.WRITE  WRITE PHT ON FILE
051.231  320                   1946          RNC                     RETURN IF NO ERROR
051.232  303 324 042           1947          JMP     ERROR
```

```
051.235                 1950          XTEXT   DNS


                        1952X **      $DNS - DECODE NUMERIC SWITCH.
                        1953X *
                        1954X *       $DNS DECODES A NUMERIC SWITCH OF THE FORM:
                        1955X *
                        1956X *       :NNN
                        1957X *
                        1958X *       A POSTRADIX OF D, Q, O, OR B IS ALLOWED. IF THE VALUE
                        1959X *       IS SYNTATICALLY VALID, IT IS REPLACED WITH BLANKS.
                        1960X *
                        1961X *       ENTRY   (HL) = ADDRESS IF ':'
                        1962X *               (A) = DEFAULT BASE (2, 8 OR 10)
                        1963X *       EXIT    'C' CLEAR IF OK
                        1964X *                (HL) ADVANCED PAST VALUE
                        1965X *                VALUE BLANKED
                        1966X *                (DE) = VALUE
                        1967X *                'C' SET IF ERROR
                        1968X *       USES    ALL
                        1969X
                        1970X
051.235  076 012       1971X $DNS.   MVI     A,10            BASE 10 DEFAULT
051.237  107           1972X $DNS    MOV     B,A             (B) = DEFAULT BASE
051.240  176           1973X         MOV     A,M
051.241  376 072       1974X         CPI     ':'
051.243  067           1975X         STC
051.244  300           1976X         RNE                     NOT ':'
051.245  345           1977X         PUSH    H               SAVE ADDRESS OF SWITCH START
051.246  043           1978X         INX     H
051.247  170           1979X         MOV     A,B
051.250  315 274 051   1980X         CALL    $DNV            DECODE NUMERIC VALUE
051.253  301           1981X         POP     B               (BC) = ADDRESS OF ':'
051.254  330           1982X         RC                      ERROR
051.255  076 040       1983X $DNS1   MVI     A,' '
051.257  002           1984X         STAX    B               BLANK LINE
051.260  003           1985X         INX     B               INCREMENT ADDRESS
051.261  175           1986X         MOV     A,L
051.262  271           1987X         CMP     C
051.263  302 255 051   1988X         JNE     $DNS1
051.266  170           1989X         MOV     A,B
051.267  274           1990X         CMP     H               SEE IF IN RIGHT BANK
051.270  302 255 051   1991X         JNE     $DNS1
051.273  311           1992X         RET                     RETURN WITH 'C' CLEAR AND VALUE
051.274                 1993          XTEXT   DNV
```

```
                                1995X **      $DNV - DECODE NUMERIC VALUE.
                                1996X *
                                1997X *       $DNV DECODES A NUMERIC VALUE (IN THE FORM OF AN ASCII STRING)
                                1998X *       INTO A BINARY NUMBER. THE MAXIMUM MAGNITUDE IS
                                1999X *       65535D.
                                2000X *
                                2001X *       THE NUMBER MAY CONTAIN A POSTRADIX OF 'B' (BINARY)
                                2002X *       'O' OR 'Q' (OCTAL) OR 'D' (DECIMAL)
                                2003X *
                                2004X *       ENTRY   (HL) = ADDRESS OF FIRST BYTE OF NUMBER
                                2005X *               (A) = DEFAULT BASE (2 FOR BINARY, 10 FOR DECIMAL, ETC.)
                                2006X *       EXIT    'C' CLEAR IF OK
                                2007X *                (HL) ADVANCED PAST NUMBER (AND POSTRADIX)
                                2008X *                (DE) = VALUE
                                2009X *                'C' SET IF ERROR
                                2010X *       USES    ALL
                                2011X
                                2012X
051.274  062 011 052            2013X $DNV    STA     $DNVA           SET DEFAULT BASE
051.277  104                    2014X         MOV     B,H
051.300  115                    2015X         MOV     C,L             (BC) = TEXT ADDRESS
                                2016X
                                2017X *       SCAN FOR POSTRADIX
                                2018X
051.301  176                    2019X $DNV1   MOV     A,M
051.302  315 055 052            2020X         CALL    $CVD.           CHECK FOR VALID DECIMAL DIGIT
051.305  043                    2021X         INX     H
051.306  322 301 051            2022X         JNC     $DNV1           MORE TO GO
051.311  053                    2023X         DCX     H               REMOVE EXTRA INCREMENT
051.312  171                    2024X         MOV     A,C
051.313  275                    2025X         CMP     L               SEE IF THERE WERE ANY NUMBERS
051.314  067                    2026X         STC                     ASSUME NOT
051.315  310                    2027X         RE                      ERROR
                                2028X
                                2029X *       OUT OF NUMBERS. SEE IF POSTRADIX FOLLOWS
                                2030X
051.316  176                    2031X         MOV     A,M             (A) = PROPOSED POSTRADIX
051.317  345                    2032X         PUSH    H               SAVE END ADDRESS
051.320  041 012 052            2033X         LXI     H,$DNVB
051.323  247                    2034X         ANA     A
051.324  312 344 051            2035X         JZ      $DNV2           NO POSTRADIX
051.327  315 023 052            2036X         CALL    $TBLS
051.332  176                    2037X         MOV     A,M
051.333  302 344 051            2038X         JNE     $DNV2           NOT POSTRADIX
051.336  341                    2039X         POP     H
051.337  043                    2040X         INX     H               SKIP POSTRADIX
051.340  345                    2041X         PUSH    H
051.341  062 011 052            2042X         STA     $DNVA           SET NEW POSTRADIX
051.344  021 000 000            2043X $DNV2   LXI     D,0             (DE) = ACCUMULATOR
                                2044X
                                2045X *       BUILD NUMBER
                                2046X
051.347  072 011 052            2047X $DNV3   LDA     $DNVA           (A) = BASE
051.352  365                    2048X         PUSH    PSW             SAVE BASE
051.353  315 007 031            2049X         CALL    $MU86           MULTIPLY
051.356  321                    2050X         POP     D               (D) = BASE
```

```
051.357  332 007 052  2051X          JC      $DNV4           OVERFLOW
051.362  012          2052X          LDAX    B               (A) = DIGIT
051.363  326 060      2053X          SUI     '0'
051.365  003          2054X          INX     B
051.366  272          2055X          CMP     D               COMPARE TO BASE
051.367  077          2056X          CMC
051.370  332 007 052  2057X          JC      $DNV4           TOO LARGE A DIGIT
051.373  315 101 030  2058X          CALL    $DADA.          ADD TO VALUE
051.376  353          2059X          XCHG                    (DE) = VALUE
051.377  012          2060X          LDAX    B
052.000  315 055 052  2061X          CALL    $CVD.
052.003  322 347 051  2062X          JNC     $DNV3           MORE TO GO
052.006  247          2063X          ANA     A               CLEAR CARRY
052.007  341          2064X $DNV4    POP     H               RESTORE POINTER
052.010  311          2065X          RET                     EXIT
                      2066X
052.011  000          2067X $DNVA    DB      0               DEFAULT BASE
052.012  102 002      2068X $DNVB    DB      'B',2           POSTRADIX TABLE
052.014  117 010      2069X          DB      'O',8
052.016  121 010      2070X          DB      'Q',8
052.020  104 012      2071X          DB      'D',10
052.022  000          2072X          DB      0
052.023               2073           XTEXT   TBLS


                      2075X **       $TBLS - TABLE SEARCH
                      2076X *
                      2077X *        TABLE FORMAT
                      2078X *
                      2079X *        DB      KEY1,VAL1,
                      2080X *        .       .
                      2081X *        .       .
                      2082X *        DB      KEYN,VALN
                      2083X *        DB      0
                      2084X *
                      2085X *        ENTRY   (A) = PATTERN
                      2086X *                (H,L) = TABLE FWA
                      2087X *        EXIT    (A) = PATTERN IF FOUND
                      2088X *                'Z' SET IF FOUND
                      2089X *                'Z' CLEAR IF NOT FOUND OR PATTERN=0          /78.10.GC/
                      2090X *        USES    A,F,H,L
                      2091X
                      2092X
052.023  305          2093X $TBLS    PUSH    B
052.024  376 000      2094X          CPI     0                                            /78.10.GC/
052.026  312 050 052  2095X          JZ      TBL2                                         /78.10.GC/
052.031  107          2096X          MOV     B,A
052.032  176          2097X TBL1     MOV     A,M             (A) = CHARACTER
052.033  043          2098X          INX     H
052.034  270          2099X          CMP     B
052.035  312 052 052  2100X          JZ      TBL3            IF MATCH
052.040  247          2101X          ANA     A
052.041  043          2102X          INX     H               SKIP PAST
052.042  302 032 052  2103X          JNZ     TBL1            IF NOT END OF TABLE
```

