```
                        ***  EDIT - HEATH HDOS TEXT EDITOR.
                        *
                        *    ADAPTED FROM 'HOS8 - WINTEK TEXT EDITOR'
                        *
                        *    J. G. LETWIN, 12/12/77,  FOR *HEATH* CORPORATION
                        *
                        *    COPYRIGHT 12/1977, 79/05        BY *HEATH* CORPORATION.
                        *
                        *    G. Chandler             79/05   --.04.--
                        *                            79/12   --.05.--
                        *                            80/02
                        *
                        ***  EDIT PERFORMS SIMPLE LINE AND STRING EDITING FUNCTIONS.
                        *
                        *    SEE THE 'EDIT' USERS MANUAL FOR INSTRUCTIONS.
```

```
                        22  ****    ASSEMBLY CONSTANTS.
000.000                 23          XTEXT   ASCII


                        25X **      ASCII CHARACTER EQUIVALENCES.
                        26X
000.015                 27X CR      EQU     13              CARRIAGE RETURN
000.012                 28X LF      EQU     10              LINE FEED
000.200                 29X NULL    EQU     200Q            PAD CHARACTER
000.000                 30X NUL2    EQU     0
000.007                 31X BELL    EQU     7               BELL CHARACTER
000.177                 32X RUBOUT  EQU     177Q
000.010                 33X BKSP    EQU     10Q             CTL-H
000.026                 34X C.SYN   EQU     26Q             SYNC
000.002                 35X C.STX   EQU     2               STX
000.047                 36X QUOTE   EQU     47Q
000.011                 37X TAB     EQU     11Q
000.033                 38X ESC     EQU     33Q
000.012                 39X NL      EQU     12Q             NEW LINE (HDOS SYSTEMS)
000.212                 40X ENL     EQU     NL+200Q         NL + END-OF-LINE-FLAG
000.014                 41X FF      EQU     14Q             FORM FEED
000.001                 42X CTLA    EQU     01Q             CTL-A
000.002                 43X CTLB    EQU     02Q             CTL-B
000.003                 44X CTLC    EQU     03Q             CTL-C
000.004                 45X CTLD    EQU     04Q             CTL-D
000.017                 46X CTLO    EQU     17Q             CTL-O
000.020                 47X CTLP    EQU     20Q             CTL-P
000.021                 48X CTLQ    EQU     21Q             CTL-Q
000.023                 49X CTLS    EQU     23Q             CTL-S
000.032                 50X CTLZ    EQU     32Q             CTL-Z


                        52  **      COMMAND OPTIONS.
                        53
000.001                 54  OPT.A   EQU     1               PRINT LINE AFTER PROCESS
000.002                 55  OPT.B   EQU     2               PRINT LINE BEFORE PROCESSING


                        57  **      MACHINE INSTRUCTIONS.
                        58
000.072                 59  MI.LDA  EQU     072Q
000.000                 60  MI.NOP  EQU     000Q
000.311                 61  MI.RET  EQU     311Q
                        62
                        63  ****
```

```
000.000                 65          XTEXT   FBDEF

                        67X **      FILE BLOCK DEFINITIONS.
                        68X
000.000                 69X         ORG     0
000.000                 70X FB.CHA  DS      1           CHANNEL NUMBER
000.001                 71X FB.FLG  DS      1           FLAGS
000.002                 72X FB.FWA  DS      2           BUFFER FWA
000.004                 73X FB.PTR  DS      2           BUFFER POINTER
000.006                 74X FB.LIM  DS      2           LIMIT OF DATA IN BUFFER (READ OPERATIONS)
000.010                 75X FB.LWA  DS      2           LWA OF BUFFER
000.012                 76X FB.NAM  DS      4+8+4+1     NAME OF FILE
000.021                 77X FB.NAML EQU     *-FB.NAM
000.033                 78X FBENL   EQU     *           ENTRY LENGTH
000.033                 79          XTEXT   HOSDEF

                        81X **      HOSDEF - DEFINE HOS PARAMETER.
                        82X *
                        83X
                        84X
000.026                 85X VERS    EQU     1*16+6      VERSION 1.6
                        86X
000.377                 87X SYSCALL EQU     377Q        SYSCALL INSTRUCTION
                        88X
                        89X
000.000                 90X         ORG     0
                        91X
                        92X *       RESIDENT FUNCTIONS
                        93X
000.000                 94X .EXIT   DS      1           EXIT (MUST BE FIRST)
000.001                 95X .SCIN   DS      1           SCIN
000.002                 96X .SCOUT  DS      1           SCOUT
000.003                 97X .PRINT  DS      1           PRINT
000.004                 98X .READ   DS      1           READ
000.005                 99X .WRITE  DS      1           WRITE
000.006                100X .CONSL  DS      1           SET/CLEAR CONSOLE OPTIONS
000.007                101X .CLRCO  DS      1           CLEAR CONSOLE BUFFER
000.010                102X .LOADO  DS      1           LOAD AN OVERLAY
000.011                103X .VERS   DS      1           RETURN HDOS VERSION NUMBER
000.012                104X .SYSRES DS      1           PRECEDING FUNCTIONS ARE RESIDENT
                       105X
                       106X
                       107X *       *HDOSOVLO.SYS*  FUNCTIONS
                       108X
000.040                109X         ORG     40A
                       110X
000.040                111X .LINK   DS      1           LINK  (MUST BE FIRST)
000.041                112X .CTLC   DS      1           CTL-C
000.042                113X .OPENR  DS      1           OPENR
000.043                114X .OPENW  DS      1           OPENW
000.044                115X .OPENU  DS      1           OPENU
```

```
000.045                116X .OPENC  DS      1               OPENC
000.046                117X .CLOSE  DS      1               CLOSE
000.047                118X .POSIT  DS      1               POSITION
000.050                119X .DELET  DS      1               DELETE
000.051                120X .RENAM  DS      1               RENAME
000.052                121X .SETTP  DS      1               SETTOP
000.053                122X .DECODE DS      1               NAME DECODE
000.054                123X .NAME   DS      1               GET FILE NAME FROM CHANNEL
000.055                124X .CLEAR  DS      1               CLEAR CHAN
000.056                125X .CLEARA DS      1               CLEAR ALL CHANS
000.057                126X .ERROR  DS      1               LOOKUP ERROR
000.060                127X .CHFLG  DS      1               CHANGE FLAGS
000.061                128X .DISMT  DS      1               FLAG SYSTEM DISK DISMOUNTED
000.062                129X .LOADD  DS      1               LOAD DEVICE DRIVER
                       130X
                       131X
                       132X *       *HDOSOVL1.SYS*  FUNCTIONS
                       133X
000.200                134X         ORG     200Q
                       135X
000.200                136X .MOUNT  DS      1               MOUNT  (MUST BE FIRST)
000.201                137X .DMOUN  DS      1               DISMOUNT
000.202                138X .MONMS  DS      1               MOUNT/NO MESSAGE
000.203                139X .DMNMS  DS      1               DISMOUNT/NO MESSAGE
000.204                140X .RESET  DS      1               RESET = DISMOUNT/MOUNT OF UNIT
000.205                141          XTEXT   HOSEQU


                       143X **      HDOS SYSTEM EQUIVALENCES.
                       144X *
                       145X
024.000                146X S.GRT0  EQU     24000A          SYSTEM AREA FOR  GRT0
025.000                147X S.GRT1  EQU     25000A          SYSTEM AREA FOR  GRT1
026.000                148X S.GRT2  EQU     26000A          SYSTEM AREA FOR  GRT2
                       149X
030.000                150X ROMBOOT EQU     30000A          ROM BOOT ENTRY
                       151X
040.100                152X         ORG     40100A          FREE SPACE FROM PAM-8
                       153X
040.100                154X         DS      8               JUMP TO SYSTEM EXIT
040.110                155X D.CON   DS      16              DISK CONSTANTS
040.130                156X SYDD    EQU     *               SYSTEM DISK ENTRY POINT
040.130                157X D.VEC   DS      24*3            SYSTEM ROM ENTRY VECTORS
040.240                158X D.RAM   DS      31              SYSTEM ROM WORK AREA
040.277                159X S.VAL   DS      36              SYSTEM VALUES
040.343                160X S.INT   DS      115             SYSTEM INTERNAL WORK AREAS
041.126                161X         DS      16
041.146                162X S.SOVR  DS      2               STACK OVERFLOW WARNING
041.150                163X         DS      42200A-*        SYSTEM STACK
001.032                164X STACKL  EQU     *-S.SOVR        STACK SIZE
                       165X
042.200                166X STACK   EQU     *               LWA+1 SYSTEM STACK
042.200                167X USERFWA EQU     *               USER FWA
042.200                168          XTEXT   ESVAL
```

```
                    170X **     S.VAL - SYSTEM VALUE DEFINTIONS.
                    171X *
                    172X *      THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEN.
                    173X *
                    174X *      THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.
                    175X
                    176X
040,277             177X        ORG     S.VAL
                    178X
040,277             179X S.DATE DS      9           SYSTEM DATE (IN ASCII)
040.310             180X S.DATC DS      2           CODED DATE
040.312             181X S.TIME DS      4           TIME FROM MIDNIGHT (IN TICS)
040.316             182X S.HIMEM DS     2           HARDWARE HIGH MEMORY ADRESS+1
                    183X
040.320             184X S.SYSM DS      2           FWA RESIDENT SYSTEM
                    185X
040.322             186X S.USRM DS      2           LWA USER MEMORY
                    187X
040.324             188X S.OMAX DS      2           MAX OVERLAY SIZE FOR SYSTEM
                    189X
                    190X
                    191X **     THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL
                    192X
000,200             193X CSL.ECH EQU    10000000B   SUPPRESS ECHO
000.002             194X CSL.WRP EQU    00000010B   WRAP LINES AT WIDTH
000,001             195X CSL.CHR EQU    00000001B   OPERATE IN CHARACTER MODE
                    196X
000,000             197X I.CSLMD EQU    0           S.CSLMD IS FIRST BYTE
040.326             198X S.CSLMD DS     1           CONSOLE MODE
                    199X
000.200             200X CTP.BKS EQU    10000000B   TERMINAL PROCESSES BACKSPACES
000.040             201X CTP.MLI EQU    00100000B   MAP LOWER CASE TO UPPER ON INPUT
000.020             202X CTP.MLO EQU    00010000B   MAP LOWER CASE TO UPPER ON OUTPUT
000.010             203X CTP.2SB EQU    00001000B   TERMINAL NEEDS TWO STOP BITS
000.002             204X CTP.BKM EQU    00000010B   MAP BKSP (UPON INPUT) TO RUBOUT
000.001             205X CTP.TAB EQU    00000001B   TERMINAL SUPPORTS TAB CHARACTERS
                    206X
000.001             207X I.CONTY EQU    1           S.CONTY IS 2ND BYTE
000.000             208X        ERRNZ   *-S.CSLMD-I.CONTY
040.327             209X S.CONTY DS     1           CONSOLE TYPE FLAGS
000.002             210X I.CUSOR EQU    2           S.CUSOR IS 3RD BYTE
000.000             211X        ERRNZ   *-S.CSLMD-I.CUSOR
040.330             212X S.CUSOR DS     1           CURRENT CURSOR POSITION
000.003             213X I.CONWI EQU    3           S.CONWI IS 4TH BYTE
000.000             214X        ERRNZ   *-S.CSLMD-I.CONWI
040.331             215X S.CONWI DS     1           CONSOLE WIDTH
                    216X
000.001             217X CO.FLG EQU     00000001B   CTL-O FLAG
000.200             218X CS.FLG EQU     10000000B   CTL-S FLAG
                    219X
000.004             220X I.CONFL EQU    4           S.CONFL IS 5TH BYTE
000.000             221X        ERRNZ   *-S.CSLMD-I.CONFL
040.332             222X S.CONFL DS     1           CONSOLE FLAGS
                    223X
040.333             224X S.CAADR DS     2           ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040.335             225X S.CCTAB DS     6           ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
```

```
040.343                 226           XTEXT   ABSDEF

                        228X **       ABS FORMAT EQUIVALENCES.
                        229X
000.000                 230X          ORG     0
                        231X
000.000                 232X ABS.ID   DS      1       3770 = BINARY FILE FLAG
000.001                 233X          DS      1       FILE TYPE (FT.ABS)
000.002                 234X ABS.LDA  DS      2       LOAD ADDRESS
000.004                 235X ABS.LEN  DS      2       LENGTH OF ENTIRE RECORD
000.006                 236X ABS.ENT  DS      2       ENTRY POINT
                        237X
000.010                 238X ABS.COD  DS      0       CODE STARTS HERE
000.010                 239           XTEXT   ECDEF


                        241X **       ERROR CODE DEFINITIONS,
                        242X
000.000                 243X          ORG     0
000.000                 244X          DS      1       NO ERROR #0
000.001                 245X EC.EOF   DS      1       END OF FILE
000.002                 246X EC.EOM   DS      1       END OF MEDIA
000.003                 247X EC.ILC   DS      1       ILLEGAL SYSCALL CODE
000.004                 248X EC.CNA   DS      1       CHANNEL NOT AVAILABLE
000.005                 249X EC.DNS   DS      1       DEVICE NOT SUITABLE
000.006                 250X EC.IDN   DS      1       ILLEGAL DEVICE NAME
000.007                 251X EC.IFN   DS      1       ILLEGAL FILE NAME
000.010                 252X EC.NRD   DS      1       NO ROOM FOR DEVICE DRIVER
000.011                 253X EC.FNO   DS      1       CHANNEL NOT OPEN
000.012                 254X EC.ILR   DS      1       ILLEGAL REQUEST
000.013                 255X EC.FUC   DS      1       FILE USAGE CONFLICT
000.014                 256X EC.FNF   DS      1       FILE NAME NOT FOUND
000.015                 257X EC.UND   DS      1       UNKNOWN DEVICE
000.016                 258X EC.ICN   DS      1       ILLEGAL CHANNEL NUMBER
000.017                 259X EC.DIF   DS      1       DIRECTORY FULL
000.020                 260X EC.IFC   DS      1       ILLEGAL FILE CONTENTS
000.021                 261X EC.NEM   DS      1       NOT ENOUGH MEMORY
000.022                 262X EC.RF    DS      1       READ FAILURE
000.023                 263X EC.WF    DS      1       WRITE FAILURE
000.024                 264X EC.WPV   DS      1       WRITE PROTECTION VIOLATION
000.025                 265X EC.WP    DS      1       DISK WRITE PROTECTED
000.026                 266X EC.FAP   DS      1       FILE ALREADY PRESENT
000.027                 267X EC.DDA   DS      1       DEVICE DRIVER ABORT
000.030                 268X EC.FL    DS      1       FILE LOCKED
000.031                 269X EC.FAO   DS      1       FILE ALREADY OPEN
000.032                 270X EC.IS    DS      1       ILLEGAL SWITCH
000.033                 271X EC.UUN   DS      1       UNKNOWN UNIT NUMBER
000.034                 272X EC.FNR   DS      1       FILE NAME REQUIRED
000.035                 273X EC.DIW   DS      1       DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036                 274X EC.UNA   DS      1       UNIT NOT AVAILABLE
000.037                 275X EC.ILV   DS      1       ILLEGAL VALUE
000.040                 276X EC.ILO   DS      1       ILLEGAL OPTION
000.041                 277X EC.VPM   DS      1       VOLUME PRESENTLY MOUNTED ON DEVICE
```

```
000.042                 278X EC.NVM  DS      1               NO VOLUME PRESENTLY MOUNTED
000.043                 279X EC.FOD  DS      1               FILE OPEN ON DEVICE
000.044                 280X EC.NPM  DS      1               NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045                 281X EC.DNI  DS      1               DISK NOT INITIALIZED
000.046                 282X EC.DNR  DS      1               DISK IS NOT READABLE
000.047                 283X EC.DSC  DS      1               DISK STRUCTURE IS CORRUPT
000.050                 284X EC.NCV  DS      1               NOT CORRECT VERSION OF HDOS
000.051                 285X EC.NOS  DS      1               NO OPERATING SYSTEM MOUNTED
000.052                 286X EC.IOI  DS      1               ILLEGAL OVERLAY INDEX
000.053                 287X EC.OTL  DS      1               OVERLAY TO LARGE
000.054                 288          XTEXT   FILDEF

                        290X **      FILDEF - FILE TYPE DEFINITIONS.
                        291X *
                        292X *       DB      377Q,FT.XXX
                        293X
                        294X
000.000                 295X FT.ABS  EQU     0               ABSOLUTE BINARY
000.001                 296X FT.PIC  EQU     1               POSITION INDEPENDANT CODE
000.002                 297X FT.REL  EQU     2               RELOCATABLE CODE
000.003                 298X FT.BAC  EQU     3               COMPILED BASIC CODE
000.054                 299          XTEXT   DIRDEF

                        301X **      DIRECTORY ENTRY FORMAT.
                        302X
000.000                 303X         ORG     0
                        304X
                        305X
000.377                 306X DF.EMP  EQU     377Q            FLAGS ENTRY EMPTY
000.376                 307X DF.CLR  EQU     376Q            FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
                        308X
000.000                 309X DIR.NAM DS      8               NAME
000.010                 310X DIR.EXT DS      3               EXTENSION
000.013                 311X DIR.PRO DS      1               PROJECT
000.014                 312X DIR.VER DS      1               VERSION
000.015                 313X DIRIDL  EQU     *               FILE IDENTIFICATION LENGTH
                        314X
000.015                 315X DIR.CLU DS      1               CLUSTER FACTOR
000.016                 316X DIR.FLG DS      1               FLAGS
000.017                 317X         DS      1               RESERVED
000.020                 318X DIR.FGN DS      1               FIRST GROUP NUMBER
000.021                 319X DIR.LGN DS      1               LAST GROUP NUMBER
000.022                 320X DIR.LSI DS      1               LAST SECTOR INDEX (IN LAST GROUP)
000.023                 321X DIR.CRD DS      2               CREATION DATE
000.025                 322X DIR.ALD DS      2               LAST ALTERATION DATE
                        323X
000.027                 324X DIRELEN EQU     *               DIRECTORY ENTRY LENGTH
000.027                 325          XTEXT   OVLDEF
```

```
                    327X **     OVERLAY TABLE ENTRYS.
                    328X
000.000             329X        ORG     0
                    330X
000.000             331X OVL.COD DS     2           FIRST SECTOR OF OVERLAY CODE
000.002             332X OVL.SIZ DS     2           OVERLAY SIZE
000.004             333X OVL.ENT DS     2           OVERLAY ENTRY POINT
000.006             334X OVL.FLB DS     1           OVERLAY FLAG BYTE
000.007             335X         DS     1           DUMMY BYTE TO ROUND TABLE SIZE UP TO 8
000.010             336X OVL.ENS EQU    *           OVERLAY ENTRY SIZE
                    337X
                    338X *      OVERLAY INDICES
                    339X
000.000             340X        ORG     0
                    341X
000.000             342X OVL0    DS     1
000.001             343X OVL1    DS     1
000.002             344          XTEXT  IOCDEF


                    346X **     I/O CHANNEL DEFINITIONS.
                    347X
000.000             348X        ORG     0
                    349X
000.000             350X IOC.LNK DS     2           ADDRESS OF NEXT CHANNEL, =0 IF LAST
000.002             351X IOC.DDA DS     2           THREAD JUMP TO DEVICE DRIVER (VIA DEV TABLE)
                    352X
000.004             353X IOC.FLG DS     1           FILE TYPE FLAGS
000.001             354X FT.DD   EQU    00000001B   =1 IF DIRECTORY DEVICE
000.002             355X FT.OR   EQU    00000010B   =1 IF OPEN FOR READ
000.004             356X FT.OW   EQU    00000100B   =1 IF OPEN FOR WRITE
000.010             357X FT.OU   EQU    00001000B   =1 IF OPEN FOR UPDATE
000.003             358X IOC.SQL EQU    *-IOC.DDA   LENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
                    359X
000.005             360X IOC.GRT DS     2           ADDRESS OF GROUP RESERVATION TABLE
000.007             361X IOC.SPG DS     1           SECTORS PER GROUP, THIS DEVICE
000.010             362X IOC.CGN DS     1           CURRENT GROUP NUMBER
000.011             363X IOC.CSI DS     1           CURRENT SECTOR INDEX (IN CURRENT GROUP)
000.012             364X IOC.LGN DS     1           LAST GROUP NUMBER
000.013             365X IOC.LSI DS     1           LAST SECTOR INDEX (IN LAST GROUP)
000.010             366X IOC.DRL EQU    *-IOC.FLG   LENGTH OF INFO NORMALLY COPIED BACK TO
                    367X *                          THE CHANNEL TABLE
000.014             368X IOC.DTA DS     2           DEVICE TABLE ADDRESS FOR THIS DEVICE
000.016             369X IOC.DES DS     2           SECTOR NUMBER OF DIRECTORY ENTRY
000.020             370X IOC.DEV DS     2           DEVICE CODE
000.022             371X IOC.UNI DS     1           UNIT NUMBER (0-9)
000.021             372X IOC.DIL EQU    *-IOC.DDA   LENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)
                    373X
000.023             374X IOC.DIR DS     DIRELEN     DIRECTORY ENTRY
                    375X
000.052             376X IOCELEN EQU    *           IOC ENTRY LENGTH
                    377X
000.001             378X IOCCID  EQU    1           INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)
```

```
042.170                 380             ORG     USERFWA-ABS.COD
                        381
042.170  377 000       382             DB      377Q,FT.ABS     ABS HEADER
042.172  200 042       383             DW      USERFWA         ORG
042.174  176 017       384             DW      MEML-USERFWA    SIZE OF LOAD IMAGE
042.176  276 061       385             DW      ENTRY           ENTRY POINT
                        386
```

```
                                389
042.200                         390 START   EQU     *
042.200                         391 RESTART EQU     *               RESTART ADDRESS
                                392
                                393
                                394 *       ENTER HERE FOR RUBBOUT AND COMMANDS DONE.
                                395
042.200  076 201                396 EDIX    MVI     A,CSL.CHR+CSL.ECH       CHARACTER MODE, NO ECHO
042.202  062 326 040            397         STA     S.CSLMD         CLEAR TERMINAL CONTRL
042.205  373                    398         EI
042.206  315 153 052            399         CALL    CBE             CHECK FOR BUFFER EMPTY
042.211  315 333 053            400         CALL    MAM             SET MAXIMUM MEMORY
042.214  315 000 055            401         CALL    $CCO            CLEAR CTL-O
042.217  315 171 055            402         CALL    $GNL            GUARANTEE NEW LINE
042.222  315 136 031            403         CALL    $TYPTX
042.225  055 255                404         DB      '-','-'+200Q
042.227  257                    405         XRA     A
042.230  062 277 061            406         STA     LINE            NULL LINE
042.233  062 136 061            407         STA     CCFLG           CLEAR CTL-C DISABLE FLAG
042.236  062 137 061            408         STA     CCPEND          CLEAR PENDING CTL-C
042.241  315 044 053            409         CALL    ECC             ENABLE CTL-C
042.244  257                    410         XRA     A
042.245  062 144 061            411 EDTO    STA     PROCHA          CLEAR PROBATATION CHARACTER
                                412
                                413 *       RE-ENTER HERE FOR BACKSPACE AND ILLEGAL CHARACTERS
                                414
042.250  041 277 061            415 EDI1    LXI     H,LINE
042.253  042 142 061            416         SHLD    LINPTR
042.256  257                    417         XRA     A
042.257  062 156 053            418         STA     ENCA            CLEAR HELD CHARACTER
042.262  061 200 042            419         LXI     SP,STACK        RESTORE STACK
                                420
                                421 *       DECODE COMMAND
                                422
042.265  315 066 043            423         CALL    DCR             DECODE COMMAND RANGE
042.270  315 072 044            424         CALL    DCN             DECODE COMMAND NAME
042.273  315 310 044            425         CALL    DCQ             DECODE COMMAND QUALIFIER
042.276  315 326 044            426         CALL    DCO             DECODE COMMAND OPTION
042.301  052 124 061            427         LHLD    CRFPTR
042.304  042 130 061            428         SHLD    WRKPTR
                                429
                                430 *       PROCESS COMMAND
                                431
042.307  072 054 063            432         LDA     CMDGRP          SEE WHICH GROUP IS COMMAND
042.312  247                    433         ANA     A
042.313  072 053 063            434         LDA     PATCNT
042.316  302 323 042            435         JNZ     EDI1.5          IS IN FULL RANGE
042.321  306 006                436         ADI     CMDDSP          IS IN NO-DATA GROUP
042.323  041 200 042            437 EDI1.5  LXI     H,EDIX
042.326  345                    438         PUSH    H               SET 'RETURN ADDRESS'
042.327  315 061 031            439         CALL    $TJMP           JUMP TO PROCESSOR
                                440
                                441 *       THE FOLLOWING COMMANDS MAY BE USED ONLY IF DATA PRESENT.
                                442
042.332                         443 CMDADR  DS      0               START OF TABLE
042.332  111 045                444         DW      PRINT           PRINT
```

```
042.334  206 045          445           DW      DELETE          DELETE
042.336  116 046          446           DW      EDITC           EDIT
042.340  022 046          447           DW      REPLAC          REPLACE
042.342  311 051          448           DW      WRITE           WRITE
042.344  132 045          449           DW      XPRINT          XPRINT                          /80.02.GC/
                          450
                          451 *         THE FOLLOWING COMMANDS MAY ALWAYS BE USED
                          452
000.006                   453 CMDDSP    EQU     *-CMDADR/2
042.346  371 044          454           DW      INSERT          INSERT
042.350  203 050          455           DW      READ            READ
042.352  052 046          456           DW      PURGE           PURGE
042.354  377 046          457           DW      FLUSH           FLUSH
042.356  146 050          458           DW      NEXT            NEXT
042.360  337 050          459           DW      SEARCH          SEARCH
042.362  041 047          460           DW      NEWIN           NEWIN
042.364  235 047          461           DW      NEWOUT          NEWOUT
042.366  001 050          462           DW      XOUT            XOUT                            /80.02.GC/
042.370  112 051          463           DW      USE             USE
042.372  025 047          464           DW      BYE             BYE


                          466 **        CTL-C INTERRUPT RECEIVED.
                          467 *
                          468
042.374  315 136 031      469 INTRPT    CALL    $TYPTX
042.377  136 303          470           DB      '^','C'+200Q
043.001  072 136 061      471           LDA     CCFLG
043.004  247              472           ANA     A
043.005  302 023 043      473           JNZ     INT1                                            /78.10.GC/
043.010  377 007          474           DB      SYSCALL+.CLRCO                                  /78.10.GC/
043.012  052 245 061      475           LHLD    XOUTFB+FB.FWA                                   /80.02.GC/
043.015  042 247 061      476           SHLD    XOUTFB+FB.PTR   ZERO THE  *XOUT*  BUFFER PTR.  /80.02.GC/
043.020  303 200 042      477           JMP     EDIX            CTL-C ALLOWED                   /78.10.GC/
                          478
043.023  076 001          479 INT1      MVI     A,1
043.025  062 137 061      480           STA     CCPEND          FLAG PENDING CTLC
043.030  311              481           RET                     DISCARD FOR NOW


                          483 **        REFUSE - REFUSE ENTERED CHARACTER.
                          484 *
                          485 *         REFUSE IS CALLED WHEN AN ILLEGAL ENTRY IS DETECTED.
                          486 *         IT TYPES A BELL, REMOVES THE LAST CHARACTER FROM THE INPUT
                          487 *         LINE, AND RE-PARSES THE COMMAND.
                          488
                          489
043.031  315 136 031      490 REFUSE    CALL    $TYPTX
043.034  207              491           DB      BELL+200Q
043.035  041 277 061      492           LXI     H,LINE
043.040  315 333 054      493           CALL    SNL             SCAN TO END
043.043  053              494           DCX     H               BACKSPACE TO LAST CHARACTER
```

```
043.044  053            495          DCX     H           HAVE ADVANCED PAST LAST CHARACTER
043.045  257            496          XRA     A
043.046  167            497          MOV     M,A
043.047  303 245 042     498          JMP     EDT0        CLEAR PROBATION (BAD) CHARACTER


                        500  **      EXIT - CTL-D STRUCK (END OF FILE ON CONSOLE)
                        501  *
                        502  *       SEE IF USER REALLY WANTS TO EXIT...
