```
                        NAM             DOS-V2.C
                    *
*** ILLEGAL 'CPM' CALL          IN LINE #    3 OF "submit.a"
                        CRO             I6809
                        OPT             NOG,PAG
      0000              LEN   120
      0000              SYM   8
      0000              SETDP 0
                    *
                    ********************************
                    *
                    * DRAGONDOS
                    * Version 2.C
                    * (c) COPYRIGHT 1983
                    *
                    * By The MICRO WORKS
                    *    R. ROGERS
                    *    R. LENTZ
                    *    A. PHELPS
                    *
                    * For DRAGON DATA LTD.
                    *
                    * last revised april 06 1984
                    *
                    * NOW TAKENOVER BY VIVAWAY LTD
                    * ON BEHALF OF DRAGON DATA LTD
                    * MODS BY A.J.POULTER
                    *
                    ********************************
    E 0000          ROM     EQU   0
                    *             0=ROM CODE
                    *             1=RAM CODE
    E 0000          DRAGON  EQU   0
                    *             0=DRAGON CODE
                    *             1=COLOR CODE
                    ********************************
      0000          SKIP1   MACRO
                            FCB             $0021 BRN ..
                            ENDM

      0000          SKIP2   MACRO
                            FCB             $008C CMPX #$....
                            ENDM

      0000          EQ      MACRO
                            ORCC            #%00000100
                            ENDM

      0000          LSRD    MACRO
                            LSRA
                            RORB
                            ENDM

      0000          ASLD    MACRO
                            ASLB
                            ROLA
                            ENDM

                    ********************************
                            LIB             LOWRAM.A
                            NAM             LOWRAM
```

```
                        ********************************
                        *
                        * FILE MANAGEMENT SYSTEM
                        * FOR THE DRAGON
                        *
                        * THE MICRO WORKS
                        * LAST REVISED:
                        * SEPTEMBER 14, 1983
                        *
                        * ASSUMPTIONS:  5" DRIVES,
                        * SINGLE OR DOUBLE SIDED,
                        * 40 OR 80 TRACKS.
                        *
                        * THESE ROUTINES ARE DESIGNED TO
                        * BE CALLED BY THE BASIC INTERPRETER,
                        * BUT ARE NOT PART OF IT; THEY
                        * MAY BE EASILY CALLED BY ANY
                        * MACHINE LANGUAGE PROGRAM.
                        *
                        * RESPONSIBILITIES OF THE BASIC
                        * INTERPRETER INCLUDE:
                        *  RESERVE 1.5K AT $0600
                        *  CALL "INITVR" ON RESET
                        *  CALL "CLOSAL" AT THE "OK"
                        *
                        ********************************
                        *  DIRECTORY TRACK FORMAT:
                        *    SECTOR 1
                        *      BYTES 0-179 FREE INFO (BIT=1 =) SECTOR FREE)
                        *      BYTE 252  # TRACKS
                        *      BYTE 253  # SECTORS SIDE
                        *      BYTE 254  COMP # TRACKS
                        *      BYTE 255 COMP # SECTORS SIDE
                        * RE DESIGNED TO
                        * BE CALLED BY THE BASIC INTERPRETER,
                        * BUT ARE NOT PART OF IT; THEY
                        * MAY BE EASILY CALLED BY ANY
                        * MACHINE LANGUAGE PRDGRAM.
                        *
                        * RESPONSIBILITIES OF THE BASIC
                        * INTERPRETER INCLUDE:
                        *    RESERVE 1.5K AT $0600
                        *    CALL INITVR ON RESET
                        *   1 EXTENT
                        * +15 #2 EXTENT
                        * +18 #3 EXTENT
                        * +21 #4 EXTENT
                        * +24 BYTES IN LAST SECT
                        *  (NOTE: DLSCT=0 =) 256)
                        *  (NOTE: IF FLAGS:5 =1,
                        *  DLSCT IS POINTER TO EXTRA
                        *  EXTENTS)
* 0000                          ORG    0
D 0000                  FLAGS   RMB    1
```

```
D 0001                  DNAME   RMB   8
D 0009                  DXTNSN  RMB   3
D 000C                  DXTXT   RMB   3*4
D 0018                  DLSCT   RMB   1
E 0019                  DIRSIZ  EQU   *          SIZE OF ENTRY
E 000A                  DIRPS   EQU   256/DIRSIZ
                        *
                        *  FORMAT FOR EXTRA EXTENT BLOCK
                        *
                        *    FLAGS RMB 1 BIT 0 = 0
                        *                BIT 5 = 0 OR 1
                        *    ..... RMB 3*7 EXTENTS
                        *    ..... RMB 2 UNUSED
                        *    DLSCT RMB 1 SAME AS BEFORE
                        *
                        *  FORMAT OF FLAGS BYTE:
                        *      BIT 0 = SECTOR UNIT
                        *              0 = FILE NAME
                        *              1 = EXTENSION
                        *      BIT 1 = PROTECTED
                        *              0 = UNPROTECTED
                        *              1 = PROTECTED
                        *      BIT 2 =
                        *      BIT 3 = END OF DIR
                        *              0 = NOT END
                        *              1 = END
                        *      BIT 4 =
                        *      BIT 5 = MORE XTNSNS
                        *              0 = NO MORE EXTS
                        *              1 = MORE EXTENTS
                        *      BIT 6 =
                        *      BIT 7 = NOT VALID
                        *              0 = VALID
                        *              1 = NOT VALID
                        *
                        *  NOTE: ONLY BITS 1 & 7 ARE
                        *  MEANINGFUL WHEN IN A FIB.
                        *
                        *  FORMAT OF EACH EXTENT ENTRY:
                        *      LSN - 2 BYTES,
                        *      LENGTH - 1 BYTE.
                        *  UNUSED ENTRIES HAVE LENGTH=0.
                        *  THE DIRECTORY AND THE ALTERNATE
                        *  DIRECTORYS TRACKS ARE NOT
                        *  COUNTED INTO THE LSN
                        *
                        ********************************
```

```
                        ********************************
                        *
                        *  FILE INFO BLOCK FORMAT
                        *
                        *  (FIRST BYTE = 0 IF NO ENTRY)
                        * +0  FILE NAME
                        * +8  EXTENSION
                        * +11 DRIVE (1-4)
                        * +12 NEXT READ BYTE
                        * +15 DIRECTORY FLAGS
                        * +16 LENGTH OF FILE
                        * +19 FSN OF EXTENT 1
                        * +21 LSN OF EXTENT 1
                        * +23 SECTORS IN EXTENT 1
                        * +24 FSN OF EXTENT 2
                        * +26 LSN OF EXTENT 2
                        * +28 SECTORS IN EXTENT 2
                        * +29 NUMBER OF DIR ENTRY
                        * +30 DIR NUMBER OF LAST ENTRY
* 0000                           ORG    0
D 0000                  FNAME    RMB    8
D 0008                  XTNSN    RMB    3
D 000B                  FDRIV    RMB    1
D 000C                  RDBYTE   RMB    3
D 000F                  FFLAGS   RMB    1
D 0010                  FILLEN   RMB    3
D 0013                  XTNT1    RMB    2
D 0015                  XTLSN1   RMB    2
D 0017                  XTLEN1   RMB    1
D 0018                  XTNT2    RMB    2
D 001A                  XTLSN2   RMB    2
D 001C                  XTLEN2   RMB    1
D 001D                  DIRN     RMB    1
D 001E                  LSDIRB   RMB    1
E 001F                  SIZFIB   EQU    *
                        *
                        ********************************
                        *
                        *  PAGE BUFFER INFO
                        *
* 0000                           ORG    0
D 0000                  PAGLSN   RMB    2          LSN IN PAGE BUFFER
D 0002                  PAGVLD   RMB    1          VALID (SEE BELOW)
D 0003                  PAGDRV   RMB    1          DRIVE
D 0004                  PAGLRU   RMB    1          LRU 1-4 (1 OLDEST)
D 0005                  PAGADR   RMB    2          ADDRESS OF BUFFER
E 0007                  LENPBF   EQU    *
                        * PAGVLD:
                        *       0=INVALID (NOT IN USE)
                        *       1=VALID, CLEAN
                        *       $FF=VALID, DIRTY
                        *       $FE=VALID, DIRTY, GRACEEXPIRED
                        *
                        ********************************
```

```
                        *
     * 0000                       ORG    0
     D 0000             DIRLSN    RMB    2
     D 0002             FRELEN    RMB    1
     D 0003             FREXT     RMB    2
     D 0005             NUMFIL    RMB    1          # OF OPEN FILES
     E 0006             DRVISZ    EQU    *
                        ********************************
     E 0014             DIRTRK    EQU    20         DIRECTORY TRACK
     E 0010             ALTTRK    EQU    16         ALTERNATE DIR
     E 000A             NOFIB     EQU    10         # OF FIBS

                        ********************************
```

```
*******************************
*
*  THESE ERROR CODES ARE RETURNED IN
*  THE B REGISTER FROM THE VARIOUS
*  ROUTINES.  ALL ROUTINES WHICH
*  CAN RETURN AN ERROR CODE END
*  WITH A "TST B", SO THAT THE
*  CALL MAY BE IMMEDIATELY FOLLOWED
*  BY A "BNE ERROR".
*
*  THE CODES MARKED WITH A "*"
*  ARE NON-FATAL AND ARE FOR INFORMATION
*  ONLY;  ALL OTHERS INDICATE
*  THAT THE REQUESTED OPERATION
*  WAS NOT PERFORMED AND OTHER
*  RETURNED INFORMATION IS NOT
*  NECESSARILY VALID.
*
*ERRNEF EQU 47 NON-EXISTANT FILE
* (FROM "SEARCH)
*ERRILN EQU 19 ILLEGAL FILE NAME
* (FROM "PARSE)
*ERRMNY EQU 45 TOO MANY FILE OPEN
* (FROM "SEARCH)
*ERRFNF EQU 27 FILE NOT FOUND
* (FROM DELETE, LENFIL, ETC.)
*ERREOF EQU 49 READ PAST EOF
* (FROM READ)
*ERRDKF EQU 23 DISK FULL
* (FROM EXTEND)
*ERRALL EQU 29 FILE EXISTS
* (FROM RCREAT)
*ERRDRF EQU 53 DIRECTORY FULL
* (FROM FEMPDR)
*ERRWEF EQU 54 WRITE PAST EOF
* (FROM WRITE)
*ERRIVD EQU 55 INVALID DIRECTORY
* (FROM DKINF)
*ERRPRT EQU 25 FILE PROTECTED
* (FROM WRITE, DELETE)
*
*ERRWPT EQU 31 DISK WRT PROTECTED
*ERRTMO EQU 13 I/O ERROR TIMEOUT
*ERRIOE EQU 37 I/O ERROR (CRC)
*ERRSNF EQU 35 SECT NOT FOUND
*           (FROM THE I/O CALL)
*******************************
```

```
                        ********************************
                        *                              *
                        * DRAGONDOS  version 1.B       *
                        *                              *
                        ********************************
                        *
* 00EA                          ORG    $00EA
D 00EA                  DD.CMD  RMB    1           COMMAND
D 00EB                  DD.UNT  RMB    1           UNIT # (1-4)
D 00EC                  DD.TRK  RMB    1           TRACK
D 00ED                  DD.SEC  RMB    1           SECTOR
D 00EE                  DD.BUF  RMB    2           BUFFER ADR
D 00F0                  DD.STA  RMB    1           STATUS
D 00F1                  FIBNO   RMB    1           CURRENT FIB #
D 00F2                  FWBYTE  RMB    1           NUMBER OF BYTES IN BUFFER
D 00F3                  FWFOR   RMB    1           NUMBER OF BYTES TO READ/WRITE
D 00F4                  FWFLAG  RMB    1           DO CARE ABOUT LENGTH, $00=NO
D 00F5                  RWFLAG  RMB    1           $00=READ,
                        *              $01=WRITE,
                        *              $FF=VERIFYING
D 00F6                  LOCK    RMB    1           LOCK OUT IRQ
D 00F7                  SYSDSK  RMB    1           SET WHEN FORMATING SYSTEM DISK

                        * DISK COMMAND BLOCK
* 0600                          ORG    $600
E 0600                  DBLOCK  EQU    *

                        *            SUPER KLUDGE !!?
D 0600                  THRASH  RMB    1
D 0601                  TEMPS   RMB    2           TEMP STORAGE FORS REGISTER
D 0603                  TEMP.C  RMB    1           TEMP FOR FREAD
D 0604                  TEMP.L  RMB    1           TEMP FOR FREAD

D 0605                  DD.TMR  RMB    1           MTR TIMER
D 0606                  DD.OPT  RMB    1           DISK OPTION
D 0607                  CTLSAV  RMB    1           LATCH TEMP
D 0608                  VERFLG  RMB    1           VERIFY  0=ON
D 0609                  ERRMSK  RMB    1           2797 ERROR MASK
D 060A                  DEFDRV  RMB    1           DEFAULT DRIVE NUMBER

D 060B                  FWBUF   RMB    2           BUFFER POINTER FOR FWRITE

D 060D                  BGNBUF  RMB    2
D 060F                  ENDBUF  RMB    2
D 0611                  RLFLAG  RMB    1           RUN/LOAD FLAG
D 0612                  RDTYPE  RMB    1           FREAD/FLREAD FLAG

                        * VARIABLES FOR 'AUTO'
D 0613                  AUTFLG  RMB    1           AUTO LINE NUMBERING FLAG
E 060D                  AUTLIN  EQU    BGNBUF      AUTO LINE NUMBERING CURRENT LINE NUMBER
E 060F                  AUTINC  EQU    ENDBUF      AUTO LINE NUMBERING INCREMENT COUNTER

                        * VARIABLES FOR 'ON ERROR'
D 0614                  ONERRF  RMB    1           FLAG FOR ERROR TRAPING TURNED ON
```

```
D 0615                  ONERRL  RMB   2          LINE NUMBER TO EXECUTE ON ERROR
D 0617                  ERLINE  RMB   2          LINE NUMBER THE ERROR OCCURED IN
D 0619                  ERRNUM  RMB   1          THE ERROR THAT OCCURED
D 061A                  ERRBEG  RMB   2          THE POINTER TO THE BEGINNING OF THE STATEMENT THAT ERRORED.