```
052.045  053          2104X          DCX     H
052.046  053          2105X          DCX     H
052.047  257          2106X          XRA     A               SET TO ZERO FOR OLD USERS       /78.10.GC/
052.050  376 001      2107X TBL2     CPI     1               CLEAR ZERO                      /78.10.GC/
                      2108X
                      2109X *        DONE
                      2110X
052.052  301          2111X TBL3     POP     B
052.053  311          2112X          RET
052.054               2113           XTEXT   MU86


                      2115X **       $MU86 - MULTIPLY 8X16 UNSIGNED.
                      2116X *
                      2117X *        $MU86 MULTIPLIES A 16 BIT VALUE BY A 8
                      2118X *        BIT VALUE.
                      2119X *
                      2120X *        ENTRY   (A) = MULTIPLIER
                      2121X *                (DE) = MULTIPLICAND
                      2122X *        EXIT    (HL) = RESULT
                      2123X *                'Z' SET IF NOT OVERFLOW
                      2124X *        USES    A,F,H,L
                      2125X
                      2126X
031.007               2127X $MU86    EQU     31007A          IN H17 ROM
052.054               2128           XTEXT   CVD


                      2130X **       $CVD - CHECK FOR VALID DIGIT.
                      2131X *
                      2132X *        CVD EXAMINES A DIGIT TO SEE IF IT IS A VALID DECIMAL DIGIT.
                      2133X *
                      2134X *        ENTRY   (HL) = ADDRESS OF CHARACTER
                      2135X *        EXIT    'C' SET IF ILLEGAL
                      2136X *                (A) = VALUE
                      2137X *        USES    A,F
                      2138X
                      2139X
052.054  176          2140X $CVD     MOV     A,M             (A) = CHARACTER
052.055  326 060      2141X $CVD.    SUI     '0'
052.057  330          2142X          RC                      ILLEGAL
052.060  376 012      2143X          CPI     9+1
052.062  077          2144X          CMC
052.063  311          2145X          RET
052.064               2146           XTEXT   DRS
```

```
                        2148X **      $DRS - DECODE AND REMOVE SWITCHES.
                        2149X *
                        2150X *       $DRS IS CALLED TO DECODE COMMAND SWITCHES FROM A LINE
                        2151X *       OF TEXT. SWITCHES TAKE THE FORM:
                        2152X *
                        2153X *       /XXXXX
                        2154X *
                        2155X *       AFTER A SWITCH HAS BEEN LOCATED, IT (AND THE PRECEDING '/')
                        2156X *       ARE REPLACED WITH BLANKS.
                        2157X *
                        2158X *       VALID SWITCH DESCRIPTIONS ARE ENCODED INTO A TABLE
                        2159X *       SUPPLIED BY THE CALLER, IN THE FORMAT:
                        2160X *
                        2161X *       DB      'X...X'         REQUIRED SWITCH CHARACTERS
                        2162X *       DB      'C'+200Q,...,'C'+200Q   OPTIONAL CHARACTERS
                        2163X *       DB      200Q            END OF CHARACTERS
                        2164X *       DW      ADDR            PROCESSOR ADDRESS (CALLED WHEN SWITCH DETECTED)
                        2165X *
                        2166X *       DB      'Y...Y'         NEXT SWITCH
                        2167X *       .       .
                        2168X *       .       .
                        2169X *       .       .
                        2170X *
                        2171X *       DB      0               FLAGS END OF TABLE
                        2172X *
                        2173X *       SWITCHES MUST BE FOLLOWED BY A ':', A '/' (ANOTHER SWITCH)
                        2174X *       A ',', OR A 00 BYTE.
                        2175X *
                        2176X *       UPON DETECTION OF A VALID SWITCH, $DRS CALLS THE USER PROCESS
                        2177X *       ROUTINE, UPON ENTRY,
                        2178X *       (HL) = ADDRESS OF THE FIRST BYTE FOLLOWING THE SWTICH
                        2179X *       'Z' CLEAR IF CHARACTER = '/', ',', OR 00
                        2180X *       'Z' SET IF CHARACTER = ':'
                        2181X *
                        2182X *       THE USER ROUTINE CAN DECODE SWITCH SUB-OPTIONS, IF DESIRED.
                        2183X *       THE USER ROUTINE MAY USE ALL REGISTERS.
                        2184X *
                        2185X *       ENTRY   (DE) = SWITCH TABLE FWA
                        2186X *               (HL) = LINE FWA
                        2187X *       EXIT    'C' CLEAR IF OK
                        2188X *               'C' SET IF ERROR
                        2189X *                (HL) = ADDRESS OF START OF BAD SWITCH
                        2190X *                (A) = ERROR CODE
                        2191X *       USES    ALL
                        2192X
                        2193X
052.064                 2194X $DRS    EQU     *
                        2195X
                        2196X *       LOOK FOR SWITCHES
                        2197X
052.064  176            2198X $DRS1   MOV     A,M
052.065  247            2199X         ANA     A
052.066  310            2200X         RZ                      END OF LINE
052.067  043            2201X         INX     H
052.070  376 057        2202X         CPI     '/'
052.072  302 064 052    2203X         JNE     $DRS1           NOT A SWITCH
```

```
052.075  042 261 052  2204X          SHLD    $DRSB           ($DRSB) = SWITCH FWA (AFTER '/')
                      2205X
                      2206X *        GOT A SWITCH. LOOK FOR A MATCH IN THE CALLER'S TABLE
                      2207X
052.100  325          2208X          PUSH    D               SAVE TABLE FWA
052.101  052 261 052  2209X $DRS2    LHLD    $DRSB           (HL) = SWITCH FWA
052.104  032          2210X $DRS3    LDAX    D               (A) = TABLE ENTRY
052.105  346 177      2211X          ANI     177Q
052.107  312 157 052  2212X          JZ      $DRS6           GOT A MATCH
052.112  276          2213X          CMP     M
052.113  302 123 052  2214X          JNE     $DRS4           NO MATCH
052.116  023          2215X          INX     D
052.117  043          2216X          INX     H
052.120  303 104 052  2217X          JMP     $DRS3           SEE IF MORE MATCH
                      2218X
                      2219X *        HAVE MIS-MATCH. SEE IF THE MISSING CHARACTER IS SIGNIFICANT
                      2220X
052.123  176          2221X $DRS4    MOV     A,M             (A) = LINE CHARACTER WE COULDNT MATCH
052.124  315 230 052  2222X          CALL    $DRS15          SEE IF OK TERMINATOR
052.127  302 137 052  2223X          JNE     $DRS4.5         NO MATCH ON THIS SWITCH
052.132  032          2224X          LDAX    D               (A) = NEXT CHARACTER IN SWITCH PATTERN
052.133  247          2225X          ANA     A
052.134  372 157 052  2226X          JM      $DRS6           HAVE SUFFICIENT MATCH
052.137  315 243 052  2227X $DRS4.5  CALL    $DRS20          SKIP TABLE ENTRY
052.142  032          2228X          LDAX    D
052.143  247          2229X          ANA     A
052.144  302 101 052  2230X          JNZ     $DRS2           MORE SWITCHES IN TABLE TO CHECK
                      2231X
                      2232X *        BAD SWITCH
                      2233X
052.147  321          2234X $DRS5    POP     D               RESTORE STACK
052.150  052 261 052  2235X          LHLD    $DRSB           POINT TO BAD SWITCH
052.153  067          2236X          STC
052.154  076 032      2237X          MVI     A,EC.IS         ILLEGAL SWITCH
052.156  311          2238X          RET
                      2239X
                      2240X *        HAVE SWITCH. CHECK IT'S FOLLOWING CHARACTER
                      2241X
052.157  315 263 052  2242X $DRS6    CALL    $SOB            SKIP OVER BLANKS
052.162  176          2243X          MOV     A,M
052.163  315 230 052  2244X          CALL    $DRS15          CHECK CHARACTER
052.166  302 147 052  2245X          JNE     $DRS5           IN ERROR
052.171  315 243 052  2246X          CALL    $DRS20          GET PROCESSOR ADDRESS
052.174  021 206 052  2247X          LXI     D,$DRS7
052.177  345          2248X          PUSH    H               SAVE (HL)
052.200  325          2249X          PUSH    D               SET RETURN ADDRESS FOR TABLE CODE
052.201  305          2250X          PUSH    B               SAVE PROCESSOR ADDRESS
052.202  176          2251X          MOV     A,M             (A) = NEXT CHARACTER
052.203  376 072      2252X          CPI     ':'             SET CONDITION CODES
052.205  311          2253X          RET                     CALL USER PROCESS
                      2254X
                      2255X *        USER PROCESS RETURNS HERE
                      2256X
052.206  321          2257X $DRS7    POP     D               (DE) = LAST CHARACTER OF SWITCH+1
052.207  052 261 052  2258X          LHLD    $DRSB           (HL) = FIRST CHARACTER OF SWITCH AFTER /
052.212  053          2259X          DCX     H               (HL) = ADDRESS OF '/'
```