                        503
                        504
043.052  315 110 052    505  EXIT    CALL    AYS         ARE YOU SURE?
043.055  332 063 043    506          JC      EXIT1       CTL-D AGAIN
043.060  302 200 042    507          JNE     RESTART     NOT SURE
043.063  257            508  EXIT1   XRA     A
043.064  377 000         509          DB      SYSCALL,,EXIT   EXIT WITH EVERYTHING OPEN
```

```
                                513  **      DCR - DECODE COMMAND RANGE.
                                514  *
                                515  *       DCR IS CALLED TO DETERMINE THE COMMAND RANGE.
                                516  *
                                517  *       CAN BE EITHER
                                518  *
                                519  *       =                       PREVIOUS RANGE
                                520  *       ' '                     ALL TEXT
                                521  *       EXPR                    LINE EXPRESSION
                                522  *
                                523  *       ENTRY   NONE
                                524  *       EXIT    CRFPTR,CRLPTR,WRKPTR SETUP
                                525  *       USES    ALL
                                526
                                527
043.066                         528  DCR     EQU     *
043.066  052 132 061            529          LHLD    PCFPTR
043.071  042 124 061            530          SHLD    CRFPTR
043.074  052 134 061            531          LHLD    PCLPTR
043.077  042 126 061            532          SHLD    CRLPTR          SET DEFAULT RANGE TO RANGE OF PREVIOUS
043.102  174                    533          MOV     A,H
043.103  265                    534          ORA     L
043.104  310                    535          RZ                      IF NO DATA, DONT ALLOW RANGE
043.105  052 120 061            536          LHLD    FILPTR
043.110  315 064 053            537          CALL    ENC             EXAMINE NEXT CHARACTER
043.113  376 040                538          CPI     ' '
043.115  302 141 043            539          JNE     DCR1            NOT BLANK
                                540
                                541  *       IS BLANK, ENTIRE RANGE.
                                542
043.120  042 124 061            543          SHLD    CRFPTR
043.123  052 122 061            544          LHLD    LALPTR
043.126  174                    545          MOV     A,H
043.127  265                    546          ORA     L
043.130  304 322 054            547          CNZ     SLB             SCAN LINE BACKWARDS (IF ANY TEXT)
043.133  042 126 061            548          SHLD    CRLPTR
043.136  303 205 053            549          JMP     GNC             READ BLANK AND EXIT
                                550
043.141  376 075                551  DCR1    CPI     '='
043.143  312 205 053            552          JE      GNC             IS OLD RANGE, READ = AND EXIT
043.146  174                    553          MOV     A,H
043.147  265                    554          ORA     L
043.150  310                    555          RZ                      NO TEXT, DONT ALLOW EXPRESSION
                                556
                                557  *       MUST BE EXPRESSION
                                558
043.151  315 235 043            559          CALL    DRE             DECODE RANGE EXPRESSION
043.154  042 124 061            560          SHLD    CRFPTR          SET FIRST COMMAND
043.157  042 126 061            561          SHLD    CRLPTR          ASSUME IS ONE LINE COMMAND
043.162  315 064 053            562          CALL    ENC
043.165  376 054                563          CPI     ','
043.167  300                    564          RNE                     NO 2ND EXPRESSION
043.170  315 205 053            565          CALL    GNC             READ ,
043.173  345                    566          PUSH    H               SAVE BEGINNING OF RANGE
043.174  315 235 043            567          CALL    DRE             DECODE RANGE EXPRESSION
043.177  042 126 061            568          SHLD    CRLPTR          SET LAST
```

```
043.202  321            569          POP     D               (DE) = FIRST
                        570
                        571  *       MAKE SURE 1ST IS LESS THAN OR EQUAL TO LAST
                        572
043.203  175            573          MOV     A,L
043.204  223            574          SUB     E
043.205  174            575          MOV     A,H
043.206  232            576          SBB     D
043.207  320            577          RNC                     IS OK
043.210  315 136 031    578          CALL    $TYPTX
043.213  012 007 106    579          DB      NL,BELL,'First <= Las','t'+200Q
043.232  303 031 043    580          JMP     REFUSE
```

```
                                584  **      DRE - DECODE RANGE EXPRESSION.
                                585  *
                                586  *       DRE DECODES A COMMAND RANGE EXPRESSION.
                                587  *
                                588  *       TOKENS VALID AS 1ST TOKEN, ONLY
                                589  *
                                590  *       NULL    CURRENT 1ST LINE
                                591  *       $       LAST LINE IN BUFFER
                                592  *       ^       1ST LINE IN BUFFER
                                593  *
                                594  *       TOKENS VALID ANYWHERE
                                595  *
                                596  *       'STR' LINE CONTAINING STRING
                                597  *
                                598  *       TOKENS NOT VALID AT HEAD OF STRING
                                599  *
                                600  *       NNN     LINE COUNT
                                601  *
                                602  *       OPERATORS
                                603  *
                                604  *       +       SCAN FORWARD
                                605  *       -       SCAN BACKWARDS
                                606  *
                                607  *       ENTRY   NONE
                                608  *       EXIT    (HL) = RESULTANT LINE POINTER
                                609  *       USES    ALL
                                610
                                611
043.235                         612  DRE     EQU     *
043.235  076 377               613          MVI     A,-1
043.237  062 145 061           614          STA     SRCDIR          SET INITIAL DIRECTION FORWARD
                                615
                                616  *       DECODE INITIAL TOKEN.
                                617
043.242  315 064 053           618          CALL    ENC             PEEK AT CHARACTER
043.245  052 120 061           619          LHLD    FILPTR
043.250  376 136               620          CPI     '^'
043.252  312 302 043           621          JE      DRE1            START AT TOP
043.255  052 122 061           622          LHLD    LALPTR          ASSUME LAST
043.260  365                   623          PUSH    PSW             SAVE (A)
043.261  315 322 054           624          CALL    SLB             SCAN LINE BACKWARDS
043.264  361                   625          POP     PSW
043.265  376 044               626          CPI     '$'
043.267  312 302 043           627          JE      DRE1            NOT TO START AT BOTTOM
043.272  052 124 061           628          LHLD    CRFPTR
043.275  376 047               629          CPI     QUOTE
043.277  312 372 043           630          JE      DRE7            IS QUOTED STRING
043.302  314 205 053           631  DRE1    CZ      GNC             ACCEPT CHARACTER OF $ OR ARROW
                                632
043.305  042 130 061           633  DRE3    SHLD    WRKPTR          SET CURRENT LINE ADDRESS
                                634
                                635  *       DECODE OPERATOR
                                636
043.310  315 064 053           637  DRE4    CALL    ENC             EXAMINE NEXT CHARACTER
043.313  326 053               638          SUI     '+'
043.315  312 331 043           639          JZ      DRE5            IS FORWARD SEARCH
```

```
043.320  376 002        640          CPI     '-'='+'
043.322  312 331 043    641          JE      DRE5            IS BACKWARD SEARCH
043.325  052 130 061    642          LHLD    WRKPTR          (HL) = LINE RANGE
043.330  311            643          RET                     EXIT WITH LINE POINTER
                        644
043.331  075            645  DRE5    DCR     A
043.332  062 145 061    646          STA     SRCDIR
043.335  315 205 053    647          CALL    GNC             READ + OR -
                        648
                        649  **      DECODE NEXT TOKEN.
                        650
043.340  315 064 053    651          CALL    ENC             EXAMINE CHARACTER
043.343  376 047        652          CPI     QUOTE
043.345  312 375 043    653          JE      DRE8            QUOTED STRING
                        654
                        655  *       HAVE NNN - STEP OVER LINES
                        656
043.350  315 265 052    657          CALL    DIN             MUST BE DECIMAL NUMBER
043.353  170            658  DRE6    MOV     A,B
043.354  261            659          ORA     C
043.355  312 305 043    660          JZ      DRE3            HAVE STEPPED ENOUGH LINES
043.360  013            661          DCX     B
043.361  315 020 044    662          CALL    MLP             MOVE LINE POINTER
043.364  042 130 061    663          SHLD    WRKPTR
043.367  303 353 043    664          JMP     DRE6
                        665
                        666  *       HAVE STRING VALUE.
                        667
043.372  042 130 061    668  DRE7    SHLD    WRKPTR
043.375  041 001 063    669  DRE8    LXI     H,QUALS         USE QUALS AREA FOR SCRATCH
044.000  314 073 054    670          CZ      RQS             READ QUOTED STRING
044.003  315 322 053    671          CALL    LQS             LOCATE QUOTED STRING
044.006  312 310 043    672          JE      DRE4            FOUND
044.011  315 020 044    673          CALL    MLP             MOVE LINE POINTER
044.014  264            674          ORA     H
044.015  303 372 043    675          JMP     DRE7            SEARCH AGAIN


                        677  **      MLP - MOVE LINE POINTER.
                        678  *
                        679  *       MLP MOVES THE LINE POINTER FORWARDS OR BACKWARDS ONE LINE,
                        680  *       DEPENDING UPON 'SRCDIR'.
                        681  *
                        682  *       IF SRCDIR <  0, FORWARDS
                        683  *       IF SRCDIR => 0, BACKWARDS
                        684  *
                        685  *       IF RUN OFF THE NED OF TEXT, EXIT TO 'REFUSE'
                        686  *
                        687  *       ENTRY   (HL) = LINE POINTER
                        688  *       EXIT    (HL) = NEW LINE POINTER
                        689  *       USES    A,F
                        690
                        691
044.020  325            692  MLP     PUSH    D
```

```
044.021  052 130 061   693          LHLD    WRKPTR
044.024  072 145 061   694          LDA     SRCDIR
044.027  247           695          ANA     A
044.030  362 053 044   696          JP      MLP1            BACKWARDS
044.033  315 333 054   697          CALL    SNL             SCAN TO NEXT LINE
044.036  353           698          XCHG
044.037  052 122 061   699          LHLD    LALPTR
044.042  353           700          XCHG
044.043  315 216 030   701          CALL    $CDEHL          COMPARE TO BOTTOM
044.046  321           702          POP     D
044.047  312 031 043   703          JE      REFUSE          IF ALREADY AT BOTTOM
044.052  311           704          RET
                       705
                       706  *       BACKWARDS
                       707
044.053  353           708  MLP1    XCHG
044.054  052 120 061   709          LHLD    FILPTR
044.057  353           710          XCHG
044.060  315 216 030   711          CALL    $CDEHL          SEE IF AT TOP
044.063  312 031 043   712          JE      REFUSE
044.066  321           713          POP     D
044.067  303 322 054   714          JMP     SLB             SCAN LINE BACKWARDS AND RETURN
```

```
                                717  **      DCN - DECODE COMMAND NAME.
                                718  *
                                719  *       DCN DECODES AND COMPLETES THE COMMAND NAME.
                                720  *
                                721  *       ENTRY   NONE
                                722  *       EXIT    (A) = COMMAND INDEX
                                723
                                724
044.072                         725  DCN     EQU     *
044.072  315 064 053            726          CALL    ENC             PRE-READ 1ST COMMAND CHARACTER
044.075  052 142 061            727          LHLD    LINPTR
044.100  053                    728          DCX     H
044.101  042 157 044            729          SHLD    DCNA            SET LINE POINTER
044.104  257                    730          XRA     A
044.105  062 156 053            731          STA     ENCA
044.110  303 116 044            732          JMP     CMD3
                                733
                                734  *       INPUT 1 CHARACTER
                                735
044.113  315 205 053            736  CMD2    CALL    GNC             GET NEXT CHARACTER
                                737
                                738  *       CLEAR NXTCHA, PATCNT
                                739
044.116  041 000 377            740  CMD3    LXI     H,377000A
044.121  042 052 063            741          SHLD    NXTCHA
                                742
044.124  021 251 060            743          LXI     D,CMDTAB
044.127  052 124 061            744          LHLD    CREPTR
044.132  174                    745          MOV     A,H
044.133  265                    746          ORA     L               SEE IF ANY DATA
044.134  062 054 063            747          STA     CMDGRP          SET COMMAND GROUP
044.137  302 145 044            748          JNZ     CMD4            HAVE DATA
044.142  021 312 060            749          LXI     D,CMDTAB.       RESTRICT COMMAND RANGE
                                750
                                751  *       CHECK AGAINST NEXT COMMAND DESCRIPTION.
                                752
044.145  041 053 063            753  CMD4    LXI     H,PATCNT
044.150  064                    754          INR     M
044.151  353                    755          XCHG
044.152  315 333 054            756          CALL    SNL             SCAN FOR NEW LINE
044.155  353                    757          XCHG
044.156  001 000 000            758          LXI     B,0             (BC) = COMMAND TEXT ADDRESS
044.157                         759  DCNA    EQU     *-2
044.161  032                    760          LDAX    D
044.162  247                    761          ANA     A
044.163  302 212 044            762          JNZ     CMD5            HAVE COMMAND ELEMENT
                                763
                                764  *       NO MORE COMMANDS, HAVE:
                                765  *
                                766  *       1) NO MATCHES, OR
                                767  *       2) A UNIQUE NEXT CHARACTER
                                768
044.166  072 052 063            769          LDA     NXTCHA
044.171  247                    770          ANA     A
044.172  312 031 043            771          JZ      REFUSE          NO MATCHES - ILLEGAL
044.175  052 142 061            772          LHLD    LINPTR
```

```
044.200  167            773          MOV     M,A
044.201  043            774          INX     H
044.202  066 000        775          MVI     M,0
044.204  062 144 061    776          STA     PROCHA
044.207  303 113 044    777          JMP     CMD2
                        778
                        779  *       CHECK NEXT TABLE ELEMENT FOR MATCH
                        780
044.212  012            781  CMD5    LDAX    B               (A) = NEXT LINE CHARACTER
044.213  247            782          ANA     A
044.214  302 265 044    783          JNZ     CMD7            IF SOME
                        784
                        785  *       NO MORE TEXT, SEE IF CAN ANTICIPATE NEXT CHARACTER
                        786
044.217  072 144 061    787          LDA     PROCHA
044.222  247            788          ANA     A
044.223  304 345 055    789          CNZ     $WCHAR
044.226  257            790  CMD6    XRA     A
044.227  062 144 061    791          STA     PROCHA          CLEAR PROBATION CHARACTER
044.232  140            792  CMD6.5  MOV     H,B
044.233  151            793          MOV     L,C             (HL) = NEW LINE POINTER
044.234  042 142 061    794          SHLD    LINPTR          SKIP OVER CHARACTERS ACCEPTED
044.237  032            795          LDAX    D               (A) = COMMAND ELEMENT
044.240  247            796          ANA     A
044.241  310            797          RZ                      EXIT IF ENTIRE COMMAND MATCHED
044.242  041 052 063    798          LXI     H,NXTCHA
                        799
                        800  *       SEE IF THIS IS THE FIRST COMPLETION CHARACTER,
                        801  *       OR IF IT IS THE SAME CHARACTER AS PREVIOUSLY FOUND
                        802
044.245  276            803          CMP     M
044.246  312 145 044    804          JZ      CMD4            SAME AS PREVIOUS, CAN COMPLETE
044.251  325            805          PUSH    D
044.252  127            806          MOV     D,A
044.253  206            807          ADD     M
044.254  167            808          MOV     M,A
044.255  272            809          CMP     D               SEE IF NXTCHA WAS 0
044.256  321            810          POP     D
044.257  312 145 044    811          JZ      CMD4            CAN COMPLETE
044.262  303 113 044    812          JMP     CMD2            CANNOT COMPLETE
                        813
                        814  *       HAVE PATTERN AND TEXT, SEE IF MATCH.
                        815
044.265  032            816  CMD7    LDAX    D
044.266  247            817          ANA     A
044.267  312 232 044    818          JZ      CMD6.5          TOTAL MATCH - PRETEND RAN OUT OF TEXT
044.272  147            819          MOV     H,A             (H) = NEXT REQUIRED CHARACTER
044.273  012            820          LDAX    B               (A) = NEXT TEXT ELEMENT
044.274  315 205 055    821          CALL    $MCU            MAP CHARACTER TO UPPER CASE
044.277  003            822          INX     B               ASSUME MATCH
044.300  274            823          CMP     H
044.301  302 145 044    824          JNZ     CMD4            NO MATCH
044.304  023            825          INX     D
044.305  303 212 044    826          JMP     CMD5
```

```
                                    830  **      DCQ - DECODE COMMAND QUALIFIER.
                                    831  *
                                    832  *       DCQ READS AN OPTIONAL QUALIFICATION STRING FOLLOWING A
                                    833  *       COMMAND
                                    834  *
                                    835  *       COMMAND'STRING'
                                    836  *
                                    837  *       ENTRY   NONE
                                    838  *       EXIT    QUALS = STRING (NULL IF NONE)
                                    839
                                    840
044,310  041 001 063               841  DCQ     LXI     H,QUALS
044.313  066 000                   842          MVI     M,0             NULL IT
044,315  315 064 053               843          CALL    ENC             CHECK NEXT CHARACTER
044.320  376 047                   844          CPI     QUOTE
044,322  300                       845          RNE                     NO QUALIFIER
044.323  303 073 054               846          JMP     ROS             READ QUOTED STRING AND RETURN
```

```
                                  850  **      DCO - DECODE COMMAND OPTIONS.
                                  851  *
                                  852  *       DCO DECODES THE COMMAND OPTION SPECIFICATION.
                                  853  *
                                  854  *       COMMANDOPTION
                                  855  *
                                  856  *       WHERE OPT = A - PRINT LINE AFTER
                                  857  *                   B - PRINT LINE BEFORE
                                  858  *                   N - PRINT LINE NUMBERS
                                  859
                                  860
044,326  041 146 061              861  DCO     LXI     H,OPTS
044,331  066 000                  862          MVI     M,0             CLEAR OPTIONS
044,333  315 064 053              863  DCO1    CALL    ENC             CHECK NEXT CHARACTER
044,336  315 205 055              864          CALL    $MCU            MAP CHARACTER TO UPPER CASE
044,341  376 101                  865          CPI     'A'
044,343  312 351 044              866          JE      DCO2            IF 'A'
044,346  376 102                  867          CPI     'B'
044,350  300                      868          RNE                     NOT OPTION
044,351  346 003                  869  DCO2    ANI     OPT.A+OPT.B
044,353  107                      870          MOV     B,A             (B) = OPTION
044,354  246                      871          ANA     M
044,355  302 031 043              872          JNZ     REFUSE          ALREADY SET
044,360  170                      873          MOV     A,B
044,361  266                      874          ORA     M               SET IN FLAGS
044,362  167                      875          MOV     M,A
044,363  315 205 053              876          CALL    GNC             ACCEPT 'A' OR 'B'
044,366  303 333 044              877          JMP     DCO1
```

```
                                    881  **      INSERT - INSERT TEXT INTO BUFFER.
                                    882  *
                                    883  *       ISNERT RECOGNIZES TWO SPECIAL CASES:
                                    884  *
                                    885  *       1) IF NO TEXT EXISTS, INITIALIZE STRUCTURE
                                    886  *       2) IF THE LINE NUMBER IS ' ', INSERT BEFORD THE 1ST LINE
                                    887
                                    888
044.371                             889  INSERT  EQU     *
044.371  315 041 054               890          CALL    RCR             REQUIRE CARRIAGE RETURN
044.374  052 130 061               891          LHLD    WRKPTR
044.377  174                       892          MOV     A,H
045.000  265                       893          ORA     L
045.001  302 035 045               894          JNZ     INS1            HAVE PRE-EXISTING TEXT
                                    895
                                    896  *       READ 1ST LINE INTO EMPTY STRUCTURE
                                    897
045.004  315 072 052               898          CALL    ATL             READ TEXT
045.007  315 255 052               899          CALL    DCC             DISABLE CTL-C
045.012  353                       900          XCHG                    (DE) = TEXT ADDRESS
045.013  041 077 070               901          LXI     H,BUFFER
045.016  315 070 046               902          CALL    SAP             SET ALL POINTERS
045.021  345                       903          PUSH    H
045.022  117                       904          MOV     C,A
045.023  006 000                   905          MVI     B,0             (BC) = LEN
045.025  011                       906          DAD     B
045.026  042 122 061               907          SHLD    LALPTR
045.031  341                       908          POP     H
045.032  303 077 045               909          JMP     INS3
                                    910
045.035  315 030 054               911  INS1    CALL    PLB             PRINT LINE BEFORE
045.040  072 277 061               912          LDA     LINE
045.043  376 040                   913          CPI     ' '
045.045  304 333 054               914  INS2    CNZ     SNL             (HL) = ADDRESS TO INSERT TEXT
045.050  315 044 053               915          CALL    ECC             RE-ENABLE CTL-C
045.053  315 171 052               916          CALL    CBO             CHECK FOR BUFFER OVERFLOW
                                    917
                                    918  *       INSERT A NEW LINE
                                    919
045.056  042 130 061               920          SHLD    WRKPTR
045.061  353                       921          XCHG
045.062  315 072 052               922          CALL    ATL             ACCEPT TEXT LINE
045.065  315 255 052               923          CALL    DCC             DISABLE CTL-C
045.070  353                       924          XCHG
045.071  117                       925          MOV     C,A
045.072  315 244 053               926          CALL    ITBK            INSERT TEXT BLOCK               /80.02.GC/
045.075  006 000                   927          MVI     B,0
045.077  315 252 030               928  INS3    CALL    $MOVE           MOVE TEXT IN
045.102  052 130 061               929          LHLD    WRKPTR
045.105  264                       930          ORA     H               CLEAR 'Z'
045.106  303 045 045               931          JMP     INS2
```

```
                                   935  **      PRINT - PRINT TEXT LINES.
                                   936  *
                                   937
045.111  315 041 054              938  PRINT   CALL    RCR             REQUIRE CARRIAGE RETURN
045.114  315 231 054              939  PRI1    CALL    SEL             SCAN FOR ELIGIBLE LINE
045.117  310                      940          RZ                      IF NO MORE
045.120  315 345 054              941          CALL    TTX             TYPE SOURCE TEXT
045.121                           942  PRIA    EQU     *-2             PROCESSOR ADDRESS
045.123  315 045 052              943          CALL    ACL             ADVANCE COMMAND LINE
045.126  302 114 045              944          JNZ     PRI1
045.131  311                      945          RET                     DONE
```

```
                                949  **      XPRINT  -  PROCESS XPRINT COMMAND                       /80.02.GC/
                                950  *
                                951  *       XPRINT processes the XPRINT command which outputs
                                952  *       text to a specified alternate file.  The most
                                953  *       useful application of which, being a listing to
                                954  *       an alternate printer.
                                955  *
                                956
045.132                         957  XPRINT  EQU     *
045.132  315 041 054           958          CALL    RCR
                                959
045.135  072 244 061           960          LDA     XOUTFB+FB,FLG
045.140  346 004               961          ANI     FT.OW
045.142  312 017 052           962          JZ      WRI4            REQUIRE AN OUTPUT FILE
                                963
045.145  315 231 054           964  XPR1    CALL    SEL
045.150  312 164 045           965          JZ      XPR2            NO MORE LINES
                                966
                                967  *       OUTPUT THE SPECIFIED LINE TO THE XPRINT DEVICE
                                968
045.153  315 173 045           969          CALL    XPR4            OUTPUT A LINE
                                970
045.156  315 045 052           971          CALL    ACL             ADVANCE ONE LINE
045.161  302 145 045           972          JNZ     XPR1
                                973
                                974  *       FLUSH THE OUPUT TO THE SPECIFIED DEVICE
                                975
045.164                         976  XPR2    EQU     *
                                977
045.164  041 243 061           978          LXI     H,XOUTFB        USE XOUT FILE BUFFER
045.167  315 306 057           979          CALL    $FWBRK          BREAKOUTPUT
                                980
045.172  311                   981          RET


                                983  **      OUTPUT A LINE
                                984
045.173                         985  XPR4    EQU     *
045.173  345                   986          PUSH    H
045.174  353                   987          XCHG                    DE = ADDRESS OF LINE
045.175  041 243 061           988          LXI     H,XOUTFB        HL = FILE BUFFER
045.200  315 057 057           989          CALL    $FWRIL          WRITE LINE
045.203  341                   990          POP     H               RESTORE LINE ADDRESS
045.204  311                   991          RET
                                992
045.205  000                   993  XPRA    DB      0               FLUSH CHARACTER
000.001                         994  XPRAL   EQU     *-XPRA          LENGTH ( SHOULD BE ONE TO LEAVE BUFFER EMPTY )
```

```
                                998  **      DELETE - DELETE LINE RANGE.
                                999
                               1000
045.206  072 277 061          1001  DELETE  LDA     LINE
045.211  376 040              1002          CPI     ' '
045.213  312 031 043          1003          JE      REFUSE          <BLANK>DELETE ILLEGAL
045.216  315 041 054          1004          CALL    RCR             REQUIRE CARRIAGE RETURN
                               1005
                               1006  *       ENTERED FROM *WRITE* HERE
                               1007
045.221  072 001 063          1008  DEL0    LDA     QUALS
045.224  247                  1009          ANA     A
045.225  312 331 045          1010          JZ      DEL3            AM TO DELETE A BLOCK OF TEXT
045.230  315 044 053          1011  DEL1    CALL    ECC             ENABLE CTL-C
045.233  315 231 054          1012          CALL    SEL             SCAN FOR ELIGIBLE LINE          /10.04.77/
045.236  312 277 045          1013          JZ      DEL2            DONE                            /10.04.77/
045.241  345                  1014          PUSH    H               SAVE ADDRESS                    /10.04.77/
045.242  052 130 061          1015          LHLD    WRKPTR                                          /10.04.77/
045.245  353                  1016          XCHG
045.246  052 126 061          1017          LHLD    CRLPTR          SEE IF AT LAST TEXT LINE
045.251  173                  1018          MOV     A,E
045.252  225                  1019          SUB     L
045.253  172                  1020          MOV     A,D
045.254  234                  1021          SBB     H
045.255  341                  1022          POP     H               (HL) = TEXT POINTER             /10.04.77/
045.256  365                  1023          PUSH    PSW             SAVE RESULT FOR LATER TEST
045.257  315 030 054          1024          CALL    PLB             PRINT LINE BEFORE
045.262  315 255 052          1025          CALL    DCC             DISABLE CTL-C
045.265  315 361 054          1026          CALL    $CLL            COMPUTE LINE LENGTH
045.270  315 337 052          1027          CALL    DTBK            DELETE TEXT BLOCK               /80.02.GC/
045.273  361                  1028          POP     PSW             RESTORE CONDITION AFTER TEST
045.274  332 230 045          1029          JC      DEL1            MORE TO GO
                               1030
                               1031  *       ALL DONE, CLEAR PREVIOUS COMMAND RANGE TO FORCE NEW RANGE
                               1032
045.277                        1033  DEL2    EQU     *                                               /80.02.GC/
045.277  052 122 061          1034          LHLD    LALPTR                                          /80.02.GC/
045.302  353                  1035          XCHG                    DE = END OF LAST + 1            /80.02.GC/
045.303  052 124 061          1036          LHLD    CRFPTR          HL = CURRENT FIRST POINTER      /80.02.GC/
045.306  315 216 055          1037          CALL    HLCPDE          COMPARE                         /80.02.GC/
045.311  332 322 045          1038          JC      DEL2.5          HL < DE                         /80.02.GC/
                               1039
045.314  052 122 061          1040          LHLD    LALPTR                                          /80.02.GC/
045.317  315 322 054          1041          CALL    SLB             SCAN BACK ONE LINE              /80.02.GC/
                               1042
045.322  042 132 061          1043  DEL2.5  SHLD    PCFPTR          SET PREVIOUS RANGE TO FIRST LINE
045.325  042 134 061          1044          SHLD    PCLPTR
045.330  311                  1045          RET                     EXIT
                               1046
                               1047  *       NO QUALIFIER STRING, WILL THEREFORE DELETE AN ENTIRE BLOCK.