                        ********************************

D 061C                  DRVINF  RMB   4*DRVISZ
D 0634                  PBFINF  RMB   4*LENPBF   BUFFER INFO
D 0650                  TNAME   RMB   11         TEMP FILE NAME
D 065B                  TDRV    RMB   1          DRIVE FOR TNAME
D 065C                  SRCLSN  RMB   2          COUNTER FOR LSN INDIRECTORY SEARCH
D 065E                  TBFADR  RMB   2          SAVE BUFFER ADDRESS
D 0660                  TEMP1   RMB   1          EXTENT FOUND
D 0661                  RLENG   RMB   2          LENGTH (FROM Y)
D 0663                  WBYTE   RMB   1          WHICH BYTE (FROMB)
D 0664                  WRBYTE  RMB   3          CURRENT WRITE ADDRESS FOR FWRITE
D 0667                  PAGPLC  RMB   2          PAGE BUFFER ADDRESS
D 0669                  RSECT   RMB   2          CURRENT SECTOR
D 066B                  TOTAL   RMB   2          TOTAL SECTORS FOUND
D 066D                  TARGET  RMB   2          FSN LOOKING FOR
D 066F                  SCTEXT  RMB   2          CURRENT SECTOR
D 0671                  BUFAD   RMB   2
D 0673                  NMBYTE  RMB   2
D 0675                  FLSCT   RMB   2
D 0677                  FLBYT   RMB   1
D 0678                  SAVFIB  RMB   2
D 067A                  SAVEXT  RMB   3
D 067D                  TFIBN   RMB   1
D 067E                  TEMP2   RMB   1
D 067F                  DIRBFP  RMB   2
D 0681                  TEMP3   RMB   1
D 0682                  LASDIR  RMB   1
D 0683                  NEWUSR  RMB   20
E 0697                  ENDVAR  SET   *

                        ********************************
```

```
                        ********************************
                        *
                        *    DRIVE DISCRIPTOR TABLE
                        *
                        ********************************
 E 0696                 LUVALD  EQU    *-1
 D 0697                         RMB    4           TABLE VALID FLAG
 E 069A                 LUTRAK  EQU    *-1
 D 069B                         RMB    4           CURRENT TRACK
 E 069E                 LUSTPR  EQU    *-1
 D 069F                         RMB    4           STEP RATES
 E 06A2                 LUMTRK  EQU    *-1
 D 06A3                         RMB    4
 E 06A6                 LUMSEC  EQU    *-1
 D 06A7                         RMB    4           MAX SECTORS/TRACK

 D 06AB                 DTYDIR  RMB    18          DIR SECTOR DIRTY
                        ********************************
                        *
                        *      FILE CONTROL BLOCKS (10)
                        *
                        ********************************
                        *
                        * +0  FILE NAME
                        * +9  FILE EXTENSION
                        * +12 DRIVE NUMBER
                        * +13 NEXT READ BYTE
                        * +16 DIR FLAGS
                        * +17 LENGTH OF FILE
                        * +20 FSN OF EXTENT 1
                        * +22 LSN OF EXTENT 1
                        * +24 SECTORS IN EXTENT 1
                        * +25 FSN OF EXTENT 2
                        * +27 LSN OF EXTENT 2
                        * +29 SECTORS IN EXTENT 2
                        * +30 LSN OF DIR ENTRY
                        * +31 POSITION IN DIRSEC
                        *
                        ********************************
 E 06BD                 FIBS    EQU    *
 D 06BD                         RMB    SIZFIB*NOFIB
 E 07F3                 ENDFBS  EQU    *

                        ********************************
                        *
                        *      DISK I/O BUFFERS (4)
                        *
                        ********************************
 * 0800                         ORG    $0800
 D 0800                 BUF.01  RMB    $400
 E 0800                 FM.BUF  EQU    BUF.01      DATA TABLE FOR TRACK WRITE

                        ********************************
                        *
```

```
                        *      BASIC GRAPHICS PAGES
                        *      AND PROGRAM STORAGE
                        *
                        * ))) NOTE :
                        *      DOS DOES NOT USE ANY
                        *      RAM ABOVE THIS LOCATION
                        *
                        ********************************


                        ********************************
                        *      CONDITIONAL EQUATES
                        ********************************

E 90A1                  VDCRLF  EQU   $90A1
E 90F5                  VDSPAC  EQU   $90F5
E 90F8                  VDQUES  EQU   $90F8

E 8331                  GETSTK  EQU   $8331
E 841F                  RUNC    EQU   $841F
E 842B                  RUNCC   EQU   $842B
E 831C                  BLTU    EQU   $831C
E 83FF                  FINLIN  EQU   $83FF
E 89AC                  SYNCHK  EQU   $89AC
E 9165                  NORMFC  EQU   $9165
E BAED                  SNDON   EQU   $BAED
E BE12                  DOSND   EQU   $BE12
E 85EE                  DEXPTR  EQU   $85EE
E 8371                  READY   EQU   $8371

E 9DD9                  FILINE  EQU   $9DD9

E 849F                  NEWSTT  EQU   $849F
E BA77                  GRNCLS  EQU   $BA77      CLEAR THE SREEN
E 8434                  STKINI  EQU   $8434
E BDDC                  CASOFF  EQU   $BDDC
E BAC3                  SNDENB  EQU   $BAC3
E 84DA                  BASRTS  EQU   $84DA
E 85E7                  GOTO    EQU   $85E7
E 957A                  PRTLIN  EQU   $957A
E B54A                  CHROUT  EQU   $B54A
E B505                  KBCHAR  EQU   $B505
E 869A                  LINGET  EQU   $869A
E 83ED                  CHEAD   EQU   $83ED
E 8417                  SCRTCH  EQU   $8417      PERFORM A NEW COMMAND
E B44F                  BASRST  EQU   $B44F      RESTART VECTOR FOR BASIC
E 90E5                  STROUT  EQU   $90E5      OUTPUT A STRING OF TEXT
E 8371                  IDLE    EQU   $8371      BASIC IDLELOOP
E 9D3D                  BASIRQ  EQU   $9D3D      BASIC IRQ HANDLER
E 851B                  ISCNTC  EQU   $851B      CHECK CTL C DEPRESSED?

E 8882                  TMERR   EQU   $8882
E B851                  DNERR   EQU   $B851
E 8342                  OMERR   EQU   $8342
E 89B4                  SNERR   EQU   $89B4
```

```
E 8B8D                  FCERR   EQU   $8B8D
E B848                  FMERR   EQU   $B848

E 00E5                  ATTOK   EQU   $00E5
E 0088                  ONTOK   EQU   $0088
E 00BC                  TOTOK   EQU   $00BC
E 00C2                  OFFTOK  EQU   $00C2
E 0080                  FORTOK  EQU   $0080
E 0081                  GOTOK   EQU   $0081

E 9C76                  ERRDIR  EQU   $9C76
E B7CC                  BLKMOV  EQU   $B7CC      BLOCK MOVEFROM [X] TO [U] FOR [B]
E 84ED                  DOJUMP  EQU   $84ED      JUMP TO ADDRESS IN TABLE
E 00CE                  LOWTOK  EQU   $00CE      LOWEST DISK COMMAND TOKEN
E 8E51                  GETBYT  EQU   $8E51      GET BYTE FROM BASIC LINE
E 8E7E                  COMBYT  EQU   $8E7E      GET COMMA AND THEN A BYTE, ERROR IF NO COMMA
E 89AA                  CHKCOM  EQU   $89AA      LOOK FOR A COMMA, IF NOT FOUND ERROR
E 8A94                  PTRGET  EQU   $8A94      FIND POINTERTO VARIABLE
E 8877                  CHKSTR  EQU   $8877      CHECK VARIABLE JUST FOUND FOR STRING
E 8887                  FRMEVL  EQU   $8887
E 8C52                  STR$    EQU   $8C52      MAKE A STRING FOR USE
E 8D9A                  FRESTR  EQU   $8D9A
E 8D9F                  FRETMP  EQU   $8D9F
E 8E83                  GETWRD  EQU   $8E83      GET 16 BIT INTEGER
E 8C37                  GIVABF  EQU   $8C37      SEND BACK A 16 BIT INTEGER
E 0044                  LOWFUN  EQU   $0044      2*LOWEST DISK FUNCTION TOKEN
E 8874                  FUNRET  EQU   $8874      FUNCTION RETURN - CHECK TYPE MISMATCH
E 013C                  FERAML  EQU   $013C      FUNCTION EXPANSION RAM LOCATION
E 0137                  CERAML  EQU   $0137      COMMAND EXPANSION RAM LOCATION
E B5D3                  GOBACK  EQU   $B5D3
E 8CD7                  GARBA2  EQU   $8CD7
E 877A                  READAT  EQU   $877A
E 879A                  READRT  EQU   $879A
E A66B                  BITAB   EQU   $A66B
E 8CB3                  GETSPA  EQU   $8CB3
E B4B2                  BTITLE  EQU   $B4B2
E 835E                  INER1   EQU   $835E
E 8358                  INER2   EQU   $8358
E 9587                  UNPAC   EQU   $9587
E 903D                  PRINT   EQU   $903D
E 8514                  RESTOR  EQU   $8514
E 00B0                  USRPTR  EQU   $00B0
E B7EE                  CLOSRT  EQU   $B7EE
E AA87                  SETGRA  EQU   $AA87
E 8DEA                  ASC2    EQU   $8DEA
E B7D4                  GETCON  EQU   $B7D4
E B835                  OPNDEV  EQU   $B835
E 8424                  CLEARC  EQU   $8424
E 85E7                  LUKALL  EQU   $85E7
E 82A9                  BERRLS  EQU   $82A9
                        ********************************
                        *
                        * UNCONDITIONAL EQUATES
                        *
                        ********************************
```

```
E 0000                  BRKFLG  EQU     $0000
E 0003                  BCOUNT  EQU     $0003
E 0006                  STRFLG  EQU     $0006       $00=NUMERIC,$FF=STRING

E 0019                  TXTTAB  EQU     $0019       POINTS TO THE BEGINNING OF BASIC PROGRAM
E 001B                  VARTAB  EQU     $001B       POINTS TO THE BEGINNING OF BASIC VARIABLES
E 001D                  ARRTAB  EQU     $001D       POINTS TO THE BEGINNING OF BASIC ARRAYS
E 001D                  ARYTAB  EQU     ARRTAB
E 001F                  ARRTOP  EQU     $001F       POINTS TO THE TOP(LAST) OF THE ARRAYS
E 001F                  STREND  EQU     ARRTOP
E 0021                  STKTOP  EQU     $0021       TOP OF THE STACK

E 0023                  FRETOP  EQU     $0023
E 0027                  MEMSIZ  EQU     $0027
E 002B                  LINNUM  EQU     $002B       LINE NUMBER STORAGE FOR BASIC GOTO/GOSUB
E 002F                  TXTBEG  EQU     $002F       POINTS TO THE BEGINNING OF THE CURRENT COMMAND

E 0039                  VARPNT  EQU     $0039
E 0041                  HIGHDS  EQU     $0041
E 0043                  HIGHTR  EQU     $0043
E 0045                  LOWDS   EQU     $0045
E 0047                  LOWTR   EQU     $0047
E 0052                  FAC     EQU     $0052       FLOATING POINT ACCUMULATOR
E 004F                  OSLOC   EQU     $4F         SCRATCH FOR BOOTS 'OS'
E 0063                  FAC2    EQU     $0063       ROUNDING FAC
E 0068                  CURLIN  EQU     $0068       CURRENT LINENUMBER FOR BASIC
E 006F                  DEVNUM  EQU     $006F       CURRENT DEVICE NUMBER
E 0085                  VALTYP  EQU     $0085       VARIABLE TYPE
E 008A                  DBLZER  EQU     $008A       TWO BYTE ZERO MAINTAINED FOR BASIC
E 008C                  SNDFRQ  EQU     $008C       SOUND FREQ.