```
                                2260X
                                2261X *         REPLACE SWITCH WITH BLANKS
                                2262X
     052.213  066 040           2263X $DRS8     MVI     M,' '
     052.215  043               2264X           INX     H
     052.216  315 216 030       2265X           CALL    $CDEHL
     052.221  302 213 052       2266X           JNE     $DRS8           NOT THERE YET
     052.224  321               2267X           POP     D               (DE) = SWITCH TABLE FWA
     052.225  303 064 052       2268X           JMP     $DRS1           LOOK FOR MORE SWITCHES

                                2270X **        $DRS15 - CHECK FOR VALID DELIMITER CHARACTER.
                                2271X *
                                2272X *         $DRS15 CHECKS THE NEXT TEXT CHARACTER TO SEE IF IT IS
                                2273X *
                                2274X *         00, '/', ',', ':'
                                2275X *
                                2276X *         ENTRY   (A) = CHARACTER
                                2277X *         EXIT    'Z' SET IFF CHARACTER IS ONE OF THE ABOVE
                                2278X *         USES    F
                                2279X
     052.230  247               2280X $DRS15    ANA     A
     052.231  310               2281X           RZ                      IS 00
     052.232  376 057           2282X           CPI     '/'
     052.234  310               2283X           RE
     052.235  376 054           2284X           CPI     ','
     052.237  310               2285X           RE
     052.240  376 072           2286X           CPI     ':'
     052.242  311               2287X           RET

                                2289X **        $DRS20 - GET PROCESSOR ADDRESS.
                                2290X *
                                2291X *         $DRS20 IS CALLED TO GET THE PROCESSOR ADDRESS FIELD OUT OF
                                2292X *         AN ENTRY IN THE SWITCH TABLE. THE CALLER SUPPLIES A POINTER
                                2293X *         TO SOMEWHERE IN THE TEXT PART OF THE SWITCH DESCRIPTION;
                                2294X *         $DRS20 ADVANCES THE POINTER TO THE PROCESSOR ADDRESS.
                                2295X *
                                2296X *         ENTRY   (DE) = POINTER TO TEXT PART OF SWITCH ENTRY
                                2297X *         EXIT    (DE) = POINTER TO 1ST BYTE OF NEXT SWITCH TABLE ENTRY
                                2298X *                 (BC) = PROCESSOR ADDRESS FROM TABLE
                                2299X *         USES    A,F,B,C,D,E
                                2300X
                                2301X
     052.243  032               2302X $DRS20    LDAX    D
     052.244  023               2303X           INX     D
     052.245  376 200           2304X           CPI     200Q
     052.247  302 243 052       2305X           JNE     $DRS20
     052.252  032               2306X           LDAX    D               (A) = LOW BYTE OF PROCESSOR ADDRESS
     052.253  117               2307X           MOV     C,A
     052.254  023               2308X           INX     D
     052.255  032               2309X           LDAX    D
     052.256  107               2310X           MOV     B,A             (BC) = PROCESSOR ADDRESS
     052.257  023               2311X           INX     D
     052.260  311               2312X           RET
                                2313X
```

```
052.261  000 000      2314X $DRSB   DW      0               POINTER TO SWITCH BEING PROCESSED
052.263               2315          XTEXT   SOB


                      2317X **      $SOB - SKIP OVER BLANKS.
                      2318X *
                      2319X *       $SOB IS CALLED TO SKIP AN ARBITRARILY LONG STRING OF BLANKS AND TABS.
                      2320X *
                      2321X *       ENTRY   (HL) = FWA OF (POSSIBLE) BLANK STRING
                      2322X *       EXIT    (HL) = LWA+1 OF BLANK STRING (UNCHANGED IF NO BLANKS)
                      2323X *               (A) = FIRST NON-BLANK, NON-TAB CHARACTER EEN
                      2324X *       USES    A,F,H,L
                      2325X
                      2326X
052.263  053         2327X $SOB    DCX     H               PRE-DECREMENT
052.264  043         2328X $SOB1   INX     H
052.265  176         2329X         MOV     A,M
052.266  376 040     2330X         CPI     ' '
052.270  312 264 052 2331X         JE      $SOB1           GOT BLANK
052.273  376 011     2332X         CPI     TAB
052.275  312 264 052 2333X         JE      $SOB1           GOT TAB
052.300  311         2334X         RET
052.301               2335          XTEXT   CHL


                      2337X **      $CHL - COMPLEMENT (HL).
                      2338X *
                      2339X *       (HL) = -(HL)            TWO'S COMPLEMENT
                      2340X *
                      2341X *       ENTRY   NONE
                      2342X *       EXIT    NONE
                      2343X *       USES    A,F,H,L
                      2344X
                      2345X
030.224               2346X $CHL    EQU     30224A          IN H17 ROM
052.301               2347          XTEXT   MOVE


                      2349X **      $MOVE - MOVE DATA
                      2350X *
                      2351X *       $MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                      2352X *       IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                      2353X *       FIRST TO LAST.
                      2354X *
                      2355X *       IF THE MOVE IS TO A HIGHER ADDRESS; THE BYTES ARE MOVED FROM
                      2356X *       LAST TO FIRST.
                      2357X *
                      2358X *       THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                      2359X *
                      2360X *       ENTRY   (BC) = COUNT
```

```
                                2361X *                 (DE) = FROM
                                2362X *                 (HL) = TO
                                2363X *         EXIT    MOVED
                                2364X *                 (DE) = ADDRESS OF NEXT FROM BYTE
                                2365X *                 (HL) = ADDRESS OF NEXT *TO* BYTE
                                2366X *                 'C' CLEAR
                                2367X *         USES    ALL
                                2368X
                                2369X
      030,252                   2370X $MOVE     EQU     30252A          IN H17 ROM
      052,301                   2371            XTEXT   CCO



                                2373X **        $CCO - CLEAR CONTROL-O
                                2374X *
                                2375X *         $CCO IS CALLED TO CLEAR THE EFFECT OF THE CTL-O CHARACTER.
                                2376X *
                                2377X *         ENTRY   NONE
                                2378X *         EXIT    NONE
                                2379X *         USES    NONE
                                2380X
                                2381X
      052,301  315 054 031      2382X $CCO      CALL    $SAVALL         SAVE REGISTERS
      052,304  076 004          2383X           MVI     A,I.CONFL
      052,306  001 001 000      2384X           LXI     B,CO.FLG        CLEAR CO.FLG
      052,311  377 006          2385X           DB      SYSCALL+.CONSL
      052,313  303 047 031      2386X           JMP     $RSTALL         RESTORE REGISTERS AND RETURN
      052,316                   2387            XTEXT   SAVALL



                                2389X **        $RSTALL - RESTORE ALL REGISTERS.
                                2390X *
                                2391X *         $RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND
                                2392X *         RETURNS TO THE PREVIOUS CALLER.
                                2393X *
                                2394X *         ENTRY   (SP) = PSW
                                2395X *                 (SP+2) = BC
                                2396X *                 (SP+4) = DE
                                2397X *                 (SP+6) = HL
                                2398X *                 (SP+8) = RET
                                2399X *         EXIT    TO *RET*, REGISTERS RESTORED
                                2400X *         USES    ALL
                                2401X
                                2402X
      031,047                   2403X $RSTALL   EQU     31047A          IN H17 ROM
```