                               1048  *       LOCATE THAT BLOCK, AND DELETE ALL IN ONE SWOOP (RUNS A HECK OF A
                               1049  *       LOT FASTER!)
                               1050
045.331  315 255 052          1051  DEL3    CALL    DCC             DISABLE CTL-C
045.334  052 130 061          1052          LHLD    WRKPTR
045.337  042 020 046          1053          SHLD    DELA            SAVE FWA OF BLOCK
```

```
045.342  001 000 000  1054           LXI    B,0             (BC) = BYTES TO DELETE
                      1055
045.345  052 126 061  1056  DEL4     LHLD   CRLPTR          SEE IF THE LAST LINE IN THE RANGE
045.350  353          1057           XCHG
045.351  052 130 061  1058           LHLD   WRKPTR
045.354  175          1059           MOV    A,L
045.355  223          1060           SUB    E
045.356  174          1061           MOV    A,H
045.357  232          1062           SBB    D
045.360  365          1063           PUSH   PSW             SAVE RESULT
045.361  315 030 054  1064           CALL   PLB             PRINT LINE BEFORE
045.364  315 361 054  1065           CALL   $CLL            COMPUTE LINE LENTH
045.367  315 072 030  1066           CALL   $DADA           (HL) = LINE LWA+1
045.372  201          1067           ADD    C
045.373  117          1068           MOV    C,A
045.374  170          1069           MOV    A,B
045.375  316 000      1070           ACI    0
045.377  107          1071           MOV    B,A             ADD LENGTH TO (BC)
046.000  042 130 061  1072           SHLD   WRKPTR          ADVANCE POINTER
046.003  361          1073           POP    PSW             (PSW) = RESULTS OF WRKPTR-CRLPTR
046.004  332 345 045  1074           JC     DEL4            IF NOT ALL DONE
                      1075
                      1076  *        DELETE (BC) BYTES AT (DELA)
                      1077
046.007  052 020 046  1078           LHLD   DELA
046.012  315 350 052  1079           CALL   DTBK.           DELETE A TEXT BLOCK            /80.02.6C/
046.015  303 277 045  1080           JMP    DEL2            FINISH UP
                      1081
046.020  000 000      1082  DELA     DW     0               FWA OF BLOCK TO DELETE
```

```
                        1086  **      REPLACE - PROCESS REPLACE COMMAND.
                        1087  *
                        1088
                        1089
046.022  315 041 054   1090  REPLAC  CALL    RCR             REQUIRE CARRIAGE RETURN
046.025  315 226 054   1091  REP1    CALL    SEL.            SCAN FOR ELIGIBLE LINE
046.030  310           1092          RZ                      DONE
046.031  315 030 054   1093          CALL    PLB             PRINT LINE BEFORE
046.034  315 072 052   1094          CALL    ATL             ACCEPT TEXT LINE
046.037  117           1095          MOV     C,A
046.040  315 144 054   1096          CALL    RSL             REPLACE SINGLE LINE
046.043  315 045 052   1097          CALL    ACL             ADVANCE COMMAND LINE
046.046  310           1098          RZ
046.047  303 025 046   1099          JMP     REP1
```

```
                                1103  **      PURGE - PURGE TEXT BUFFER.
                                1104  *
                                1105  *       PURGE DELETES ALL TEXT, AND INITIALIZES THE DATA STRUCTURE.
                                1106  *
                                1107  *       THE NUMBER OF FREE BYTES REMAINING IS TYPED OUT.
                                1108
                                1109
046.052  315 041 054            1110  PURGE   CALL    RCR             REQUIRE CARRIAGE RETURN
046.055  315 110 052            1111          CALL    AYS             ARE YOU SURE
046.060  330                    1112          RC                      NOT SURE
046.061  300                    1113          RNE                     NOT SURE
                                1114
                                1115  **      PURGE. - PURGE WITHOUT WARNING.
                                1116  *
                                1117
046.062                         1118  PURGE.  EQU     *
046.062  041 000 000            1119          LXI     H,0
046.065  315 255 052            1120          CALL    DCC             DISABLE CTL-C



                                1122  **      SAP - SET ALL POINERS.
                                1123  *
                                1124  *       SAP SETS THE FOLLOWING POINTERS TO A SINGLE VALUE:
                                1125  *
                                1126  *       FILPTR  FIRST LINE POINTER
                                1127  *       LALPTR  LAST LINE POINTER
                                1128  *       CRFPTR  COMMAND FIRST LINE POINTER
                                1129  *       CRLPTR  COMMAND LAST LINE POINTER
                                1130  *       WRKPTR  WROKING POINTER
                                1131  *
                                1132  *       ENTRY   (HL) = VALUE
                                1133  *       EXIT    NONE
                                1134  *       USES    NONE
                                1135
                                1136
046.070  042 120 061            1137  SAP     SHLD    FILPTR
046.073  042 122 061            1138          SHLD    LALPTR
046.076  042 124 061            1139          SHLD    CRFPTR
046.101  042 126 061            1140          SHLD    CRLPTR
046.104  042 130 061            1141          SHLD    WRKPTR
046.107  042 132 061            1142          SHLD    PCFPTR
046.112  042 134 061            1143          SHLD    PCLPTR
046.115  311                    1144          RET
```

```
                                1148  **      EDITC - PROCESS EDIT COMMAND.
                                1149  *
                                1150  *       EDIT/FROM/TO/COUNT
                                1151
                                1152
046.116                         1153  EDITC   EQU     *
046.116  315 217 053            1154          CALL    GTC             GET DELIMITER
046.121  107                    1155          MOV     B,A             (B) = DELIMITER
                                1156
                                1157  *       READ /FROM/
                                1158
046.122  041 257 062            1159          LXI     H,EDIA
046.125  315 345 046            1160          CALL    RDS             READ DELIMITED STRING
046.130  171                    1161          MOV     A,C             (A) = LEN
046.131  247                    1162          ANA     A
046.132  312 031 043            1163          JZ      REFUSE          NULL IS ILLEGAL
                                1164
                                1165  *       READ /TO/ STRING
                                1166
046.135  041 330 062            1167          LXI     H,EDIB
046.140  121                    1168          MOV     D,C             (D) = LENGTH OF /FROM/
046.141  315 345 046            1169          CALL    RDS             READ DELIMITED STRING
046.144  102                    1170          MOV     B,D             (B) = LEN(FROM), (C) = LEN(TO)
046.145  305                    1171          PUSH    B               SAVE
046.146  001 000 000            1172          LXI     B,0
046.151  315 064 053            1173          CALL    ENC
046.154  376 052                1174          CPI     '*'
046.156  302 167 046            1175          JNE     EDI0            TO PROCESS ALL OF THEM
046.161  315 205 053            1176          CALL    GNC
046.164  303 175 046            1177          JMP     EDI2
                                1178
046.167  003                    1179  EDI0    INX     B               DEFAULT COUNT = 1
046.170  376 012                1180          CPI     NL
046.172  304 265 052            1181          CNE     DDN             DECODE IF DECIMAL
046.175  315 041 054            1182  EDI2    CALL    RCR             REQUIRE CARRIAGE RETURN
                                1183
                                1184  *       GET NEXT LINE
                                1185
046.200  315 226 054            1186  EDI3    CALL    SEL.            SCAN FOR ELIGIBLE LIN
046.203  312 335 046            1187          JZ      EDI5            ALL DONE
046.206  052 130 061            1188          LHLD    WRKPTR
046.211  315 361 054            1189          CALL    $CLL            COMPUTE LINE LENGTH
046.214  305                    1190          PUSH    B               SAVE REPEAT COUNT
046.215  117                    1191          MOV     C,A
046.216  006 000                1192          MVI     B,0             (BC) = LINE LENGTH
046.220  353                    1193          XCHG                    (DE) = FROM
046.221  041 067 062            1194          LXI     H,WRKSTR
046.224  345                    1195          PUSH    H               SAVE DEST ADDRESS
046.225  315 252 030            1196          CALL    $MOVE           MOVE INTO WRKSTR
046.230  341                    1197          POP     H               (HL) = $WRKSTR
046.231  301                    1198          POP     B               (BC) = REPEAT COUNT
046.232  021 257 062            1199          LXI     D,EDIA
046.235  315 264 054            1200          CALL    SFS             SEE IF SOURCE STRING IS PRESENT
046.240  302 335 046            1201          JNZ     EDI5            NOT FOUND
046.243  353                    1202          XCHG                    SAVE (HL) IN (DE)
046.244  315 030 054            1203          CALL    PLB             PRINT LINE BEFORE
```

```
046.247  353            1204            XCHG                    RESTORE (HL)
                        1205
                        1206  *         REPLACE STRING
                        1207
046.250  321            1208            POP     D               (D) = LEN(FROM), (E) = LEN(TO)
046.251  305            1209            PUSH    B               SAVE REPLACEMENT COUNTS
046.252  325            1210            PUSH    D               SAVE LENGTHS
046.253  345            1211            PUSH    H               SAVE ADDRESS OF MATCH
                        1212
                        1213  *         SOURCE LINE IS HEAD MATCH TAIL
                        1214  *
                        1215  *         MOVE TAIL TO ITS NEW POSITION TO MAKE ROOM FOR /TO/
                        1216
046.254  112            1217            MOV     C,D             (BC) = LEN(FROM)
046.255  006 000        1218            MVI     B,0
046.257  120            1219            MOV     D,B             (DE) = LEN(TO)
046.260  031            1220            DAD     D               (HL) = NEW TIAL ADDRESS
046.261  353            1221            XCHG
046.262  341            1222            POP     H
046.263  345            1223            PUSH    H
046.264  011            1224            DAD     B               (HL) = CURRENT TAIL ADDRESS
046.265  315 361 054     1225            CALL    $CLL            COMPUTE LINE LENGTH
046.270  006 000        1226            MVI     B,0
046.272  117            1227            MOV     C,A             (BC) = LENGTH OF TAIL
046.273  353            1228            XCHG
046.274  315 252 030     1229            CALL    $MOVE           MOVE TAIL
046.277  341            1230            POP     H               (HL) = MATCH ADDRES
046.300  301            1231            POP     B               (BC) = LENGTHS
046.301  305            1232            PUSH    B
046.302  006 000        1233            MVI     B,0
046.304  021 330 062     1234            LXI     D,EDIB
046.307  315 252 030     1235            CALL    $MOVE           COPY INTO PLACE
                        1236
                        1237  *         COMPRESS STRING AND PUT BACK IN BUFFER
                        1238
046.312  041 067 062     1239            LXI     H,WRKSTR
046.315  315 361 054     1240            CALL    $CLL            COMPUTE LINE LENGTH
046.320  117            1241            MOV     C,A             (C) = LENGTH
046.321  315 144 054     1242            CALL    RSL             REPLACE SINGLE LINE
                        1243
                        1244  *         DECREMENT REQUEST COUNT
                        1245
046.324  321            1246            POP     D
046.325  301            1247            POP     B
046.326  325            1248            PUSH    D
046.327  013            1249            DCX     B
                        1250
                        1251  *         SEE IF MORE TO GO
                        1252
046.330  170            1253            MOV     A,B
046.331  261            1254            ORA     C
046.332  312 343 046     1255            JZ      EDI6            NO MORE LINES TO CONSIDER
                        1256
046.335  315 045 052     1257  EDI5      CALL    ACL             ADVANCE COMMAND LINE
046.340  302 200 046     1258            JNZ     EDI3            MORE TO GO
046.343  301            1259  EDI6      POP     B
```

```
046.344  311              1260          RET

                         1262  **      RDS - READ DELIMITED STRING.
                         1263  *
                         1264  *       ENTRY   (B) = DELIMITER
                         1265  *               (HL) = ADDRESS FOR STRING
                         1266  *       EXIT    (HL) UNCHANGED
                         1267  *               (C) = LENGTH OF STRING
                         1268  *       USES    A,F,C
                         1269
                         1270
046.345  016 377        1271  RDS     MVI     C,377Q
046.347  345              1272          PUSH    H
046.350  325              1273          PUSH    D
046.351  026 050         1274          MVI     D,40            (D) = MAX COUNT
046.353  025              1275  RDS1    DCR     D
046.354  312 031 043     1276          JZ      REFUSE          TOO MANY
046.357  315 217 053     1277          CALL    GTC             GET TEXT CHARACTER
046.362  167              1278          MOV     M,A
046.363  043              1279          INX     H
046.364  014              1280          INR     C
046.365  270              1281          CMP     B
046.366  302 353 046     1282          JNE     RDS1            NOT DELIMITER
                         1283
                         1284  *       OUT OF STRING
                         1285
046.371  053              1286          DCX     H
046.372  066 000         1287          MVI     M,0             END IT
046.374  321              1288          POP     D               RESTORE (DE)
046.375  341              1289          POP     H
046.376  311              1290          RET
```

```
                        1294  **      FLUSH - PROCESS FLUSH COMMAND.
                        1295  *
                        1296
                        1297
     046.377            1298  FLUSH   EQU     *               ENTRY POINT
     046.377 315 041 054 1299          CALL    RCR             REQUIRE CARRIAGE RETURN
     047.002 072 156 061 1300  FLUSH1  LDA     INFB+FB.FLG
     047.005 365         1301          PUSH    PSW             SAVE FLAG
     047.006 315 151 050 1302          CALL    NEXT.           MOVE DATA THROUGH
     047.011 361         1303          POP     PSW
     047.012 346 002     1304          ANI     FT.OR
     047.014 302 002 047 1305          JNZ     FLUSH1          NOT AT EOF YET
                        1306
                        1307  *       HAVE READ EOF, WRITE ALL.
                        1308
     047.017 041 210 061 1309          LXI     H,OUTFB
     047.022 303 147 056 1310          JMP     $FCLO           CLOSE AND EXIT
```

```
                                1313  ***     BYE - EXIT EDITOR.
                                1314  *
                                1315  *       BYE (CR)
                                1316  *
                                1317  *       BYE FLUSHES OUT THE EXISTING FILES, AND EXITS.
                                1318
                                1319
047.025  315 377 046           1320  BYE     CALL    FLUSH
047.030  041 243 061           1321          LXI     H,XOUTFB        CLOSE  *XOUT*  FILE            /80.02.GC/
047.033  315 147 056           1322          CALL    $FCLO                                           /80.02.GC/
047.036  257                   1323          XRA     A
047.037  377 000                1324          DB      SYSCALL,,EXIT           EXIT
```

```
                        1328  **      NEWIN - PROCESS NEWIN COMMAND.
                        1329  *
                        1330
                        1331
    047.041             1332  NEWIN   EQU     *
                        1333
                        1334  *       SET NEW 'IN' FILE
                        1335
    047.041 315 217 053 1336          CALL    GTC             GET DELIMITER
    047.044 376 012     1337          CPI     NL
    047.046 312 031 043 1338          JE      REFUSE          NO NAME
    047.051 107         1339          MOV     B,A
    047.052 041 257 062 1340          LXI     H,EDIA
    047.055 315 345 046 1341          CALL    RDS             READ DELIMITED STRING
    047.060 315 151 055 1342          CALL    $MLU            MAP LINE TO UPPER CASE
    047.063 315 041 054 1343          CALL    RCR             REQUIRE CARRIAGE RETURN
    047.066 315 350 053 1344          CALL    MIM             REQUEST MINIMUM MEMORY
    047.071 076 021     1345          MVI     A,FB.NAML
    047.073 271         1346          CMP     C               SEE IF TOO LONG A NAME GIVEN
    047.074 332 204 047 1347          JC      NEWIN4          TOO LONG
    047.077 072 156 061 1348          LDA     INFB+FB.FLG
    047.102 346 002     1349          ANI     FT.OR
    047.104 312 155 047 1350          JZ      NEWIN1          NOT ALREADY OPEN
    047.107 315 136 031 1351          CALL    $TYPTX
    047.112 012 117 154 1352          DB      NL,'Old Input File Not Finished.',' '+200Q
    047.150 315 110 052 1353          CALL    AYS             ARE YOU SURE?
    047.153 330         1354          RC                      NOT SURE
    047.154 300         1355          RNE                     NOT SURE
    047.155 041 155 061 1356  NEWIN1  LXI     H,INFB
    047.160 315 147 056 1357          CALL    $FCLO           CLOSE OLD ONE
    047.163 345         1358          PUSH    H
    047.164 315 271 055 1359          CALL    $MOVEL
    047.167 021 000     1360          DW      FB.NAML
    047.171 257 062     1361          DW      EDIA
    047.173 167 061     1362          DW      FB.NAM+INFB     SET NAME
    047.175 341         1363          POP     H
    047.176 021 147 061 1364          LXI     D,DEFALT
    047.201 303 007 056 1365          JMP     $FOPER          OPEN FOR READ AND EXIT
                        1366
                        1367  *       ILLEGAL FILE NAME GIVEN
                        1368
    047.204 315 136 031 1369  NEWIN4  CALL    $TYPTX
    047.207 007 111 154 1370          DB      BELL,'Illegal File Name',' '+200Q
    047.232 303 200 042 1371          JMP     EDIX
```

```
                          1375  **      NEWOUT,'NAME'
                          1376  *
                          1377
                          1378
047.235                   1379  NEWOUT  EQU     *
                          1380
                          1381  *       SET NEW 'OUT' FILE
                          1382
047.235  315 217 053     1383          CALL    GTC             GET DELIMITER
047.240  107             1384          MOV     B,A             (B) = DELIMITER
047.241  376 012         1385          CPI     NL
047.243  312 031 043     1386          JE      REFUSE          NO NEW FILE
047.246  041 257 062     1387          LXI     H,EDIA
047.251  315 345 046     1388          CALL    RDS             READ DELIMITED STRING
047.254  315 151 055     1389          CALL    $MLU            MAP LINE TO UPPER CASE
047.257  315 041 054     1390          CALL    RCR             REQUIRE CARRIAGE RETURN
047.262  315 350 053     1391          CALL    MIM             REQUEST MINIMUM MEMROY
047.265  076 021         1392          MVI     A,FB.NAML
047.267  271             1393          CMP     C
047.270  332 204 047     1394          JC      NEWIN4          TOO MANY CHARACTERS FOR FILE NAME
047.273  072 211 061     1395          LDA     OUTFB+FB.FLG
047.276  346 004         1396          ANI     FT.OW
047.300  312 352 047     1397          JZ      NEWO1           OUTPUT CLOSED
047.303  315 136 031     1398          CALL    $TYPTX
047.306  012 117 154     1399          DB      NL,'Old Output File Not Finished.',' '+200Q
047.345  315 110 052     1400          CALL    AYS             SURE?
047.350  330             1401          RC                      NOT SURE
047.351  300             1402          RNE                     NOT SURE
047.352  041 210 061     1403  NEWO1   LXI     H,OUTFB
047.355  315 147 056     1404          CALL    $FCLO           CLOSE OLD STUFF
047.360  345             1405          PUSH    H
047.361  315 271 055     1406          CALL    $MOVEL
047.364  021 000         1407          DW      FB.NAML
047.366  257 062         1408          DW      EDIA
047.370  222 061         1409          DW      OUTFB+FB.NAM
047.372  341             1410          POP     H               (HL) = FB ADDRESS
047.373  021 147 061     1411          LXI     D,DEFALT
047.376  303 016 056     1412          JMP     $FOPEW          OPEN FOR WRITE AND EXIT
```

```
                                1416  **      XOUT    -  PROCESS XOUT COMMAND                 /80.02.GC/
                                1417  *
                                1418  *       XOUT closes any currently specified XPRINT channel,
                                1419  *       and opens the newly specified one.
                                1420  *
                                1421
     050.001                    1422  XOUT    EQU     *
                                1423
                                1424  *       SET NEW 'OUT' FILE
                                1425
     050.001  315 217 053       1426          CALL    GTC
     050.004  107               1427          MOV     B,A
     050.005  376 012           1428          CPI     NL
     050.007  312 031 043       1429          JE      REFUSE          NO NEW FILE
                                1430
     050.012  041 257 062       1431          LXI     H,EDIA
     050.015  315 345 046       1432          CALL    RDS             READ DELIMITED STRING
     050.020  315 151 055       1433          CALL    $MLU            MAP TO UPPER CASE
     050.023  315 041 054       1434          CALL    RCR             GET NEWLINE
     050.026  315 350 053       1435          CALL    MIM             MINIMUM MEMORY
                                1436
     050.031  076 021           1437          MVI     A,FB.NAML
     050.033  271               1438          CMP     C
     050.034  332 204 047       1439          JC      NEWIN4          TOO MANY CHARACTERS
                                1440
     050.037  072 244 061       1441          LDA     XOUTFB+FB.FLG
     050.042  346 004           1442          ANI     FT.OW
     050.044  312 117 050       1443          JZ      XOUT1           OUTPUT CLOSED
                                1444
     050.047  315 136 031       1445          CALL    $TYPTX
     050.052  012 117 154       1446          DB      NL,'Old XOUT File is not finished.',' '+200Q
     050.112  315 110 052       1447          CALL    AYS             SURE?
     050.115  330               1448          RC                      NOT SURE
     050.116  300               1449          RNE                     NOT SURE
                                1450
     050.117  041 243 061       1451  XOUT1   LXI     H,XOUTFB
     050.122  315 147 056       1452          CALL    $FCLO           CLOSE THE OLD ONES
     050.125  345               1453          PUSH    H
     050.126  315 271 055       1454          CALL    $MOVEL
     050.131  021 000           1455          DW      FB.NAML
     050.133  257 062           1456          DW      EDIA
     050.135  255 061           1457          DW      XOUTFB+FB.NAM
     050.137  341               1458          POP     H
     050.140  021 147 061       1459          LXI     D,DEFALT
     050.143  303 016 056       1460          JMP     $FOPEW          OPEN FOR WRITE AND EXIT
```

```
                                1464  **      NEXT - PROCESS 'NEXT' COMMAND.
                                1465  *
                                1466
                                1467
050.146                         1468  NEXT    EQU     *
050.146  315 041 054            1469          CALL    RCR             REQUIRE CARRIAGE RETURN
050.151                         1470  NEXT.   EQU     *
050.151  052 122 061            1471          LHLD    LALPTR
050.154  174                    1472          MOV     A,H
050.155  265                    1473          ORA     L
050.156  312 215 050            1474          JZ      READ.           NOTHING TO WRITE
050.161  315 322 054            1475          CALL    SLB             SCAN LINE BACKWARDS
050.164  042 126 061            1476          SHLD    CRLPTR
050.167  042 130 061            1477          SHLD    WRKPTR
050.172  042 124 061            1478          SHLD    CRFPTR
050.175  315 314 051            1479          CALL    WRITE.          WRITE ALL
050.200  303 215 050            1480          JMP     READ.           LOAD BACK UP
```

```
                        1484  **      READ - READ LINES FROM FILE.
                        1485  *
                        1486
050.203                 1487  READ    EQU     *
050.203  315 041 054    1488          CALL    RCR             REQUIRE CARRIAGE RETURN
050.206  315 215 050    1489          CALL    READ.
050.211  332 215 052    1490          JC      CBOI            NO ROOM
050.214  311            1491          RET
                        1492
050.215  072 156 061    1493  READ.   LDA     INFB+FB.FLG
050.220  346 002        1494          ANI     FT.OR
050.222  312 307 050    1495          JZ      READ2           AT EOF
050.225  052 122 061    1496  READO   LHLD    LALPTR          (HL) = LAST LINE POINTER
050.230  174            1497          MOV     A,H
050.231  265            1498          ORA     L
050.232  302 243 050    1499          JNZ     READ1           NOT EMPTY
050.235  041 077 070    1500          LXI     H,BUFFER
050.240  315 070 046    1501          CALL    SAP             SET ALL POINTERS IF NOT TEXT YET
050.243  021 000 002    1502  READ1   LXI     D,512           (DE) = ROOM TO LEAVE IN BUFFER
050.246  031            1503          DAD     D
050.247  353            1504          XCHG                    (DE) = PROPOSED NEW LALPTR
050.250  052 140 061    1505          LHLD    BUFMAX
050.253  175            1506          MOV     A,L             SEE IF WOULD EXCEDE MEMORY
050.254  223            1507          SUB     E
050.255  174            1508          MOV     A,H
050.256  232            1509          SBB     D
050.257  330            1510          RC                      CBOI => NO ROOM
                        1511
                        1512  *       HAVE ROOM. READ A LINE.
                        1513
050.260  052 122 061    1514          LHLD    LALPTR
050.263  353            1515          XCHG
050.264  001 200 000    1516          LXI     B,128
050.267  041 155 061    1517          LXI     H,INFB
050.272  315 254 056    1518          CALL    $FREAL          READ LINE
050.275  332 307 050    1519          JC      READ2           EOF
050.300  353            1520          XCHG                    (HL) = NEW LWA+1
050.301  042 122 061    1521          SHLD    LALPTR          UPDATE POINTER
050.304  303 225 050    1522          JMP     READO           READ SOME MORE
                        1523
                        1524  *       AT EOF
                        1525
050.307  315 136 031    1526  READ2   CALL    $TYPTX
050.312  012 105 156    1527          DB      NL,'End of Fil','e'+200Q
050.326  041 155 061    1528          LXI     H,INFB
050.331  315 147 056    1529          CALL    $FCLO           CLOSE BUFFER: AM DONE
050.334  067            1530          STC
050.335  077            1531          CMC                     CLEAR CARRY
050.336  311            1532          RET
```

```
                                1536  **      SEARCH - PROCESS SEARCH COMMAND.
                                1537  *
                                1538
                                1539
      050.337                   1540  SEARCH  EQU     *
                                1541
                                1542  *       DECODE SEARCH STRING
                                1543
      050.337  315 217 053      1544          CALL    GTC             GET DELIMITER
      050.342  107              1545          MOV     B,A             (B) = DELIMITER
      050.343  041 257 062      1546          LXI     H,EDIA
      050.346  315 345 046      1547          CALL    RDS             READ DELIMITER STRING
      050.351  171              1548          MOV     A,C
      050.352  247              1549          ANA     A
      050.353  312 031 043      1550          JZ      REFUSE          NULL STRING IS ILLEGAL
      050.356  315 041 054      1551          CALL    RCR             REQUIRE CR
                                1552
                                1553  *       TRY TO FIND LINE.
                                1554
      050.361  052 122 061      1555  SEA0    LHLD    LALPTR
      050.364  174              1556          MOV     A,H
      050.365  265              1557          ORA     L
      050.366  312 027 051      1558          JZ      SEA2            NO DATA IN BUFFER
      050.371  315 322 054      1559          CALL    SLB             SCAN LINE BACKWARDS
      050.374  042 126 061      1560          SHLD    CRLPTR          SET COMMAND LIMIT
      050.377  315 226 054      1561  SEA1    CALL    SEL.            SCAN FOR ELIGIBLE LINE
      051.002  312 027 051      1562          JZ      SEA2            NONE IN BUFFER
      051.005  052 130 061      1563          LHLD    WRKPTR          (HL) = ADDRESS OF TEXT LINE
      051.010  021 257 062      1564          LXI     D,EDIA
      051.013  315 264 054      1565          CALL    SFS             SEE IF FOUND
      051.016  312 056 051      1566          JZ      SEA3            FOUND IT
      051.021  315 045 052      1567          CALL    ACL             ADVANCE LINE
      051.024  302 377 050      1568          JNZ     SEA1            MORE GO TO
                                1569
                                1570  *       NOT FOUND IN THIS BUFFER.