E 009D                  EXECAD  EQU     $009D       ADDRESS FOR THE 'EXEC' COMMAND
E 009F                  GETCHR  EQU     $009F       GETS THE NEXT CHARACTER OF A BASIC LINE
E 00A5                  CHRGOT  EQU     $00A5       GETS THE CURRENT CHARACTER
E 00A6                  TXTPTR  EQU     $00A6       POINTS AT THE CURRENT CHARACTER IN A COMMAND
E 00B7                  SCTOP   EQU     $00B7       POINTS AT THE TOP OF THE VIDEO SCREEN
E 00BA                  SCBASE  EQU     $00BA       POINTS AT THE FIRST GRAPHICS PAGE
E 02DA                  BUFLNM  EQU     $02DA       KEYBOARD BUFFER LINE NUMBER
E 02DD                  LINBUF  EQU     $02DD       LINE INPUT BUFFER
E 02DD                  BUF     EQU     $02DD
E 03DA                  FACBUF  EQU     $03DA       BUFFER FOR UNPACKING A NUMBER
E 2600                  BOOTBF  EQU     $2600       OS9 BOOT BUFFER
E E000                  EILOC   EQU     $E000       EXPANSION INTERFACE ROM LOCATION
E F9FF                  MAXLIN  EQU     63999       MAXIMUM LINENUMBER FOR A BASIC PROGRAM
                        ********************************
                        * FORMAT VARIABLES
* 00F2                          ORG     FWBYTE
D 00F2                  FM.TRK  RMB     1           MAXIMUM TRACK ONFORMAT
D 00F3                  FM.SID  RMB     1           NUMBER OF SIDES TO FORMAT
D 00F4                  FM.SDE  RMB     1           SIDE BEING FORMATED

                        ********************************
                        *       VECTORS AND OPCODES
E 0071                  RSTFLG  EQU     $0071       RESTART FLAG
```

```
E 0072                  RSTHCK  EQU     $0072       RESTART VECTOR
E 0109                  NMIVEC  EQU     $0109       NMI VECTOR
E 010C                  IRQVEC  EQU     $010C       IRQ VECTOR
E 0106                  SWIVEC  EQU     $0106       SWI VECTOR
E 015E                  HOOKS   EQU     $015E       BASIC HOOK TABLE

                        ********************************
                        *       STATUS AND COMMAND BITS
E 0004                  TR00    EQU     4           TRACK 00 BIT
E 0001                  BUSY    EQU     1           BUSY STATUS BIT
E 0004                  VFY     EQU     4           VERIFY BIT

                        ********************************
                        *       2797 CONTROLLER
E FF2F                  WD.CSR  EQU     $FF2F       CMD REG
E FF2E                  WD.TRK  EQU     $FF2E       TRACK REG
E FF2D                  WD.SEC  EQU     $FF2D       SECTOR REG
E FF2C                  WD.DAT  EQU     $FF2C       DATA REG
E FF24                  A.DSKSLT EQU    $FF24       DISK SELECT AND MOTOR CONTROL
E FF25                  NMICNT  EQU     $FF25       NMI ENABLE (CA2) 0=DISABLE

                        ********************************
                        *       LATCH CONTROL BITS
E 0000                  MOTRON  EQU     #$10        motor on bit
E 0000                  MOTOFF  EQU     #$EF        motor off
E 0040                  ENPRE   EQU     $40         PRECOMP ENABLE
E 00BF                  NOPRE   EQU     $BF         PRECOMP DISABLE

                        ********************************
                        *       2797  COMMAND BYTES
E 0000                  RESCMD  EQU     $00         RESTORE
E 0010                  SEKCMD  EQU     $10         SEEK
E 0088                  RSCMD   EQU     $88         READ SECTOR
E 00C0                  RACMD   EQU     $C0         READ ADDRESS
E 00A8                  WSCMD   EQU     $A8         WRITE SECTOR
E 00F4                  WTCMD   EQU     $F4         WRITE TRACK
E 00D0                  ABTCMD  EQU     $D0         ABORT

                        ********************************
                        *       2797  ERROR MASKS
E 0019                  TY1ERR  EQU     $19         TYPE 1
E 003F                  RSERR   EQU     $3F         READ
E 003F                  VERERR  EQU     RSERR       VERIFY
E 00DF                  WVERR   EQU     $DF         WRITE W/VERIFY
E 005F                  WSERR   EQU     $5F         WRITE SECTOR
E 0047                  WTERR   EQU     $47         WRITE TRACK
E 0050                  DORERR  EQU     $50         DORK
E 00FC                  CMDERR  EQU     $FC         COMMAND
E 00FE                  TIMERR  EQU     $FE         TIME-OUT
E 00FD                  UNTERR  EQU     $FD         UNIT NUMBER

                        ********************************
                        *       DISK I/O ROUTINE COMMANDS
E 0000                  DIO.RE  EQU     0           RESTORE
E 0001                  DIO.SE  EQU     1           SEEK
```

```
E 0002                  DIO.RS  EQU   2           READ SECTOR
E 0003                  DIO.WV  EQU   3           WRITE WITH VERIFY
E 0004                  DIO.WS  EQU   4           WRITE SECTOR
E 0005                  DIO.WT  EQU   5           WRITE TRACK
E 0006                  DIO.RA  EQU   6           READ ADDRESS
E 0007                  DIO.VE  EQU   7           VERIFY SECTOR
E 0008                  DIO.DK  EQU   8           WRITE DEFECTIVE SECTOR

                        ********************************
                        *      INTERRUPT CONTROL
E 0020                  NMIOK   EQU   $20         NMI ENABLE
E 0035                  IRQOK   EQU   $35         IRQ ENABLE
E 0037                  FIRQOK  EQU   $37         FIRQ ENABLE

                        ********************************
                        *      PIA ADDRESSES
E FF21                  PB2CRA  EQU   $FF21       PIA2 CONTROL REGISTER A
E FF26                  PB2DAT  EQU   $FF26       PIA2 DATA REGISTER (2797 DRQ's)
E FF27                  PB2CRB  EQU   $FF27       PIA2 CONTROL REGISTER B
E FF1D                  PIABEG  EQU   $FF1D
E FF25                  PIAEND  EQU   $FF25

                        ********************************
                        *      CONSTANTS
E 00D2                  TIMOUT  EQU   210         DISK TIME-OUT  VALUE

                        ********************************
                                END
                                LIB               DISKINIT.A
                                NAM               DISKINIT
```

```
                        ********************************
* C000                          ORG    $C000
E C000                  PROBEG  EQU    *
C C000 44 4B                    FCC    'DK'

  C002   20      26 C02A        BRA    INIT
                        ********************************
                        *
                        *  JUMP TABLE:
                        *  ROUTINES AVAILABLE TO THE
                        *  BASIC INTERPRETER AND TO THE
                        *  ASSEMBLY-LANGUAGE PROGRAMMER.
                        *
C C004 C169                     FDB    DISK
C C006 00EA                     FDB    DD.CMD
C C008 C83E                     FDB    PARSE       SCAN AND VERIFY FILE NAME
C C00A C8FF                     FDB    SEARCH      LOCATE OR MAKE FIB
C C00C CF91                     FDB    CREATE      CREATE NEW FILE, BACKUP OLD
C C00E CEF6                     FDB    LENFIL      REPORT LENGTH OF FILE
C C010 CF38                     FDB    CLOSAL      CLOSE ALL FILES
C C012 CF4E                     FDB    CLOSE1      CLOSE ONE FILE
C C014 CA14                     FDB    READ        READ FROM FILE
C C016 CBD6                     FDB    WRITE       WRITE TO FILE
C C018 D1C9                     FDB    GETFRE      GET FREE SPACE ON DRIVE
C C01A D033                     FDB    DELETE      DELETE FILE
C C01C D119                     FDB    PROTEC      SET/CLEAR FILE PROTECT
C C01E D14B                     FDB    RNAME       RENAME FILE
C C020 D250                     FDB    GETDIR      GET DIRECTORY ENTRY
C C022 D2C3                     FDB    READSB      READ SECTORTO BUFFER
C C024 C7B4                     FDB    BACKDR      CLEAN UP BUFFERS AND COPY DIRECTORY
C C026 D386                     FDB    RLSN        READ SECTOR TO USER BUFFER
C C028 D376                     FDB    WLSN        WRITE SECTOR FROM USER BUFFER

                        ********************************
```

```
                        ********************************
                        *
                        *  POWER UP INITIALIZATION
                        *
                        ********************************
                        *
                        *      CLEAR DISK RAM
                        *
                        ********************************
C02A  8E   0600         INIT    LDX   #DBLOCK
C02D  1F     12                 TFR   X,Y
C02F  6F 80             INIT1   CLR   ,X+
C031  31 3F                     LEAY  -1,Y
C033  26     FA C02F            BNE   INIT1
C035  1F     10                 TFR   X,D
C037  1F     89                 TFR   A,B
C039  CB     18                 ADDB  #$18
C03B  D7     19                 STB   TXTTAB
C03D  BD   AA87                 JSR   SETGRA
C040  96     BA                 LDA   ($BA
C042  8B     06                 ADDA  #$06
C044  97     B7                 STA   ($B7

                        ********************************
                        *
                        *      INIT LOWRAM
                        *
                        ********************************
C046  8E   DE34                 LDX   #INTABL     Get from address
C049  E6 80             INIT2   LDB   ,X+         Get number of bytes to move
C04B  27     07 C054            BEQ   INIT3       Skip if zero
C04D  EE 81                     LDU   ,X++        Get move to address
C04F  BD   B7CC                 JSR   BLKMOV      Do the move
C052  20     F5 C049            BRA   INIT2       Loop till done
                        *
                        *
C054  C6     00         INIT3   LDB   #$00
C056  F7   0607                 STB   CTLSAV      NMIS ENABLED
C059  73   0608                 COM   VERFLG      VERIFY=ON

C05C  8E   0683                 LDX   #NEWUSR
C05F  9F     B0                 STX   USRPTR
C061  CE   B88D                 LDU   #FCERR
C064  C6     0A                 LDB   #10
C066  EF 81             INIT4   STU   ,X++
C068  5A                        DECB
C069  26     FB C066            BNE   INIT4
                        * SET DEFAULT DRIVE TO 1
C06B  7C   060A                 INC   DEFDRV

                        ********************************
                        *
                        * INIT FILE MANGEMENT VARS
                        *
```

```
                        ********************************
C06E   8D      48 C0B8          BSR    INITVR

                        ********************************
                        *
                        *      SET UP BASIC HOOKS
                        *
                        ********************************
C070   8E    015E               LDX    #HOOKS
C073 108E    DEAB               LDY    #HKTABL
C077   CC    137E               LDD    #$137E
C07A   E7 80           INIT5    STB    ,X+
C07C   EE A1                    LDU    ,Y++
C07E   EF 81                    STU    ,X++
C080   4A                       DECA
C081   26      F7 C07A          BNE    INIT5

                        ********************************
                        *
                        * LOOK FOR THE EXPANSION ROM
                        *
                        ********************************
C083   8E    4549               LDX    #$4549
C086   BC    E000               CMPX   EILOC
C089 1027    1F75 E002          LBEQ   EILOC+2

                        ********************************
                        *
                        *      SET UP RESTART VECTOR
                        *
                        ********************************
C08D   8E    C75C               LDX    #DK.RST
C090   9F      72               STX    <RSTHCK
C092   86      55               LDA    #$55
C094   97      71               STA    <RSTFLG

                        ********************************
                        *
                        *      DISPLAY THE BANNER
                        *
                        ********************************
C096   1C      AF               ANDCC  #$AF        Enable IRQ and FIRQ
C098   8E    B4B2               LDX    #BTITLE
C09B   BD    90E5               JSR    STROUT
C09E   0C      EB               INC    DD.UNT
C0A0   8E    8000               LDX    #$8000      Delay count
C0A3   4F                       CLRA
C0A4   3D              INIT6    MUL                Delay loop
C0A5   30 1F                    LEAX   -1,X
C0A7   26      FB C0A4          BNE    INIT6       Loop till count = zero
C0A9   BD    C165               JSR    GO.RST
C0AC   BD    BA77               JSR    GRNCLS
C0AF   8E    DFD3               LDX    #TITLE-1
C0B2   BD    90E5               JSR    STROUT
C0B5   7E    8371               JMP    IDLE
```

****

```
                        ********************************
                        *
                        *  INITIALIZE VARIABLES
                        *
                        *  CALLED ON POWER-UP OR RESET
                        *
                        ********************************
                        * MAKE THE FDC SHUT-UP !
C0B8  86      D0        INITVR  LDA     #ABTCMD
C0BA  B7    FF2F              STA     WD.CSR

                        *  INITIALIZE DRIVE INFO TABLES
C0BD  8E    0696              LDX     #LUVALD
C0C0  6F 84                   CLR     ,X
C0C2  6F 01                   CLR     1,X
C0C4  6F 02                   CLR     2,X
C0C6  6F 03                   CLR     3,X

                        *  INITIALIZE # FILES OPEN PER DRIVE
C0C8  8E    0621              LDX     #DRVINF+NUMFIL
C0CB  6F 84                   CLR     0,X
C0CD  6F 06                   CLR     DRVISZ*1,X
C0CF  6F 0C                   CLR     DRVISZ*2,X
C0D1  6F 88   12              CLR     DRVISZ*3,X

                        *  INITIALIZE VARIABLES:
C0D4  0F      F6              CLR     LOCK            UPDATE OK
                        *  INITIALIZE DIR TRACK BACKUP
C0D6  8E    06AB              LDX     #DTYDIR
C0D9  6F 80           INIT7   CLR     ,X+
C0DB  8C    07F3              CMPX    #ENDFBS
C0DE  26      F9 C0D9         BNE     INIT7
                        *  INITIALIZE PAGE BUFFER INFO
C0E0  C6      04              LDB     #4
C0E2  8E    0634              LDX     #PBFINF
C0E5  CE    0800              LDU     #BUF.01
C0E8  6F 02           INIT8   CLR     PAGVLD,X
C0EA  E7 04                   STB     PAGLRU,X        ORDER OF LRU
C0EC  EF 05                   STU     PAGADR,X
C0EE  33 C9 0100              LEAU    256,U
C0F2  30 07                   LEAX    LENPBF,X
C0F4  5A                      DECB
C0F5  26      F1 C0E8         BNE     INIT8
C0F7  39                      RTS

                        ********************************
                              END
                              LIB             DISKROM.A
                              NAM             DISKROM.A
```

```
                        ********************************
                        *
                        *      GENERAL DISK CALLS
                        *
                        ********************************
                        * READ ADDRESS
COF8  C6      06        GO.RDA  LDB    #DIO.RA
COFA                            SKIP2

                        ********************************
                        * WRITE TRACK
COFB  C6      05        GO.WRT  LDB    #DIO.WT
COFD                            SKIP2

                        ********************************
                        * WRITE SECTOR
COFE  C6      04        GO.WRS  LDB    #DIO.WS
C100                            SKIP2

                        ********************************
                        * WRITE SECTOR WITH VERIFY
C101  C6      03        GO.WRV  LDB    #DIO.WV
C103                            SKIP2

                        ********************************
                        * READ SECTOR
C104  C6      02        GO.RDS  LDB    #DIO.RS

                        ********************************
C106  34      02        EDISK   PSHS   A           SAVE REGISTERS
C108  32 7E             ED.THR  LEAS   -2,S
C10A  7F    0600                LDA    THRASH
C10D  86      03                LDA    #3          NUMBER OF RETRYS
C10F  ED E4                     STD    0,S
C111  8D      4F C162 RETRY2    BSR    GO.SEK      SEEK TRACK
C113  25      09 C11E           BCS    DO.TRY
C115  E6 61                     LDB    1,S         GET CALLERS CMD
C117  D7      EA                STB    (DD.CMD
C119  BD    C169                JSR    DISK
C11C  24      3C C15A           BCC    ED.RTS      !NO ERROR!