```
                        2405X **      $SAVALL - SAVE ALL REGISTERS ON STACK.
                        2406X *
                        2407X *       $SAVALL SAVES ALL THE REGISTERS ON THE STACK.
                        2408X *
                        2409X *       ENTRY   NONE
                        2410X *       EXIT    (SP) = PSW
                        2411X *               (SP+2) = BC
                        2412X *               (SP+4) = DE
                        2413X *               (SP+6) = HL
                        2414X *       USES    H,L
                        2415X
                        2416X
031.054                 2417X $SAVALL EQU     31054A          IN H17 ROM
052.316                 2418          XTEXT   HLIHL


                        2420X **      $HLIHL - LOAD HL INDIRECT THROUGH HL.
                        2421X *
                        2422X *       (HL) = ((HL))
                        2423X *
                        2424X *       ENTRY   NONE
                        2425X *       EXIT    NONE
                        2426X *       USES    A,H,L
                        2427X
030.211                 2428X $HLIHL  EQU     30211A          IN H17 ROM
052.316                 2429          XTEXT   CDEHL


                        2431X **      $CDEHL - COMPARE (DE) TO (HL)
                        2432X *
                        2433X *       $CDEHL COMPARES (DE) TO (HL) FOR EQUALITY.
                        2434X *
                        2435X *       ENTRY   NONE
                        2436X *       EXIT    'Z' SET IF (DE) = (HL)
                        2437X *       USES    A,F
                        2438X
                        2439X
030.216                 2440X $CDEHL  EQU     30216A          IN H17 ROM
052.316                 2441          XTEXT   CRC


                        2443X **      $CRC - COMPUTE CRC16
                        2444X *
                        2445X *
                        2446X *       COMPUTE THE CRC16 CHECKSUM.
                        2447X *
                        2448X *       ENTRY   (HL) = CURRENT CHECKSUM
                        2449X *               (A) = BYTE
                        2450X *       EXIT    (HL) UPDATED
                        2451X *               (A) UNCHANGED.
```

```
                        2452X *         USES    F,H,L
                        2453X
                        2454X
      052.316  305      2455X $CRC      PUSH    B               SAVE (BC)
      052.317  006 010  2456X           MVI     B,8             (B) = BIT COUNT
      052.321  007      2457X $CRC1     RLC
      052.322  117      2458X           MOV     C,A             (C) = BIT
      052.323  175      2459X           MOV     A,L
      052.324  207      2460X           ADD     A
      052.325  157      2461X           MOV     L,A
      052.326  174      2462X           MOV     A,H
      052.327  027      2463X           RAL
      052.330  147      2464X           MOV     H,A
      052.331  027      2465X           RAL
      052.332  251      2466X           XRA     C
      052.333  017      2467X           RRC
      052.334  322 347 052 2468X        JNC     $CRC2           IF NOT TO XOR
      052.337  174      2469X           MOV     A,H
      052.340  356 200  2470X           XRI     200Q
      052.342  147      2471X           MOV     H,A
      052.343  175      2472X           MOV     A,L
      052.344  356 005  2473X           XRI     5Q
      052.346  157      2474X           MOV     L,A
      052.347  171      2475X $CRC2     MOV     A,C
      052.350  005      2476X           DCR     B
      052.351  302 321 052 2477X       JNZ     $CRC1           IF MORE TO GO
      052.354  301      2478X           POP     B               RESTORE (BC)
      052.355  311      2479X           RET                     EXIT
      052.356           2480            XTEXT   COMP


                        2482X **        $COMP - COMPARE TWO CHARACTER STRINGS.
                        2483X *
                        2484X *         $COMP COMPARES TWO BYTE STRINGS.
                        2485X *
                        2486X *         ENTRY   (C) = COMPARE COUNT
                        2487X *                 (DE) = FWA OF STRING #1
                        2488X *                 (HL) = FWA OF STRING #2
                        2489X *         EXIT    'Z' CLEAR, IS MIS-MATCH
                        2490X *                  (C) = LENGTH REMAINING
                        2491X *                  (DE) = ADDRESS OF MISMATCH IN STRING#1
                        2492X *                  (HL) = ADDRESS OF MISMATCH IN STRING #2
                        2493X *                 'C' SET, HAVE MATCH
                        2494X *                  (C) = 0
                        2495X *                  (DE) = (DE) + (OC)
                        2496X *                  (HL) = (HL) + (OC)
                        2497X *         USES    A,F,C,D,E,H,L
                        2498X
                        2499X
      030.060           2500X $COMP     EQU     30060A          IN H17 ROM
      052.356           2501            XTEXT   MCU
```

```
                        2503X **      MCU - MAP LOWER CASE TO UPPER CASE.
                        2504X *
                        2505X *       MCU MAPS A LOWER CASE ALPHABETIC TO UPPER
                        2506X *       CASE.
                        2507X *
                        2508X *       ENTRY   (A) = CHARACTER
                        2509X *       EXIT    (A) = CHARACTER RESULT
                        2510X *       USES    A,F
                        2511X
                        2512X
052.356  376 141        2513X $MCU    CPI     'a'
052.360  330           2514X         RC                      NOT LOWER CASE
052.361  376 173       2515X         CPI     'z'+1
052.363  320           2516X         RNC                     NOT LOWER CASE
052.364  326 040       2517X         SUI     'a'-'A'
052.366  311           2518X         RET
052.367                2519          XTEXT   TYPTX


                        2521X **      $TYPTX - TYPE TEXT.
                        2522X *
                        2523X *       $TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
                        2524X *
                        2525X *       IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED,
                        2526X *       A BYTE WITH THE 200Q BIT SET IS THE LAST BYTE IN THE MESSAGE.
                        2527X *
                        2528X *       ENTRY   (RET) = TEXT
                        2529X *       EXIT    TO (RET+LENGTH)
                        2530X *       USES    A,F
                        2531X
                        2532X
031.136                 2533X $TYPTX  EQU     31136A          IN H17 ROM
                        2534X
031.144                 2535X $TYPTX. EQU     31144A          IN H17 ROM
052.367                 2536          XTEXT   RCHAR


                        2538X **      $RCHAR - READ SINGLE CHARACTER FROM CONSOLE.
                        2539X *
                        2540X *       ENTRY   NONE
                        2541X *       EXIT    (A) = CHARACTER
                        2542X *       USES    A,F
                        2543X
                        2544X
052.367  377 001        2545X $RCHAR  DB      SYSCALL,.SCIN
052.371  332 367 052    2546X         JC      $RCHAR          NOT READY
052.374  311           2547X         RET
                        2548X
052.375  377 002       2549X $WCHAR  DB      SYSCALL,.SCOUT
052.377  311           2550X         RET
053.000                 2551          XTEXT   DADA
```

```
                        2553X **      $DADA - PERFORM (H,L) = (H,L) + (0,A)
                        2554X *
                        2555X *       ENTRY   (H,L) = BEFORE VALUE
                        2556X *               (A) = BEFORE VALUE
                        2557X *       EXIT    (H,L) = (H,L) + (0,A)
                        2558X *               'C' SET IF OVERFLOW
                        2559X *       USES    F,H,L
                        2560X
                        2561X
    030.072             2562X $DADA   EQU     30072A          IN H17 ROM
    053.000             2563          XTEXT   DADA2



                        2565X **      $DADA, - ADD (0,A) TO (H,L)
                        2566X *
                        2567X *       ENTRY   NONE
                        2568X *       EXIT    (HL) = (HL) + (0A)
                        2569X *       USES    A,F,H,L
                        2570X
                        2571X
    030.101             2572X $DADA.  EQU     30101A          IN H17 ROM
    053.000             2573          XTEXT   DU66



                        2575X **      $DU66 - UNSIGNED 16 / 16 DIVIDE.
                        2576X *
                        2577X *       (HL) = (BC)/(DE)
                        2578X *
                        2579X *       ENTRY   (BC), (DE) PRESET
                        2580X *       EXIT    (HL) = RESULT
                        2581X *               (DE) = REMAINDER
                        2582X *       USES    ALL
                        2583X
                        2584X
    030.106             2585X $DU66   EQU     30106A          IN H17 ROM
    053.000             2586          XTEXT   MLU



                        2588X **      MLU - MAP LOWER CASE LINE TO UPPER CASE.
                        2589X *
                        2590X *       MLU MAPS THE LOWER CASE ALPHABETICS IN A LINE TO UPPER CASE.
                        2591X *
                        2592X *       ENTRY   (HL)  = LINE FWA
                        2593X *       EXIT    NONE
                        2594X *       USES    NONE
                        2595X
                        2596X
    053.000  365        2597X $MLU    PUSH    PSW             SAVE (PSW)
    053.001  345        2598X         PUSH    H               SAVE FWA
    053.002  053        2599X         DCX     H               ANTICIPATE INX H
```