                                1571
      051.027  072 156 061      1572  SEA2    LDA     INFB+FB.FLG
      051.032  346 002          1573          ANI     FT.OR
      051.034  312 074 051      1574          JZ      SEA4            AT END OF FILE
      051.037  315 151 050      1575          CALL    NEXT.           ADVANCE TEXT
      051.042  052 120 061      1576          LHLD    FILPTR
      051.045  042 124 061      1577          SHLD    CRFPTR
      051.050  042 130 061      1578          SHLD    WRKPTR
      051.053  303 361 050      1579          JMP     SEA0
                                1580
                                1581  *       FOUND IT
                                1582
      051.056  363              1583  SEA3    DI                      LOCK OUT CTL-C
      051.057  052 130 061      1584          LHLD    WRKPTR
      051.062  042 132 061      1585          SHLD    PCFPTR
      051.065  042 134 061      1586          SHLD    PCLPTR          SET BOUNDS TO FOUND LINE
      051.070  373              1587          EI                      RE-ALLOW CTL-C
      051.071  303 020 054      1588          JMP     PLA             PRINT LINE AFTER
                                1589
                                1590  *       NOT FOUND ANYWHERE.
                                1591
```

```
051.074  315 136 031  1592  SEA4    CALL    $TYPTX
051.077  012 116 157  1593          DB      NL,'Not Foun','d'+200Q
051.111  311          1594          RET
```

```
                        1598  **      USE - TYPE MEMORY STATISTICS.
                        1599  *
                        1600
                        1601
051.112                 1602  USE     EQU     *
051.112  315 041 054   1603          CALL    RCR             REQUIRE CARRIAGE RETURN
051.115  001 000 000   1604          LXI     B,0             (BC) = LINE COUNT
                        1605
051.120  315 231 054   1606  USE1    CALL    SEL             SCAN FOR ELIGIBLE LINE
051.123  312 143 051   1607          JZ      USE2            NO MORE
051.126  003           1608          INX     B               COUNT LINE
051.127  315 030 054   1609          CALL    PLB             PRINT LINE BEFORE
051.132  315 020 054   1610          CALL    PLA             PRINT LINE AFTER
051.135  315 045 052   1611          CALL    ACL             ADVANCE COMMAND LINE
051.140  302 120 051   1612          JNZ     USE1            LOOP IF MORE IN RANGE
                        1613
                        1614  *       (BC) = COUNT OF LINES WITHIN RANGE
                        1615
051.143  076 005       1616  USE2    MVI     A,5
051.145  041 246 051   1617          LXI     H,USEB
051.150  315 157 031   1618          CALL    $UDD
051.153  052 120 061   1619          LHLD    FILPTR
051.156  353           1620          XCHG                    (DE) = FIRST TEXT BYTE ADDRESS
051.157  052 122 061   1621          LHLD    LALPTR          (HL) = LAST TEXT BYTE ADDRESS
051.162  345           1622          PUSH    H               SAVE
051.163  175           1623          MOV     A,L
051.164  223           1624          SUB     E
051.165  117           1625          MOV     C,A
051.166  174           1626          MOV     A,H
051.167  232           1627          SBB     D
051.170  107           1628          MOV     B,A             (BC) = BYTES USED
051.171  076 005       1629          MVI     A,5
051.173  041 264 051   1630          LXI     H,USEC
051.176  315 157 031   1631          CALL    $UDD
051.201  321           1632          POP     D               (DE) = LAST
051.202  172           1633          MOV     A,D
051.203  263           1634          ORA     E
051.204  302 212 051   1635          JNZ     USE3            NON-ZERO
051.207  021 077 070   1636          LXI     D,BUFFER
051.212                 1637  USE3    EQU     *
051.212  052 140 061   1638          LHLD    BUFMAX          (HL) = MAX BUFFER SIZE
051.215  175           1639          MOV     A,L
051.216  223           1640          SUB     E
051.217  117           1641          MOV     C,A
051.220  174           1642          MOV     A,H
051.221  232           1643          SBB     D
051.222  107           1644          MOV     B,A             (BC) = AMOUNT UNUSED
051.223  076 005       1645          MVI     A,5
051.225  041 302 051   1646          LXI     H,USED
051.230  315 157 031   1647          CALL    $UDD            UNPACK COUNT
051.233  315 136 031   1648          CALL    $TYPTX
051.236  114 151 156   1649          DB      'Lines = '
051.246  130 130 130   1650  USEB    DB      'XXXXX',NL,'Used  = '
051.264  130 130 130   1651  USEC    DB      'XXXXX',NL,'Free  = '
051.302  130 130 130   1652  USED    DB      'XXXXX',ENL
051.310  311           1653          RET
```

```
                        1657  **      WRITE - WRITE LINES TO OUTPUT FILE.
                        1658  *
                        1659  *.      WRITE TEXT BLOCKS FROM THE TOP OF THE BUFFER UNTIL THE CURRENT
                        1660  *       LINE
                        1661
                        1662
051.311                 1663  WRITE   EQU     *
051.311  315 041 054    1664          CALL    RCR
051.314  076 000        1665  WRITE.  MVI     A,MI.NOP        DELETE TEXT AFTER WRITE
051.316  062 374 051    1666  WRI..   STA     WRIA            SET FLAG
051.321  052 120 061    1667          LHLD    FILPTR
051.324  042 130 061    1668          SHLD    WRKPTR          START AT TOP OF TEXT
051.327  072 211 061    1669          LDA     OUTFB+FB.FLG
051.332  346 004        1670          ANI     FT.OW
051.334  312 017 052    1671          JZ      WRI4            REQUIRE NEWOUT
                        1672
                        1673  *       SEE IF MORE TEXT TO WRITE.
                        1674
051.337  052 124 061    1675          LHLD    CRFPTR
051.342  174            1676          MOV     A,H
051.343  265            1677          ORA     L
051.344  312 374 051    1678          JZ      WRI3            NO DATA
                        1679
                        1680  *       WRITE ANOTHER LINE
                        1681
051.347  052 130 061    1682          LHLD    WRKPTR
051.352  353            1683          XCHG                    (DE) = CURRENT LINE
051.353  052 124 061    1684  WRI1    LHLD    CRFPTR          (HL) = LIMIT
051.356  315 216 030    1685          CALL    $CDEHL          COMPARE
051.361  365            1686          PUSH    PSW             SAVE RESULTS
051.362  041 210 061    1687          LXI     H,OUTFB
051.365  315 057 057    1688          CALL    $FWRIL          WRITE LINE
051.370  361            1689          POP     PSW             (A) = RESULTS OF TEST
051.371  302 353 051    1690          JNE     WRI1            MORE TO TO
                        1691
                        1692  *       END OF WRITTING. DELETE LINES WRITTEN.
                        1693
051.374                 1694  WRI3    EQU     *
051.374  000            1695  WRIA    NOP                     SET TO *RET* FOR SAVE
051.375  052 124 061    1696          LHLD    CRFPTR
052.000  042 126 061    1697          SHLD    CRLPTR          SET LINES WRITTEN AS COMMAND RANGE
052.003  052 120 061    1698          LHLD    FILPTR
052.006  042 124 061    1699          SHLD    CRFPTR
052.011  042 130 061    1700          SHLD    WRKPTR
052.014  303 221 045    1701          JMP     DELO            DELETE
                        1702
                        1703  *       REQUIRE NEWOUT
                        1704
052.017  315 136 031    1705  WRI4    CALL    $TYPTX
052.022  012 007 116    1706          DB      NL,BELL,'No Output Fil','e'+200Q
052.042  303 200 042    1707          JMP     EDIX
```

```
                                1711  **      ACL - ADVANCE COMMAND LINE.
                                1712  *
                                1713  *       ACL ADVANCES WRKPTR TO THE NEXT COMMAND LINE.
                                1714  *
                                1715  *       EXIT    (WRKPTR) UPDATED
                                1716  *               (HL) = (WRKPTR)
                                1717  *               'Z' SET IF AT END OF RANGE
                                1718  *       USES    A,F,H,L
                                1719
                                1720
052.045  325                   1721  ACL       PUSH    D
052.046  052 126 061            1722            LHLD    CRLPTR
052.051  353                   1723            XCHG
052.052  052 130 061            1724            LHLD    WRKPTR
052.055  315 216 030            1725            CALL    $CDEHL          COMPARE
052.060  321                   1726            POP     D
052.061  310                   1727            RZ                      IF AT END
052.062  315 333 054            1728            CALL    SNL             SCAN TO NEXT LINE
052.065  042 130 061            1729            SHLD    WRKPTR
052.070  264                   1730            ORA     H               CLEAR 'Z'
052.071  311                   1731            RET


                                1733  **      ATL - ACCEPT TEXT LINE.
                                1734  *
                                1735  *       ATL READS A LINE OF TEXT FROM THE CONSOLE INTO *LINE*.
                                1736  *
                                1737  *       THE LINE IS TERMINATED BY A 00 BYTE.
                                1738  *
                                1739  *       ENTRY   NONE
                                1740  *       EXIT    (HL) = *LINE
                                1741  *               (A) = BYTE COUNT
                                1742  *       USES    A,F,H,L
                                1743
                                1744
052.072  041 277 061            1745  ATL       LXI     H,LINE
052.075  257                   1746            XRA     A
052.076  062 326 040            1747            STA     $,CSLMD         SET LINE-MODE INPUT
052.101  315 233 055            1748            CALL    $RTL            READ LINE
052.104  320                   1749            RNC                     NOT CTL-D
052.105  303 052 043            1750            JMP     EXIT            CTL-D STRUCK


                                1752  **      AYS - ASK ARE YOU SURE?
                                1753  *
                                1754  *       AYS PROMPTS THE USER, 'SURE?'
                                1755  *       AND GETS HIS REPLY.
                                1756  *
                                1757  *       ENTRY   NONE
                                1758  *       EXIT    'C' SET IF CTL-D
                                1759  *               'C' CLEAR IF NOT CTL-D
                                1760  *                'Z' SET IF SURE
```

```
                        1761  *       USES    ALL
                        1762
                        1763
052.110  315 136 031   1764  AYS     CALL    $TYPTX
052.113  007 101 162   1765          DB      BELL,'Are You Sure?',' '+2000
052.132  315 337 055   1766          CALL    $RCHAR
052.135  315 345 055   1767          CALL    $WCHAR          ECHO
052.140  315 205 055   1768          CALL    $MCU            MAP TO UPPER
052.143  376 004       1769          CPI     CTLD
052.145  067           1770          STC                     ASSUME CTL-D
052.146  310           1771          RE                      CTL-D
052.147  326 131       1772          SUI     'Y'             SEE IF 'Y'
052.151  247           1773          ANA     A               CLEAR CARRY
052.152  311           1774          RET                     RETURN WITH CODES SET


                        1776  **      CBE - CHECK FOR BUFFER EMPTY.
                        1777  *
                        1778  *       IF FILPTR=LALPTR, ZERO POINTERS,
                        1779
052.153  052 120 061   1780  CBE     LHLD    FILPTR
052.156  353           1781          XCHG
052.157  052 122 061   1782          LHLD    LALPTR
052.162  315 216 030   1783          CALL    $CDEHL
052.165  300           1784          RNE                     NOT EMPTY
052.166  303 062 046   1785          JMP     PURGE.          HAVE DELETED ALL.


                        1787  **      CBO - CHECK BUFFER OVERFLOW
                        1788  *
                        1789  *       CBO IS CALLED BY COMMANDS WHICH MAY INCREASE THE SIZE
                        1790  *       OF THE BUFFER TEXT. IF THERE IS NOT ROOM ENOUGH FOR
                        1791  *       THE MAXIMUM SIZE INCREASE (120 CHARACTERS), AN OVERFLOW
                        1792  *       IS DECLARED.
                        1793  *
                        1794  *       ENTRY   NONE
                        1795  *       EXIT    TO (RET) IF OK
                        1796  *       USES    A,F
                        1797
052.171  345           1798  CBO     PUSH    H
052.172  325           1799          PUSH    D
052.173  052 122 061   1800          LHLD    LALPTR
052.176  021 170 000   1801          LXI     D,120
052.201  031           1802          DAD     D
052.202  353           1803          XCHG                    (DE) = NEW LIMIT
052.203  052 140 061   1804          LHLD    BUFMAX
052.206  175           1805          MOV     A,L
052.207  223           1806          SUB     E
052.210  174           1807          MOV     A,H
052.211  232           1808          SBB     D
052.212  321           1809          POP     D
052.213  341           1810          POP     H
052.214  320           1811          RNC                     IS OK
```

```
052.215  315 136 031  1812  CBO1    CALL    $TYPTX
052.220  012 007 116  1813          DB      NL,BELL,'Not Enough RA',/M'+200Q
052.240  303 200 042  1814          JMP     EDIX            ABORT COMMAND


                      1816  **      CDV - CHECK DECIMAL VALIDITY.
                      1817  *
                      1818  *       CDV EXAMINES THE NEXT CHARACTER TO SEE IF IT IS A DECIMAL
                      1819  *       DIGIT.
                      1820  *
                      1821  *       ENTRY   NONE
                      1822  *       EXIT    NEXT CHARACTER NOT READ
                      1823  *               'C' SET IF OK
                      1824  *                (A) = DIGIT VALUE (0=9)
                      1825  *               'C' SET IF NOT DECIMAL DIGIT
                      1826
                      1827
052.243  315 064 053  1828  CDV     CALL    ENC             EXAMINE NEXT CHARACTER
052.246  326 060      1829          SUI     '0'
052.250  330          1830          RC
052.251  376 012      1831          CPI     9+1
052.253  077          1832          CMC
052.254  311          1833          RET


                      1835  **      DCC - DISABLE CTL-C PROCESSING.
                      1836  *
                      1837  *       DCC IS CALLED WHEN A PROCESSOR IS ABOUT TO ENTER SENSITIVE CODE.
                      1838  *       CTL-C'S WILL BE HELD UNTIL A COMPANION CALL TO 'ECC' IS MADE.
                      1839  *
                      1840  *       ENTRY   NONE
                      1841  *       EXIT    NONE
                      1842  *       USES    NONE
                      1843
052.255  365          1844  DCC     PUSH    PSW
052.256  076 001      1845          MVI     A,1
052.260  062 136 061  1846          STA     CCFLG           FLAG DISABLED
052.263  361          1847          POP     PSW
052.264  311          1848          RET


                      1850  **      DDN - DECODE DECIMAL NUMBER.
                      1851  *
                      1852  *       ENTRY   NONE
                      1853  *       EXIT    (BC) = VALUE (IF NON-NULL)
                      1854  *               TO 'REFUSE' IF NULL
                      1855  *       USES    A,B,C,F
                      1856
                      1857
052.265  345          1858  DDN     PUSH    H
```

```
052.266 325           1859          PUSH    D
052.267 315 243 052   1860          CALL    CDV             CHECK DECIMAL VALUE
052.272 332 031 043   1861          JC      REFUSE          NOT DECIMAL DIGIT
052.275 021 000 000   1862          LXI     D,0             (DE) = ACCUMULATOR
052.300 315 243 052   1863  DDN1    CALL    CDV             CHECK DECIMAL VALUE
052.303 332 332 052   1864          JC      DDN2            NO MORE DIGITS
052.306 315 324 030   1865          CALL    $MUI0           (HL) = (DE)*10
052.311 332 031 043   1866          JC      REFUSE          OVERFLOW
052.314 137           1867          MOV     E,A
052.315 026 000       1868          MVI     D,0             (DE) = DIGIT VALUE
052.317 031           1869          DAD     D
052.320 332 031 043   1870          JC      REFUSE          NO GOOD
052.323 353           1871          XCHG                    (DE) = VALUE
052.324 315 205 053   1872          CALL    GNC             READ DECIMAL VALUE
052.327 303 300 052   1873          JMP     DDN1            ACCEPT ANOTHER
                      1874
                      1875  *       NUMBER ACCUMULATED, RETURN.
                      1876
052.332 102           1877  DDN2    MOV     B,D
052.333 113           1878          MOV     C,E
052.334 321           1879          POP     D
052.335 341           1880          POP     H
052.336 311           1881          RET


                      1883  **      DTBK    -  DELETE TEXT BLOCK                    /80.02.GC/
                      1884  *
                      1885  *       DTBK DELETES THE SPECIFIED TEXT BLOCK FROM THE TABLE
                      1886  *
                      1887  *
                      1888  *       ENTRY:  A       =  COUNT
                      1889  *               HL      =  ADDRESS IN BLOCK
                      1890  *
                      1891  *       EXIT:   NONE
                      1892  *
                      1893  *       USES:   PSW
                      1894  *
                      1895
052.337 305           1896  DTBK    PUSH    B
052.340 117           1897          MOV     C,A
052.341 006 000       1898          MVI     B,0             BC = FULL WORD COUNT
052.343 315 350 052   1899          CALL    DTBK.
052.346 301           1900          POP     B
052.347 311           1901          RET
```

```
                        1903  **      BC      = FULL WORD COUNT
                        1904  *
                        1905
052,350  345           1906  DTBK,   PUSH    H
052,351  325           1907          PUSH    D
052,352  353           1908          XCHG                    DE = BUFFER ADDRESS
                        1909
                        1910  *       FIX POINTERS THAT WILL MOVE
                        1911
052,353  052 126 061   1912          LHLD    CRLPTR          HL = CURRENT RANGE LAST POINTER
052,356  315 216 055   1913          CALL    HLCPDE
052,361  332 375 052   1914          JC      DTBK1           DELETION IS NOT IN RANGE
052,364  312 375 052   1915          JZ      DTBK1           DELETION IS NOT IN RANGE
                        1916
052,367  315 026 053   1917          CALL    DTBK3           HL = HL - BC
052,372  042 126 061   1918          SHLD    CRLPTR
052,375                1919  DTBK1   EQU     *
                        1920
052,375  052 122 061   1921          LHLD    LALPTR
053,000  345           1922          PUSH    H
053,001  315 026 053   1923          CALL    DTBK3           HL = HL - BC
053,004  042 122 061   1924          SHLD    LALPTR
053,007  341           1925          POP     H
                        1926
053,010  353           1927          XCHG                    HL = ADDRESS IN BUFFER
053,011  345           1928          PUSH    H               SAVE DESTINATION
053,012  011           1929          DAD     B
053,013  353           1930          XCHG                    DE = SOURCE ADDRESS
053,014  315 035 053   1931          CALL    DTBK4           BC = HL - DE
053,017  341           1932          POP     H               HL = DESTINATION ADDRESS
                        1933
053,020  315 252 030   1934  DTBK2   CALL    $MOVE
                        1935
053,023  321           1936          POP     D
053,024  341           1937          POP     H
053,025  311           1938          RET



053,026  175           1940  DTBK3   MOV     A,L
053,027  221           1941          SUB     C
053,030  157           1942          MOV     L,A
053,031  174           1943          MOV     A,H
053,032  230           1944          SBB     B
053,033  147           1945          MOV     H,A
053,034  311           1946          RET
```

```
053.035  175            1948  DTBK4   MOV     A,L
053.036  223            1949          SUB     E
053.037  117            1950          MOV     C,A
053.040  174            1951          MOV     A,H
053.041  232            1952          SBB     D
053.042  107            1953          MOV     B,A
053.043  311            1954          RET


                        1956  **      ECC - ENABLE CTL-C.
                        1957  *
                        1958  *       ECC IS CALLED TO RESTORE CTL-C PROCESSING AFTER
                        1959  *       A CALL TO *DCC*
                        1960  *
                        1961  *       IF A CTL-C WAS HIT IN THE INTERIM, IT WILL BE PROCESSED NOW.
                        1962  *
                        1963  *       ENTRY   NONE
                        1964  *       EXIT    TO CTL-C PROCESSOR IF ONE WAS STRUCK
                        1965  *       USES    NONE
                        1966
                        1967
053.044  365            1968  ECC     PUSH    PSW
053.045  363            1969          DI                      INTERLOCK
053.046  257            1970          XRA     A
053.047  062 136 061    1971          STA     CCFLG           CLEAR FLAG
053.052  072 137 061    1972          LDA     CCPEND
053.055  373            1973          EI
053.056  247            1974          ANA     A
053.057  302 374 042    1975          JNZ     INTRPT          PROCESS THAT NOW
053.062  361            1976          POP     PSW
053.063  311            1977          RET


                        1979  **      ENC - EXAMINE NEXT CHARACTER.
                        1980  *
                        1981  *       ENC RETURNS A PREVIEW OF THE NEXT INPUT CHARACTER. THE CHARACTER
                        1982  *       'POINTER' IS NOT UPDATED.
                        1983  *
                        1984  *       ENTRY   NONE
                        1985  *       EXIT    (A) = CHARACTER
                        1986  *       USES    A,F
                        1987
                        1988
053.064  072 156 053    1989  ENC     LDA     ENCA
053.067  247            1990          ANA     A
053.070  300            1991          RNZ                     HAVE CHARACTER
                        1992
                        1993  *       MUST READ ANOTHER CHARACTER FROM LINE OR TERMINAL.
                        1994
053.071  345            1995          PUSH    H
053.072  052 142 061    1996          LHLD    LINPTR
053.075  175            1997          MOV     A,L
```

```
053.076  074          1998          INR     A
053.077  365          1999          PUSH    PSW             SAVE FOR LATER COMPARE
053.100  176          2000          MOV     A,M             (A) = CHARACTER
053.101  043          2001          INX     H
053.102  247          2002          ANA     A
053.103  302 137 053  2003          JNZ     ENC1            GOT CHARACTER IN LINE
                      2004
                      2005  *       MUST READ ANOTHER CHARACTER FROM TERMINAL
                      2006
053.106  072 144 061  2007          LDA     PROCHA
053.111  247          2008          ANA     A
053.112  304 345 055  2009          CNZ     $WCHAR          ECHO PROBATION CHARACTER
053.115  315 015 055  2010          CALL    $INCHA          READ ANOTHER CHARACTER
053.120  376 004      2011          CPI     CTLD
053.122  312 052 043  2012          JE      EXIT            IS CTL-D
053.125  052 142 061  2013          LHLD    LINPTR
053.130  167          2014          MOV     M,A             STORE IN LINE
053.131  062 144 061  2015          STA     PROCHA          PUT ON 'PROBATION'
053.134  043          2016          INX     H
053.135  066 000      2017          MVI     M,0
053.137  042 142 061  2018  ENC1    SHLD    LINPTR          UPDATE LINE POINTER
053.142  062 156 053  2019          STA     ENCA            SET PRE-READ CHARACTER
053.145  147          2020          MOV     H,A             SAVE CHARACTER
053.146  361          2021          POP     PSW             (A) = PREVIOUS *L* VALUE+1
053.147  275          2022          CMP     L
053.150  302 250 042  2023          JNE     EDI1            BACKSPACE OR RUBBOUT
053.153  174          2024          MOV     A,H             (A) = SAVED CHARACTER
053.154  341          2025          POP     H               RESTORE (HL)
053.155  311          2026          RET
                      2027
053.156  000          2028  ENCA    DB      0               HELD CHARACTER


                      2030  **      ERROR - PROCESS ERROR MESSAGES.
                      2031  *
                      2032  *       ERROR IS CALLED WHEN A FILE ERROR OCCURS.
                      2033  *       IT EXITS TO *RESTART*, WHICH CLEANS THE STACK.
                      2034  *
                      2035  *       ENTRY   (A) = ERROR CODE
                      2036  *       EXIT    TO RESTART
                      2037  *       USES    ALL
                      2038
                      2039
053.157  365          2040  ERROR   PUSH    PSW             SAVE CODE
053.160  315 136 031  2041          CALL    $TYPTX
053.163  012 007 105  2042          DB      NL,BELL,'Error -',' '+200Q
053.175  361          2043          POP     PSW
053.176  046 012      2044          MVI     H,NL
053.200  377 057      2045          DB      SYSCALL,.ERROR
053.202  303 200 042  2046          JMP     RESTART
```

```
                                2048  **      GNC - GET NEXT CHARACTER.
                                2049  *
                                2050  *       GNC READS THE NEXT CHARACTER, AND ADVANCES THE POINTER.
                                2051  *
                                2052  *       ENTRY   NONE
                                2053  *       EXIT    (A) = CHARACTER
                                2054  *       USES    A,F
                                2055
                                2056
053.205  315 064 053            2057  GNC     CALL    ENC             EXAMINE NEXT
053.210  365                    2058          PUSH    PSW             SAVE CHARACTER
053.211  257                    2059          XRA     A
053.212  062 156 053            2060          STA     ENCA            CLEAR HELD CHARACTER
053.215  361                    2061          POP     PSW
053.216  311                    2062          RET


                                2064  **      GTC - GET TEXT CHARACTER.
                                2065  *
                                2066  *       GTC GETS A CHARACTER FROM THE INPUT STREAM, AND REQUIRES IT TO B
                                2067  *       PRINTABLE CHARACTER.
                                2068  *
                                2069  *       ENTRY   NONE
                                2070  *       EXIT    (A) = CHARACTER
                                2071  *       USES    A,F
                                2072
                                2073
053.217  315 064 053            2074  GTC     CALL    ENC
053.222  376 011                2075          CPI     TAB
053.224  312 205 053            2076          JE      GNC             ALLOW TABS
053.227  376 014                2077          CPI     FF
053.231  312 205 053            2078          JE      GNC             ALLOW FORM FEEDS
053.234  376 040                2079          CPI     20H
053.236  332 031 043            2080          JC      REFUSE          BAD
053.241  303 205 053            2081          JMP     GNC             GET IT AND RETURN


                                2083  **      ITBK    -  INSERT TEXT BLOCK                    /80.02.GC/
                                2084  *
                                2085  *       ITBK INSERTS THE SPECIFIED NUMBER OF BYTES INTO
                                2086  *       THE SPECIFIED TEXT BLOCK AT THE SPECIFIED ADDRESS.
                                2087  *
                                2088  *
                                2089  *       ENTRY:  A       = COUNT
                                2090  *               HL      = ADDRESS IN BUFFER
                                2091  *
                                2092  *       EXIT:   NONE
                                2093  *
                                2094  *       USES:   PSW
                                2095  *
                                2096
053.244  305                    2097  ITBK    PUSH    B
```

```
053.245  117           2098           MOV     C,A
053.246  006 000       2099           MVI     B,0             BC = FULL WORD COUNT
053.250  315 255 053   2100           CALL    ITBK.
053.253  301           2101           POP     B
053.254  311           2102           RET


                       2104   **      BC      = FULL WORD COUNT
                       2105   *
                       2106
053.255  345           2107   ITBK.   PUSH    H
053.256  325           2108           PUSH    D
053.257  353           2109           XCHG                    DE = ADDRESS IN BUFFER
                       2110
                       2111   *       FIX MOVING POINTERS
                       2112
053.260  052 126 061   2113           LHLD    CRLPTR
053.263  315 216 055   2114           CALL    HLCPDE
053.266  332 300 053   2115           JC      ITBK1           DELETION IS NOT IN RANGE
053.271  312 300 053   2116           JZ      ITBK1           DELETION IS NOT IN RANGE
                       2117
053.274  011           2118           DAD     B
053.275  042 126 061   2119           SHLD    CRLPTR          UPDATE CURRENT RANGE LAST POINTER
053.300                2120   ITBK1   EQU     *
                       2121
053.300  052 122 061   2122           LHLD    LALPTR
053.303  345           2123           PUSH    H
053.304  011           2124           DAD     B
053.305  042 122 061   2125           SHLD    LALPTR
053.310  341           2126           POP     H
                       2127
053.311  305           2128           PUSH    B               SAVE COUNT
053.312  315 035 053   2129           CALL    DTBK4           BC = HL - DE
053.315  341           2130           POP     H               HL = COUNT
053.316  031           2131           DAD     D               HL = HL + DE = DESTINATION
                       2132
053.317  303 020 053   2133           JMP     DTBK2           MOVE IT OUT


                       2135   **      LQS - LOCATE QUOTED STRING.
                       2136   *
                       2137   *       LQS FINDS A QUOTED STRING IN A TEXT LINE.
                       2138   *
                       2139   *       THE LINE IS EXPANDED INTO WRKSTR, AND THE SEARCH IS MADE.