                        *
                        * SHOULD WE RETRY THIS ERROR ?
                        *  ERR.NR, NOT READY
                        *  ERR.WP, WRITE PROTECTED
                        *
C11E  C1      84        DO.TRY  CMPB   #ERR.WP
C120  27      27 C149           BEQ    ED.ERR

C122  6A E4                     DEC    ,S
C124  27      15 C13B           BEQ    ED.ERP
C126  E6 E4                     LDB    ,S
C128  54                        LSRB
C129  24      08 C133           BCC    ED.TKO
```

```
                        ********************************
                        * STEP-OUT / STEP-IN
C12B  0C      EC                INC    (DD.TRK
C12D  8D      33 C162           BSR    GO.SEK
C12F  0A      EC                DEC    (DD.TRK
C131  20      DE C111           BRA    RETRY2

                        ********************************
                        * RESTORE / RESEEK
C133  96      EC        ED.TKO  LDA    (DD.TRK
C135  8D      2E C165           BSR    GO.RST
C137  97      EC                STA    (DD.TRK
C139  20      D6 C111           BRA    RETRY2

                        ********************************
C13B  C1      80        ED.ERP  CMPB   #ERR.NR
C13D  26      18 C157           BNE    INIT9
C13F  7D    0600                TST    THRASH
C142  26      13 C157           BNE    INIT9
C144  73    0600                COM    THRASH
C147  20      0A C153           BRA    INIT10
C149  96      EC        ED.ERR  LDA    DD.TRK
C14B  81      14                CMPA   #DIRTRK
C14D  26      08 C157           BNE    INIT9
C14F  86      10                LDA    #ALTTRK
C151  97      EC                STA    DD.TRK
C153  E6 61           INIT10    LDB    1,S          GET USER CMMD
C155  20      B6 C10D           BRA    ED.THR
C157  1A      01        INIT9   SEC
C159                            SKIP1
C15A  5F              ED.RTS    CLRB
C15B  32 62                     LEAS   2,S
C15D  35      82                PULS   A,PC
                        ********************************
```

```
                        ********************************
                        * VERIFY SECTOR
C15F  C6      07        GO.VER  LDB     #DIO.VE
C161                            SKIP2

                        ********************************
                        * SEEK A TRACK
C162  C6      01        GO.SEK  LDB     #DIO.SE
C164                            SKIP2

                        ********************************
                        * RESTORE HEAD
C165  C6      00        GO.RST  LDB     #DIO.RE
C167  D7      EA                STB     <DD.CMD
```

```
                        ********************************
                        *                              *
                        *      DISK I/O                *
                        *                              *
                        * This is the main disk I/O    *
                        * entry point. All registers   *
                        * are preserved except         *
                        * condition codes and B on the *
                        * return,                      *
                        *                              *
                        * Call :                       *
                        *  DD.UNT = Drive number       *
                        *  DD.SEC = Sector number      *
                        *  DD.TRK = Track number       *
                        *  DD.BUF = Buffer pointer     *
                        *  DD.CMD = Function to execute*
                        *                              *
                        *     $00: RESTORE             *
                        *     $01: SEEK                *
                        *     $02: READ SECTOR         *
                        *     $03: WRITE SECTOR        *
                        *          - OPTIONAL VERIFY   *
                        *     $04: WRITE SECTOR        *
                        *          - NO VERIFY         *
                        *     $05: WRITE TRACK         *
                        *     $06: READ ADDRESS        *
                        *     $07: VERIFY              *
                        *                              *
                        * Exit conditions :            *
                        *                              *
                        * DD.STA = WD2797 status       *
                        *          register or zero if *
                        *          no error occured.   *
                        * ACCB   = Error code for basic*
                        *          or zero if no error *
                        *          occured.            *
                        * Carry flag = 0 if no error   *
                        *              1 if an error   *
                        *                              *
                        ********************************

C169  34     7B      DISK    PSHS   CC,A,DP,X,Y,U SAVE THE WORLD
C16B  1A     50              ORCC   #$50          Disable IRQ and FIRQ
C16D  66 E4                  ROR    ,S            remove carry
C16F  0F     F0              CLR    (DD.STA       no error yet
C171  96     EA              LDA    (DD.CMD       get command
C173  81     07              CMPA   #(DTBEND-DSKTBL)/2-1 valid command
C175  23     02 C179         BLS    CMDOK         cmmd ok.
C177  20     08 C181         BRA    BUMCMD        hop out invalid command

C179  BD   C287      CMDOK   JSR    STUNIT        SET UNIT #, FIRE UP MOTOR
C17C  24     0E C18C         BCC    GETSER        GET SERIOUS

C17E  C6     FD              LDB    #UNTERR       Bad unit number
```

```
C180                            SKIP2
C181  C6     FC       BUMCMD    LDB    #CMDERR     Bad command
C183  1A     01                SEC
C185  69 E4                    ROL    ,S
C187  D7     F0                STB    <DD.STA
C189  16    00BE C24A          LBRA   DGONE

                        ********************************
                        * Now that we know that the
                        * call is for real, we'll setup
                        * the hardware to have a go at
                        * it.
                        ********************************
                        * Stack the pia control
                        * registers and disable misc
                        * IRQs at the PIA, since SYNC
                        * can see any of them and will
                        * wander off if it does

C18C  8E    FF01       GETSER    LDX    #$FF01      Point at the first PIA
C18F  A6 84            STAKKR    LDA    ,X          Get the register
C191  34     02                  PSHS   A           Save it
C193  84     FC                  ANDA   #$FC        Disable IRQs
C195  A7 81                      STA    ,X++        Put it back
C197  A6 84                      LDA    ,X          Get the register
C199  34     02                  PSHS   A           Save it
C19B  84     FC                  ANDA   #$FC        Disable IRQs
C19D  A7 84                      STA    ,X          Put it back
C19F  30 88  1E                  LEAX   $1E,X       Next PIA
C1A2  8C    FF23                CMPX   #PB2CRB-4   Is that all
C1A5  25     E8 C18F            BLO    STAKKR      No, not yet

C1A7  B6    FF27               LDA    NMICNT+2    get B control register
C1AA  8A     03                ORA    #$3         enable CB1 ie DRQ from FDC interrupts
C1AC  B7    FF27               STA    NMICNT+2

                        * We will automatically
                        * retry 3 times

C1AF  86     03                LDA    #3
C1B1  34     02                PSHS   A

                        ********************************
                        * Which side are we to
                        * deal with?
                        * If DD.SEC is bigger than 18
                        * it's side 2, since there are
                        * only 18 sectors per track.
                        *

C1B3  96     ED                LDA    <DD.SEC     Sector #
C1B5  81     12                CMPA   #18         side 2?
C1B7  23     0A C1C3           BLS    ONESID      No, side 1

                        * Adjust sector # to be in
```

```
                        * range and set side bit
                        *
C1B9   80      12               SUBA   #18
C1BB   C6      02               LDB    #2
C1BD   FA    0606               ORB    DD.OPT
C1C0   F7    0606               STB    DD.OPT
C1C3   B7    FF2D      ONESID   STA    WD.SEC

                        * Setup is done so here
                        * goes nothing
                        *
                        * SET DIRECT PAGE TO $FF
                        *
C1C6 109E      8A      TRY.IT   LDY    DBLZER      Make zero
C1C9   9E      EE               LDX    (DD.BUF     Buffer
C1CB   D6      EA               LDB    (DD.CMD     What to do
C1CD   58                       ASLB
C1CE   B1    FF2D               CMPA   WD.SEC      HAS IT TAKEN THE SECTOR NUMBER
C1D1   26      F0 C1C3          BNE    ONESID      LOOP TILL IT HAS
C1D3   86      FF               LDA    #-1
C1D5   1F      8B               TFR    A,DP
C1D7   CE    DDEA               LDU    #DSKTBL
C1DA   AD D5                    JSR    [B,U]       GO DO IT

                        *       WE CAN RETURN TO HERE THROUGH 2 PATHS
                        *       1: TYPE 1 ERROR TIMEOUT - SEC, A=HDWERR CODE
                        *       2: NMI - CLC, A=2797 STATUS

C1DC   97      F0               STA    (DD.STA     HDWERR CODEOR RAW STATUS
C1DE   24      02 C1E2          BCC    RETRYS
C1E0   20      2E C210          BRA    DEXIT       ON HDWERR, FORGET IT

                        *       WE GOT SOME RESPONSE FROM THE 2797; SEE IF
                        *       WE NEED TO DO ANY RETRIES

C1E2   B4    0609      RETRYS   ANDA   ERRMSK
C1E5   97      F0               STA    (DD.STA     MASKED STATUS
C1E7   27      0E C1F7          BEQ    WRTVRF      IF OK, SEE ABOUT VERIFYING

                        *
                        *       WE DON'T WANT TO DO RETRIES ON A BAD VERIFY.
                        *       5 REVOLUTIONS IS ENOUGH.
                        *

C1E9   96      EA               LDA    (DD.CMD
C1EB   81      07               CMPA   #DIO.VE
C1ED   27      04 C1F3          BEQ    NORTRY

C1EF   6A E4                    DEC    ,S          ELSE RETRY IT
C1F1   26      D3 C1C6          BNE    TRY.IT      UNTIL TIREDOUT
C1F3   1A      01       NORTRY  SEC                SET THE POOPED FLAG
C1F5   20      19 C210          BRA    DEXIT       AND SPLIT

                        *       IF WE NEED TO VERIFY AFTER A WRITE, NOW'S THE TIME.
```

```
C1F7  7D   0609     WRTVRF  TST   ERRMSK
C1FA  2A     14 C210        BPL   DEXIT
C1FC  96     EC             LDA   (DD.TRK
C1FE  81     14             CMPA  #DIRTRK
C200  27     0B C20D        BEQ   VANYWY
C202  81     10             CMPA  #ALTTRK
C204  27     07 C20D        BEQ   VANYWY
C206  7D   0608     VRFWRT  TST   VERFLG     ARE VERIFY AND WRITE SECTOR
C209  1C     FE             CLC
C20B  2A     03 C210        BPL   DEXIT      IF NOT, WE'RE DONE;

                    *      ELSE DO THE VERIFY RECURSIVELY AND LEAVE WITH
                    *      WHATEVER DISK STATUS ANDCONDITION CODES
                    *      THE VERIFY OPERATION GENERATED.

C20D  17   FF4F C15F VANYWY  LBSR  GO.VER
C210  32 61          DEXIT   LEAS  1,S        REMOVE COUNTER
C212  69 64                  ROL   4,S

                    * UPDATE CURRENT DRIVE'S
                    * TRACK POINTER
C214  8E   069A              LDX   #LUTRAK
C217  D6     EB              LDB   (DD.UNT
C219  3A                     ABX
C21A  B6   FF2E              LDA   )WD.TRK
C21D  91     EC              CMPA  (DD.TRK    DOES SEEK ERROR EXIST
C21F  27     11 C232         BEQ   NSEEKR
C221  F6   0609              LDB   ERRMSK
C224  C1     19              CMPB  #TY1ERR
C226  26     0A C232         BNE   NSEEKR
C228  C6     FF              LDB   #$FF
C22A  D7     F0              STB   (DD.STA
C22C  66 64                  ROR   4,S
C22E  1A     01              SEC
C230  69 64                  ROL   4,S
C232  A7 84          NSEEKR  STA   0,X

                    * PUT THE PIA CONTROL REGS BACK
                    * WHERE THEY WERE
C234  8E   FF23              LDX   #$FF23
C237  35     02      UNSTKR  PULS  A
C239  A7 84                  STA   ,X
C23B  35     02              PULS  A
C23D  A7 83                  STA   ,--X
C23F  30 88   E2             LEAX  -$1E,X
C242  8C   FF01              CMPX  #$FF01
C245  22     F0 C237         BHI   UNSTKR

C247  7F   0606              CLR   DD.OPT      GOOD ONE TIME ONLY, FOR SAFETY'S SAKE

C24A  5F             DGONE   CLRB
C24B  35     7B              PULS  CC,A,DP,X,Y,U GIT ITAND GO
C24D  24     37 C286         BCC   HOME1       !NO ERROR!
                    * FALL THRU TO GET AN ERROR
```

****

```
                        ********************************
                        * HANDLE AN ERROR FROM DISK
C24F  D6      F0                LDB   (DD.STA
C251  27      33 C286           BEQ   HOME1
C253  2A      20 C275           BPL   BITERR
                        * PARAMETER ERROR ?
C255  C1      FC                CMPB  #CMDERR
C257  26      04 C25D           BNE   HOP1
C259  C6      A4                LDB   #ERR.PR
C25B  20      27 C284           BRA   HOP4
                        * DRIVE NUMBER ERROR ?
C25D  C1      FD        HOP1    CMPB  #UNTERR
C25F  26      04 C265           BNE   HOP2
C261  C6      28                LDB   #ERR.DN
C263  20      1F C284           BRA   HOP4
                        * DRIVE OUT AVAILABLE ERROR ?
C265  C1      FE        HOP2    CMPB  #TIMERR
C267  26      04 C26D           BNE   HOP3
C269  C6      80                LDB   #ERR.NR
C26B  20      17 C284           BRA   HOP4
                        * SEEK ERROR ?
C26D  C1      FF        HOP3    CMPB  #$FF
C26F  26      11 C282           BNE   DSKUND
C271  C6      82                LDB   #ERR.SK
C273  20      0F C284           BRA   HOP4
                        *
C275  1F      98        BITERR  TFR   B,A
C277  C6      82                LDB   #ERR.WP-2
C279  49                        ROLA
C27A  CB      02        NEXTER  ADDB  #2
C27C  49                        ROLA
C27D  4D                        TSTA
C27E  25      04 C284           BCS   HOP4
C280  26      F8 C27A           BNE   NEXTER
C282  C6      A6        DSKUND  LDB   #ERR.UN
C284  1A      01        HOP4    SEC
C286  39                HOME1   RTS
                        ********************************
```

```
                        ********************************
                        *       SET UNIT
                        *  THIS ROUTINE SETS THE UNIT
                        *  NUMBER AND INSURES
                        *  THAT THE MOTOR IS ACTIVE.