```
053.003  043            2600X $MLU1   INX     H
053.004  176            2601X         MOV     A,M             (A)= CHARACTER
053.005  315 356 052    2602X         CALL    $MCU            MAP CHAR TO UPPER
053.010  167            2603X         MOV     M,A
053.011  247            2604X         ANA     A
053.012  302 003 053    2605X         JNZ     $MLU1           MORE TO GO
053.015  341            2606X         POP     H               RESTORE (HL)
053.016  361            2607X         POP     PSW             RESTORE (PSW)
053.017  311            2608X         RET
053.020                 2609          XTEXT   TYPCH



                        2611X **      $TYPCH - TYPE SINGLE CHARACTER.
                        2612X *
                        2613X *       ENTRY   (RET) = CHARACTER
                        2614X *       EXIT    TO (RET)+1
                        2615X *               (A) = CHARACTER TYPED
                        2616X
                        2617X
053.020  343            2618X $TYPCH  XTHL                    (HL) = RETURN ADDRESS
053.021  176            2619X         MOV     A,M             (A) = CHARACTER
053.022  043            2620X         INX     H
053.023  343            2621X         XTHL                    RESTORE ADVANCED EXIT ADDRESS
                        2622X
                        2623X **      $TYPC. - TYPE SINGLE CHARACTER.
                        2624X *
                        2625X *       ENTRY   (A) = CHARACTER
                        2626X *       EXIT    TO (RET)
                        2627X
053.024  377 002        2628X $TYPC.  DB      SYSCALL,.SCOUT
053.026  311            2629X         RET
053.027                 2630          XTEXT   RTL



                        2632X **      $RTL - READ TEXT LINE.
                        2633X *
                        2634X *       $RTL READS A LINE FROM THE TERMINAL.
                        2635X *
                        2636X *       CHARACTER ARE ACCEPTED FROM THE TERMINAL, RUBOUT AND BACKSPACE
                        2637X *       CHARACTERS ARE PROCESSED. WHEN A CARRIAGE RETURN IS ENTERED,
                        2638X *       $RTL RETURNS.
                        2639X *
                        2640X *       ENTRY   (HL) = BUFFER FWA
                        2641X *       EXIT    'C' CLEAR IF OK
                        2642X *                DATA IN BUFFER
                        2643X *                (A) = TEXT LENGTH
                        2644X *               'C' SET IF CTL-D STRUCK
                        2645X *       USES    A,F
                        2646X
                        2647X
053.027  315 036 053    2648X $RTL.   CALL    $RTL            $RTL IN UPPER CASE
053.032  330            2649X         RC                      CTL-D
```

```
053.033  303 000 053  2650X          JMP     $MLU            MAP LINE TO UPPER CASE
                      2651X
053.036               2652X $RTL     EQU     *
053.036  345          2653X          PUSH    H               SAVE FWA
053.037  315 367 052  2654X $RTL1    CALL    $RCHAR
053.042  376 004      2655X          CPI     CTLD
053.044  312 071 053  2656X          JE      $RTL2           CTL-D STRUCK
053.047  167          2657X          MOV     M,A
053.050  043          2658X          INX     H
053.051  376 012      2659X          CPI     NL
053.053  302 037 053  2660X          JNE     $RTL1
053.056  053          2661X          DCX     H
053.057  066 000      2662X          MVI     M,0
053.061  043          2663X          INX     H
                      2664X
                      2665X *        ALL DONE. COMPUTE LENGTH
                      2666X
053.062  353          2667X          XCHG                    (DE) = LWA+1
053.063  343          2668X          XTHL                    (HL) = FWA
053.064  173          2669X          MOV     A,E
053.065  225          2670X          SUB     L               (A) = LENGTH
053.066  247          2671X          ANA     A               CLEAR CARRY
053.067  321          2672X          POP     D               RESTORE (DE)
053.070  311          2673X          RET
                      2674X
                      2675X *        CTL-D STRUCK
                      2676X
053.071  341          2677X $RTL2    POP     H               (HL) = FWA
053.072  067          2678X          STC
053.073  311          2679X          RET
053.074               2680           XTEXT   UDDN


                      2682X **       $UDDN - UNPACK DECIMAL DIGITS.
                      2683X *
                      2684X *        UDDN CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
                      2685X *        DECIMAL DIGITS. THE RESULT IS NULL FILLED TO THE LEFT.
                      2686X *
                      2687X *        ENTRY   (B,C) = ADDRESS VALUE
                      2688X *                (A) = DIGIT COUNT
                      2689X *                (H,L) = MEMORY ADDRESS
                      2690X *        EXIT    (HL) = (HL) + (A)
                      2691X *        USES    ALL
                      2692X
                      2693X
053.074               2694X $UDDN    EQU     *
053.074  315 072 030  2695X          CALL    $DADA
053.077  345          2696X          PUSH    H               SAVE FINAL (H,L) VALUE
                      2697X
053.100  365          2698X UDDN1    PUSH    PSW
053.101  345          2699X          PUSH    H
053.102  021 012 000  2700X          LXI     D,10
053.105  315 106 030  2701X          CALL    $DU66           (H,L) = VALUE/10
053.110  104          2702X          MOV     B,H
```

```
053.111  115          2703X          MOV     C,L             (BC) = QUOTIENT
053.112  341          2704X          POP     H
053.113  076 060      2705X          MVI     A,'0'
053.115  203          2706X          ADD     E               ADD REMAINDER
053.116  053          2707X          DCX     H
053.117  167          2708X          MOV     M,A             STORE DIGIT
053.120  170          2709X          MOV     A,B
053.121  261          2710X          ORA     C
053.122  312 134 053  2711X          JZ      UDDN2           ALL ZEROS
053.125  361          2712X          POP     PSW
053.126  075          2713X          DCR     A
053.127  302 100 053  2714X          JNZ     UDDN1           IF MORE TO GO
                      2715X
                      2716X *        ALL DONE, EXIT
                      2717X
053.132  341          2718X UDDN1.5  POP     H               RESTORE H
053.133  311          2719X          RET                     RETURN
                      2720X
                      2721X *        DIGITS LEADING THIS ONE ARE ZERO, STORE NULLS INSTEAD.
                      2722X
053.134  361          2723X UDDN2    POP     PSW
053.135  075          2724X UDDN3    DCR     A
053.136  312 132 053  2725X          JE      UDDN1.5         ALL DONE
053.141  053          2726X          DCX     H
053.142  066 000      2727X          MVI     M,0
053.144  303 135 053  2728X          JMP     UDDN3
053.147               2729           XTEXT   ZERO


                      2731X **       $ZERO - ZERO MEMORY
                      2732X *
                      2733X *        $ZERO ZEROS A BLOCK OF MEMORY.
                      2734X *
                      2735X *        ENTRY   (HL) = ADDRESS
                      2736X *                (B) = COUNT
                      2737X *        EXIT    (A) = 0
                      2738X *        USES    A,B,F,H,L
                      2739X
                      2740X
031.212               2741X $ZERO    EQU     31212A          IN H17 ROM
053.147               2742           XTEXT   TOD


                      2744X **       $TOD - TYPE OCTAL DIGITS.
                      2745X *
                      2746X *        $TOD TYPES AN OCTAL BYTE AS 3 OCTAL DIGITS, ZERO FILL.
                      2747X *
                      2748X *        ENTRY   (A) = VALUE
                      2749X *        EXIT    VALUE TYPED
                      2750X *        USES    A,F
                      2751X
                      2752X
```

```
053.147  305          2753X $TOD    PUSH    B
053.150  006 003      2754X         MVI     B,3
053.152  247          2755X         ANA     A               CLEAR CARRY
                      2756X
053.153  027          2757X TOD1    RAL
053.154  027          2758X         RAL
053.155  027          2759X         RAL
053.156  365          2760X         PUSH    PSW
053.157  346 007      2761X         ANI     7
053.161  306 060      2762X         ADI     '0'
053.163  315 024 053  2763X         CALL    $TYPC.          TYPE CHARACTER
053.166  361          2764X         POP     PSW
053.167  005          2765X         DCR     B
053.170  302 153 053  2766X         JNZ     TOD1            IF MORE TO GO
053.173  301          2767X         POP     B
053.174  311          2768X         RET                     EXIT
```