                       2140   *
                       2141   *       ENTRY   (HL) = ADDRSS OF STRING
                       2142   *       EXIT    'Z' SET IF FOUND
                       2143   *               (DE) = ADDRESS IN LINWRK, IF FOUND
                       2144   *               (HL) UNCHANGED
                       2145   *       USES    A,F,D,E
                       2146
                       2147
```

```
053.322  353           2148  LQS     XCHG
053.323  052 130 061   2149          LHLD    WRKPTR          POINT TO TEXT
053.326  315 264 054   2150          CALL    SFS             SEARCH FOR STRING
053.331  353           2151          XCHG
053.332  311           2152          RET


                       2154  **      MAM - REQUEST MAXIMUM MEMORY ALLOCATION.
                       2155  *
                       2156  *       MAM REQUESTS THE MAXIMUM MEMORY AVAILABLE SO THAT THE HDOS OVERLAY
                       2157  *       CAN REMAIN RESIDENT.
                       2158  *
                       2159  *       THE SPACE IS GIVEN TO *BUFFER*.
                       2160  *
                       2161  *       * * NOTE * * - SOME OF THE MOVE AND MANAGEMENT ROUTINES
                       2162  *       USED BY *EDIT* CANNOT HANDLE TRANSFERS OF >32768, THEREFORE
                       2163  *       MAM REFUSES TO ALLOCATE MORE THAN 32000 TO THE BUFFER.
                       2164  *       DONT CHANGE THIS WITHOUT CAREFULLY CHECKING THINGS.
                       2165  *
                       2166  *       * * NOTE * * - THIS HOPEFULLY HAS BEEN FIXED AS OF  /80.02,GC/
                       2167  *
                       2168  *       ENTRY   NONE
                       2169  *       EXIT    NONE
                       2170  *       USES    NONE
                       2171  *
                       2172
053.333  315 054 031   2173  MAM     CALL    $SAVALL
053.336  052 320 040   2174          LHLD    S.SYSM
053.341  021 366 377   2175          LXI     D,-10                                   /79.05.sc/
053.344  031           2176          DAD     D                                       /79.05.sc/
053.345  303 372 053   2177          JMP     MIM1            REQUEST AND STORE       /80.02.GC/


                       2179  **      MIM - REQUEST MINIMUM MEMORY.
                       2180  *
                       2181  *       MIM SETS THE CURRENT PROGRAM SIZE TO THE MINIMUM POSSIBLE
                       2182  *       (IMMEDIATELY ABOVE THE LAST TEXT IN MEMORY)
                       2183  *
                       2184  *       ENTRY   NONE
                       2185  *       EXIT    NONE
                       2186  *       USES    NONE
                       2187
                       2188
053.350  315 054 031   2189  MIM     CALL    $SAVALL
053.353  052 122 061   2190          LHLD    LALPTR
053.356  174           2191          MOV     A,H
053.357  265           2192          ORA     L
053.360  302 366 053   2193          JNZ     MIM0            HAVE TEXT
                       2194
                       2195  *       NO TEXT, JUST LOOK AT BUFFER SIZE
                       2196
053.363  041 077 070   2197          LXI     H,BUFFER
```

```
                                2198
053,366  021 040 000  2199  MIM0    LXI     D,32
053,371  031          2200          DAD     D               ADD SOME SLOP
053,372  042 140 061  2201  MIM1    SHLD    BUFMAX
053,375  353          2202          XCHG                    (DE) = NEW LIMIT
053,376  052 322 040  2203          LHLD    $.USRM
054,001  315 216 030  2204          CALL    $CDEHL          SEE IF ALREADY HAVE THAT AMOUNT
054,004  312 047 031  2205          JE      $RSTALL         DONT ASK, WE HAVE IT!
054,007  353          2206          XCHG                    (HL) = AMOUNT TO ASK FOR
054,010  377 052      2207          DB      SYSCALL,.SETTP
054,012  322 047 031  2208          JNC     $RSTALL         IF OK, RESTORE AND EXIT
054,015  303 157 053  2209          JMP     ERROR


                                2211  **      PLA - PRINT LINE AFTER.
                                2212  *
                                2213  *       PLA PRINTES THE LINE IF THE *A* OPTION HAS BEEN SPECIFIED.
                                2214  *
                                2215  *       ENTRY   (WRKPTR) = LINE POINTER
                                2216  *       EXIT    NONE
                                2217  *       USES    A,F
                                2218
                                2219
054,020  072 146 061  2220  PLA     LDA     OPTS
000,000                 2221          ERRNZ   OPT.A-1
054,023  037           2222          RAR
054,024  320           2223          RNC                     NOT SET
054,025  303 342 054   2224          JMP     TTX.            TYPE TEXT


                                2226  **      PLB - PRINT LINE BEFORE.
                                2227  *
                                2228  *       PLB PRINTS THE WORKING LINE IF TGE *BEFORE* OPTION IS
                                2229  *       SELECTED.
                                2230  *
                                2231  *       ENTRY   (WRKPTR) = NEXT LINE TO CONSIDER
                                2232  *       EXIT    (HL) = (WRKPTR)
                                2233  *       USES    A,F,H,L
                                2234
                                2235
054,030  072 146 061  2236  PLB     LDA     OPTS
054,033  346 002      2237          ANI     OPT.B
054,035  310          2238          RZ                      NOT SET
054,036  303 342 054  2239          JMP     TTX.            TYPE TEXT
```

```
                                2241  **      RCR - REQUIRE CARRIAGE RETURN.
                                2242  *
                                2243  *       RCR IS CALLED BY THOSE COMMANDS WHICH END WITH A CARRIAGE
                                2244  *       RETURN, TOO MAKE SURE THAT CARRIAGE RETURN WAS ENTERED.
                                2245  *
                                2246  *       ENTRY   NONE
                                2247  *       EXIT    NONE
                                2248  *       USES    A,F
                                2249
                                2250
 054.041  315 205 053          2251  RCR     CALL    GNC
 054.044  376 012              2252          CPI     NL
 054.046  302 031 043          2253          JNE     REFUSE          NO GOOD
 054.051  315 001 056          2254          CALL    $CRLF           ECHO CRLF
 054.054  345                  2255          PUSH    H               SAVE (HL)
 054.055  052 124 061          2256          LHLD    CRFPTR
 054.060  042 132 061          2257          SHLD    PCFPTR          SAVE PREVIOUS COMMAND BOUNDS
 054.063  052 126 061          2258          LHLD    CRLPTR
 054.066  042 134 061          2259          SHLD    PCLPTR
 054.071  341                  2260          POP     H
 054.072  311                  2261          RET


                                2263  **      RQS - READ QUOTED STRING
                                2264  *
                                2265  *       RQS READS A QUOTED STRING FROM THE INPUT LINE, AND PLACES
                                2266  *       IT IN MEMORY.
                                2267  *
                                2268  *       ENTRY   (HL) = ADDRESS FOR STRING
                                2269  *       EXIT    (HL) = UNCHANGED
                                2270  *               STRING IN MEMORY
                                2271  *       USES    A,F
                                2272
                                2273
 054.073  345                  2274  RQS     PUSH    H
 054.074  325                  2275          PUSH    D               SAVE (DE)
 054.075  315 205 053          2276          CALL    GNC             READ INITIAL QUOTE
 054.100  026 050              2277          MVI     D,40
                                2278
                                2279  *       READ ANOTHER CHARACTER
                                2280
 054.102  025                  2281  RQS1    DCR     D
 054.103  312 031 043          2282          JZ      REFUSE          TOO MANY CHARACTERS
 054.106  315 217 053          2283          CALL    GTC             GET TEXT CHARACTER
 054.111  376 047              2284          CPI     QUOTE
 054.113  167                  2285          MOV     M,A             STORE IN MEMORY
 054.114  043                  2286          INX     H
 054.115  302 102 054          2287          JNE     RQS1            NOT QUOTE
                                2288
                                2289  *       HAVE QUOTE
                                2290
 054.120  315 064 053          2291          CALL    ENC             EXAMINE NEXT
 054.123  376 047              2292          CPI     QUOTE
 054.125  302 136 054          2293          JNE     RQS2            SONGLE QUOTE - EXIT
```

```
                                2294
                                2295  *       HAVE DOUBLE QUOTE
                                2296
054.130  315 217 053            2297          CALL    GTC             READ '
054.133  303 102 054            2298          JMP     RQS1
                                2299
                                2300  *       END OF STRING
                                2301
054.136  053                    2302  RQS2    DCX     H
054.137  066 000                2303          MVI     M,0             END STRING
054.141  321                    2304          POP     D
054.142  341                    2305          POP     H
054.143  311                    2306          RET


                                2308  **      RSL - REPLACE SINGLE LINE.
                                2309  *
                                2310  *       RSL REPLACES A SINGLE LINE IN THE TEXT BLOCK WITH A LINE
                                2311  *       IN MEMORY.
                                2312  *
                                2313  *       ENTRY   (HL) = REPLACEMENT LINE ADDRESS
                                2314  *               (C) = LENGTH
                                2315  *               (WRKPTR) = ADDRESS IN BLOCK OF LINE TO REPLACE
                                2316  *       EXIT    LINE REPLACED
                                2317  *       USES
                                2318
                                2319
054.144  315 255 052            2320  RSL     CALL    DCC             DISABLE CTL-C
054.147  353                    2321          XCHG
054.150  052 130 061            2322          LHLD    WRKPTR
054.153  315 361 054            2323          CALL    $CLL            CHECK OLD LINE LENGTH
054.156  221                    2324          SUB     C               OLD - NEW               /80.02.GC/
054.157  332 170 054            2325          JC      RSL1            OLD < NEW               /80.02.GC/
                                2326
                                2327  *       OLD >= NEW, DELETE EXTRA BYTES                  /80.02.GC/
                                2328
054.162  315 337 052            2329          CALL    DTBK            DELETE BLOCK            /80.02.GC/
054.165  303 175 054            2330          JMP     RSL2                                    /80.02.GC/
                                2331
                                2332  *       OLD <  NEW, INSERT EXTRA BYTES                  /80.02.GC/
                                2333
054.170  057                    2334  RSL1    CMA                                             /80.02.GC/
054.171  074                    2335          INR     A                                       /80.02.GC/
054.172  315 244 053            2336          CALL    ITBK            INSERT BLOCK            /80.02.GC/
000.000                         2337          ERRNZ   *-RSL2                                  /80.02.GC/
                                2338
                                2339  *       MOVE THE TEXT ACTUALLY IN
                                2340
054.175                         2341  RSL2    EQU     *                                       /80.02.GC/
054.175  006 000                2342          MVI     B,0
054.177  315 252 030            2343          CALL    $MOVE           ISNERT LINE
054.202  315 044 053            2344          CALL    ECC             RESTORE CTL-C PROCESSING
054.205  303 020 054            2345          JMP     PLA             PRINT LINE AFTER AND RETURN
```

```
                                2347  **      R8N - READ 8 BIT NUMBER.
                                2348  *
                                2349  *       R8N READS AN 8 BIT NUMBER FROM THE COMMAND STREAM.
                                2350  *
                                2351  *       ENTRY   NONE
                                2352  *       EXIT    (A) = VALUE
                                2353  *               TO 'REFUSE' IF BAD
                                2354  *       USES    A,B,C,F
                                2355
                                2356
054.210  315 265 052            2357  R8N     CALL    DDN             DECODE NUMBER
054.213  170                    2358          MOV     A,B
054.214  247                    2359          ANA     A
054.215  302 031 043            2360          JNZ     REFUSE          TOO LARGE
054.220  171                    2361          MOV     A,C             (A) = VALUE
054.221  311                    2362          RET


                                2364  **      SEL - SCAN FOR ELIGIBLE LINE.
                                2365  *
                                2366  *       SEL SCANS TO FIND THE NEXT LINE MEETING THE QUALIFIER STRING.
                                2367  *
                                2368  *       * * NOTE * *  'DELETE' ASSUMES THAT SEL ONLY CHECKS FOR
                                2369  *               QUALIFIER STRINGS IN Q'QUALS', AND SKIPS
                                2370  *               CALLING SEL IF 'QUALS' IS 00. THIS MUST BE MODIFIED IF MORE
                                2371  *               QUALIFIACTION SPECIFICATIONS ARE ALLOWED IN THE FUTURE.
                                2372  *
                                2373  *       ENTRY   (WRKPTR) = NEXT LINE TO CONSIDER
                                2374  *       EXIT    (WRKPTR) = NEXT LINE TO PROCESS
                                2375  *               (HL) = (WRKPTR)
                                2376  *               'Z' SET IF NO MORE LINES
                                2377  *       USES    A,F,H,L
                                2378
                                2379
054.222  315 045 052            2380  SEL1    CALL    ACL             ADVANCE COMMAND LINE
054.225  310                    2381          RZ                      DONE
                                2382
054.226  315 171 052            2383  SEL.    CALL    CBO             CHECK FOR BUFFER OVERFLOW
054.231  052 130 061            2384  SEL     LHLD    WRKPTR
054.234  174                    2385          MOV     A,H
054.235  265                    2386          ORA     L
054.236  310                    2387          RZ                      NO TEXT ESISTS
054.237  041 001 063            2388          LXI     H,QUALS
054.242  176                    2389          MOV     A,M
054.243  247                    2390          ANA     A
054.244  312 257 054            2391          JZ      SEL2            NO QUAL STRING
                                2392
                                2393  *       SEE IF MEET QUALIFIER STRING
                                2394
054.247  325                    2395          PUSH    D
054.250  315 322 053            2396          CALL    LQS             LOCATE QUOTED STRING
054.253  321                    2397          POP     D
054.254  302 222 054            2398          JNZ     SEL1            DONT HAVE IT
                                2399
```

```
                        2400  *       HAVE QUALIFIED LINE.
                        2401
054.257  052 130 061    2402  SEL2    LHLD    WRKPTR
054.262  264            2403          ORA     H               CLEAR 'Z'
054.263  311            2404          RET


                        2406  **      SFS - SEARCH FOR STRING.
                        2407  *
                        2408  *       SFS SCANS AN EXPANDED CHARACTER STRING FOR A MATCH FOR
                        2409  *       SOME PATTERN STRING
                        2410  *
                        2411  *       ENTRY   (DE) = STRING ADDRESS
                        2412  *               (HL) = LINE ADDRESS
                        2413  *       EXIT    (DE) UNCHANGED
                        2414  *               (HL) = ADDRESS OF 1ST MATCH CHARACTER
                        2415  *       USES    A,F,H,L
                        2416
                        2417
054.264  325            2418  SFS     PUSH    D               SAVE STRING ADDRESS
054.265  345            2419          PUSH    H
054.266  176            2420          MOV     A,M
054.267  247            2421          ANA     A
054.270  076 001        2422          MVI     A,1
054.272  312 316 054    2423          JZ      SFS2            NOT FOUND - NO MORE TEXT
                        2424
                        2425  *       COMPARE STRINGS
                        2426
054.275  032            2427  SFS1    LDAX    D
054.276  247            2428          ANA     A
054.277  312 316 054    2429          JZ      SFS2            A MATCH
054.302  276            2430          CMP     M
054.303  023            2431          INX     D
054.304  043            2432          INX     H
054.305  312 275 054    2433          JE      SFS1            KEEP TRYING
                        2434
                        2435  *       A FAILURE
                        2436
054.310  341            2437          POP     H
054.311  321            2438          POP     D
054.312  043            2439          INX     H
054.313  303 264 054    2440          JMP     SFS
                        2441
054.316  341            2442  SFS2    POP     H
054.317  321            2443          POP     D
054.320  247            2444          ANA     A               SET 'Z' IF FOUND
054.321  311            2445          RET
```

```
                                2447  **      SLB - SCAN LINE BACKWARDS.
                                2448  *
                                2449  *       SLB SCANS BACKWARDS OVER THE PREVIOUS LINE.
                                2450  *
                                2451  *       ENTRY   (HL) = 1ST BYTE OF CURRENT LINE
                                2452  *       EXIT    (HL) = FIRST BYTE OF PREVIOUS LINE
                                2453  *       USES    A;F;H;L
                                2454
                                2455
054.322  053                   2456  SLB     DCX     H
054.323  053                   2457  SLB1    DCX     H
054.324  176                   2458          MOV     A,M
054.325  247                   2459          ANA     A
054.326  302 323 054            2460          JNZ     SLB1
054.331  043                   2461          INX     H
054.332  311                   2462          RET


                                2464  **      SNL - SCAN TO NEXT LINE.
                                2465  *
                                2466  *       SNL SCANS THE TEXT BLOCK FOR THE NEXT LINE.
                                2467  *
                                2468  *       ENTRY   (HL) = START OF CURRENT LINE
                                2469  *       EXIT    (HL) = START OF NEXT LINE
                                2470  *       USES    A;F;H
                                2471
                                2472
054.333  176                   2473  SNL     MOV     A,M
054.334  043                   2474          INX     H
054.335  247                   2475          ANA     A
054.336  302 333 054            2476          JNZ     SNL
054.341  311                   2477          RET


                                2479  **      TTX - TYPE TEXT LINE.
                                2480  *
                                2481  *       TTX TYPES THE TEXT FOR A LINE.
                                2482  *
                                2483  *       ENTRY   (HL) = FIRST BYTE
                                2484  *       EXIT    (HL) UNCHANGED
                                2485  *       USES    A;F
                                2486
                                2487
054.342  052 130 061            2488  TTX.    LHLD    WRKPTR
054.345  315 361 054            2489  TTX     CALL    $CLL            COMPUTE LENGTH
054.350  345                   2490          PUSH    H               SAVE ADDRESS
054.351  075                   2491          DCR     A               REMOVE COUNT OF '00'
054.352  315 314 055            2492          CALL    $TYPCC          TYPE IT
054.355  341                   2493          POP     H
054.356  303 001 056            2494          JMP     $CRLF
```

```
054.361                 2497            XTEXT   CLL


                        2499X **        CLL - COMPUTE LINE LENGTH.
                        2500X *
                        2501X *         CLL COUNTS THE NUMBER OF CHARACTERS IN A SOURCE LINE.
                        2502X *         THE LINE IS TERMINATED BY A 00 BYTE; THE 00 BYTE IS ENCLUDED
                        2503X *         IN THE COUNT.
                        2504X *
                        2505X *         ENTRY   (HL) = FWA OF LINE
                        2506X *         EXIT    (HL) UNCHANGED
                        2507X *                 (A) = LENGTH OF LINE
                        2508X *         USES    A,F
                        2509X
                        2510X
054.361 345             2511X $CLL      PUSH    H               SAVE STARTING ADDRESS
054.362 325             2512X           PUSH    D
054.363 026 000         2513X           MVI     D,0
                        2514X
054.365 176             2515X CLL1      MOV     A,M
054.366 024             2516X           INR     D
054.367 247             2517X           ANA     A
054.370 043             2518X           INX     H
054.371 302 365 054     2519X           JNZ     CLL1            SCAN FOR END
054.374 172             2520X           MOV     A,D
054.375 321             2521X           POP     D
054.376 341             2522X           POP     H
054.377 311             2523X           RET
055.000                 2524            XTEXT   CCO


                        2526X **        $CCO - CLEAR CONTROL-O
                        2527X *
                        2528X *         $CCO IS CALLED TO CLEAR THE EFFECT OF THE CTL-O CHARACTER.
                        2529X *
                        2530X *         ENTRY   NONE
                        2531X *         EXIT    NONE
                        2532X *         USES    NONE
                        2533X
                        2534X
055.000 315 054 031     2535X $CCO      CALL    $SAVALL         SAVE REGISTERS
055.003 076 004         2536X           MVI     A,I.CONFL
055.005 001 001 000     2537X           LXI     B,CO.FLG        CLEAR CO.FLG
055.010 377 006         2538X           DB      SYSCALL,.CONSL
055.012 303 047 031     2539X           JMP     $RSTALL         RESTORE REGISTERS AND RETURN
055.015                 2540            XTEXT   INCHA
```

```
                                2542X **      $INCHA - READ ONE CHARACTER.
                                2543X *
                                2544X *       $INCHA READS ONE CHARACTER FROM THE TERMINAL.
                                2545X *
                                2546X *       CHAR = CTL-U: ERASE LINE
                                2547X *            = BKSP: BACKSPACE CHARACTER
                                2548X *              = RUBOUT: BACKSPACE CHARACTER
                                2549X
                                2550X ********************************************************************************
                                2551X **
P   000.001                     2552X         ERRNZ   I                 THIS ROUTINE IS OBSOLETE
                                2553X
                                2554X ********************************************************************************
                                2555X
                                2556X
    055.015  315 337 055        2557X $INCHA  CALL    $RCHAR            READ A CHARACTER
    055.020  376 010            2558X         CPI     BKSP
    055.022  312 063 055        2559X         JE      INC0              IS BKSP
    055.025  376 177            2560X         CPI     RUBOUT
    055.027  312 063 055        2561X         JE      INC0              IS RUBOUT
    055.032  365                2562X         PUSH    PSW               SAVE CODE
    055.033  072 150 055        2563X         LDA     $INCHAA           (A) = RUBOUT FLAG
    055.036  247                2564X         ANA     A
    055.037  304 345 055        2565X         CNZ     $WCHAR            ECHO RUBOUT CHAR, IF ANY
    055.042  257                2566X         XRA     A
    055.043  062 150 055        2567X         STA     $INCHAA           CLEAR FLAG
    055.046  361                2568X         POP     PSW
    055.047  376 025            2569X         CPI     'U'-'@'
    055.051  300                2570X         RNE                       NOT CTL-U, RETURN
                                2571X
                                2572X *       IS CTL-U
                                2573X
    055.052  041 277 061        2574X         LXI     H,LINE
    055.055  315 001 056        2575X         CALL    $CRLF
    055.060  303 112 055        2576X         JMP     INC1              CLEAR LINE AND SET LINPTR
                                2577X
                                2578X *       IS BKSP
                                2579X
    055.063  052 142 061        2580X INC0    LHLD    LINPTR
    055.066  076 277            2581X         MVI     A,$LINE
    055.070  275                2582X         CMP     L
    055.071  312 015 055        2583X         JE      $INCHA            IF ALREADY AT FRONT
    055.074  053                2584X         DCX     H
    055.075  072 327 040        2585X         LDA     S.CONTY           SEE IF BACKSPACING
    055.100  247                2586X         ANA     A
    055.101  362 122 055        2587X         JP      INC3              IS NON-CRT
    055.104  315 136 031        2588X         CALL    $TYPTX
    055.107  010 040 210        2589X         DB      BKSP,' ',BKSP+200Q      BACKSPACE FOR CRT
    055.112  042 142 061        2590X INC1    SHLD    LINPTR
    055.115  066 000            2591X         MVI     M,0               CLEAR ENTRY
    055.117  303 015 055        2592X         JMP     $INCHA            AGAIN
                                2593X
                                2594X *       BACKSPACE FOR NON-CRT
                                2595X
    055.122  072 150 055        2596X INC3    LDA     $INCHAA           (A) = FLAG
    055.125  247                2597X         ANA     A
```

```
055,126  302 141 055  2598X          JNZ     INC4            AM STILL BACKSPACING
055,131  076 057      2599X          MVI     A,'/'
055,133  062 150 055  2600X          STA     $INCHAA         SET FLAG
055,136  315 345 055  2601X          CALL    $WCHAR          TYPE
055,141  176         2602X INC4     MOV     A,M
055,142  315 345 055  2603X          CALL    $WCHAR          SHOW CHARACTER BEING REMOVED
055,145  303 112 055  2604X          JMP     INC1            CLEAR IT
                      2605X
055,150  000         2606X $INCHAA  DB      0               RUBOUT FLAG
055,151               2607           XTEXT   UDD


                      2609X **       $UDD - UNPACK DECIMAL DIGITS.
                      2610X *
                      2611X *        UDD CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
                      2612X *        DECIMAL DIGITS. THE RESULT IS ZERO FILLED.
                      2613X *
                      2614X *        ENTRY   (B,C) = ADDRESS VALUE
                      2615X *                (A) = DIGIT COUNT
                      2616X *                (H,L) = MEMORY ADDRESS
                      2617X *        EXIT    (HL) = (HL) + (A)
                      2618X *        USES    ALL
                      2619X
                      2620X
031,157               2621X $UDD     EQU     31157A          IN H17 ROM
055,151               2622           XTEXT   MLU


                      2624X **       MLU - MAP LOWER CASE LINE TO UPPER CASE.
                      2625X *
                      2626X *        MLU MAPS THE LOWER CASE ALPHABETICS IN A LINE TO UPPER CASE.
                      2627X *
                      2628X *        ENTRY   (HL)  = LINE FWA
                      2629X *        EXIT    NONE
                      2630X *        USES    NONE
                      2631X
                      2632X
055,151  365         2633X $MLU     PUSH    PSW             SAVE (PSW)
055,152  345         2634X          PUSH    H               SAVE FWA
055,153  053         2635X          DCX     H               ANTICIPATE INX H
055,154  043         2636X $MLU1    INX     H
055,155  176         2637X          MOV     A,M             (A)= CHARACTER
055,156  315 205 055  2638X          CALL    $MCU            MAP CHAR TO UPPER
055,161  167         2639X          MOV     M,A
055,162  247         2640X          ANA     A
055,163  302 154 055  2641X          JNZ     $MLU1           MORE TO GO
055,166  341         2642X          POP     H               RESTORE (HL)
055,167  361         2643X          POP     PSW             RESTORE (PSW)
055,170  311         2644X          RET
055,171               2645           XTEXT   GNL
```

```
                                2647X **      $GNL - GUARANTEE NEW LINE.
                                2648X *
                                2649X *       $GNL GUARANTEES THE START OF A NEW LINE BY ISSUING A CRLF
                                2650X *       IF THE CURSOR IS NOT AT COLUMN 1..
                                2651X *
                                2652X *       ENTRY   NONE
                                2653X *       EXIT    NONE
                                2654X *       USES    ALL
                                2655X
                                2656X
055.171  076 002               2657X $GNL    MVI     A,I.CUSOR
055.173  001 000 000           2658X         LXI     B,0
055.176  377 006               2659X         DB      SYSCALL,.CONSL          READ CURSOR
055.200  075                   2660X         DCR     A
055.201  310                   2661X         RZ                      AT COLUMN 1
055.202  303 001 056           2662X         JMP     $CRLF           NEW LINE
055.205                        2663          XTEXT   MCU


                                2665X **      MCU - MAP LOWER CASE TO UPPER CASE.
                                2666X *
                                2667X *       MCU MAPS A LOWER CASE ALPHABETIC TO UPPER
                                2668X *       CASE.
                                2669X *
                                2670X *       ENTRY   (A) = CHARACTER
                                2671X *       EXIT    (A) = CHARACTER RESULT
                                2672X *       USES    A,F
                                2673X
                                2674X
055.205  376 141               2675X $MCU    CPI     'a'
055.207  330                   2676X         RC                      NOT LOWER CASE
055.210  376 173               2677X         CPI     'z'+1
055.212  320                   2678X         RNC                     NOT LOWER CASE
055.213  326 040               2679X         SUI     'a'-'A'
055.215  311                   2680X         RET
055.216                        2681          XTEXT   CHL


                                2683X **      $CHL - COMPLEMENT (HL).
                                2684X *
                                2685X *       (HL) = -(HL)            TWO'S COMPLEMENT
                                2686X *
                                2687X *       ENTRY   NONE
                                2688X *       EXIT    NONE
                                2689X *       USES    A,F,H,L
                                2690X
                                2691X
030.224                         2692X $CHL    EQU     30224A          IN H17 ROM
055.216                         2693          XTEXT   HLCPDE                          /80.02.GC/
                                2694X **      HLCPDE - (HL) COMPARED TO (DE)
                                2695X *
                                2696X *       THIS ROUTINE IS DOUBLE WORD COMPARE OF REGISTER PAIRS (DE) AND (HL).