                        ********************************
C287  86    D0      STUNIT  LDA   #ABTCMD
C289  B7  FF2F              STA   WD.CSR
C28C  96    EB              LDA   <DD.UNT     get drive number
C28E  27    04 C294         BEQ   BUMUNT
C290  81    04              CMPA  #4          max 4 drives
C292  23    03 C297         BLS   GOODUN
C294  1A    01      BUMUNT  SEC
C296  39            FAKEHK  RTS

C297  C6    01      GOODUN  LDB   #1          default drive 1
C299  58            UNLOOP  LSLB              make drive select bit
C29A  4A                    DECA              decrement counter
C29B  26    FC C299         BNE   UNLOOP      loop till bit found
C29D  54                    LSRB              correct over shift
C29E  CA    10              ORB   #$10         motor on bit
C2A0  34    04              PSHS  B
C2A2  B6  0607              LDA   CTLSAV
C2A5  84    F0              ANDA  #$F0        MASK UNIT #
C2A7  AA E0                 ORA   ,S+         OR IN NEW UNIT
C2A9  B7  0607              STA   CTLSAV      save current control
C2AC  17  1989 DC38         LBSR  SELDRV      select drive and motor on
C2AF  B6  FF25              LDA   NMICNT      get NMI control register
C2B2  8A    38              ORA   #$38        enable NMI
C2B4  B7  FF25              STA   NMICNT
                        *
                        * NOTE: MOTOR ON DELAY IS NOT
                        *       USED ON PURPOSE.
                        *
                        *      TELL 2797 ABOUT WHICH TRACK ITS REALLY ON

C2B7  8E  069A              LDX   #LUTRAK
C2BA  96    EB              LDA   <DD.UNT
C2BC  A6 86                 LDA   A,X         GET TRACK FOR THIS UNIT
C2BE  B7  FF2E              STA   WD.TRK      AND TELL THE CHIP

C2C1  86    D2              LDA   #TIMOUT
C2C3  B7  0605              STA   DD.TMR
C2C6  5F                    CLRB
C2C7  39                    RTS

                        ********************************
```

```
                        ********************************
                        *      DO RESTORE (#0)
                        ********************************
C2C8  4F          RSTHED  CLRA
C2C9  B7  00EC            STA   )DD.TRK
C2CC  20    12 C2E0       BRA   DOTYP1

                        ********************************
                        *  DO SEEK (#1)
                        *  DD.TRK MUST BE SET WHEN THIS
                        *  ROUTINE IS CALLED.
                        ********************************
C2CE  B6  00EC    SEEK    LDA   )DD.TRK
C2D1  91    2E            CMPA  (WD.TRK
C2D3  26    07 C2DC       BNE   DOSEEK
C2D5  4F                  CLRA                AND CARRY
C2D6  B7  0609            STA   ERRMSK
C2D9  1F    8B            TFR   A,DP
C2DB  39                  RTS

C2DC  97    2C    DOSEEK  STA   (WD.DAT
C2DE  86    10            LDA   #SEKCMD

                        ********************************
                        *      TYPE ONE WD COMMANDS.
                        *      WILL RETURN TY1ERR IF NO
                        *      NMI IS RECEIVED IN 5 SEC
                        ********************************
C2E0  C6    19    DOTYP1  LDB   #TY1ERR
C2E2  F7  0609            STB   ERRMSK

C2E5  8E  069E            LDX   #LUSTPR     TABLE OF RATES
C2E8  F6  00EB            LDB   )DD.UNT     THIS UNIT
C2EB  AA 85               ORA   B,X         RATE FOR THIS UNIT FROM THE TABLE

C2ED  97    2F            STA   (WD.CSR

C2EF  3D          DOTP11  MUL
C2F0  3D                  MUL
C2F1  31 3F               LEAY  -1,Y
C2F3  26    FA C2EF       BNE   DOTP11

                        *      NMI NEVER SHOWED: GO AWAY MAD
C2F5  20    4B C342       BRA   TOUTER
                        ********************************
```

```
                        ********************************
                        *
                        *      READ OPERATIONS
                        *
                        *      READ SECTOR (#2)
                        *      READ ADDRESS (#6)
                        *
                        *  DATA REQUESTED IN DD.SEC OR FIRST VALID ADDRESS
                        *  ARE READ INTO BUFFER DEFINED BY DD.BUF
                        *
                        *  WILL RETURN RSERR IF NG
                        ********************************
C2F7  86     C0         RDADR   LDA   #RACMD      READ ADDRESS
C2F9                            SKIP2
C2FA  86     88         RDSEC   LDA   #RSCMD      READ SECTOR
C2FC  C6     3F                 LDB   #RSERR
C2FE  F7   0609                 STB   ERRMSK
C301  C6     05                 LDB   #5
C303  BA   0606                 ORA   DD.OPT      RS OPTIONS
C306  97     2F                 STA   <WD.CSR

C308  96     27         RDTIME  LDA   <PB2CRB
C30A  2B     0E C31A            BMI   RDFIRQ
C30C  31 3F                     LEAY  -1,Y
C30E  26     F8 C308            BNE   RDTIME
C310  96     26                 LDA   <PB2DAT
C312  2B     05 C319            BMI   RDLOOP
C314  5A                        DECB
C315  26     F1 C308            BNE   RDTIME
C317  20     29 C342            BRA   TOUTER

C319  13                RDLOOP  SYNC
C31A  D6     26         RDFIRQ  LDB   <PB2DAT     CLEAR INTERRUPT
C31C  96     2C                 LDA   <WD.DAT     GET DATA
C31E  A7 80                     STA   ,X+
C320  20     F7 C319            BRA   RDLOOP

                        ********************************
```

```
                        ********************************
                        *      VERIFY (#7)
                        *  READ FIRST TWO BYTES OF SECTOR INTO SCRATCH AREA (FPAC)
                        *  THEN FLUSH REST OF SECTOR DATA AND CHECK STATUS
                        *  WILL RETURN VERERR IF NG
                        ********************************
C322  8E   004F       VERF     LDX    #OSLOC
C325  86     3F                LDA    #VERERR
C327  B7   0609                STA    ERRMSK
C32A  C6     05                LDB    #5
C32C  86     88                LDA    #RSCMD
C32E  BA   0606                ORA    DD.OPT
C331  97     2F                STA    (WD.CSR

C333  96     27       VRTIME   LDA    (PB2CRB
C335  2B     17 C34E           BMI    VRFIRQ
C337  31 3F                    LEAY   -1,Y
C339  26     F8 C333           BNE    VRTIME
C33B  96     26                LDA    (PB2DAT
C33D  2B     0F C34E           BMI    VRFIRQ
C33F  5A                       DECB
C340  26     F1 C333           BNE    VRTIME

C342  86     D0       TOUTER   LDA    #ABTCMD
C344  97     2F                STA    (WD.CSR
C346  4F                       CLRA
C347  1F     8B                TFR    A,DP
C349  86     FE                LDA    #TIMERR
C34B  1A     01                SEC
C34D  39                       RTS

                        *      THE FIRST TWO BYTES OF THE SECTOR BEING
                        *      VERIFIED ARE SAVED IN BASIC'S FPAC (SCRATCH)
                        *      FOR THE NON-CLOBBERING BOOT ROUTINE TO CHECK

C34E  96     2C       VRFIRQ   LDA    (WD.DAT
C350  0D     26                TST    (PB2DAT
C352  A7 80                    STA    ,X+
C354  13                       SYNC
C355  96     2C                LDA    (WD.DAT
C357  0D     26                TST    (PB2DAT
C359  A7 84                    STA    ,X

                        *      AND NOW THE MAIN FLUSH LOOP...

C35B  13                VLOOP    SYNC
C35C  0D     26                TST    (PB2DAT     CLEAR INTERRUPT
C35E  96     2C                LDA    (WD.DAT     GET DATA
C360  20     F9 C35B           BRA    VLOOP

                        ********************************
```

```
                        ********************************
                        *      WRITE SECTOR OPERATIONS
                        *
                        *      WRITE WITH OPTIONAL VERIFY (#3)
                        *      JUST WRITE - NO VERIFY (#4)
                        *
                        *  FROM THE BUFFER DEFINED BY
                        *  DD.BUF INTO DD.SEC
                        *  WILL RETURN WVERR OR WSERR IF NG
                        ********************************
C362  86     5F         WRTSEC  LDA   #WSERR      NO VERIFY NOHOW
C364                            SKIP2
C365  86     DF         WRTVER  LDA   #WVERR      DO VERIFY IF VERFLG IS SET
C367  B7   0609                 STA   ERRMSK
C36A  8D     3D C3A9            BSR   WCOMP
C36C  86     A8                 LDA   #WSCMD
C36E  BA   0606                 ORA   DD.OPT      DISK OPTION
C371  97     2F                 STA   <WD.CSR
C373  A6 80                     LDA   ,X+

C375  D6     27         WRTIME  LDB   <PB2CRB
C377  2B     09 C382            BMI   WRFIRQ
C379  31 3F                     LEAY  -1,Y
C37B  26     F8 C375            BNE   WRTIME
C37D  20     C3 C342            BRA   TOUTER

C37F  A6 80             WRLOOP  LDA   ,X+
C381  13                        SYNC
C382  97     2C         WRFIRQ  STA   <WD.DAT
C384  96     26                 LDA   <PB2DAT
C386  20     F7 C37F            BRA   WRLOOP

                        ********************************
```

```
                        ********************************
                        *      WRITE TRACK (#5)        *
                        *  will return WTERR if N.F.G. *
                        *                              *
                        *    The format for the write  *
                        * buffer is somewhat strange   *
                        * for this routine. The reason *
                        * for this is so we can format *
                        * a diskette without hosing    *
                        * the poor users basic program.*
                        *   The method of run-lenght   *
                        * coding requires only $22B    *
                        * bytes instead of the normale *
                        * $0000 bytes.                 *
                        *                              *
                        ********************************
C388  86     47     WRTRAK  LDA   #WTERR
C38A  B7   0609             STA   ERRMSK
C38D  8D     1A C3A9        BSR   WCOMP
C38F  86     F4             LDA   #WTCMD
C391  BA   0606             ORA   DD.OPT
C394  97     2F             STA   (WD.CSR
C396  EC 81          HOP5   LDD   ,X++          *-8
C398  13             HOP6   SYNC                *-4
C399  91     26             CMPA  (PB2DAT      *-4 CLEAR INTERRUPT
C39B  D7     2C             STB   (WD.DAT      *-4 OUTPUT DATA
C39D  4A                    DECA               *-2
C39E  26     F8 C398        BNE   HOP6         *-3
C3A0  A6 80                 LDA   ,X+          *-6
C3A2  13                    SYNC               *-4
C3A3  91     26             CMPA  (PB2DAT      *-2 CLEAR INTERRUPT
C3A5  97     2C             STA   (WD.DAT      *-4 OUTPUT DATA
C3A7  20     ED C396        BRA   HOP5         *-3

                        ********************************
```

```
                        ********************************
                        *      WRITE PRECOMP
                        *  ENABLED ABOVE TRACK 16
                        ********************************
C3A9  96     2E         WCOMP   LDA    (WD.TRK
C3AB  81     10                 CMPA   #16
C3AD  23     07 C3B6            BLS    NOCOMP
C3AF  B6   0607                 LDA    CTLSAV
C3B2  8A     40                 ORA    #ENPRE
C3B4  20     05 C3BB            BRA    COMPEX
C3B6  B6   0607         NOCOMP  LDA    CTLSAV
C3B9  84     BF                 ANDA   #NOPRE
C3BB  17   187A DC38    COMPEX  LBSR   SELDRV      output pre-comp control bit
C3BE  39                        RTS

                        ********************************
                                END
                                LIB             FORMAT.A
                                NAM             FORMAT.A
```

```
                        ********************************
                        * FORMAT UNIT,TRACKS,SIDES
                        *       1-4  40-80  1-2
                        *
                        * FORMAT DISK COMMAND
                        *
                        ********************************

C3BF   34    01      BOOTDSK  PSHS   CC          SAVE CONDITION CODES
C3C1   0A    F7               DEC    <SYSDSK     SET SYSTEM DISK FORMAT
C3C3   35    01               PULS   CC
                        *
C3C5   27    23 C3EA FORMAT   BEQ    NFMARG      IF NO ARGS,FORMAT DEFAULT UNIT, 80 TRKS, 1 SIDE
                        *
                        * GET UNIT NUMBER
                        *
C3C7   BD    C709             JSR    DRVNUM      GET DRIVE NUMBER
C3CA   D7    EB               STB    <DD.UNT     SAVE DRIVE NUMBER
                        *
                        * GET # OF SIDES
                        *
C3CC   BD    C590             JSR    OPTCHR      GET NUMBER OF TRACKS
C3CF   27    1E C3EF          BEQ    NFMSID      SKIP IF NONE ENTERED DEFAULT SINGLE SIDED 80 TRACKS
C3D1   5A                     DECB
C3D2   C1    01               CMPB   #1
C3D4   22    11 C3E7          BHI    HOP7        SKIP IF MORE THEN TWO SIDES REQUESTED
C3D6   D7    F4               STB    <FM.SDE     SAVE TWO SIDED
                        *
                        * GET TRACKS
                        *
C3D8   BD    C590             JSR    OPTCHR      GET NUMBER OF TRACKS
C3DB   27    14 C3F1          BEQ    NFMTRK      SKIP IF NONE ENTERED DEFAULT 80 TRACK
C3DD   C1    28               CMPB   #40         (40 OR 80)
C3DF   27    12 C3F3          BEQ    FMRGOT
C3E1   00    F4               NEG    <FM.SDE     MAKE NEGATIVE IF TWO SIDED 80 TRACK
C3E3   C1    50               CMPB   #80
C3E5   27    0C C3F3          BEQ    FMRGOT
C3E7   7E    C716      HOP7   JMP    ER.PRM      ONLY 40 OR 80 TRACKS ALLOWED
C3EA   F6    060A      NFMARG LDB    DEFDRV      GET DEFAULT DRIVE NUMBER
C3ED   D7    EB               STB    <DD.UNT
C3EF   0F    F4        NFMSID CLR    <FM.SDE     DEFAULT SINGLE SIDED 80 TRACK
C3F1   C6    50        NFMTRK LDB    #80
C3F3   D7    F2        FMRGOT STB    <FM.TRK
C3F5   BD    D742             JSR    HKCLOS      CLOSE ALL OPEN FILES
C3F8 1026    0086 C482        LBNE   FM.ERO      SKIP IF ERROR
C3FC   8E    0800             LDX    #FM.BUF     POINT AT SECTOR BUFFER
C3FF   9F    EE               STX    <DD.BUF     SAVE POINTER

                        *       WE HAVE TO CHECK HERE FOR A MOTOR ON DELAY.