```
                                2771  ***     THIS SECTION DESCRIBES THE PATCH HISTORY TABLE.  THE ACTUAL TABLE
                                2772  *       WILL BE READ INTO THE BUFFER AREA *PHT*, DECLARED BELOW.
                                2773  *
                                2774  *       THE PATCH HISTORY TABLE (PHT) IS USED TO KEEP TRACK OF WHAT PATCHES
                                2775  *       HAVE BEEN ENTERED INTO A FILE.  SYSTEM FILES ALL HAVE
                                2776  *       BUILT-ON PHT'S, ANY FILE WITHOUT ONE IS THEREFORE NOT A SYSTEM
                                2777  *       FILE.
                                2778  *
                                2779  *       Note that the PHT stuck on the end of a binary file,
                                2780  *       after the binary information.  Thus, the PHT is not loaded into memory
                                2781  *       during normal execution.
                                2782  *
                                2783  *       The PHT occupies an entire sector, always the last sector in the file.
                                2784  *       If the patches cause the file to be lengthened, then the
                                2785  *       old PHT will be overlaid, and the new one added to the end of the
                                2786  *       longer file.
                                2787
                                2788
                                2789
053.175                         2790  PHTFORM DS      0               START OF 'STANDARD FORM' FOR PHT
000.000                         2791  PHT.HDR EQU     *-PHTFORM
053.175  376 001               2792          DB      376Q,001Q       PHT HEADER
053.177  120 110 124           2793          DB      'PHT/HSG'        IDENTIFIES THE REAL THING!
000.011                         2794  PHTHDRL EQU     *-PHTFORM               PHT HEADER LENGTH
000.011                         2795  PHT.DAT EQU     *-PHTFORM
053.206  122 127 055           2796          DB      'RW-JGL-RW'     DATE OF LAST PATCH IN DD-MMM-YY
000.022                         2797  PHT.CNT EQU     *-PHTFORM
053.217  000                    2798          DB      0               NUMBER OF PATCHES MADE
000.023                         2799  PHT.HIS EQU     *-PHTFORM
053.220  000 000 000 000 000 0  2800          DB      0,0,0,0,0,0,0,0 BIT-BY-BIT PATCH HISTORY (PATCH 0-63)
000.033                         2801  PHTL    EQU     *-PHTFORM       PHT ENTIRE LENGTH
```

```
                                2804  **      PATCH SERIES VALUES.
                                2805  *
                                2806  *       THESE CELLS KEEP TRACK OF THE CURRENT PATCH ADDRESS
                                2807  *       AND ASSOCIATED VALUES
                                2808
053.230  000 000                2809  SKEW    DW      0               CORRECTION FACTOR TO MAP PROGRAM ADDRESS
053.232  000                    2810  SDISP   DB      0               SECTORS TO DISPLACE INTO FILE TO FIND PROGRAM
                                2811  *                                INTO FILE BYTE NUMBER
053.233  000 000                2812  CPA     DW      0               CURRENT PATCH ADDRESS
053.235  000                    2813  SIVB    DB      0               CURRENT SECTOR IN VIEW BUFFER
053.236  000                    2814  SIVBV   DB      0               <>0 IF SIVB VALID
053.237  000                    2815  SIVBA   DB      0               <>0 IF SECTOR-IN-VIEWBFR ALTERED


                                2817  **      MISCELANIOUS WORK AREAS
                                2818
053.240  123 131 060            2819  DEFALT  DB      'SYOABS'        DEFAULT FOR FILE NAMES
053.246  000 060                2820  PLPTR   DW      PATLIST         POINTER TO NEXT FREE BYTE IN PATLIST
053.250  000 000                2821  PLMAX   DW      0               MAX SIZE OF PATCH LIST
053.252  000                    2822  FILTYP  DB      0               FT.ABS OR FT.PIC


                                2824  **      PATCH VALUES (MEANINGFUL ONLY FOR SYSTEM PATCHES)
                                2825
053.253                         2826  PATPRQ  DS      5               PATCH PREREQUESITE LIST
000.005                         2827  PATPRQL EQU     *-PATPRQ        NUMBER OF BYTES IN LIST
053.260                         2828  PATCHID DS      1               PATCH ID VALUE


                                2830  **      PHT FLAGS
                                2831
053.261  000                    2832  PHIST   DB      0               <>0 IF PHT PRESENT
053.262  000 000                2833  PGMFWA  DW      0               FWA OF USER PROGRAM
053.264  000 000                2834  PGMLWA  DW      0               LWA OF USER PROGRAM
053.266  000 000                2835  FILSIZ  DW      0               SIZE OF FILE
                                2836
000.001                         2837          IF      .SYS.
                                2838  CHECKC  DB      0               <>0 IF /CHECK SWITCH SELECTED
                                2839  PHTSWI  DB      0               <>0 IF /PHT SWITCH SELECTED
                                2840  PHT     DS      256             FILL PHT BUFFER WITH COPYRIGHT MESSAGE
                                2841          ENDIF
                                2842
053.270                         2843  .PAT1.  DS      32              PATCH AREA WHICH DOESNT REQUIRE EXPANSION OF FILE
053.330  014                    2844          DB      FF
053.331  266 271 337            2845          DB      377Q-'I',377Q-'F',377Q-' ',377Q-'U',377Q-' ',377Q-'C',377Q-'N'
053.340  337 255 273            2846          DB      377Q-' ',377Q-'R',377Q-'D',377Q-' ',377Q-'T',377Q-'H',377Q-'S'
053.347  337 252 337            2847          DB      377Q-' ',377Q-'U',377Q-' ',377Q-'C',377Q-'N',377Q-' ',377Q-'G'
053.356  253 337 276            2848          DB      377Q-'T',377Q-' ',377Q-'A',377Q-' ',377Q-'G',377Q-'D',377Q-' '
053.365  265 275 336            2849          DB      377Q-'J',377Q-'B',377Q-'!',FF
                                2850
                                2851
                                2852
                                2853  **      BUFFERS
```

```
                        2854
        053.371         2855  MEML    EQU     *               END OF LOAD IMAGE
                        2856
        053.371         2857  .PAT.   DS      64              PATCH AREA
                        2858
        054.071         2859  LINE    DS      80
        054.211         2860          DS      *+255/256*256-* PUT VIEWBFR ON EVEN BOUNDARY
        055.000         2861  VIEWBFR DS      256             BUFFER TO PROVIDE 'OLD VALUE' DISPLAY
        000.000         2862          ERRNZ   #VIEWBFR        MUST BE ON EVEN BOUNDARY
        056.000         2863  BUFFER  DS      256             WORKING BUFFER
        000.001         2864          IF      .SYS.
                        2865          ELSE                    PUT BUFFER OFF END, NORMALLY
        057.000         2866  PHT     DS      256             PATCH HISTORY TABLE
                        2867          ENDIF


                        2869  **      PATCH LIST.
                        2870  *
                        2871
        060.000         2872  PATLIST EQU     *               PATCH LIST STARTS HERE
                        2873
        060.000         2874          END
ASSEMBLY COMPLETE
 2874 STATEMENTS
    0 ERRORS DETECTED
11872 BYTES FREE
```

```
$CCO     052301    615    2382L
$CDEHL   030216   1265    1302   2265   2440E
$CHL     030224    786    1095   2346E
$COMP    030060    816    2500E
$CRC     052316   1445    2455L
$CRC1    052321   2457L   2477
$CRC2    052347   2468    2475L
$CVD     052054   2140L
$CVD.    052055   2020    2061   2141L
$DADA    030072   2562E   2695
$DADA.   030101   1914    2058   2572E
$DNS     051237   1972L
$DNS.    051235    878    1971L
$DNS1    051255   1983L   1988   1991
$DNV     051274   1980    2013L
$DNV1    051301   2019L   2022
$DNV2    051344   2035    2038   2043L
$DNV3    051347   2047L   2062
$DNV4    052007   2051    2057   2064L
$DNVA    052011   2013    2042   2047   2067L
$DNVB    052012   2033    2068L
$DRS     052064    856    2194E
$DRS1    052064   2198L   2203   2268
$DRS15   052230   2222    2244   2280L
$DRS2    052101   2209L   2230
$DRS20   052243   2227    2246   2302L  2305
$DRS3    052104   2210L   2217
$DRS4    052123   2214    2221L
$DRS4.5  052137   2223    2227L
$DRS5    052147   2234L   2245
$DRS6    052157   2212    2226   2242L
$DRS7    052206   2247    2257L
$DRS8    052213   2263L   2266
$DRSB    052261   2204    2209   2235   2258   2314L
$DU66    030106   2585E   2701
$HLIHL   030211   1472    2428E
$MCU     052356   2513L   2602
$MLU     053000   2597L   2650
$MLU1    053003   2600L   2605
$MOVE    030252    979    2370E
$MU86    031007   2049    2127E
$RCHAR   052367   2545L   2546   2654
$RSTALL  031047   2386    2403E
$RTL     053036   2648    2652E
$RTL.    053027    729    1066   1162   1689   2648L
$RTL1    053037   2654L   2660
$RTL2    053071   2656    2677L
$SAVALL  031054   2382    2417E
$SOB     052263   2242    2327L
$SOB1    052264   2328L   2331   2333
$TBLS    052023   2036    2093L
$TOD     053147   1148    1150   1154   2753L
$TYPC.   053024   2628L   2763
$TYPCH   053020   2618L
$TYPTX   031136    616     628    671    726    924    933    960    970   1063   1106   1112   1151
                  1155    1204   1223   1256   1267   1391   1397   2533E
$TYPTX.  031144   2535E
$UDDN    053074   1019    2694E
```