```

```
                        2697X *
                        2698X *       ENTRY:  (HL)$(DE) SET UP
                        2699X *
                        2700X *       EXIT:   (PSW) =
                        2701X *                       'Z' SET   IF (HL) =  (DE)
                        2702X *                       'C' SET   IF (HL) <  (DE)
                        2703X *                       'C' CLEAR IF (HL) >= (DE)
                        2704X *
                        2705X *
                        2706X *       USES:   (PSW)
                        2707X *
                        2708X
055.216  174           2709X HLCPDE  MOV     A,H
055.217  272           2710X         CMP     D       'C' SET => (A) < (D)
055.220  300           2711X         RNZ
055.221  175           2712X         MOV     A,L
055.222  273           2713X         CMP     E       'C' SET => (L) < (E)
055.223  311           2714X         RET
055.224                2715          XTEXT   SAVALL


                        2717X **      $RSTALL - RESTORE ALL REGISTERS.
                        2718X *
                        2719X *       $RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND
                        2720X *       RETURNS TO THE PREVIOUS CALLER.
                        2721X *
                        2722X *       ENTRY   (SP) = PSW
                        2723X *               (SP+2) = BC
                        2724X *               (SP+4) = DE
                        2725X *               (SP+6) = HL
                        2726X *               (SP+8) = RET
                        2727X *       EXIT    TO *RET*, REGISTERS RESTORED
                        2728X *       USES    ALL
                        2729X
                        2730X
031.047                 2731X $RSTALL EQU     31047A          IN H17 ROM


                        2733X **      $SAVALL - SAVE ALL REGISTERS ON STACK.
                        2734X *
                        2735X *       $SAVALL SAVES ALL THE REGISTERS ON THE STACK.
                        2736X *
                        2737X *       ENTRY   NONE
                        2738X *       EXIT    (SP) = PSW
                        2739X *               (SP+2) = BC
                        2740X *               (SP+4) = DE
                        2741X *               (SP+6) = HL
                        2742X *       USES    H,L
                        2743X
                        2744X
031.054                 2745X $SAVALL EQU     31054A          IN H17 ROM
055.224                 2746          XTEXT   RTL
```

```
                                2748X **      $RTL - READ TEXT LINE.
                                2749X *
                                2750X *       $RTL READS A LINE FROM THE TERMINAL.
                                2751X *
                                2752X *       CHARACTER ARE ACCEPTED FROM THE TERMINAL, RUBOUT AND BACKSPACE
                                2753X *       CHARACTERS ARE PROCESSED. WHEN A CARRIAGE RETURN IS ENTERED,
                                2754X *       $RTL RETURNS.
                                2755X *
                                2756X *       ENTRY   (HL) = BUFFER FWA
                                2757X *       EXIT    'C' CLEAR IF OK
                                2758X *                DATA IN BUFFER
                                2759X *                (A) = TEXT LENGTH
                                2760X *                'C' SET IF CTL-D STRUCK
                                2761X *       USES    A,F
                                2762X
                                2763X
055.224  315 233 055           2764X $RTL:   CALL    $RTL              $RTL IN UPPER CASE
055.227  330                   2765X         RC                      CTL-D
055.230  303 151 055           2766X         JMP     $MLU              MAP LINE TO UPPER CASE
                                2767X
055.233                        2768X $RTL    EQU     *
055.233  345                   2769X         PUSH    H                 SAVE FWA
055.234  315 337 055           2770X $RTL1   CALL    $RCHAR
055.237  376 004               2771X         CPI     CTLD
055.241  312 266 055           2772X         JE      $RTL2             CTL-D STRUCK
055.244  167                   2773X         MOV     M,A
055.245  043                   2774X         INX     H
055.246  376 012               2775X         CPI     NL
055.250  302 234 055           2776X         JNE     $RTL1
055.253  053                   2777X         DCX     H
055.254  066 000               2778X         MVI     M,0
055.256  043                   2779X         INX     H
                                2780X
                                2781X *       ALL DONE, COMPUTE LENGTH
                                2782X
055.257  353                   2783X         XCHG                      (DE) = LWA+1
055.260  343                   2784X         XTHL                      (HL) = FWA
055.261  173                   2785X         MOV     A,E
055.262  225                   2786X         SUB     L                 (A) = LENGTH
055.263  247                   2787X         ANA     A                 CLEAR CARRY
055.264  321                   2788X         POP     D                 RESTORE (DE)
055.265  311                   2789X         RET
                                2790X
                                2791X *       CTL-D STRUCK
                                2792X
055.266  341                   2793X $RTL2   POP     H                 (HL) = FWA
055.267  067                   2794X         STC
055.270  311                   2795X         RET
055.271                        2796          XTEXT   MOVEL
```

```
                                2798X **      $MOVEL - MOVE DATA
                                2799X *
                                2800X *       $MOVEL MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                                2801X *       IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                                2802X *       FIRST TO LAST.
                                2803X *
                                2804X *       IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                                2805X *       LAST TO FIRST.
                                2806X *
                                2807X *       THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                                2808X *
                                2809X *       CALL    $MOVEL
                                2810X *       DW      COUNT
                                2811X *       DW      FROM
                                2812X *       DW      TO
                                2813X *
                                2814X *       ENTRY   ((SP)) = RET
                                2815X *               (RET+0) = COUNT (WORD VALUE)
                                2816X *               (RET+2) = FROM
                                2817X *               (RET+4) = TO
                                2818X *       EXIT    TO (RET+6)
                                2819X *               (DE) = ADDRESS OF NEXT FROM BYTE
                                2820X *               (HL) = ADDRESS OF NEXT *TO* BYTE
                                2821X *               'C' CLEAR
                                2822X *       USES    ALL
                                2823X
                                2824X
    055.271  341                2825X $MOVEL  POP     H               (HL) = RET
    055.272  116                2826X         MOV     C,M
    055.273  043                2827X         INX     H
    055.274  106                2828X         MOV     B,M             (BC) = COUNT
    055.275  043                2829X         INX     H
    055.276  136                2830X         MOV     E,M
    055.277  043                2831X         INX     H
    055.300  126                2832X         MOV     D,M             (DE) = FROM
    055.301  043                2833X         INX     H
    055.302  325                2834X         PUSH    D               ((SP)) = FROM
    055.303  136                2835X         MOV     E,M
    055.304  043                2836X         INX     H
    055.305  126                2837X         MOV     D,M             (DE) = TO
    055.306  043                2838X         INX     H
    055.307  343                2839X         XTHL                    ((SP)) = RET, (HL) = FROM
    055.310  353                2840X         XCHG                    (DE) = FROM , (HL) = TO
    055.311  303 252 030        2841X         JMP     $MOVE           MOVE IT
    055.314                     2842          XTEXT   TYPCC


                                2844X **      $TYPCC - TYPE A CHARACTER STRING BY COUNT.
                                2845X *
                                2846X *       $TYPCC TYPES A STRING OF CHARACTERS. THE CALLER SUPPLIES
                                2847X *       THE CHARACTER ADDRESS AND COUNT.
                                2848X *
                                2849X *       ENTRY   (HL) = ADDRESS
                                2850X *               (A) = COUNT
```

```
                        2851X *       EXIT    (HL) = LAST CHARACTER ADDRESS+1
                        2852X *       USES    A,F,H,L
                        2853X
                        2854X
055.314                 2855X $TYPCC  EQU     *
055.314  247           2856X         ANA     A
055.315  310           2857X         RZ                      NOTHING TO TYPE
055.316  365           2858X         PUSH    PSW             SAVE COUNT
055.317  176           2859X         MOV     A,M             (A) = CHARACTER
055.320  043           2860X         INX     H
055.321  377 002       2861X         DB      SYSCALL,.SCOUT
055.323  361           2862X         POP     PSW
055.324  075           2863X         DCR     A
055.325  303 314 055   2864X         JMP     $TYPCC
055.330                2865          XTEXT   TYPCH


                        2867X **      $TYPCH - TYPE SINGLE CHARACTER.
                        2868X *
                        2869X *       ENTRY   (RET) = CHARACTER
                        2870X *       EXIT    TO (RET)+1
                        2871X *               (A) = CHARACTER TYPED
                        2872X
                        2873X
055.330  343           2874X $TYPCH  XTHL                    (HL) = RETURN ADDRESS
055.331  176           2875X         MOV     A,M             (A) = CHARACTER
055.332  043           2876X         INX     H
055.333  343           2877X         XTHL                    RESTORE ADVANCED EXIT ADDRESS
                        2878X
                        2879X **      $TYPC. - TYPE SINGLE CHARACTER.
                        2880X *
                        2881X *       ENTRY   (A) = CHARACTER
                        2882X *       EXIT    TO (RET)
                        2883X
055.334  377 002       2884X $TYPC.  DB      SYSCALL,.SCOUT
055.336  311           2885X         RET
055.337                2886          XTEXT   RCHAR


                        2888X **      $RCHAR - READ SINGLE CHARACTER FROM CONSOLE.
                        2889X *
                        2890X *       ENTRY   NONE
                        2891X *       EXIT    (A) = CHARACTER
                        2892X *       USES    A,F
                        2893X
                        2894X
055.337  377 001       2895X $RCHAR  DB      SYSCALL,.SCIN
055.341  332 337 055   2896X         JC      $RCHAR          NOT READY
055.344  311           2897X         RET
                        2898X
055.345  377 002       2899X $WCHAR  DB      SYSCALL,.SCOUT
055.347  311           2900X         RET
```

```
055.350                 2901          XTEXT   INDL


                        2903X **      $INDL - INDEXED LOAD,
                        2904X *
                        2905X *       $INDL LOADS DE WITH THE TWO BYTES AT (HL)+DISPLACMENT
                        2906X *
                        2907X *       THIS ACTS AS AN INDEXED FULL WORD LOAD,
                        2908X *
                        2909X *       (DE) = ( (HL) + DSPLACEMENT )
                        2910X *
                        2911X *       ENTRY   ((RET)) = DISPLACMENT (FULL WORD)
                        2912X *               (HL) = TABLE ADDRESS
                        2913X *       EXIT    TO (RET+2)
                        2914X *       USES    A,F,D,E
                        2915X
                        2916X
030.234                 2917X $INDL   EQU     30234A          IN H17 ROM
055.350                 2918          XTEXT   TBLS


                        2920X **      $TBLS - TABLE SEARCH
                        2921X *
                        2922X *       TABLE FORMAT
                        2923X *
                        2924X *       DB      KEY1,VAL1,
                        2925X *       .       .
                        2926X *       .       .
                        2927X *       DB      KEYN,VALN
                        2928X *       DB      0
                        2929X *
                        2930X *       ENTRY   (A) = PATTERN
                        2931X *               (H,L) = TABLE FWA
                        2932X *       EXIT    (A) = PATTERN IF FOUND
                        2933X *               'Z' SET IF FOUND
                        2934X *               'Z' CLEAR IF NOT FOUND OR PATTERN=0          /78.10.GC/
                        2935X *       USES    A,F,H,L
                        2936X
                        2937X
055.350  305           2938X $TBLS   PUSH    B
055.351  376 000       2939X         CPI     0                                            /78.10.GC/
055.353  312 375 055   2940X         JZ      TBL2                                         /78.10.GC/
055.356  107           2941X         MOV     B,A
055.357  176           2942X TBL1    MOV     A,M             (A) = CHARACTER
055.360  043           2943X         INX     H
055.361  270           2944X         CMP     B
055.362  312 377 055   2945X         JZ      TBL3            IF MATCH
055.365  247           2946X         ANA     A
055.366  043           2947X         INX     H               SKIP PAST
055.367  302 357 055   2948X         JNZ     TBL1            IF NOT END OF TABLE
055.372  053           2949X         DCX     H
055.373  053           2950X         DCX     H
```

```
055.374  257          2951X          XRA     A               SET TO ZERO FOR OLD USERS       /78.10.6C/
055.375  376 001      2952X TBL2     CPI     1               CLEAR ZERO                      /78.10.6C/
                      2953X
                      2954X *        DONE
                      2955X
055.377  301          2956X TBL3     POP     B
056.000  311          2957X          RET
056.001                2958           XTEXT   CDEHL



                      2960X **       $CDEHL - COMPARE (DE) TO (HL)
                      2961X *
                      2962X *        $CDEHL COMPARES (DE) TO (HL) FOR EQUALITY.
                      2963X *
                      2964X *        ENTRY   NONE
                      2965X *        EXIT    'Z' SET IF (DE) = (HL)
                      2966X *        USES    A,F
                      2967X
                      2968X
030.216               2969X $CDEHL   EQU     30216A          IN H17 ROM
056.001               2970           XTEXT   CRLF



                      2972X **       $CRLF - TYPE CARRIAGE RETURN/ LINE FEED
                      2973X *
                      2974X *        $CRLF IS USED TO GENERATE PADDED CRLF'S.
                      2975X *
                      2976X *        ENTRY   NONE
                      2977X *        EXIT    (A) = 0
                      2978X *        USES    A,F
                      2979X
                      2980X
056.001  076 012      2981X $CRLF    MVI     A,NL
056.003  377 002      2982X          DB      SYSCALL+.SCOUT
056.005  257          2983X          XRA     A
056.006  311          2984X          RET
056.007               2985           XTEXT   DADA



                      2987X **       $DADA - PERFORM (H,L) = (H,L) + (0,A)
                      2988X *
                      2989X *        ENTRY   (H,L) = BEFORE VALUE
                      2990X *                (A) = BEFORE VALUE
                      2991X *        EXIT    (H,L) = (H,L) + (0,A)
                      2992X *                'C' SET IF OVERFLOW
                      2993X *        USES    F,H,L
                      2994X
                      2995X
030.072               2996X $DADA    EQU     30072A          IN H17 ROM
056.007               2997           XTEXT   MOVE
```

```
                        2999X **      $MOVE - MOVE DATA
                        3000X *
                        3001X *       $MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                        3002X *       IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                        3003X *       FIRST TO LAST.
                        3004X *
                        3005X *       IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                        3006X *       LAST TO FIRST.
                        3007X *
                        3008X *       THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                        3009X *
                        3010X *       ENTRY   (BC) = COUNT
                        3011X *               (DE) = FROM
                        3012X *               (HL) = TO
                        3013X *       EXIT    MOVED
                        3014X *               (DE) = ADDRESS OF NEXT FROM BYTE
                        3015X *               (HL) = ADDRESS OF NEXT *TO* BYTE
                        3016X *               'C' CLEAR
                        3017X *       USES    ALL
                        3018X
                        3019X
030,252                 3020X $MOVE   EQU     30252A          IN H17 ROM
056.007                 3021          XTEXT   MU10


                        3023X **      $MU10 - MULTIPLY UNSIGNED 16 BIT QUANTITY BY 10.
                        3024X *
                        3025X *       (HL) = (DE)*10
                        3026X *
                        3027X *       ENTRY   (DE) = MULTIPLIER
                        3028X *       EXIT    'C' CLEAR IF OK
                        3029X *                (HL) = PRODUCT
                        3030X *                'C' SET IF ERROR
                        3031X *       USES    D,E,H,L,F
                        3032X
                        3033X
030.324                 3034X $MU10   EQU     30324A          IN H17 ROM
056.007                 3035          XTEXT   TBRA


                        3037X **      $TBRA - BRANCH RELATIVE THOUGH TABLE.
                        3038X *
                        3039X *       $TBRA USES THE SUPPLIED INDEX TO SELECT A BYTE FROM THE
                        3040X *       JUMP TABLE. THE CONTENTS OF THIS BYTE ARE ADDED TO THE
                        3041X *       ADDRESS OF THE BYTE, YEILDING THE PROCESSOR ADDRESS.
                        3042X *
                        3043X *       CALL    $TBRA
                        3044X *       DB      LAB1-*          INDEX = 0 FOR LAB1
                        3045X *       DB      LAB2-*          INDEX = 1 FOR LAB2
                        3046X *       DB      LABN-*          INDEX = N-1 FOR LABN
                        3047X *
                        3048X *       ENTRY   (A) = INDEX
```

```
                        3049X *             (RET) = TABLE FWA
                        3050X *       EXIT  TO COMPUTED ADDRESS
                        3051X *       USES  F,H,L
                        3052X
                        3053X
031.076                 3054X $TBRA   EQU   31076A          IN H17 ROM
056.007                 3055          XTEXT FOPE



                        3057X **      $FOPEx - OPEN FILE BLOCK FOR I/O
                        3058X *
                        3059X *       $FOPEx IS CALLED BEFORE ANY I/O IS DONE VIA A
                        3060X *       FILE BLOCK. $FOPEx SETS UP THE FILE BLOCK, AND OPENS
                        3061X *       THE FILE VIA *HDOS*.
                        3062X *
                        3063X *       ENTRY (DE) = ADDRESS OF DEFAULT BLOCK
                        3064X *             (HL) = ADDRESS OF FILE BLOCK
                        3065X *       EXIT  TO $FERROR IF ERROR
                        3066X *             TO CALLER IF OK
                        3067X *       USES  A,F,B,C,D,E
                        3068X
                        3069X
056.007  315 034 056    3070X $FOPER  CALL  $FOPER.
056.012  320            3071X         RNC
056.013  303 167 060    3072X         JMP   $FERROR         IN ERROR
                        3073X
056.016  315 037 056    3074X $FOPEW  CALL  $FOPEW.
056.021  320            3075X         RNC
056.022  303 167 060    3076X         JMP   $FERROR         IN ERROR
                        3077X
056.025  315 042 056    3078X $FOPEU  CALL  $FOPEU.
056.030  320            3079X         RNC
056.031  303 167 060    3080X         JMP   $FERROR         IN ERROR
                        3081X
                        3082X
056.034  076 002        3083X $FOPER. MVI   A,FT.OR         FILE TYPE OF OPEN FOR READ
056.036  001            3084X         DB    001Q            LXI,B TO SKIP NEXT MVI
056.037  076 004        3085X $FOPEW. MVI   A,FT.OW         OPEN FOR WRITE
056.041  001            3086X         DB    001Q            LXI,B TO SKIP NEXT MIV
056.042  076 006        3087X $FOPEU. MVI   A,FT.OR+FT.OW
                        3088X
                        3089X *       (A) = FILE FLAGS
                        3090X
056.044  345            3091X         PUSH  H               SAVE FILE BLOCK ADDRESS
056.045  365            3092X         PUSH  PSW             SAVE NEW FLAGS
000.000                 3093X         ERRNZ FB.CHA
056.046  106            3094X         MOV   B,M             (B) = CHANNEL NUMBER
056.047  305            3095X         PUSH  B               SAVE HANNEL NUMBER
000.000                 3096X         ERRNZ FB.FLG-FB.CHA-1
056.050  043            3097X         INX   H
056.051  117            3098X         MOV   C,A             (C) = NEW FILE FLAGS
056.052  176            3099X         MOV   A,M             (A) = CURRENT TYPE
056.053  247            3100X         ANA   A
056.054  171            3101X         MOV   A,C             (A) = NEW FLAGS TO BE SET
```

```
056.055  312 067 056  3102X          JZ      $FOPE1          NOT ALREADY OPEN
                      3103X
                      3104X *        ALREADY OPEN. SQUACK
                      3105X
056.060  301          3106X          POP     B               RESTORE (BC)
056.061  361          3107X          POP     PSW             DISCARD NEW FLAGS
056.062  341          3108X          POP     H               (HL) = FB ADDRESS
056.063  076 031      3109X          MVI     A,EC.FAO        FILE ALREADY OPEN
056.065  067          3110X          STC
056.066  311          3111X          RET
                      3112X
000.000               3113X          ERRNZ   FB.FWA-FB.FLG-1
056.067  043          3114X $FOPE1   INX     H               (HL) = $FB.FWA
056.070  116          3115X          MOV     C,M
056.071  043          3116X          INX     H
056.072  106          3117X          MOV     B,M             (BC) = FB.FWA
056.073  043          3118X          INX     H
000.000               3119X          ERRNZ   FB.PTR-FB.FWA-2
056.074  161          3120X          MOV     M,C             SET FB.PTR = FB.FWA
056.075  043          3121X          INX     H
056.076  160          3122X          MOV     M,B
056.077  043          3123X          INX     H
000.000               3124X          ERRNZ   FB.LIM-FB.PTR-2
056.100  161          3125X          MOV     M,C             SET FB.LIM = FB.FWA
056.101  043          3126X          INX     H
056.102  160          3127X          MOV     M,B
056.103  043          3128X          INX     H
000.000               3129X          ERRNZ   FB.NAM-FB.LIM-4
056.104  043          3130X          INX     H
056.105  043          3131X          INX     H               (HL) = $FB.NAM
                      3132X
                      3133X *        FILE BLOCK POINTERS SETUP. OPEN FILE
                      3134X
056.106  345          3135X          PUSH    H               SAVE NEW ADDRESS FOR NAME
056.107  041 140 056  3136X          LXI     H,$FOPEB
056.112  247          3137X          ANA     A                                       /78.10.GC/
056.113  312 122 056  3138X          JZ      $FOPE2
000.000               3139X          ERRNZ   .EXIT
056.116  315 350 055  3140X          CALL    $TBLS           FIND CODE
056.121  176          3141X          MOV     A,M
056.122  062 130 056  3142X $FOPE2   STA     $FOPEA          SET SYSCALL CODE
056.125  341          3143X          POP     H               (HL) = $FB.NAM
056.126  361          3144X          POP     PSW             (A) = CHANNEL NUMBER
056.127  377 000      3145X          DB      SYSCALL.,.EXIT
056.130               3146X $FOPEA   EQU     *-1             SYSCALL CODE
056.131  321          3147X          POP     D               (D) = NEW FLAG
056.132  341          3148X          POP     H               (HL) = FILE BLOCK ADDRESS
056.133  330          3149X          RC                      EXIT IF ERROR
056.134  043          3150X          INX     H
000.000               3151X          ERRNZ   FB.FLG-1
056.135  162          3152X          MOV     M,D             SET NEW FLAGS
056.136  053          3153X          DCX     H               RESTORE (HL)
056.137  311          3154X          RET
                      3155X
056.140  002 042      3156X $FOPEB   DB      FT.OR,.OPENR    TABLE OF SYSCALL CODES
056.142  004 043      3157X          DB      FT.OW,.OPENW
```

```
056.144  006 044       3158X          DB      FT.OR+FT.OW,.OPENU
056.146  000           3159X          DB      0               SHOULD NOT OCCUR
056.147                3160           XTEXT   FCLO


                       3162X **       $FCLO - CLOSE FILE BLOCK.
                       3163X *
                       3164X *        $FCLO IS CALLED TO TERMINATE PROCESSING THROUGH A FILE
                       3165X *        BLOCK.
                       3166X *
                       3167X *        ENTRY   (HL) = FILE BLOCK ADDRESS
                       3168X *        EXIT    TO $FERROR IF ERROR
                       3169X *                 TO CALLER IF OK
                       3170X *        USES    A,F,B,C,D,E
                       3171X
                       3172X
056.147  315 156 056   3173X $FCLO    CALL    $FCLO.
056.152  320           3174X          RNC                     NO ERROR
056.153  303 167 060   3175X          JMP     $FERROR
                       3176X
056.156  345           3177X $FCLO.    PUSH    H               SAVE FILE BLOCK ADDRESS
000.000                3178X          ERRNZ   FB.FLG-1
056.157  043           3179X          INX     H               (HL) = $FB.FLG
056.160  176           3180X          MOV     A,M
056.161  066 000       3181X          MVI     M,0             CLEAR FLAG
056.163  247           3182X          ANA     A
056.164  312 252 056   3183X          JZ      $FCLO4          FILE NOT OPEN
056.167  346 004       3184X          ANI     FT.OW
056.171  312 244 056   3185X          JZ      $FCLO3          NO WRITING, NO FLUSHING NEEDED
                       3186X
                       3187X *        WAS OPEN FOR WRITE. SEE IF NEED FLUSH THE LAST SECTOR
                       3188X
056.174  315 234 030   3189X          CALL    $INDL
056.177  003 000       3190X          DW      FB.PTR-FB.FLG
056.201  325           3191X          PUSH    D               SAVE (FB.PTR)
056.202  315 234 030   3192X          CALL    $INDL           (DE) = (FB.FWA)
056.205  001 000       3193X          DW      FB.FWA-FB.FLG
056.207  341           3194X          POP     H               (HL) = (FB.PTR)
056.210  175           3195X          MOV     A,L
056.211  223           3196X          SUB     E
056.212  117           3197X          MOV     C,A
056.213  174           3198X          MOV     A,H
056.214  232           3199X          SBB     D
056.215  107           3200X          MOV     B,A             (BC) = AMOUNT IN BLOCK
056.216  261           3201X          ORA     C
056.217  312 244 056   3202X          JZ      $FCLO3          NONE TO FLUSH
                       3203X
                       3204X *        NEED TO FLUSH BUFFER
                       3205X *
                       3206X *        (BC) = DATA AMOUNT
                       3207X *        (DE) = FWA
                       3208X *        (HL) = LWA+1
                       3209X
056.222  171           3210X          MOV     A,C
```

```
056.223  247            3211X          ANA     A
056.224  312 237 056     3212X          JZ      $FCLO2          DONT HAVE PARTIAL SECTOR
                         3213X
                         3214X *        ZERO FILL PARTIAL SECTOR
                         3215X
056.227  066 000         3216X $FCLO1   MVI     M,0
056.231  043            3217X          INX     H
056.232  014            3218X          INR     C
056.233  302 227 056     3219X          JNZ     $FCLO1
056.236  004            3220X          INR     B               COUNT ANOTHER FULL SECTOR
056.237  341            3221X $FCLO2   POP     H               (HL) = FB FWA
056.240  176            3222X          MOV     A,M             (A) = CHANNEL NUMBER
000.000                 3223X          ERRNZ   FB.CHA
056.241  345            3224X          PUSH    H
056.242  377 005         3225X          DB      SYSCALL,.WRITE          FLUSH
                         3226X
                         3227X *        READY TO CLOSE FILE.
                         3228X *
                         3229X *        'C' SET IF ERROR
                         3230X *         (A) = ERROR CODE
                         3231X
056.244  341            3232X $FCLO3   POP     H               (HL) = FILE BLOCK ADDRESS
056.245  330            3233X          RC                      ERROR
000.000                 3234X          ERRNZ   FB.CHA
056.246  176            3235X          MOV     A,M             (A) = CHANNEL NUMBER
056.247  345            3236X          PUSH    H
056.250  377 046         3237X          DB      SYSCALL,.CLOSE  CLOSE CHANNEL
056.252  341            3238X $FCLO4   POP     H               (HL) = FILE BLOCK ADDRESS
056.253  311            3239X          RET
056.254                 3240           XTEXT   FREAL



                         3242X **       $FREAL - READ BYTES FROM FILE BUFFER.
                         3243X *
                         3244X *        $FREAL IS CALLED TO READ A NUMBER OF BYTES FROM A FILE BUFFER.
                         3245X *
                         3246X *        ENTRY   (BC) = BYTE COUNT
                         3247X *                (DE) = FWA FOR BYTES
                         3248X *                (HL) = ADDRESS OF FILE BUFFER
                         3249X *        EXIT    TO *FERROR* IF ERROR
                         3250X *                TO CALLER IF OK
                         3251X *                 (BC) = UNREAD BYTE COUNT (ONLY IF EOF)
                         3252X *                 (DE) = ADDRESS OF FIRST UNUSED BYTE
                         3253X *                 'C' SET IF EOF DURING READ
                         3254X *        USES    A,F,B,C,D,E
                         3255X
                         3256X
056.254  315 267 056     3257X $FREAL   CALL    $FREAL.