                        *       IT'S THE ONLY PLACE WE CAN'T GET AWAY WITH IT.
                        *

C401   BD    C165             JSR    GO.RST      RESTORE DRIVE
C404   26    7C C482          BNE    FM.ERO      ABORT ON ERROR
```

```
C406  86   01            LDA   #1          LOOK FOR SECTOR 1
                    * CHECK DISK READY
C408  97   ED            STA   (DD.SEC
C40A  BD   C15F           JSR   GO.VER
C40D  C1   80             CMPB  #ERR.NR     ABORT ON NOT READY ERROR ELSE FALL THOUGH
C40F  27   71 C482        BEQ   FM.ERO

                    *     IT MAY OR MAY NOT FLUNK THE VERIFY. IF IT FLUNKS,
                    *     IT WILL LET THE MOTOR RUN FOR 5 REVOLUTIONS
                    *     IF IT SUCCEEDS, WE MOVE ON WITHOUT DELAY

C411  0F   F3     FM.LOP  CLR   (FM.SID     SET SIDE ONE
C413  0F   ED             CLR   (DD.SEC     BECOMES SIDE FLAG HERE..
C415  BD   C55F           JSR   FM.BLD      BUILD TRACK IN MEMORY
C418  BD   C0FB           JSR   GO.WRT      AND WRITE IT OUT
C41B  25   65 C482        BCS   FM.ERO      ABORT ON ERROR

                    *     CHECK FOR DOUBLE SIDED

C41D  0D   F4             TST   (FM.SDE     IS THIS A DOUBLE SIDED FORMAT?
C41F  27   0F C430        BEQ   FM.ONE      IF NOT THEN CONTINUE THE LOOP
C421  86   01             LDA   #1          SET FORMAT'S SIDE BIT TO SIDE ONE
C423  97   F3             STA   (FM.SID
C425  40                  NEGA              SIDE TWO SWITCH. WILL BE GREATER
C426  97   ED             STA   (DD.SEC     THAN SECT/TRACK AND WILL FORCE SIDE TWO

                    *     AND GO FORMAT THE BACK SIDE

C428  BD   C55F           JSR   FM.BLD      BUILD THE FLIP SIDE TRACK IN MEMORY
C42B  BD   C0FB           JSR   GO.WRT      AND WRITE IT OUT
C42E  25   52 C482        BCS   FM.ERO      ERROR

                    *     THEN PRESS ON

C430  0C   EC     FM.ONE  INC   (DD.TRK
C432  96   EC             LDA   (DD.TRK
C434  91   F2             CMPA  (FM.TRK
C436  25   D9 C411        BCS   FM.LOP

                    ********************************
                    * VERIFY DISK

C438  BD   C165           JSR   GO.RST      START AT THE  LOGICAL PLACE...
C43B  25   45 C482        BCS   FM.ERO

C43D  BD   C162     FM.VLP  JSR   GO.SEK    GET ON TRACK
C440  25   40 C482        BCS   FM.ERO
C442  4F                  CLRA
C443  BD   C548           JSR   TVER        CHECK 1 OR 2 SIDES, DEPENDING

C446  0C   EC             INC   (DD.TRK
C448  96   EC             LDA   (DD.TRK
C44A  91   F2             CMPA  (FM.TRK
C44C  25   EF C43D        BCS   FM.VLP      FOR NEXT TRACK IF NOT DONE
```

```
                        ****************
                        * INIT THE DIR'S
C44E  9E    EE          LDX   (DD.BUF
C450  DC    8A          LDD   DBLZER

C452  ED 81         HOP8    STD   ,X++
C454  8C  0B00              CMPX  #FM.BUF+$300
C457  26    F9 C452         BNE   HOP8
C459  86    01              LDA   #01
C45B  97    ED              STA   (DD.SEC
C45D  86    14              LDA   #DIRTRK
C45F  8D    0A C46B         BSR   HOP9        WRITE 1ST SECTOR OF DIR TRACK 20
C461  0A    ED              DEC   DD.SEC
C463  0A    EE              DEC   DD.BUF
C465  86    10              LDA   #ALTTRK
C467  8D    08 C471         BSR   HOP10       WRITE 1ST SECTOR OF DIR TRACK 16
C469  20    1D C488         BRA   HOP11
C46B  34    02      HOP9    PSHS  A
C46D  8D    70 C4DF         BSR   MAKSAT      MAKE SECTOR 1 AND 2 OF DIR TRACK
C46F  35    02              PULS  A
C471  97    EC      HOP10   STA   DD.TRK
C473  BD  C101              JSR   GO.WRV      WRITE 1ST SECTOR OF DIR
C476  25    0A C482         BCS   FM.ERO
C478  0C    ED              INC   DD.SEC
C47A  0C    EE              INC   DD.BUF
C47C  BD  C101              JSR   GO.WRV      WRITE 2ND SECTOR OF DIR
C47F  25    01 C482         BCS   FM.ERO      ABORT ON ERROR
C481  39                    RTS
C482  7F  00F7      FM.ERO  CLR   )SYSDSK     CLEAR SYSTEM DISK FLAG
C485  7E  C718              JMP   NEWERR
C488  0C    EE      HOP11   INC   DD.BUF
C48A  9E    EE              LDX   (DD.BUF
C48C  CC  890A              LDD   #$890A      25 BYTES PER DIR ENTRY 10 ENTRYS PER SECTOR
C48F  A7 84         HOP12   STA   ,X
C491  30 88  19             LEAX  DIRSIZ,X
C494  5A                    DECB
C495  26    F8 C48F         BNE   HOP12

                        * FILL THE DIRECTORYS WITH THIS
C497  8D    04 C49D         BSR   WRTDLP      WRITE DIR TRACK 16
C499  86    14              LDA   #DIRTRK     WRITE DIR TRACK 20
C49B  97    EC              STA   (DD.TRK     SAVE TRACK NUMBER
C49D  CC  1003      WRTDLP  LDD   #$1003      START ON SECTOR 3
C4A0  D7    ED              STB   (DD.SEC     16 SECTOR TO WRITE
C4A2  BD  C101      HOP13   JSR   GO.WRV      WRITE SECTOR WITH VERIFY
C4A5  25    DB C482         BCS   FM.ERO      ABORT ON ERROR
C4A7  0C    ED              INC   (DD.SEC     INCRERMENT SECTOR NUMBER
C4A9  4A                    DECA              DECREMENT SECTOR COUNTER
C4AA  26    F6 C4A2         BNE   HOP13       LOOP TILL ALL DONE
                        *
                        * USE GO.WRV IE WRITE SECTOR WITH VERIFY
                        * SECTOR IN DD.SEC TRACK IN DD.TRK
                        * WRIT 32 SECTORS STARTING ON SECTOR 3
                        * SIDE IS AUTOMATICLY TAKEN CARE OFF IN GO.WRV
                        *
```

```
C4AC  0D    F7           TST   (SYSDSK     IS THIS A SYSTEM DISK
C4AE  27    29 C4D9      BEQ   NOBOOT      SKIP IF NOT
C4B0  BD    C165         JSR   GO.RST      RESTOR DRIVE TO TRACK 0 AND CLEAR DD.TRK
C4B3  25    CD C482      BCS   FM.ERO      ABORT ON ERROR
C4B5  8E    C000         LDX   #PROBEG     POINT AT START OF DOS
C4B8  C6    03           LDB   #3          START ON SECTOR 3
C4BA  D7    ED     MAKBOOT STB   (DD.SEC    SAVE SECTOR NUMBER
C4BC  9F    EE           STX   (DD.BUF     SAVE BUFFER POINTER
C4BE  BD    C101         JSR   GO.WRV      GO WRITE SECTOR
C4C1  25    BF C482      BCS   FM.ERO      ABORT ON ERROR
C4C3  30 89 0100         LEAX  $100,X      NEXT 256 BYTES
C4C7  8C    E000         CMPX  #PROBEG+$2000 END OF DOS
C4CA  27    0D C4D9      BEQ   NOBOOT
C4CC  D6    ED           LDB   (DD.SEC     GET SECTOR NUMBER
C4CE  5C                 INCB               INCREMENT IT
C4CF  C1    13           CMPB  #18+1       ARE WE THERE
C4D1  26    E7 C4BA      BNE   MAKBOOT     LOOP TILL TRACK DONE
C4D3  0C    EC           INC   (DD.TRK     NEXT TRACK
C4D5  C6    01           LDB   #1          NEXT TRACK AND SECTOR 1
C4D7  20    E1 C4BA      BRA   MAKBOOT     LOOP TILL DONE
C4D9  7F    00F7   NOBOOT CLR  )SYSDSK     CLEAR SYSTEM DISK FLAG
C4DC  7E    C0B8         JMP   INITVR

                  ********************************
C4DF  97    EC     MAKSAT STA   (DD.TRK
C4E1  86    12           LDA   #18         SECTORS/TRACK
C4E3  C6    5A           LDB   #90         BYTES IN SAT
C4E5  0D    F4           TST   (FM.SDE     # SIDES
C4E7  27    02 C4EB      BEQ   HOP14
C4E9  58                 LSLB
C4EA  48                 LSLA
C4EB  B7    08FD   HOP14 STA   FM.BUF+253  NUMBER OF SECTORS
C4EE  43                 COMA
C4EF  B7    08FF         STA   FM.BUF+255  COMPLEMENT OF NUMBER OF SECTORS
C4F2  96    F2           LDA   (FM.TRK
C4F4  B7    08FC         STA   FM.BUF+252  NUMBER OF TRACKS
C4F7  0D    F4           TST   (FM.SDE     HAVE WE ALREADY SHIFTED
C4F9  26    05 C500      BNE   HOP400      SKIP IS SO
C4FB  81    50           CMPA  #80         80 TRACK DRIVE ?
C4FD  26    01 C500      BNE   HOP400      SKIP IF NOT
C4FF  58                 LSLB               DOUBLE BYTES IN SECTOR ALLOCATION MAP
C500  43           HOP400 COMA
C501  B7    08FE         STA   FM.BUF+254  COMPLERMENT OF TRACKS

                  * SET UP SAT
C504  9E    EE           LDX   (DD.BUF     POINT AT START OF DIR TRACK BUFFER
C506  CE    0900         LDU   #FM.BUF+$100
C509  0D    F7           TST   (SYSDSK     IS THIS A SYSTEM DISK
C50B  27    0E C51B      BEQ   NOTSYS      SKIP IF NOT
C50D  96    EB           LDA   (DD.UNT     GET DRIVE NUMBER
C50F  81    02           CMPA  #2          SYSTEM DISK ONLY ON DRIVE 1 & 2
C511  22    08 C51B      BHI   NOTSYS      SKIP IF NOT DRIVE 1 OR 2
C513  C0    05           SUBB  #5          TAKE SECTOR REQUIRED FOR BOOT
C515  30 04              LEAX  4,X         MOVE POINTER
C517  86    FC           LDA   #$FC
```

```
C519   A7 80                    STA    ,X+          NOW WE HAVE THEM
C51B   86      FF       NOTSYS  LDA    #$FF         INDICATE SECTORS FREE
C51D   A7 80            HOP15   STA    ,X+
C51F   5A                       DECB
C520   26      FB C51D          BNE    HOP15        LOOP TILL SECTOR MAP INIT
C522   CC    242D               LDD    #$242D       DEFALUT DIR ON TRACK 16 AND 20
C525   0D      F4               TST    <FM.SDE      TEST SIDE FLAG
C527   27      0D C536          BEQ    HOP18        SKIP IF SINGLE SIDE
C529   2A      08 C533          BPL    HOP17        SKIP IF TWO SIDE 40 TRACK
C52B   CC    B4FF               LDD    #$B4FF       MUST BE TWO SIDE 80 TRACK
C52E   E7 C0            HOP16   STB    ,U+
C530   4A                       DECA
C531   26      FB C52E          BNE    HOP16
C533   CC    485A      HOP17   LDD    #$485A       DIR ON TRACKS 16 AND 20
C536   DE      8A      HOP18   LDU    <DBLZER      CLEAR A & B (SAVE BYTES)
C538   34      02               PSHS   A
C53A   8D      02 C53E          BSR    FILL
C53C   35      04               PULS   B
C53E   9E      EE      FILL    LDX    <DD.BUF
C540   3A                       ABX
C541   86      FC               LDA    #$FC         STORE 00,00,FC
C543   EF 81                    STU    ,X++
C545   A7 84                    STA    ,X
C547   39              HOME2   RTS
                       ********************************
C548   0F      ED      TVER    CLR    <DD.SEC
C54A   0D      F4               TST    <FM.SDE      SINGLE SIDED
C54C   27      02 C550          BEQ    FM.VER
C54E   8D      00 C550          BSR    FM.VER       ELSE DO BOTH SIDES
C550   86      12      FM.VER  LDA    #18
C552   0C      ED      HOP19   INC    <DD.SEC
C554   BD    C15F               JSR    GO.VER       CHECK SECTOR
C557 1025    FF27 C482          LBCS   FM.ERO       ON ERROR
C55B   4A                       DECA
C55C   26      F4 C552          BNE    HOP19
C55E   39              HOME3   RTS

                       ********************************
                       * BUILD TRACK IN MEMORY

C55F   DE      EE      FM.BLD  LDU    <DD.BUF      GET BUFFER POINTER
C561   8E    DE0D               LDX    #FM.FMT
C564 108E    DDFA               LDY    #INTLEV      POINT AT SECTOR INTERLEAVE TABLE
C568   C6      0C               LDB    #$04*3       NUMBER OF BYTES TO MOVE
C56A   8D      21 C58D          BSR    HOP20        MOVE DATA AT X TO U
                       *
                       ********************************
                       * BUILD A SECTOR LOOP
                       *
C56C   8E    DE19      BSECTR  LDX    #FM.FMS      POINT AT ?