```
$WCHAR  052375  2549L
$ZERO   031212  1618    2741E
.CHFLG  000060   251L
.CLEAN  000205   266L
.CLEAR  000055   248L    659
.CLEARA 000056   249L    657
.CLOSE  000046   241L    601    826
.CLRCO  000007   225L
.CONSL  000006   224L   2385
.CTLC   000041   236L    593    697
.DAD    000206   267L
.DECODE 000053   246L    740
.DELET  000050   243L
.DISMT  000061   252L
.DMNMS  000203   264L
.DMOUN  000201   262L
.ERROR  000057   250L    620
.EXIT   000000   218L    559    704
.LINK   000040   235L
.LOADD  000062   253L
.LOADO  000010   226L
.MONMS  000202   263L
.MOUNT  000200   261L
.NAME   000054   247L
.OPEN   000063   254L
.OPENC  000045   240L
.OPENR  000042   237L    750
.OPENU  000044   239L    830
.OPENW  000043   238L
.PAT.   053371  2857L
.PAT1.  053270  2843L
.POSIT  000047   242L   1651
.PRINT  000003   221L   1021
.READ   000004   222L    760    810    1610   1860
.RENAM  000051   244L
.RESET  000204   265L
.SCIN   000001   219L   2545
.SCOUT  000002   220L   2549   2628
.SETTP  000052   245L    690
.SYS.   000001     3E    562    851    866    889    1235   1347   1683   1764   2837   2864
.SYSRES 000012   228L
.VERS   000011   227L    652
.WRITE  000005   223L   1540   1549   1893   1945
ABL     047170  1192    1194   1196   1374L
ABL1    047273  1387    1397L
ABS.COD 000010   522L    535    787
ABS.ENT 000006   520L
ABS.ID  000000   516L
ABS.LDA 000002   518L    784
ABS.LEN 000004   519L    790   1886
AEE     047324   811    1416L  1530   1605   1611
AIO.CGN 041047   451L
AIO.CHA 041116   466L
AIO.CNT 041111   462L
AIO.CSI 041050   452L
AIO.DDA 041041   447E
AIO.DES 041055   456L
AIO.DEV 041057   457L
```

```
AIO.DIR 041062    460L
AIO.DTA 041053    455L
AIO.EOF 041113    464L
AIO.EOM 041112    463L
AIO.FLG 041043    448L
AIO.GRT 041044    449L
AIO.LGN 041051    453L
AIO.LSI 041052    454L
AIO.SPG 041046    450L
AIO.TFP 041114    465L
AIO.UNI 041061    458L
AIO.VEC 041040    446L
APP     046021    579     1139E   1200    1206
APP4    046030    1146L
APP5    046112    1177L   1180
APP6    046132    1168    1190L
APP7    046165    1184    1204L
BELL    000007    478E     617     934     971    1044    1045    1107    1113    1205    1257    1268    1392
                  1398
BKSP    000010    480E
BOOT.P  000001    426E
BUFFER  056000    758      762     779     784     790    1858    1881    1886    1891    2863L
C.STX   000002    482E
C.SYN   000026    481E
CBS     047335    1233    1440L   1738
CBS1    047340    1441L   1448
CCHIT   042361    627L     695
CDB.H84 000001    369E
CDB.H85 000000    368E
CFU     047354    1293    1470L
CN.FIL  000000     45E     600     749     759     809     825     827    1471    1539    1548    1609    1650
                  1859    1892    1944
CO.FLG  000001    346E    2384
CPA     053233    1101    1140    1146    1191    1193    1197    1199    2812L
CPC     046212    586     1223L   1258
CPC2    046355    1255    1262L
CPCA    047077    1234    1264    1273L
CR      000015    474E
CS.FLG  000200    347E
CSL.CHR 000001    323E
CSL.ECH 000200    320E
CSL.RAW 000004    321E
CSL.WRP 000002    322E
CTLA    000001    489E
CTLB    000002    490E
CTLC    000003    491E     592     696
CTLD    000004    492E    2655
CTLO    000017    493E
CTLP    000020    494E
CTLQ    000021    495E
CTLS    000023    496E
CTLZ    000032    497E
CTP.2SB 000010    332E
CTP.BKM 000002    333E
CTP.BKS 000200    328E
CTP.FF  000100    329E
CTP.MLI 000040    330E
CTP.MLO 000020    331E
```

```
CTP.TAB 000001  334E
D.CON   040110  282L
D.RAM   040240  285L
D.VEC   040130  284L
DCS     044137  734   849L
DCSA    044160  855   862L
DEFALT  053240  739   747   828   2819L
DEV.DDA 000004  175L
DEV.DVG 000015  188L
DEV.DVL 000013  187L
DEV.FLG 000006  176L
DEV.JMP 000003  174L
DEV.MNU 000010  184L
DEV.MUM 000007  183L
DEV.NAM 000000  166L
DEV.RES 000002  170L
DEV.UNT 000011  185L
DEVELEN 000016  190E
DF.CLR  000376  106E
DF.EMP  000377  105E
DIR.ALD 000025  121L
DIR.CLU 000015  114L
DIR.CRD 000023  120L
DIR.EXT 000010  109L
DIR.FGN 000020  117L
DIR.FLG 000016  115L
DIR.LGN 000021  118L
DIR.LSI 000022  119L
DIR.NAM 000000  108L
DIR.PRO 000013  110L
DIR.VER 000014  111L
DIRELEN 000027  123E   155   460
DIRIDL  000015  112E
DOB     047373  1084  1087  1183  1492L
DOB1    047376  1493L 1508
DR.IM   000001  171E
DR.PR   000002  172E
DT.CH   000020  181E
DT.CR   000002  178E
DT.CW   000004  179E   744
DT.DD   000001  177E   744
DT.RN   000010  180E
DV.EL   000000  167E
DV.NU   000001  168E
EC.CNA  000004   58L
EC.DDA  000027   77L
EC.DIF  000017   69L
EC.DIW  000035   83L
EC.DNI  000045   91L
EC.DNR  000046   92L
EC.DNS  000005   59L   745
EC.DSC  000047   93L
EC.EOF  000001   55L  1418
EC.EOM  000002   56L
EC.FAO  000031   79L
EC.FAP  000026   76L
EC.FL   000030   78L
EC.FNF  000014   66L
```