056.257  320            3258X          RNC                     RETURN IF OK
056.260  376 001         3259X          CPI     EC.EOF
056.262  302 167 060     3260X          JNE     $FERROR         ERROR IS NOT EOF
056.265  067            3261X          STC
056.266  311            3262X          RET                     ERROR IS SIMPLY EOF
                         3263X
```

```
                        3264X
056.267                 3265X $FREAL. EQU   *
056.267  013            3266X         DCX   B             (BC) = COUNT NOT ENCLUDING 00 BYTE
056.270  257            3267X         XRA   A
056.271  062 166 060    3268X         STA   EOFFLG        CLEAR EOF FLAG
056.274  345            3269X         PUSH  H
056.275  315 012 060    3270X         CALL  CBT           COPY BUFFER POINTERS TO TEMP CELLS
                        3271X
                        3272X *       COPY DATA FROM BUFFER TO TARGET
                        3273X
056.300  325            3274X $REAL2  PUSH  D             SAVE TARGET ADDRESS
056.301  072 155 060    3275X         LDA   T.FLG
056.304  346 002        3276X         ANI   FT.OR
056.306  076 011        3277X         MVI   A,EC.FNO
056.310  067            3278X         STC                 ASSUME FILE NOT OPEN
056.311  312 045 057    3279X         JZ    $REAL8        ERROR
056.314  170            3280X         MOV   A,B
056.315  261            3281X         ORA   C
056.316  312 045 057    3282X         JZ    $REAL8        ALL DONE
                        3283X
                        3284X *       COMPUTE MIN( DATA IN BUFFER, DATA REQUESTED)
                        3285X
056.321  052 160 060    3286X $REAL3  LHLD  T.PTR
056.324  353            3287X         XCHG                (DE) = (FB.PTR) = ADDRESS OF DATA
056.325  052 162 060    3288X         LHLD  T.LIM         (HL) = LIMIT ADDRESS
056.330  175            3289X         MOV   A,L
056.331  223            3290X         SUB   E
056.332  157            3291X         MOV   L,A
056.333  174            3292X         MOV   A,H
056.334  232            3293X         SBB   D
056.335  147            3294X         MOV   H,A           (HL) = NUMBER OF BYTES IN BUFFER
056.336  171            3295X         MOV   A,C
056.337  225            3296X         SUB   L             COMPARE TO REQUESTED COUNT
056.340  170            3297X         MOV   A,B
056.341  234            3298X         SBB   H
056.342  322 347 056    3299X         JNC   $REAL4        LESS THAN REQUESTED COUNT
056.345  140            3300X         MOV   H,B
056.346  151            3301X         MOV   L,C           DONT TRANSFER MORE THAN LIMIT
056.347  174            3302X $REAL4  MOV   A,H
056.350  265            3303X         ORA   L
056.351  302 365 056    3304X         JNZ   $REAL6        SOME IN BUFFER
                        3305X
                        3306X *       BUFFER IS EMPTY. RE-FILL IT
                        3307X
056.354  315 072 060    3308X         CALL  $FFB          FILL FILE BUFFER
056.357  332 045 057    3309X         JC    $REAL8        ERROR CONDITION
056.362  303 321 056    3310X         JMP   $REAL3        COUNT THE DATA
                        3311X
                        3312X *       GOT THE DATA. MOVE IT FROM BUFFER TO TARGET
                        3313X *
                        3314X *       (BC) = LIMIT COUNT
                        3315X *       (DE) = FROM
                        3316X *       (HL) = COUNT
                        3317X *       ((SP)) = TO
                        3318X
056.365  171            3319X $REAL6  MOV   A,C
```

```
057.366  225          3320X          SUB     L
057.367  117          3321X          MOV     C,A
057.370  170          3322X          MOV     A,B
057.371  234          3323X          SBB     H
057.372  107          3324X          MOV     B,A             REMOVE BYTES ABOUT TO BE MOVED FROM REQUEST COUNT
057.373  305          3325X          PUSH    B
057.374  343          3326X          XTHL                    (HL) = REMAINING REQUEST COUNT
057.375  301          3327X          POP     B               (BC) = COUNT FOR THIS COPY
057.376  343          3328X          XTHL                    (HL) = TARGET ADDR, ((SP)) = REMAINING REQ. COUNT
057.377  032          3329X $REAL7   LDAX    D
057.000  023          3330X          INX     D
057.001  167          3331X          MOV     M,A
057.002  043          3332X          INX     H
057.003  247          3333X          ANA     A               SEE IF 00 BYTE
057.004  302 013 057   3334X          JNZ     $REL7.3         NOT 00
                      3335X
                      3336X *        IS 00 BYTE. IGNORE IT
                      3337X
057.007  343          3338X          XTHL
057.010  043          3339X          INX     H               ADD ONE TO UNREQUITED COUNT
057.011  343          3340X          XTHL
057.012  053          3341X          DCX     H               BACKSPACE OVER CHARACTER
057.013  013          3342X $REL7.3  DCX     B
057.014  376 012       3343X          CPI     NL
057.016  312 036 057   3344X          JE      $REL7.5         IS END OF LINE
057.021  170          3345X          MOV     A,B
057.022  261          3346X          ORA     C
057.023  302 377 056   3347X          JNZ     $REAL7          MORE TO GO
057.026  353          3348X          XCHG
057.027  042 160 060   3349X          SHLD    T.PTR           UPDATE POINTER
057.032  301          3350X          POP     B               (BC) = REMAINING COUNT
057.033  303 300 056   3351X          JMP     $REAL2          SEE IF MORE IN BUFFER
                      3352X
                      3353X *        END OF CODED LINE
                      3354X
057.036  353          3355X $REL7.5  XCHG
057.037  033          3356X          DCX     D               BACK OVER NL CHARACTER
057.040  042 160 060   3357X          SHLD    T.PTR           UPDATE POINTER
057.043  301          3358X          POP     B               (BC) = REMAINING COUNT
057.044  325          3359X          PUSH    D               SAVE TARGET LWA
                      3360X
                      3361X *        READ COMPLETE.
                      3362X *
                      3363X *        (PSW) = COMPLETION FLAGS
                      3364X
057.045  321          3365X $REAL8   POP     D               RESTORE TARGET ADDRESS
057.046  365          3366X          PUSH    PSW             SAVE RETURN CODE
057.047  257          3367X          XRA     A
057.050  022          3368X          STAX    D               FLAG END OF LINE
057.051  361          3369X          POP     PSW             RESTORE RESULT FLAGS
057.052  023          3370X          INX     D               POINT TO NEXT FREE
057.053  341          3371X $REAL9   POP     H
057.054  303 040 060   3372X          JMP     CTB             COPY TEMP POINTERS BACK TO BLOCK, EXIT
057.057               3373           XTEXT   FWRIL
```

```
                                3375X **      $FWRIL - WRITE LINE FROM FILE BUFFER.
                                3376X *
                                3377X *       $FWRIL IS CALLED TO WRITE A LINE TO A FILE BUFFER.
                                3378X *
                                3379X *       ENTRY   (DE) = FWA FOR BYTES
                                3380X *               (HL) = ADDRESS OF FILE BUFFER
                                3381X *       EXIT    TO *FERROR* IF ERROR
                                3382X *               TO CALLER IF OK
                                3383X *                (DE) = ADDRESS OF FIRST UNWRITTEN BYTE
                                3384X *       USES    A,F,B,C,D,E
                                3385X
                                3386X
057.057  315 066 057           3387X $FWRIL  CALL    $FWRIL.
057.062  320                   3388X         RNC                     RETURN IF OK
057.063  303 167 060           3389X         JMP     $FERROR         ERROR
                                3390X
                                3391X *       SCAN FOR END OF LINE
                                3392X
057.066  325                   3393X $FWRIL. PUSH    D               SAVE LINE POINTER
057.067  001 377 377           3394X         LXI     B,-1            (BC) = COUNT
057.072  032                   3395X $FWRIL1 LDAX    D
057.073  023                   3396X         INX     D
057.074  003                   3397X         INX     B
057.075  247                   3398X         ANA     A
057.076  302 072 057           3399X         JNZ     $FWRIL1         MORE TO GO
057.101  321                   3400X         POP     D
057.102  315 124 057           3401X         CALL    $FWRIB          WRITE BYTES
057.105  330                   3402X         RC                      ERROR
                                3403X
                                3404X *       WRITE 'NL' CHARACTER
                                3405X
057.106  023                   3406X         INX     D
057.107  325                   3407X         PUSH    D
057.110  001 001 000           3408X         LXI     B,1
057.113  021 123 057           3409X         LXI     D,$FWRILA
057.116  315 124 057           3410X         CALL    $FWRIB
057.121  321                   3411X         POP     D
057.122  311                   3412X         RET
                                3413X
057.123  012                   3414X $FWRILA DB      NL
057.124                         3415          XTEXT   FWRIB



                                3417X **      $FWRIB - WRITE BYTES FROM FILE BUFFER.
                                3418X *
                                3419X *       $FWRIB IS CALLED TO WRITE A NUMBER OF BYTES FROM A FILE BUFFER.
                                3420X *
                                3421X *       ENTRY   (BC) = BYTE COUNT
                                3422X *               (DE) = FWA FOR BYTES
                                3423X *               (HL) = ADDRESS OF FILE BUFFER
                                3424X *       EXIT    TO *FERROR* IF ERROR
                                3425X *               TO CALLER IF OK
                                3426X *                (DE) = ADDRESS OF FIRST UNWRITTEN BYTE
                                3427X *       USES    A,F,B,C,D,E
```

```
                        3428X
                        3429X
057.124 315 133 057     3430X $FWRIB  CALL    $FWRIB.
057.127 320             3431X         RNC                   RETURN IF OK
057.130 303 167 060     3432X         JMP     $FERROR       ERROR
                        3433X
                        3434X
057.133                 3435X $FWRIB. EQU     *
057.133 345             3436X         PUSH    H
057.134 315 012 060     3437X         CALL    CBT           COPY BUFFER POINTERS TO TEMP CELLS
                        3438X
                        3439X *       COPY DATA FROM USER AREA TO BUFFER
                        3440X
057.137 325             3441X $WRIB2  PUSH    D             SAVE AREA ADDRESS
057.140 072 155 060     3442X         LDA     T.FLG
057.143 346 004         3443X         ANI     FT.OW         SEE IF OPEN FOR WRITE
057.145 312 301 057     3444X         JZ      $WRIB8        FILE NOT OPEN FOR WRITE
057.150 170             3445X         MOV     A,B
057.151 261             3446X         ORA     C
057.152 312 301 057     3447X         JZ      $WRIB8        ALL DONE
                        3448X
                        3449X *       COMPUTE MIN( ROOM IN BUFFER, WRITE COUNT REQUESTED)
                        3450X
057.155 052 160 060     3451X $WRIB3  LHLD    T.PTR
057.160 353             3452X         XCHG                  (DE) = (FB.PTR) = ADDRESS OF ROOM
057.161 052 164 060     3453X         LHLD    T.LWA         (HL) = LIMIT ADDRESS
057.164 175             3454X         MOV     A,L
057.165 223             3455X         SUB     E
057.166 157             3456X         MOV     L,A
057.167 174             3457X         MOV     A,H
057.170 232             3458X         SBB     D
057.171 147             3459X         MOV     H,A           (HL) = BYTES OF ROOM IN BUFFER
057.172 171             3460X         MOV     A,C           COMPARE REQUESTED COUNT TO BUFFER ROOM
057.173 225             3461X         SUB     L
057.174 170             3462X         MOV     A,B
057.175 234             3463X         SBB     H
057.176 322 203 057     3464X         JNC     $WRIB4        MORE REQUESTED THEN ROOM
057.201 140             3465X         MOV     H,B
057.202 151             3466X         MOV     L,C           USE REQUESTED COUNT
057.203 174             3467X $WRIB4  MOV     A,H
057.204 265             3468X         ORA     L
057.205 302 245 057     3469X         JNZ     $WRIB6        SOME ROOM IN BUFFER
                        3470X
                        3471X *       BUFFER IS FULL, EMPTY IT
                        3472X
057.210 305             3473X         PUSH    B             SAVE COUNT
057.211 052 156 060     3474X         LHLD    T.FWA
057.214 042 160 060     3475X         SHLD    T.PTR         CLEAR REMOVAL POINTER
057.217 353             3476X         XCHG
057.220 052 164 060     3477X         LHLD    T.LWA
057.223 175             3478X         MOV     A,L
057.224 223             3479X         SUB     E
057.225 117             3480X         MOV     C,A
057.226 174             3481X         MOV     A,H
057.227 232             3482X         SBB     D
057.230 107             3483X         MOV     B,A           (BC) = DATA IN BUFFER
```

```
057.231  072 154 060  3484X          LDA     T.CHA
057.234  377 005      3485X          DB      SYSCALL,.WRITE  WRITE BUFFER
057.236  301          3486X          POP     B               (BC) = DESIRED COUNT
057.237  322 155 057  3487X          JNC     $WRIB3          GOT THE DATA
                      3488X
                      3489X *        ERROR ON WRITE.
                      3490X
057.242  303 301 057  3491X          JMP     $WRIB8          HAVE ERROR
                      3492X
                      3493X *        GOT THE DATA. MOVE IT FROM BUFFER TO TARGET
                      3494X *
                      3495X *        (BC) = REQUEST COUNT
                      3496X *        (DE) = TO
                      3497X *        (HL) = COUNT
                      3498X *        ((SP)) = FROM
                      3499X
057.245  171          3500X $WRIB6   MOV     A,C
057.246  225          3501X          SUB     L
057.247  117          3502X          MOV     C,A
057.250  170          3503X          MOV     A,B
057.251  234          3504X          SBB     H
057.252  107          3505X          MOV     B,A             REMOVE BYTES ABOUT TO BE MOVED FROM REQUEST COUNT
057.253  305          3506X          PUSH    B
057.254  343          3507X          XTHL                    (HL) = REMAINING REQUEST COUNT
057.255  301          3508X          POP     B               (BC) = COUNT FOR THIS COPY
057.256  343          3509X          XTHL                    (HL) = TARGET ADDR, ((SP)) = REMAINING REQ. COUNT
057.257  176          3510X $WRIB7   MOV     A,M
057.260  022          3511X          STAX    D
057.261  023          3512X          INX     D
057.262  043          3513X          INX     H
057.263  013          3514X          DCX     B
057.264  170          3515X          MOV     A,B
057.265  261          3516X          ORA     C
057.266  302 257 057  3517X          JNZ     $WRIB7          MORE TO GO
057.271  353          3518X          XCHG
057.272  042 160 060  3519X          SHLD    T.PTR           UPDATE POINTER
057.275  301          3520X          POP     B               (BC) = REMAINING COUNT
057.276  303 137 057  3521X          JMP     $WRIB2          SEE IF MORE IN BUFFER
                      3522X
                      3523X *        WRITE COMPLETE.
                      3524X *
                      3525X *        (PSW) = COMPLETION FLAGS
                      3526X
057.301  321          3527X $WRIB8   POP     D               RESTORE TARGET ADDRESS
057.302  341          3528X          POP     H
057.303  303 040 060  3529X          JMP     CTB             COPY TEMP POINTERS BACK TO BLOCK, EXIT
```

```
                        3531X **      $FWBRK  -  BREAKOUTPUT                                  /80.02.GC/
                        3532X *
                        3533X *       $FWBRK empties the specified buffer by filling it with NULLs
                        3534X *       and then writing it.  Note this is used to insure that block
                        3535X *       mode I/O is output if it is not really a serial device (eg.
                        3536X *       writing to AT: from *EDIT*.
                        3537X *
                        3538X *
                        3539X *       ENTRY:  HL      =  FILE BLOCK POINTER
                        3540X *
                        3541X *       EXIT:   HL      =  FILE BLOCK POINTER
                        3542X *               TO  $FERROR IF ERROR
                        3543X *
                        3544X *       USES:   PSW,BC,DE
                        3545X *
                        3546X
057.306  315 315 057    3547X $FWBRK  CALL    $FWBRK.
057.311  320            3548X         RNC                       NO ERROR
                        3549X
057.312  303 167 060    3550X         JMP     $FERROR
                        3551X
057.315  345            3552X $FWBRK. PUSH    H
057.316  315 012 060    3553X         CALL    CBT               COPY BUFFER TO TEMPORARY
057.321  315 331 057    3554X         CALL    $FWBRK1
057.324  341            3555X         POP     H
057.325  315 040 060    3556X         CALL    CTB               COPY TEMPORARY TO BUFFER
057.330  311            3557X         RET
                        3558X
057.331  052 164 060    3559X $FWBRK1 LHLD    T.LWA
057.334  353            3560X         XCHG                      DE = BUFFER LWA
057.335  052 160 060    3561X         LHLD    T.PTR             HL = BUFFER PTR
057.340  173            3562X         MOV     A,E
057.341  225            3563X         SUB     L
057.342  117            3564X         MOV     C,A
057.343  172            3565X         MOV     A,D
057.344  234            3566X         SBB     H
057.345  107            3567X         MOV     B,A               BC = DE - HL
057.346  261            3568X         ORA     C
057.347  310            3569X         RZ                        THE BUFFER IS ALREADY FLUSHED
                        3570X
                        3571X *       FILL THE BUFFER WITH NULLS
                        3572X
057.350  170            3573X FWBRK2  MOV     A,B
057.351  261            3574X         ORA     C
057.352  312 364 057    3575X         JZ      FWBRK3            NO MORE LEFT TO FILL
                        3576X
057.355  066 000        3577X         MVI     M,0
057.357  043            3578X         INX     H
057.360  013            3579X         DCX     B
057.361  303 350 057    3580X         JMP     FWBRK2
                        3581X
057.364  052 156 060    3582X FWBRK3  LHLD    T.FWA
057.367  042 160 060    3583X         SHLD    T.PTR
057.372  353            3584X         XCHG                      DE = BUFFER FWA
057.373  052 164 060    3585X         LHLD    T.LWA             HL = BUFFER LWA
057.376  175            3586X         MOV     A,L
```

```
057.377 223           3587X          SUB     E
060.000 117           3588X          MOV     C,A
060.001 174           3589X          MOV     A,H
060.002 232           3590X          SBB     D
060.003 107           3591X          MOV     B,A             BC = HL - DE  ( BC = COUNT )
060.004 072 154 060   3592X          LDA     T.CHA
060.007 377 005       3593X          DB      SYSCALL+.WRITE
060.011 311           3594X          RET
060.012               3595           XTEXT   FUTIL


                      3597X **       $FUTIL - UTILITY ROUTINES FOR FILE BLOCK ROUTINES.
                      3598X
                      3599X **       CBT - COPY BLOCK POINTERS TO TEMP CELLS.
                      3600X *
                      3601X *        ENTRY   (HL) = FILE BLOK FWA
                      3602X *        EXIT    NONE
                      3603X *        USES    A,F,H,L
                      3604X
060.012 325           3605X CBT      PUSH    D
060.013 305           3606X          PUSH    B               SAVE REGISTERS
000.000               3607X          ERRNZ   TLEN-10         ASSUME 10 BYTES TO MOVE
060.014 021 154 060   3608X          LXI     D,T.CHA         (DE) = TARGET FOR MOVE
060.017 006 005       3609X          MVI     B,10/2
060.021 176           3610X CBT1     MOV     A,M             COPY FILE BUFFER INTO WORK AREA
060.022 022           3611X          STAX    D
060.023 043           3612X          INX     H
060.024 023           3613X          INX     D
060.025 176           3614X          MOV     A,M
060.026 022           3615X          STAX    D
060.027 043           3616X          INX     H
060.030 023           3617X          INX     D
060.031 005           3618X          DCR     B
060.032 302 021 060   3619X          JNZ     CBT1            MORE TO GO
060.035 301           3620X          POP     B
060.036 321           3621X          POP     D               (DE) = DATA TARGET ADDRESS
060.037 311           3622X          RET
                      3623X
                      3624X
                      3625X **       CTB - COPY TEMP CELLS BACK TO FILE BLOCK.
                      3626X *
                      3627X *        ENTRY   (HL) = FILE BLOCK ADDRESS
                      3628X *        EXIT    NONE
                      3629X *        USES    NONE
                      3630X
060.040 365           3631X CTB      PUSH    PSW
060.041 325           3632X          PUSH    D
060.042 305           3633X          PUSH    B
060.043 345           3634X          PUSH    H               SAVE REGISTERS
060.044 006 004       3635X          MVI     B,8/2
060.046 021 154 060   3636X          LXI     D,T.CHA
060.051 032           3637X CTB1     LDAX    D
060.052 167           3638X          MOV     M,A
060.053 023           3639X          INX     D
```

```
060.054  043            3640X          INX     H
060.055  032            3641X          LDAX    D
060.056  167            3642X          MOV     M,A
060.057  023            3643X          INX     D
060.060  043            3644X          INX     H
060.061  005            3645X          DCR     B
060.062  302 051 060     3646X          JNZ     CTB1            RESTORE FILE BUFFER VALUES
060.065  341            3647X          POP     H
060.066  301            3648X          POP     B
060.067  321            3649X          POP     D
060.070  361            3650X          POP     PSW
060.071  311            3651X          RET


                        3653X **       *FFB - FILE FILE BUFFER.
                        3654X *
                        3655X *        *FFB FILLS THE FILE BUFFER BY READING FROM THE FILE.
                        3656X *
                        3657X *        ENTRY   NONE
                        3658X *        EXIT    'C' SET IF READ INCOMPLETE
                        3659X *                 (A) = ERROR CODE
                        3660X *                 'C' CLEAR IF READ COMPLETEE
                        3661X *                  DATA IN BUFFER
                        3662X *        USES    A,F,D,E,H,L
                        3663X
                        3664X
060.072  072 166 060     3665X *FFB     LDA     EOFFLG
060.075  037            3666X          RAR
060.076  330            3667X          RC                      EOF
                        3668X
                        3669X *        CAN READ MORE. DO SO
                        3670X
060.077  305            3671X          PUSH    B               SAVE COUNT
060.100  052 156 060     3672X          LHLD    T.FWA
060.103  042 160 060     3673X          SHLD    T.PTR           CLEAR REMOVAL POINTER
060.106  353            3674X          XCHG
060.107  052 164 060     3675X          LHLD    T.LWA
060.112  042 162 060     3676X          SHLD    T.LIM           SET DATA LIMIT
060.115  175            3677X          MOV     A,L
060.116  223            3678X          SUB     E
060.117  117            3679X          MOV     C,A
060.120  174            3680X          MOV     A,H
060.121  232            3681X          SBB     D
060.122  107            3682X          MOV     B,A             (BC) = ROOM IN BUFFER
060.123  072 154 060     3683X          LDA     T.CHA
060.126  377 004        3684X          DB      SYSCALL+.READ   READ BUFFER
060.130  120            3685X          MOV     D,B             (D) = SECTORS UNREAD
060.131  301            3686X          POP     B               (BC) = DESIRED COUNT
060.132  320            3687X          RNC                     GOT THE DATA
                        3688X
                        3689X *        ERROR ON READ. SEE IF EOF
                        3690X
060.133  027            3691X          RAL
060.134  062 166 060     3692X          STA     EOFFLG          SET EOF, WE HOPE
```

```
060.137  376 003       3693X          CPI     EC.EOF*2+1
060.141  037          3694X          RAR
060.142  300          3695X          RNE                     IS NOT EOF, RETURN NOW!
060.143  072 163 060  3696X          LDA     T.LIM+1
060.146  222          3697X          SUB     D
060.147  062 163 060  3698X          STA     T.LIM+1         SET AMOUNT OF DATA WE DID GET
060.152  247          3699X          ANA     A
060.153  311          3700X          RET                     EXIT WITH DATA
                      3701X
                      3702X
                      3703X **       TEMP CELLS TO HOLD FILE BLOCK POINTERS DURING I/O
                      3704X
000.000               3705X          ERRNZ   FB.CHA
060.154  000          3706X T.CHA    DB      0               CHANNEL NUMBER
000.000               3707X          ERRNZ   *-T.CHA-FB.FLG
060.155  000          3708X T.FLG    DB      0               FLAG BYTE
000.000               3709X          ERRNZ   *-T.CHA-FB.FWA
060.156  000 000      3710X T.FWA    DW      0
000.000               3711X          ERRNZ   *-T.CHA-FB.PTR
060.160  000 000      3712X T.PTR    DW      0
000.000               3713X          ERRNZ   *-T.CHA-FB.LIM
060.162  000 000      3714X T.LIM    DW      0
000.000               3715X          ERRNZ   *-T.CHA-FB.LWA
060.164  000 000      3716X T.LWA    DW      0
000.012               3717X TLEN     EQU     *-T.CHA         LENGTH OF TEMP CELLS
                      3718X
060.166  000          3719X EOFFLG   DB      0
060.167               3720           XTEXT   FERROR



                      3722X **       $FERROR - PROCESS FILE ERRORS.
                      3723X *
                      3724X *        $FERROR IS CALLED TO COMPLAIN ABOUT AN ERROR ENCOUNTERED
                      3725X *        WHEN PROCESSING FILES.
                      3726X *
                      3727X *        ENTRY   (A) = ERROR CODE
                      3728X *                (HL) = ADDRESS OF FILE NAME - FB.NAM
                      3729X *        EXIT    TO RESTART
                      3730X *        USES    ALL
                      3731X
                      3732X
060.167  365          3733X $FERROR  PUSH    PSW             SAVE CODE
060.170  315 136 031  3734X          CALL    $TYPTX
060.173  012 007 105  3735X          DB      NL,BELL,'ERROR ON FILE',' '+200Q
060.213  021 012 000  3736X          LXI     D,FB.NAM
060.216  031          3737X          DAD     D
                      3738X
                      3739X *        PRINT FILE NAME
                      3740X
060.217  176          3741X $FERR1   MOV     A,M
060.220  043          3742X          INX     H               ADVANCE MESSAGE
060.221  247          3743X          ANA     A
060.222  312 233 060  3744X          JZ      $FERR2
060.225  315 345 055  3745X          CALL    $WCHAR
```

```
060.230  303 217 060  3746X          JMP     $FERR1
                      3747X
                      3748X *        TYPE ERROR MESSAGE
                      3749X
060.233  315 136 031  3750X $FERR2   CALL    $TYPTX
060.236  040 055 240  3751X          DB      ' -',' '+2000Q
060.241  046 012      3752X          MVI     H,NL
060.243  361          3753X          POP     PSW             (A) = CODE
060.244  377 057      3754X          DB      SYSCALL,.ERROR
060.246  303 200 042  3755X          JMP     RESTART         EXIT
060.251               3756           XTEXT   TJMP


                      3758X **       $TJMP - TABLE JUMP.
                      3759X *
                      3760X *        USAGE
                      3761X *
                      3762X *        CALL    $TJMP             (A) = INDEX
                      3763X *        DW      ADDR1
                      3764X *        .         .
                      3765X *        .         .
                      3766X *        .         .
                      3767X *        DW      ADDRN
                      3768X *
                      3769X *        ENTRY   (A) = INDEX
                      3770X *        EXIT    TO PROCESSOR
                      3771X *                (A) = INDEX*2
                      3772X *        USES    NONE.
                      3773X
                      3774X
031.061               3775X $TJMP    EQU     31061A          IN H17 ROM, (A) = INDEX*2
                      3776X
031.062               3777X $TJMP.   EQU     31062A          IN H17 ROM
060.251               3778           XTEXT   TYPTX


                      3780X **       $TYPTX - TYPE TEXT.
                      3781X *
                      3782X *        $TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
                      3783X *
                      3784X *        IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED,
                      3785X *        A BYTE WITH THE 200Q BIT SET IS THE LAST BYTE IN THE MESSAGE.
                      3786X *
                      3787X *        ENTRY   (RET) = TEXT
                      3788X *        EXIT    TO (RET+LENGTH)
                      3789X *        USES    A,F
                      3790X
                      3791X
031.136               3792X $TYPTX   EQU     31136A          IN H17 ROM
                      3793X
031.144               3794X $TYPTX.  EQU     31144A          IN H17 ROM
```

```
                        3797  **      CMDTAB - COMMAND TABLE.