C56F   C6      06               LDB    #$2*3        NUMBER OF BYTES TO MOVE
C571   8D      1A C58D          BSR    HOP20        MOVE DATA AT X TO U
C573   86      01               LDA    #$01         * COUNT BYTE
C575   D6      EC               LDB    <DD.TRK
C577   ED C1                    STD    ,U++
```

```
C579  D6     F3            LDB   (FM.SID
C57B  E7 C0                STB   ,U+
C57D  E6 A0                LDB   ,Y+          * GET SECTOR #
C57F  ED C1                STD   ,U++
C581  A7 C0                STA   ,U+
C583  C6     12            LDB   #$06*3
C585  8D     06 C58D       BSR   HOP20
C587  6D A4                TST   ,Y           LAST SECTOR?
C589  26     E1 C56C       BNE   BSECTR
C58B  C6     03            LDB   #$01*3
C58D  7E   B7CC     HOP20   JMP   BLKMOV

                           ********************************
C590  9D     A5     OPTCHR  JSR   CHRGOT
C592  27     CA C55E       BEQ   HOME3
C594  BD   89AA              JSR   CHKCOM
C597  7E   C6B9              JMP   NZRBYT
                           ********************************
```

```
                        ********************************
                        *
                        *      BACKUP
                        *
                        * COMMAND SYNTAX:
                        *  BACKUP [d [TO d][,s][,t]]
                        *
                        *  d = 1 TO 4, DRIVE NUBER
                        *  s = 1 OR 2, NUMBER OF SIDES
                        *  t = 40 OR 80 NUMBER OF TRACKS
                        *
                        ********************************
                        * STACK FRAME
                        *
                        * 0015 : MEMSEC, RAM BUFFER SIZE IN SECTORS
                        * 0014 : BUPSEC, SECTORS/TRACKS
                        *      :-----------------------
                        * 0012 : RAM POINTER           :
                        * 0010 : STRING POINTER        :
                        * 0009 : SECTOR                : DESTINATION DRIVE
                        * 0008 : TRACK                 :
                        * 0007 : UNIT                  :
                        *      :-----------------------
                        * 0005 : RAM POINTER           :
                        * 0003 : STRING POINTER        :
                        * 0002 : SECTOR                : SOURCE DRIVE
                        * 0001 : TRACK                 :
                        * 0000 : UNIT                  :
                        *      :-----------------------
                        * U IS USED AS A POINTER TO THE
                        * BEGINNING OF THE FRAME. BELOW
                        * THIS POINTER IS STACK USAGE
                        * FOR SUBROUTINES, ETC. WITHIN
                        * BACKUP.
                        *
                        ********************************
E C59A                  BACKUP  EQU   *
                        *     JSR   NEW
  C59A   32 70                    LEAS  -16,S
  C59C   1F      43               TFR   S,U
  C59E   1F      30               TFR   U,D
  C5A0   83    0040               SUBD  #64
  C5A3   93      1F               SUBD  ARRTOP
  C5A5 102B    BD99 8342          LBMI  OMERR
  C5A9   81      01               CMPA  #1
  C5AB 102D    BD93 8342          LBLT  OMERR
  C5AF   A7 4F                    STA   15,U
  C5B1   86      12               LDA   #18
  C5B3   A7 4E                    STA   14,U
  C5B5   DC      1F               LDD   ARRTOP
  C5B7   ED 45                    STD   5,U
  C5B9   ED 4C                    STD   12,U
  C5BB   CC    DF6D               LDD   #SSTRG-1
  C5BE   ED 43                    STD   3,U
```

```
C5C0   CC    DF7C              LDD   #DESTRG-1
C5C3   ED 4A                   STD   10,U
C5C5   CC    0001              LDD   #1
C5C8   ED 41                   STD   1,U
C5CA   ED 48                   STD   8,U
C5CC   B6    060A              LDA   DEFDRV
C5CF   A7 C4                   STA   0,U
C5D1   A7 47                   STA   7,U
C5D3 108E    02D0              LDY   #720
C5D7   9D      A5              JSR   CHRGOT
C5D9   27      4A C625         BEQ   GOON
C5DB   BD    C6B9              JSR   NZRBYT
C5DE   C1      04              CMPB  #4
C5E0 1022    012F C713         LBHI  ER.DNE
C5E4   E7 C4                   STB   0,U
C5E6   E7 47                   STB   7,U
C5E8   9D      A5              JSR   CHRGOT
C5EA   27      39 C625         BEQ   GOON
C5EC   81      BC              CMPA  #TOTOK
C5EE   26      19 C609         BNE   BUSNER
C5F0   9D      9F              JSR   GETCHR
C5F2   BD    C6B9              JSR   NZRBYT
C5F5   C1      04              CMPB  #4
C5F7 1022    0118 C713         LBHI  ER.DNE
C5FB   E7 47                   STB   7,U
C5FD   8D      91 C590         BSR   OPTCHR
C5FF   27      24 C625         BEQ   GOON
C601   C1      02              CMPB  #2
C603   27      07 C60C         BEQ   HOP21
C605   C1      01              CMPB  #1
C607   27      09 C612         BEQ   HOP22
C609   7E    89B4      BUSNER   JMP   SNERR
C60C   1F      20      HOP21    TFR   Y,D
C60E   31 AB                   LEAY  D,Y
C610   68 4E                   ASL   14,U
C612   BD    C590      HOP22    JSR   OPTCHR
C615   27      0E C625         BEQ   GOON
C617   C1      50              CMPB  #80
C619   27      06 C621         BEQ   HOP23
C61B   C1      28              CMPB  #40
C61D   27      06 C625         BEQ   GOON
C61F   20      E8 C609         BRA   BUSNER
C621   1F      20      HOP23    TFR   Y,D
C623   31 AB                   LEAY  D,Y
C625   4F              GOON     CLRA
C626   31 21           BUPRMT   LEAY  1,Y
C628   30 C4                   LEAX  0,U
C62A   8D      6D C699         BSR   PROMPT
C62C   31 3F            BULOOP  LEAY  -1,Y
C62E   26      06 C636         BNE   TESTA
C630   8D      1D C64F         BSR   EMPTYY
C632   32 C8   10              LEAS  16,U
C635   39              HOME4    RTS

C636   A1 4F            TESTA   CMPA  15,U
```

```
C638  26    0E C648          BNE   MORA
C63A  8D    13 C64F          BSR   EMPTYY
C63C  34    06               PSHS  D
C63E  DC    1F               LDD   ARRTOP
C640  ED 4C                  STD   12,U
C642  ED 45                  STD   5,U
C644  35    06               PULS  D
C646  20    DE C626          BRA   BUPRMT

C648  C6    02       MORA    LDB   #DIO.RS
C64A  8D    12 C65E          BSR   BUOPRS
C64C  4C                     INCA
C64D  20    DD C62C          BRA   BULOOP

C64F  4D             EMPTYY  TSTA
C650  27    E3 C635          BEQ   HOME4
C652  30 47                  LEAX  7,U
C654  8D    43 C699          BSR   PROMPT
C656  C6    03               LDB   #DIO.WV
C658  8D    04 C65E EMLOOP   BSR   BUOPRS
C65A  4A                     DECA
C65B  26    FB C658          BNE   EMLOOP
C65D  39             HOME5   RTS

C65E  34    06       BUOPRS  PSHS  A,B
C660  A6 84                  LDA   0,X
C662  97    EB               STA   (DD.UNT
C664  EC 05                  LDD   5,X
C666  DD    EE               STD   DD.BUF
C668  EC 01                  LDD   1,X
C66A  DD    EC               STD   (DD.TRK
C66C  E6 61                  LDB   1,S
C66E  BD    C106             JSR   EDISK
C671  24    14 C687          BCC   HOP25
                     * ERROR - ERROR - ERROR
C673  F7    0603             STB   TEMP.C
C676  A6 61                  LDA   1,S          DISK CMMD
C678  81    02               CMPA  #DIO.RS
C67A  26    05 C681          BNE   HOP24        BOMB TIME
C67C  35    16               PULS  A,B,X
C67E  BD    C64F             JSR   EMPTYY
C681  F6    0603     HOP24   LDB   TEMP.C
C684  7E    C718             JMP   NEWERR
                     *
C687  6C 02          HOP25   INC   2,X
C689  A6 02                  LDA   2,X
C68B  A1 4E                  CMPA  14,U
C68D  23    06 C695          BLS   OPRS1
C68F  86    01               LDA   #1
C691  A7 02                  STA   2,X
C693  6C 01                  INC   1,X
C695  6C 05          OPRS1   INC   5,X
C697  35    86               PULS  A,B,PC

C699  E6 C4          PROMPT  LDB   0,U
```

```
C69B   E1 47              CMPB   7,U
C69D   26    BE C65D      BNE    HOME5
C69F   34    72           PSHS   A,X,Y,U
C6A1   BD    BA77         JSR    GRNCLS
C6A4   AE 61              LDX    1,S
C6A6   AE 03              LDX    3,X
C6A8   BD    90E5         JSR    STROUT
C6AB   8E    DF90         LDX    #DIRSTR-1
C6AE   BD    90E5         JSR    STROUT
C6B1   BD    B505         JSR    KBCHAR
C6B4   BD    BA77         JSR    GRNCLS
C6B7   35    F2           PULS   A,X,Y,U,PC

                  ********************************
                          END
                          LIB            BIFACE.A
                          NAM            BIFACE
```

```
                        ********************************
C6B9   34     60        NZRBYT  PSHS   Y,U
C6BB   BD   8E51                JSR    GETBYT
C6BE   5D                       TSTB
C6BF   26     03 C6C4           BNE    HOP26
C6C1   7E   8B8D                JMP    FCERR
C6C4   35     E0        HOP26   PULS   Y,U,PC

                        ********************************
                        *
                        * BASIC INTERFACE MODULE
                        *
                        ********************************
                        * INTERPRETER ENTRY
C6C6   81     FF        CMDITP  CMPA   #$FF
C6C8 1027   C2E8 89B4           LBEQ   SNERR
C6CC   80     CE                SUBA   #LOWTOK
C6CE   2A     03 C6D3           BPL    CTOKHI
C6D0   7E   89B4        BISNER  JMP    SNERR

C6D3   81     1A        CTOKHI  CMPA   #(CMDEND-CMDADD)/2 IS THE TOKEN
C6D5   24     06 C6DD           BHS    CMDEXP      TOO BIG FOR US?

C6D7   8E   DE69                LDX    #CMDADD
C6DA   7E   84ED                JMP    DOJUMP      DO JUMP

                        * HERE WE ALLOW FOR THE EXPANSION INTERFACE

C6DD   6E 9F 0137       CMDEXP  JMP    [CERAML]    FOR EXPANSION COMMANDS

                        ********************************
C6E1   C0     44        FNCITP  SUBB   #LOWFUN     OUR LOWEST FUNCTION TOKEN*2
C6E3   2A     02 C6E7           BPL    FTOKHI      IF HIGH ENOUGH
C6E5   20     E9 C6D0           BRA    BISNER      SNERR AT THE USER

C6E7   C1     0E        FTOKHI  CMPB   #(FNCEND-FNCADD) IS IT TOO BIG ?
C6E9   24     08 C6F3           BHS    FNCEXP

                        *       GO EVALUATE THE FUNCTION THEN CHECK TYPE MATCH
                        *       ON RETURN TO BASIC

C6EB   8E   DE9D                LDX    #FNCADD     OUR TABLE IS HERE
C6EE   AD 95                    JSR    [B,X]
C6F0   7E   8874                JMP    FUNRET

                        *       EXPANSION INTERFACE...
C6F3   6E 9F 013C       FNCEXP  JMP    [FERAML]    FOR EXPANSION FUNCTIONS

                        ********************************
                        *
C6F7   8E   0634        CLEAN4  LDX    #PBFINF
C6FA   BD   D330        HOP27   JSR    CLEANP
C6FD   26     19 C718           BNE    NEWERR
C6FF   6F 02                    CLR    PABVLD,X
```

```
C701  30 07                LEAX  LENPBF,X
C703  8C   0650            CMPX  #PBFINF+(LENPBF*4)
C706  25     F2 C6FA       BLO   HOP27
C708  39           HOME6   RTS

                   ********************************
C709  BD   8E51    DRVNUM  JSR   GETBYT
C70C  5D                   TSTB
C70D  27     04 C713       BEQ   ER.DNE
C70F  C1     04            CMPB  #4
C711  23     F5 C708       BLS   HOME6

                   ********************************
```

```
                        ********************************
C713  C6    28          ER.DNE  LDB   #ERR.DN
C715                            SKIP2
C716  C6    A4          ER.PRM  LDB   #ERR.PR

                        ********************************
                        *
                        *     ERROR HANDLER
                        *
                        ********************************
E C718                  DKERR   EQU   *
  C718  F7  0619        NEWERR  STB   ERRNUM     SAVE ERROR
  C71B  9E    68                LDX   CURLIN     SAVE LINE #
  C71D  BF  0617                STX   ERLINE

  C720  BD  8434                JSR   STKINI     RE-INIT THE STACK
  C723  0F    F6                CLR   LOCK       TIME OUT
  C725  BD  BDDC                JSR   CASOFF     TURN OFF THE CASSETTE MOTOR
  C728  BD  BAC3                JSR   SNDENB     TURN ON THE6 BIT SOUND
  C72B  0F    6F                CLR   <$006F     SET THE DEVICE NUMBER TO THE SCREEN
  C72D  7D  0614                TST   ONERRF     IS ERROR TRAPING TURNED ON
  C730  2A    09 C73B           BPL   NEWER1     NO,
  C732  9E    68                LDX   CURLIN
  C734  30    01                INX
  C736  26    14 C74C           BNE   DONERR
  C738  BD  90A1                JSR   VDCRLF     MOVE THE CURSOR DOWN

  C73B  BD  90F8        NEWER1  JSR   VDQUES
  C73E  8E  82A9                LDX   #BERRLS
  C741  F6  0619                LDB   ERRNUM
  C744  2A    03 C749           BPL   HOP28
  C746  8E  DF2C                LDX   #ERRTAB-$80
  C749  7E  835E        HOP28   JMP   INER1
                        ********************************
                        * ON ERROR GOTO ...