```
EC.FNO  000011    63L
EC.FNR  000034    82L
EC.FOD  000043    89L
EC.FUC  000013    65L
EC.ICN  000016    68L
EC.IDN  000006    60L
EC.IFC  000020    70L     764     771
EC.IFN  000007    61L
EC.ILC  000003    57L
EC.ILO  000040    86L
EC.ILR  000012    64L
EC.ILV  000037    85L
EC.IOI  000052    96L
EC.IS   000032    80L     879    2237
EC.NCV  000050    94L     700
EC.NEM  000021    71L
EC.NOS  000051    95L
EC.NPM  000044    90L
EC.NRD  000010    62L
EC.NVM  000042    88L
EC.OTL  000053    97L
EC.RF   000022    72L
EC.UNA  000036    84L
EC.UND  000015    67L
EC.UUN  000033    81L
EC.VPM  000041    87L
EC.WF   000023    73L
EC.WP   000025    75L
EC.WPV  000024    74L
ENL     000212   487E     672     934     971    1045    1107    1113    1205    1257    1269    1392    1398
EPF     047101   594     1290E
EPF1    047110  1297L
EPF2    047120  1301L    1331
EPF3    047151  1320     1324L
ERROR   042324   611L     691     741     746     751     761     765     772     831     857     880    1419
                1541     1550    1856    1861    1889    1895    1941    1947
ERROR.  042332   611      614L    702
ESC     000033   485E
FF      000014   488E    1743    1746    2844    2849
FILSIZ  053266   792     2835L
FILTYP  053252   767     1875    2822L
FT.ABS  000000   530E     536     768    1876
FT.BAC  000003   533E
FT.DD   000001   134E
FT.OC   000020   138E
FT.OR   000002   135E
FT.OU   000010   137E    1475
FT.OW   000004   136E    1475
FT.PIC  000001   531E     770
FT.REL  000002   532E
FVB     050023   595     1523L   1542    1596
FVB1    050067  1531     1546L
GPA     045215   577     1063L   1108    1114
GPA1    045246  1070L    1073
GPA2    045256  1078L    1081
GPA3    045363  1085     1088    1112L
GPC     044315   572      958L    972
GPC1    045022   965      976L
```

```
GPC3     045046    986L    1035
GPC4     045065   1003L    1007    1028
GPC5     045136    989     1005    1032L
GPCA     045154   1020     1044L
GPCB     045176   1017     1045L
GPCC     045214    959     1023    1039    1047L
GPI      044211    570      924L    935
GPI2     044305    929      939L
I.CONFL  000004    349E     350    2383
I.CONTY  000001    336E     337
I.CONWI  000003    342E     343
I.CSLMD  000000    325E
I.CUSOR  000002    339E     340
IOC.CGN  000010    143L
IOC.CSI  000011    144L
IOC.DDA  000002    131L     139     153
IOC.DES  000016    150L
IOC.DEV  000020    151L
IOC.DIL  000021    153E
IOC.DIR  000023    155L
IOC.DRL  000010    147E
IOC.DTA  000014    149L
IOC.FLG  000004    133L     147    1473
IOC.GRT  000005    141L
IOC.LGN  000012    145L
IOC.LNK  000000    130L
IOC.LSI  000013    146L
IOC.SPG  000007    142L
IOC.SQL  000003    139E
IOC.UNI  000022    152L
IOCCTD   000001    159E
IOCELEN  000052    157E
LBF      050111   1141    1324    1573L
LBF1     050145   1587    1596L
LBF4     050150   1584    1600L
LBF5     050206   1606    1616L
LBF6     050216   1612    1622L
LF       000012    475E
LINE     054071    728     748     829     858     939     976    1065    1068    1161    1165    1173    1262
                  1688    1695    1734    2859L
MEML     053371    538    2855E
NL       000012    486E     487     617     619     672     672    1064    1224    1269    1743    2659
NUL2     000000    477E
NULL     000200    476E
OFP      043144    556     726L
OFP1     043332    789     784L
OFP1.5   043355    780     792L
OFP2     044051    805     812     823L
OFPA     044077    738     742     835L
OVL.IN   000001    393E
OVL.NUM  000014    395E
OVL.RES  000002    394E
OVL.UCS  000200    396E
PATCH    042200    547E
PATCH1   042214    553     556L
PATCH2   042225    557     561E
PATCH3   042244    569     577L    580
PATCH4   042252    579L
```

```
PATCH5   042260   578    584L
PATCHID  053260   940    1907   2828L
PATLIST  060000   666    1225   1297   2820   2872E
PATPRQ   053253   978    983    2826L  2827
PATPRQL  000005   950    962    977    985    2827E
PGMFWA   053262   778    785    794    1867   2833L
PGMLWA   053264   797    1298   1323   1865   1933   2834L
PHIST    053261   567    584    597    665    818    824    1291   2832L
PHT      057000   808    814    984    1913   1943   2866L
PHT.CNT  000022   2797E
PHT.DAT  000011   2795E
PHT.HDR  000000   813    2791E
PHT.HIS  000023   984    1913   2799E
PHTFORM  053175   813    2790L  2791   2794   2795   2797   2799   2801
PHTHDRL  000011   815    2794E
PHTL     000033   2801E
PIC.COD  000006   509L
PIC.ID   000000   504L
PIC.LEN  000002   506L   779    1881
PIC.PTR  000004   507L
PLMAX    053250   689    1379   2821L
PLPTR    053246   667    1226   1301   1374   1377   2820L
PPF      050235   756    804    1529   1604   1644L  1855   1888   1940
PRS      042372   552    648E
PRSERR   043132   702L
PRSERR1  043130   653    655    700L
QUOTE    000047   483E
RES      050253   927    963    1253   1682E
RES1     050276   1701L  1730
RES2     050340   1726   1728L
RESA     050364   1697   1727   1745L
RESAE    051024   1725   1746L
RESTART  042200   549E   554    612    630    1041   1399
ROMBOOT  030000   277E
RUBOUT   000177   479E
S.BAUD   040344   370L
S.BDA    041120   468L
S.BOOTF  041034   425L
S.CAADR  040333   353L
S.CACC   041006   409L
S.CCTAB  040335   354L
S.CDB    040343   367L
S.CFWA   040352   377L   1470
S.CODE   041007   410L
S.CONFL  040332   351L
S.CONTY  040327   338L
S.CONWI  040331   344L
S.CSLMD  040326   326L   337    340    343    350
S.CUSDR  040330   341L
S.DATC   040310   307L
S.DATE   040277   306L
S.DCS    041033   423L
S.DDDTA  040366   388L
S.DDGRP  040364   385L
S.DDLDA  040360   383L
S.DDLEN  040362   384L
S.DDOPC  040370   389L
S.DFWA   040354   378L
```

```
S.DIREA 041016      417L
S.DLINK 040346      375L
S.FASER 041013      416L
S.FCI   041021      418L
S.GRT0  024000      273E
S.GRT1  025000      274E
S.GRT2  026000      275E
S.GUP   041027      420L
S.HIMEM 040316      309L
S.INT   040343      287L      363
S.JUMPS 041010      414L
S.MOUNT 041032      422L
S.OFWA  040350      376L
S.OMAX  040324      315L      678
S.OSN   041004      405L
S.OVLE  041000      402L
S.OVLFL 040371      398L
S.OVLS  040376      401L
S.OVSTK 041035      430L
S.RFWA  040356      379L
S.SCI   041024      419L
S.SCR   041121      469L
S.SDD   041010      415L
S.SOVR  041146      289L      291
S.SSN   041002      404L
S.SYSM  040320      311L      676
S.TIME  040312      308L
S.UCSF  040372      399L
S.UCSL  040374      400L
S.USRM  040322      313L
S.VAL   040277      286L      304
SDISP   053232      850       882     1644    2810L
SIVB    053235     1526      1580     1581    1624    2813L
SIVBA   053237     1329      1523     1552    2815L
SIVBV   053236      664      1581     1626    2814L
SKEW    053230      777       789      799    1094    1576    1935    2809L
STACK   042200      293E      551
STACKL  001032      291E
START   042200      539       548E
SW.DISP 044174      864       878L
SYDD    040130      283E
SYSCALL 000377      211E      559      593     601     620     652     657     659     690     697     704     740
                    750       760      810     826     830    1021    1540    1549    1610    1651    1860    1893    1945
                   2385      2545     2549    2628
TAB     000011      484E     2332
TBL1    052032     2097L     2103
TBL2    052050     2095      2107L
TBL3    052052     2100      2111L
TOD1    053153     2757L     2766
UBH     051025      598      1854L
UBH1    051107     1877      1886L
UBH2    051112     1882      1887L
UDDN1   053100     2698L     2714
UDDN1.5 053132     2718L     2725
UDDN2   053134     2711      2723L
UDDN3   053135     2724L     2728
UNT.DIS 000006      200L
UNT.FLG 000000      196L
```

PATCH -- PATCH SYSTEM AND USER FILES XREF V1.1

CROSS REFERENCE TABLE PAGE 70

```
UNT.GRT 000002    198L
UNT.GTS 000004    199L
UNT.SIZ 000010    202E
UNT.SPG 000001    197L
USERFWA 042200    294E    535    537    538
VERS    000040    209E    654
VIEWBFR 055000   1547    1589   1608   1617   1627   2861L  2862
WPH     051141    599    1907L
WPH.    051175   1933L
WPH1    051166   1923L   1925
```

23294 BYTES FREE