                        3798  *
                        3799
060.251                 3800  CMDTAB  EQU     *
060.251  000           3801          DB      0               DUMY FIRST COMMAND
060.252  120 122 111    3802          DB      'PRINT',0
060.260  104 105 114    3803          DB      'DELETE',0
060.267  105 104 111    3804          DB      'EDIT',0
060.274  122 105 120    3805          DB      'REPLACE',0
060.304  127 122 111    3806          DB      'WRITE',0
060.312                 3807  CMDTAB. EQU     *               THESE COMMANDS ALLOWED WITH NO TEXT
060.312  130 120 122    3808          DB      'XPRINT',0      IS DUMY COMMAND FOR 2ND GROUP, REAL COMMAND FOR 1ST
060.321  111 116 123    3809          DB      'INSERT',0
060.330  122 105 101    3810          DB      'READ',0
060.335  102 114 111    3811          DB      'BLITZ',0
060.343  106 114 125    3812          DB      'FLUSH',0
060.351  116 105 130    3813          DB      'NEXT',0
060.356  123 105 101    3814          DB      'SEARCH',0
060.365  116 105 127    3815          DB      'NEWIN',0
060.373  116 105 127    3816          DB      'NEWOUT',0
061.002  130 117 125    3817          DB      'XOUT',0
061.007  125 123 105    3818          DB      'USE',0
061.013  102 131 105    3819          DB      'BYE',0
061.017  000           3820          DB      0
```

```
061.020             3823  PATCH   DS      64
```

```
                        3827  **      LINE POINTERS INTO TEXT PAGE.
                        3828
061.120  000 000        3829  FILPTR  DW      0               ADDRESS OF 1ST LINE IN BUFFER
061,122  000 000        3830  LALPTR  DW      0               ADDRESS OF END OFLAST LINE IN BUFFER +1
061.124  000 000        3831  CRFPTR  DW      0               COMMAND RANGE 1ST LINE POINTER
061,126  000 000        3832  CRLPTR  DW      0               COMMAND RANGE LAST LINE POINTER
061.130  000 000        3833  WRKPTR  DW      0               COMMAND RANGE WORKING POINTER
061,132  000 000        3834  PCFPTR  DW      0               PREVIOUS COMMAND 'FIRST' POINTER
061.134  000 000        3835  PCLPTR  DW      0               PREVIOUS COMMAND 'LAST' POINTER
                        3836
061.136  000            3837  CCFLG   DB      0               <>0 IF CTL-C DISABLED
061,137  000            3838  CCPEND  DB      0               <>0 IF CTL-C HIT DURING DISABLED PERIOD
                        3839
061,140  000 000        3840  BUFMAX  DW      0               MAX ADDRESS FOR *BUFFER*


                        3842  **      CELLS AND POINTERS
                        3843
061,142  000 000        3844  LINPTR  DW      0               LINE POINTER
061.144  000            3845  PROCHA  DB      0               PROBATION CHARACTER
061,145  000            3846  SRCDIR  DB      0               SEARCH DIRECTON
061.146  000            3847  OPTS    DB      0               OPTION FLAGS
                        3848
                        3849  *       FILE BUFFERS
                        3850
061.147  123 131 060    3851  DEFALT  DB      'SY0',0,0,0     DEFAULT DEVICE AND EXTENSION
                        3852
061.155                 3853  INFB    DS      0               INPUT FILE BUFFER
061,155  001            3854          DB      1               CHANNEL NUMBER
061.156  000            3855          DB      0               FLAGS
061.157  076 063        3856          DW      INBUF
061.161  076 063        3857          DW      INBUF
061.163  076 063        3858          DW      INBUF
061.165  076 065        3859          DW      INBUFE
061,167                 3860          DS      FB.NAML         NAME
                        3861
061,210                 3862  OUTFB   DS      0               OUTPUT FILE BUFFER
061.210  000            3863          DB      0
061.211  000            3864          DB      0               FLAGS
061.212  076 065        3865          DW      OUTBUF
061.214  076 065        3866          DW      OUTBUF
061.216  076 065        3867          DW      OUTBUF
061.220  076 067        3868          DW      OUTBUFE
061.222                 3869          DS      FB.NAML         NAME
                        3870
061.243                 3871  XOUTFB  DS      0               XOUT FILE BUFFER
061.243  002            3872          DB      2
061.244  000            3873          DB      0               FLAGS
061.245  076 067        3874          DW      XOTBUF
061.247  076 067        3875          DW      XOTBUF
061.251  076 067        3876          DW      XOTBUF
061.253  076 070        3877          DW      XOTBUFE
061.255                 3878          DS      FB.NAML
```

```
                        3882  **      PRS - PERFORM PRESET PROCESSING.
                        3883  *
                        3884  *       THIS CODE IS ONLY USED UPON ENTRY, AND THEN IS OVERLAID BY BUFFERS.
                        3885  *
                        3886  *       IT 1) TYPES THE BANNER MESSAGE
                        3887  *             2) DETERMINES THE MEMORY SIZE
                        3888  *             3) PRESETS THE TEXT PAGE TO NULL
                        3889  *
                        3890  *       ENTRY   NONE
                        3891  *       EXIT    DATA STRUCTURE INITIALIZED
                        3892
                        3893
061.276                 3894  ENTRY   EQU     *
061.276  257            3895          XRA     A
061.277  062 076 070     3896          STA     BUFFER-1        SET DUMY END-OF-LINE FOR BUFFER
061.302  062 276 061     3897          STA     LINE-1          SETUP 00 BYTE REQUIRED BEFORE *LINE*
                        3898
                        3899  *       CHECK VERSIONS AND LOAD *HDOSOVLO.SYS*
                        3900
061.305  377 011        3901          DB      SYSCALL+.VERS
061.307  332 365 061     3902          JC      PRSERR1         PROBABLY NO .VERS SYSTEM CALL
061.312  376 026        3903          CPI     VERS
061.314  302 365 061     3904          JNZ     PRSERR1         NOT THE CORRECT VERSION
                        3905
                        3906  *       SETUP HIGH MEMORY
                        3907
061.317  315 333 053     3908          CALL    MAM             SET MAXIMUM MEMORY SIZE
                        3909
                        3910  *       SETUP CTL-C PROCESSING
                        3911
061.322  041 374 042     3912          LXI     H+INTRPT
061.325  076 003        3913          MVI     A+CTLC
061.327  377 041        3914          DB      SYSCALL+.CTLC
061.331  315 136 031     3915          CALL    $TYPTX
061.334  105 104 111     3916          DB      'EDIT Issue #103.05.00',ENL
061.342  303 200 042     3917          JMP     START           STARTUP
                        3918
061.365  076 050        3919  PRSERR1 MVI     A+EC+NCV        NOT THE CORRECT VERSION
                        3920
061.367  046 012        3921  ENTEXT  MVI     H+NL
061.371  377 057        3922          DB      SYSCALL+.ERROR  THERE WAS AN ERROR UPON ENTRY
061.373  257            3923          XRA     A
061.374  377 000        3924          DB      SYSCALL+.EXIT
                        3925
                        3926  **      BUFFERS OVERLAYING PRS
                        3927
061.376                 3928  MEML    EQU     *               END OF LOAD IMAGE
061.276                 3929          ORG     ENTRY
```

```
                        3931  **      STRING AND TEXT STORE AREAS
                        3932
      061.276           3933          DS      1               REQUIRED 0 BEFORE 'LINE'
                        3934
      061.277           3935  LINE    DS      120             LINE BUFFER
      061.277           3936  FNRA    EQU     LINE            $FNR WORK AREA
      062.067           3937  WRKSTR  DS      120             EXPANDED STRING WORK AREA
      062.257           3938  EDIA    DS      41              EDIT WORK AREA
      062.330           3939  EDIB    DS      41              EDIT WORK AREA
      063.001           3940  QUALS   DS      41              QUALIFIER STRING
      063.052           3941  NXTCHA  DS      1               NEXT COMMAND CHARACTER
      063.053           3942  PATCNT  DS      1               INDEX OF CURRENT PATTERN
      063.054           3943  CMDGRP  DS      1               ZERO IF RESTRICTED COMMAND GROUP
                        3944
      063.055           3945  $FOPWRK DS      FB.NAML         USED BY $FOPEX
                        3946
      063.076           3947  INBUF   DS      512
      065.076           3948  INBUFE  EQU     *
                        3949
      065.076           3950  OUTBUF  DS      512
      067.076           3951  OUTBUFE EQU     *
                        3952
      067.076           3953  XOTBUF  DS      256
      070.076           3954  XOTBUFE EQU     *


                        3956  **      TEXT BUFFER.
                        3957
      070.076           3958          DS      1               0 BYTE NEEDED FOR BACKWARDS SCAN OF 1ST LINE
      070.077           3959  BUFFER  DS      0


      070.077           3961          END
ASSEMBLY COMPLETE
 3961 STATEMENTS
    1 ERRORS DETECTED
11538 BYTES FREE
```

```
$CCO     055000    401     2535L
$CDEHL   030216    701     711     1685    1725    1783    2204    2969E
$CHL     030224   2692E
$CLL     054361   1026    1065    1189    1225    1240    2323    2489    2511L
$CRLF    056001   2254    2494    2575    2662    2981L
$DADA    030072   1066    2996E
$FCLO    056147   1310    1322    1357    1404    1452    1529    3173L
$FCLO.   056156   3173    3177L
$FCLO1   056227   3216L   3219
$FCLO2   056237   3212    3221L
$FCLO3   056244   3185    3202    3232L
$FCLO4   056252   3183    3238L
$FERR1   060217   3741L   3746
$FERR2   060233   3744    3750L
$FERROR  060167   3072    3076    3080    3175    3260    3389    3432    3550    3733L
$FFB     060072   3308    3665L
$FOPE1   056067   3102    3114L
$FOPE2   056122   3138    3142L
$FOPEA   056130   3142    3146E
$FOPEB   056140   3136    3156L
$FOPER   056007   1365    3070L
$FOPER.  056034   3070    3083L
$FOPEU   056025   3078L
$FOPEU.  056042   3078    3087L
$FOPEW   056016   1412    1460    3074L
$FOPEW.  056037   3074    3085L
$FOPWRK  063055   3945L
$FREAL   056254   1518    3257L
$FREAL.  056267   3257    3265E
$FWBRK   057306    979    3547L
$FWBRK.  057315   3547    3552L
$FWBRK1  057331   3554    3559L
$FWRIB   057124   3401    3410    3430L
$FWRIB.  057133   3430    3435E
$FWRIL   057057    989    1688    3387L
$FWRIL.  057066   3387    3393L
$FWRIL1  057072   3395L   3399
$FWRILA  057123   3409    3414L
$GNL     055171    402    2657L
$INCHA   055015   2010    2557L   2583    2592
$INCHAA  055150   2563    2567    2596    2600    2606L
$INDL    030234   2917E   3189    3192
$MCU     055205    821     864    1768    2638    2675L
$MLU     055151   1342    1389    1433    2633L   2766
$MLU1    055154   2636L   2641
$MOVE    030252    928    1196    1229    1235    1934    2343    2841    3020E
$MOVEL   055271   1359    1406    1454    2825L
$MUIO    030324   1865    3034E
$RCHAR   055337   1766    2557    2770    2895L   2896
$REAL2   056300   3274L   3351
$REAL3   056321   3286L   3310
$REAL4   056347   3299    3302L
$REAL6   056365   3304    3319L
$REAL7   056377   3329L   3347
$REAL8   057045   3279    3282    3309    3365L
$REAL9   057053   3371L
$REL7.3  057013   3334    3342L
$REL7.5  057036   3344    3355L
```

```
$RSTALL 031047   2205    2208    2539    2731E
$RTL    055233   1748    2764    2768E
$RTL.   055224   2764L
$RTL1   055234   2770L   2776
$RTL2   055266   2772    2793L
$SAVALL 031054   2173    2189    2535    2745E
$TBLS   055350   2938L   3140
$TBRA   031076   3054E
$TJMP   031061    439    3775E
$TJMP.  031062   3777E
$TYPC.  055334   2884L
$TYPCC  055314   2492    2855E   2864
$TYPCH  055330   2874L
$TYPTX  031136    403     469     490     578    1351    1369    1398    1445    1526    1592    1648    1705
                 1764    1812    2041    2588    3734    3750    3792E   3915
$TYPTX. 031144   3794E
$UDD    031157   1618    1631    1647    2621E
$WCHAR  055345    789    1767    2009    2565    2601    2603    2899L   3745
$WRIB2  057137   3441L   3521
$WRIB3  057155   3451L   3487
$WRIB4  057203   3464    3467L
$WRIB6  057245   3469    3500L
$WRIB7  057257   3510L   3517
$WRIB8  057301   3444    3447    3491    3527L
.CHFLG  000060    127L
.CLEAR  000055    124L
.CLEARA 000056    125L
.CLOSE  000046    117L   3237
.CLRCO  000007    101L    474
.CONSL  000006    100L   2538    2659
.CTLC   000041    112L   3914
.DECODE 000053    122L
.DELET  000050    119L
.DISMT  000061    128L
.DMNMS  000203    139L
.DMOUN  000201    137L
.ERROR  000057    126L   2045    3754    3922
.EXIT   000000     94L    509    1324    3139    3145    3924
.LINK   000040    111L
.LOADD  000062    129L
.LOADO  000010    102L
.MONMS  000202    138L
.MOUNT  000200    136L
.NAME   000054    123L
.OPENC  000045    116L
.OPENR  000042    113L   3156
.OPENU  000044    115L   3158
.OPENW  000043    114L   3157
.POSIT  000047    118L
.PRINT  000003     97L
.READ   000004     98L   3684
.RENAM  000051    120L
.RESET  000204    140L
.SCIN   000001     95L   2895
.SCOUT  000002     96L   2861    2884    2899    2982
.SETTP  000052    121L   2207
.SYSRES 000012    104L
.VERS   000011    103L   3901
```

```
.WRITE  000005    99L   3225  3485  3593
ABS.COD 000010   238L    380
ABS.ENT 000006   236L
ABS.ID  000000   232L
ABS.LDA 000002   234L
ABS.LEN 000004   235L
ACL     052045   943    971  1097  1257  1567  1611  1721L 2380
ATL     052072   898    922  1094  1745L
AYS     052110   505   1111  1353  1400  1447  1764L
BELL    000007    31E   491   579  1370  1706  1765  1813  2042  3735
BKSP    000010    33E  2558  2589  2589
BUFFER  070077   901   1500  1636  2197  3896  3959L
BUFMAX  061140  1505   1638  1804  2201  3840L
BYE     047025   464   1320L
C.STX   000002    35E
C.SYN   000026    34E
CBE     052153   399   1780L
CBO     052171   916   1798L 2383
CBO1    052215  1490   1812L
CBT     060012  3270   3437  3553  3605L
CBT1    060021  3610L  3619
CCFLG   061136   407    471  1846  1971  3837L
CCPEND  061137   408    480  1972  3838L
CDV     052243  1828L  1860  1863
CLL1    054365  2515L  2519
CMD2    044113   736L   777   812
CMD3    044116   732    740L
CMD4    044145   748    753L  804   811   824
CMD5    044212   762    781L  826
CMD6    044226   790L
CMD6.5  044232   792L   818
CMD7    044265   783    816L
CMDADR  042332   443L   453
CMDDSP  000006   436    453E
CMDGRP  063054   432    747  3943L
CMDTAB  060251   743   3800E
CMDTAB. 060312   749   3807E
CO.FLG  000001   217E  2537
CR      000015    27E
CRFPTR  061124   427    530   543   560   628   744  1036  1139  1478  1577  1675  1684
                1696   1699  2256  3831L
CRLPTR  061126   532    548   561   568  1017  1056  1140  1476  1560  1697  1722  1912
                1918   2113  2119  2258  3832L
CS.FLG  000200   218E
CSL.CHR 000001   195E   396
CSL.ECH 000200   193E   396
CSL.WRP 000002   194E
CTB     060040  3372   3529  3556  3631L
CTB1    060051  3637L  3646
CTLA    000001    42E
CTLB    000002    43E
CTLC    000003    44E  3913
CTLD    000004    45E  1769  2011  2771
CTLO    000017    46E
CTLP    000020    47E
CTLQ    000021    48E
CTLS    000023    49E
CTLZ    000032    50E
```

```
CTP.2SB 000010      203E
CTP.BKM 000002      204E
CTP.BKS 000200      200E
CTP.MLI 000040      201E
CTP.MLO 000020      202E
CTP.TAB 000001      205E
D.CON   040110      155L
D.RAM   040240      158L
D.VEC   040130      157L
DCC     052255      899     923   1025   1051   1120   1844L  2320
DCN     044072      424     725E
DCNA    044157      729     759E
DCO     044326      426     841L
DCO1    044333      863L    877
DCO2    044351      866     869L
DCQ     044310      425     841L
DCR     043066      423     528E
DCR1    043141      539     551L
DDN     052265      657     1181   1858L  2357
DDN1    052300      1863L   1873
DDN2    052332      1864    1877L
DEFALT  061147      1364    1411   1459   3851L
DEL0    045221      1008L   1701
DEL1    045230      1011L   1029
DEL2    045277      1013    1033E  1080
DEL2.5  045322      1038    1043L
DEL3    045331      1010    1051L
DEL4    045345      1056L   1074
DELA    046020      1053    1078   1082L
DELETE  045206      445     1001L
DF.CLR  000376      307E
DF.EMP  000377      306E
DIR.ALD 000025      322L
DIR.CLU 000015      315L
DIR.CRD 000023      321L
DIR.EXT 000010      310L
DIR.FGN 000020      318L
DIR.FLG 000016      316L
DIR.LGN 000021      319L
DIR.LSI 000022      320L
DIR.NAM 000000      309L
DIR.PRO 000013      311L
DIR.VER 000014      312L
DIRELEN 000027      324E    374
DIRIDL  000015      313E
DRE     043235      559     567    612E
DRE1    043302      621     627    631L
DRE3    043305      633L    660
DRE4    043310      637L    672
DRE5    043331      639     641    645L
DRE6    043353      658L    664
DRE7    043372      630     668L   675
DRE8    043375      653     669L
DTBK    052337      1027    1898L  2329
DTBK.   052350      1079    1899   1906L
DTBK1   052375      1914    1915   1919E
DTBK2   053020      1934L   2133
DTBK3   053026      1917    1923   1940L
```

```
DTBK4   053035   1931   1948L  2129
EC.CNA  000004   248L
EC.DDA  000027   267L
EC.DIF  000017   257L
EC.DIW  000035   273L
EC.DNI  000045   281L
EC.DNR  000046   282L
EC.DNS  000005   249L
EC.DSC  000047   283L
EC.EOF  000001   245L   3259   3693
EC.EOM  000002   246L
EC.FAO  000031   269L   3109
EC.FAP  000026   266L
EC.FL   000030   268L
EC.FNF  000014   256L
EC.FNO  000011   253L   3277
EC.FNR  000034   272L
EC.FOD  000043   279L
EC.FUC  000013   255L
EC.ICN  000016   258L
EC.IDN  000006   250L
EC.IFC  000020   260L
EC.IFN  000007   251L
EC.ILC  000003   247L
EC.ILO  000040   276L
EC.ILR  000012   254L
EC.ILV  000037   275L
EC.IOI  000052   286L
EC.IS   000032   270L
EC.NCV  000050   284L   3919
EC.NEM  000021   261L
EC.NOS  000051   285L
EC.NPM  000044   280L
EC.NRD  000010   252L
EC.NVM  000042   278L
EC.OTL  000053   287L
EC.RF   000022   262L
EC.UNA  000036   274L
EC.UND  000015   257L
EC.UUN  000033   271L
EC.VPM  000041   277L
EC.WF   000023   263L
EC.WP   000025   265L
EC.WPV  000024   264L
ECC     053044   409    915    1011   1968L  2344
EDIO    046167   1175   1179L
EDI1    042250   415L   2023
EDI1.5  042323   435    437L
EDI2    046175   1177   1182L
EDI3    046200   1186L  1258
EDI5    046335   1187   1201   1257L
EDI6    046343   1255   1259L
EDIA    062257   1159   1199   1340   1361   1387   1408   1431   1456   1546   1564   3938L
EDIB    062330   1167   1234   3939L
EDITC   046116   446    1153E
EDIX    042200   396L   437    477    1371   1707   1814
EDTO    042245   411L   498
ENC     053064   537    562    618    637    651    726    843    863    1173   1828   1989L  2057
```

2074 2291

```
IOC.CGN 000010   362L
IOC.CSI 000011   363L
IOC.DDA 000002   351L   358   372
IOC.DES 000016   369L
IOC.DEV 000020   370L
IOC.DIL 000021   372E
IOC.DIR 000023   374L
IOC.DRL 000010   366E
IOC.DTA 000014   368L
IOC.FLG 000004   353L   366
IOC.GRT 000005   360L
IOC.LGN 000012   364L
IOC.LNK 000000   350L
IOC.LSI 000013   365L
IOC.SPG 000007   361L
IOC.SQL 000003   358E
IOC.UNI 000022   371L
IOCCTD  000001   378E
IOCELEN 000052   376E
ITBK    053244   926   2097L  2336
ITBK.   053255  2100   2107L
ITBK1   053300  2115   2116   2120E
LALPTR  061122   544    622    699    907   1034   1040   1138   1471   1496   1514   1521   1555
                1621   1782   1800   1921   1924   2122   2125   2190   3830L
LF      000012    28E
LINE    061277   406    415    492    912   1001   1745   2574   2581   3897   3935L  3936
LINPTR  061142   416    727    772    794   1996   2013   2018   2580   2590   3844L
LQS     053322   671   2148L  2396
MAM     053333   400   2173L  3908
MEML    061376   384   3928E
MI.LDA  000072    59E
MI.NOP  000000    60E   1665
MI.RET  000311    61E
MIM     053350  1344   1391   1435   2189L
MIMO    053366  2193   2199L
MIM1    053372  2177   2201L
MLP     044020   662    673    692L
MLP1    044053   696    708L
NEWIN   047041   460   1332E
NEWIN1  047155  1350   1356L
NEWIN4  047204  1347   1369L  1394   1439
NEWO1   047352  1397   1403L
NEWOUT  047235   461   1379E
NEXT    050146   458   1468E
NEXT.   050151  1302   1470E  1575
NL      000012    39E     40    579   1180   1337   1352   1385   1399   1428   1446   1527   1593
               1650   1651   1706   1813   2042   2044   2252   2775   2981   3343   3414   3735   3752
               3921
NUL2    000000    30E
NULL    000200    29E
NXTCHA  063052   741    769    798   3941L
OPT.A   000001    54E    869   2221
OPT.B   000002    55E    869   2237
OPTS    061146   861   2220   2236   3847L
OUTBUF  065076  3865   3866   3867   3950L
OUTBUFE 067076  3868   3951E
OUTFB   061210  1309   1395   1403   1409   1669   1697   3862L
OVL.COD 000000   331L
```

```
OVL.ENS 000010  336E
OVL.ENT 000004  333L
OVL.FLB 000006  334L
OVL.SIZ 000002  332L
OVLO    000000  342L
OVL1    000001  343L
PATCH   061020  3823L
PATCNT  063053  434   753   3942L
PCFPTR  061132  529   1043  1142  1585  2257  3834L
PCLPTR  061134  531   1044  1143  1586  2259  3835L
PLA     054020  1588  1610  2220L 2345
PLB     054030  911   1024  1064  1093  1203  1609  2236L
PRI1    045114  939L  944
PRIA    045121  942E
PRINT   045111  444   938L
PROCHA  061144  411   776   787   791   2007  2015  3845L
PRSERR1 061365  3902  3904  3919L
PURGE   046052  456   1110L
PURGE.  046062  1118E 1785
QUALS   063001  669   841   1008  2388  3940L
QUOTE   000047  36E   629   652   844   2284  2292
RBN     054210  2357L
RCR     054041  890   938   958   1004  1090  1110  1182  1299  1343  1390  1434  1469
                1488  1551  1603  1664  2251L
RDS     046345  1160  1169  1271L 1341  1388  1432  1547
RDS1    046353  1275L 1282
READ    050203  455   1487E
READ.   050215  1474  1480  1489  1493L
READ0   050225  1496L 1522
READ1   050243  1499  1502L
READ2   050307  1495  1519  1526L
REFUSE  043031  490L  580   703   712   771   872   1003  1163  1276  1338  1386  1429
                1550  1861  1866  1870  2080  2253  2282  2360
REP1    046025  1091L 1099
REPLAC  046022  447   1090L
RESTART 042200  391E  507   2046  3755
ROMBOOT 030000  150E
RQS     054073  670   846   2274L
RQS1    054102  2281L 2287  2298
RQS2    054136  2293  2302L
RSL     054144  1096  1242  2320L
RSL1    054170  2325  2334L
RSL2    054175  2330  2337  2341E
RUBOUT  000177  32E   2560
S.CAADR 040333  224L
S.CCTAB 040335  225L
S.CONFL 040332  222L
S.CONTY 040327  209L  2585
S.CONWI 040331  215L
S.CSLMD 040326  198L  208   211   214   221   397   1747
S.CUSOR 040330  212L
S.DATC  040310  180L
S.DATE  040277  179L
S.GRT0  024000  146E
S.GRT1  025000  147E
S.GRT2  026000  148E
S.HIMEM 040316  182L
S.INT   040343  160L
```

```
S.OMAX  040324   188L
S.SOVR  041146   162L    164
S.SYSM  040320   184L   2174
S.TIME  040312   181L
S.USRM  040322   186L   2203
S.VAL   040277   159L    177
SAP     046070   902   1137L  1501
SEA0    050361  1555L   1579
SEA1    050377  1561L   1568
SEA2    051027  1558    1562  1572L
SEA3    051056  1566    1583L
SEA4    051074  1574    1592L
SEARCH  050337   459    1540E
SEL     054231   939     964  1012   1606   2384L
SEL.    054226  1091    1186  1561   2383L
SEL1    054222  2380L   2398
SEL2    054257  2391    2402L
SFS     054264  1200    1565  2150   2418L  2440
SFS1    054275  2427L   2433
SFS2    054316  2423    2429  2442L
SLB     054322   547     624   714   1041   1475   1559   2456L
SLB1    054323  2457L   2460
SNL     054333   493     697   756    914   1728   2473L  2476
SRCDIR  061145   614     646   694   3846L
STACK   042200   166E    419
STACKL  001032   164E
START   042200   390E   3917
SYDD    040130   156E
SYSCALL 000377    87E     474   509   1324   2045   2207   2538   2659   2861   2884   2895   2899
                 2982    3145  3225   3237   3485   3593   3684   3754   3901   3914   3922   3924
T.CHA   060154  3484    3592  3608   3636   3683  3706L  3707   3709   3711   3713   3715   3717
T.FLG   060155  3275    3442  3708L
T.FWA   060156  3474    3582  3672   3710L
T.LIM   060162  3288    3676  3696   3698   3714L
T.LWA   060164  3453    3477  3559   3585   3675  3716L
T.PTR   060160  3286    3349  3357   3451   3475   3519   3561   3583   3673  3712L
TAB     000011    37E   2075
TBL1    055357  2942L   2948
TBL2    055375  2940    2952L
TBL3    055377  2945    2956L
TLEN    000012  3607    3717E
TTX     054345   941    2489L
TTX.    054342  2224    2239  2488L
USE     051112   463    1602E
USE1    051120  1606L   1612
USE2    051143  1607    1616L
USE3    051212  1635    1637E
USEB    051246  1617    1650L
USEC    051264  1630    1651L
USED    051302  1646    1652L
USERFWA 042200   167E    380   383    384
VERS    000026    85E   3903
WRI..   051316  1666L
WRI1    051353  1684L   1690
WRI3    051374  1678    1694E
WRI4    052017   962    1671  1705L
WRIA    051374  1666    1695L
WRITE   051311   448    1663E
```

WRITE.	051314	1479	1665L											
WRKPTR	061130	428	633	642	663	668	693	891	920	929	1015	1052	1058	
		1072	1141	1188	1477	1563	1578	1584	1668	1682	1700	1724	1729	2149
		2322	2384	2402	2488	3833L								
WRKSTR	062067	1194	1239	3937L										
XOTBUF	067076	3874	3875	3876	3953L									
XOTBUFE	070076	3877	3954E											
XOUT	050001	462	1422E											
XOUT1	050117	1443	1451L											
XOUTFB	061243	475	476	960	978	988	1321	1441	1451	1457	3871L			
XPR1	045145	964L	972											
XPR2	045164	965	976E											
XPR4	045123	969	985E											
XPRA	045205	993L	994											
XPRAL	000001	994E												
XPRINT	045132	449	957E											

18154 BYTES FREE