  C74C  8E  84DA        DONERR  LDX   #BASRTS    * RE-ENTRYPOINT FOR BASIC INTERPITER
  C74F  34    10                PSHS  X          PUT THE RETURN ADDRESS ON THE STACK

  C751  7F  0614                CLR   ONERRF     ONLY ONE TRAP PER 'ERROR GOTO ...'
  C754  FC  0615                LDD   ONERRL     * GET LINE NUMBER TO FIND
  C757  DD    2B                STD   <LINNUM
  C759  7E  85E7                JMP   GOTO
                        ********************************
```

```
                        ********************************
                        * INTERRUPT & RESET HANDLER
                        ********************************
                        *                              *
                        ****    RESET HANDLER    *******
                        *                              *
                        ********************************
C75C  12                DK.RST  NOP
C75D  4F                        CLRA                SET DP...
C75E  1F     8B                 TFR    A,DP         ...TO ZERO
C760  BD   C0B8                 JSR    INITVR
C763  7F   0609                 CLR    ERRMSK       NO ERRORS PLEASE.
C766  7F   0605                 CLR    DD.TMR       TIME OUT DRIVE MOTORS
C769  7F   0613                 CLR    AUTFLG       TURN OFF THE AUTO LINE NUMBERING
C76C  86     35                 LDA    #$35
C76E  B7   FF03                 STA    $FF03        TURN THE #$%&! INTERRUPTS BACK ON
C771  7E   B44F                 JMP    BASRST

                        ********************************
                        *                              *
                        ****    NMI HANDLER    *********
                        *                              *
                        ********************************
C774  96     2F         NMIHLR  LDA    <WD.CSR
C776  5F                        CLRB
C777  1F     9B                 TFR    B,DP
C779  32 6C                     LEAS   12,S
C77B  4D                        TSTA
C77C  39                HOME7   RTS

                        ********************************
                        *                              *
                        ****    IRQ HANDLER    *********
                        *                              *
                        ********************************
C77D  4F                IRQHLR  CLRA
C77E  1F     8B                 TFR    A,DP
C780  0D     F6                 TST    LOCK         NO TIME OUT
C782  26     1A C79E            BNE    IRQEXT
C784  B6   0605                 LDA    DD.TMR       GET TIMER
C787  27     15 C79E            BEQ    IRQEXT       QUIT IF ZERO
C789  4A                        DECA                ()0; COUNT
C78A  B7   0605                 STA    DD.TMR       STASH TIMER
C78D  26     0F C79E            BNE    IRQEXT       QUIT IF ()0
C78F  8D     23 C7B4            BSR    BACKDR
C791  26     0E C7A1            BNE    IRQERR
C793  B6   0607                 LDA    CTLSAV       get drive and motor control
C796  84     00                 ANDA   #MOTOFF      turn motor off
C798  B7   0607                 STA    CTLSAV       save current control
C79B  17   149A DC38            LBSR   SELDRV       output motor off control
C79E  7E   9D3D         IRQEXT  JMP    BASIRQ
C7A1  7E   C718         IRQERR  JMP    NEWERR
                        ********************************
```

```
                        ********************************
C7A4  1F      40        CALCBF  TFR     S,D
C7A6  83    0100                SUBD    #$100
C7A9  93      1F                SUBD    ARRTOP
C7AB  2B      04 C7B1           BMI     HOP29
C7AD  5F                        CLRB
C7AE  4D                        TSTA
C7AF  26      CB C77C           BNE     HOME7
C7B1  7E    8342        HOP29   JMP     OMERR

                        ********************************
                        *
                        * BACKUP THE DIRECTORY TO THE
                        * ALTERNATE DIRECTORY,
                        *
                        * NO CALLING ARGUMENTS
                        *
                        * STACK FRAME
                        *  0008 - SECTOR NUMBER
                        *  0007 - SECTOR NUMBER
                        *  0006 - SECTOR NUMBER
                        *  0005 - SECTOR NUMBER
                        *  0004 - NUMBER OF FULL BUFFERS
                        *  0003 - CURRENT SECTOR NUMBER
                        *  0002 - USERS UNIT NUMBER
                        *  0001 - THE LAST ERROR
                        *  0000 - BIT MASK FOR 'DTYDIR'
                        *
E C7B4                  REORG   SET     *
* 0000                           ORG     0
D 0000                  BD.BIT  RMB     1           DTYDIR BIT MASK
D 0001                  BD.UNT  RMB     1           USERS UNIT #
D 0002                  BD.CBN  RMB     1           CURRENT BUFFER #
D 0003                  BD.CSN  RMB     1           CURRENT SECTOR #
D 0004                  BD.SCT  RMB     4           SECTOR #'S
E 0008                  BD.ZIS  EQU     *
* C7B4                           ORG     REORG

  C7B4  BD    C6F7      BACKDR  JSR     CLEAN4
  C7B7  32 78                   LEAS    -BD.ZIS,S
  C7B9  33 E4                   LEAU    0,S
  C7BB  31 44                   LEAY    BD.SCT,U
  C7BD  8E    0800              LDX     #BUF.01
  C7C0  9F      EE              STX     DD.BUF
  C7C2  6F 42                   CLR     BD.CBN,U    # OF BUFFS USED
  C7C4  D6      EB              LDB     DD.UNT
  C7C6  E7 41                   STB     BD.UNT,U    USERS DRIVE #
  C7C8  C6      01              LDB     #1
  C7CA  E7 C4                   STB     BD.BIT,U    BIT MASK
  C7CC  0F      EB              CLR     DD.UNT

  C7CE  8E    06AA              LDX     #DTYDIR-1
  C7D1  C6      12      LOOP2   LDB     #18         # NUMER SECTORS
  C7D3  E7 43                   STB     BD.CSN,U    STARTING SECTOR
```

```
C7D5  0C     EB              INC   DD.UNT

C7D7  E6 43          LOOP    LDB   BD.CSN,U   GET SECTOR #
C7D9  A6 85                  LDA   B,X         GET TABLE ENTRY
C7DB  A5 C4                  BITA  BD.BIT,U    BIT MASK
C7DD  27     1E C7FD         BEQ   NEX         SKIP THIS SECT
                     *
                     * BACKUP THIS SECTOR
                     *
C7DF  43                     COMA              CLEAR THAT BIT
C7E0  A4 85                  ANDA  B,X
C7E2  A7 85                  STA   B,X
C7E4  6C 42                  INC   BD.CBN,U    # OF BUFFS
C7E6  D7     ED              STB   DD.SEC
C7E8  E7 A0                  STB   ,Y+         SAVE SECTOR
C7EA  C6     14              LDB   #DIRTRK     SET TRACK #
C7EC  D7     EC              STB   DD.TRK
C7EE  BD   C104              JSR   GO.RDS
C7F1  26     23 C816         BNE   HOME77      ?ERROR?
                     *
C7F3  0C     EE              INC   DD.BUF
C7F5  E6 42                  LDB   BD.CBN,U    # OF BUFFS
C7F7  C1     04              CMPB  #4
C7F9  25     02 C7FD         BLO   NEX
                     * DUMP THE BUFFERS
C7FB  8D     1A C817         BSR   DUMPN

C7FD  6A 43          NEX     DEC   BD.CSN,U    SECTOR #
C7FF  26     D6 C7D7         BNE   LOOP

                     * NEXT DRIVE
C801  6D 42                  TST   BD.CBN,U
C803  27     02 C807         BEQ   HOP30
C805  8D     10 C817         BSR   DUMPN
C807  68 C4          HOP30   LSL   BD.BIT,U    BIT MASK
C809  A6 C4                  LDA   BD.BIT,U
C80B  81     08              CMPA  #8
C80D  23     C2 C7D1         BLS   LOOP2

                     * DONE GO AWAY
C80F  A6 41                  LDA   BD.UNT,U    USERS UNIT
C811  97     EB              STA   DD.UNT
C813  32 48                  LEAS  BD.ZIS,U
C815  5F                     CLRB
C816  39             HOME77  RTS

                     ********************************
C817  86     10      DUMPN   LDA   #ALTTRK     GO TO THE
C819  97     EC              STA   DD.TRK      ALTERNATE

C81B  0A     EE      HOP31   DEC   DD.BUF
C81D  31 3F                  LEAY  -1,Y
C81F  A6 A4                  LDA   ,Y          GET THE SECTOR #
C821  97     ED              STA   DD.SEC
C823  BD   C101              JSR   GO.WRV      WRITE IT
```

```
C826  27     03 C82B          BEQ   HOP32
C828  32 62                   LEAS  2,S
C82A  39                      RTS
C82B  6A 42           HOP32   DEC   BD.CBN,U    # OF BUFFERS
C82D  26     EC C81B          BNE   HOP31       NOT DONE YET
C82F  39                      RTS

                      ********************************
                      *                              *
                      ****     FIRQ HANDLER         ****
                      *                              *
                      ********************************
C830  7D     FF26     FIRQHD   TST   NMICNT+1     CLEAR INTERRUPT IF ALL ELSE FAILS
C833  7D     FF24              TST   A.DSKSLT     CLEAR INTERRUPT IF ALL ELSE FAILS
C836  3B                       RTI

                      ********************************
                              END
                              LIB                 LOOKUP
                              NAM                 LOOKUP
```

```
                        ********************************
                        *  FILE "LOOKUP" REV. 2 MAY 83
                        ********************************
                        *
                        *  LOOK UP FILE NAME
                        *
                        *  THIS ROUTINE IS CALLED FIRST
                        *  FOR ANY DISK OPERATION.  IT
                        *  IS CALLED WITH AN ASCII FILE
                        *  NAME, AND IT RETURNS WITH A
                        *  "FIB" NUMBER.  THIS FIB IS
                        *  THEN USED WITH ALL SUBSEQUENT
                        *  OPERATIONS WITH THAT FILE.
                        *
                        *  THE ROUTINE IS COMPOSED OF
                        *  THE ROUTINES "PARSE" AND "SEARCH".

C837  8D     05 C83E LOOKUP  BSR   PARSE
C839  26     73 C8AE         BNE   HOME8
C83B  7E   C8FF              JMP   SEARCH
                        ********************************
```

```
                        ********************************
                        *
                        *  PARSE FILE NAME
                        *
                        *  THIS ROUTINE PARSES THE NAME
                        *  FOR DRIVE, BODY, AND EXTENSION.
                        *  IT CHECKS FOR LEGALITY, AND
                        *  ADDS THE DEFAULT EXTENSION IF
                        *  NECESSARY.
                        *  IT USES THE DEFAULT DRIVE IN
                        *  DEFDRV IF NO DRIVE IS GIVEN.
                        *
                        *  VALID FORMATS ARE:
                        *   "d:filename.ext"
                        *   "filename.ext:d"
                        *   "d:filename"
                        *   "filename:d"
                        *   "filename"
                        *   "filename.ext"
                        *
                        *  CALLED WITH:
                        *    X-) NAME,
                        *    B = LENGTH OF NAME,
                        *    Y-) DEFAULT EXTENSION.
                        *
                        *  RETURN WITH:
                        *    TNAME = NAME (11 BYTES),
                        *    TDRV = DRIVE (1-4),
                        *
                        *  ERROR CODE IN B:
                        *    ERRILN - ILLEGAL NAME

C83E  B6   060A   PARSE   LDA   DEFDRV     DEFAULT DRIVE
C841  B7   065B           STA   TDRV
C844  7F   0660           CLR   TEMP1      NO EXT FOUND
C847  CE   0650           LDU   #TNAME     PUT NAME HERE
C84A  86     07           LDA   #7
C84C  6F C6         HOP33 CLR   A,U
C84E  4A                  DECA
C84F  2A     FB C84C      BPL   HOP33
C851  A6 22               LDA   2,Y        SET DEFAULT EXT
C853  B7   065A           STA   TNAME+10
C856  A6 21               LDA   1,Y
C858  B7   0659           STA   TNAME+9
C85B  A6 A4               LDA   0,Y
C85D  B7   0658           STA   TNAME+8
                        *  ERROR IF NAME LENGTH = 0 OR
                        *  IF )127 (TO AVOID SIGN PROBLEMS)
C860  C1     0E           CMPB  #14        NAME LENGTH
C862  22     48 C8AC      BHI   ILLNAM
C864  5D                  TSTB
C865  27     45 C8AC      BEQ   ILLNAM

                        *  CHECK FOR EXPLICIT DRIVE #
```

```
C867  C1    03            CMPB  #3          SHORTEST W/DRIVE
C869  25    29 C894       BLO   GNNCR       NO ROOM
C86B  C0    02            SUBB  #2
C86D  A6 85               LDA   B,X
C86F  81    3A            CMPA  #':         DRIVE AT END?
C871  26    06 C879       BNE   HOP34
C873  5C                  INCB
C874  A6 85               LDA   B,X
C876  5C                  INCB
C877  20    0A C883       BRA   HOP35
C879  CB    02      HOP34 ADDB  #2
C87B  A6 01               LDA   1,X
C87D  81    3A            CMPA  #':
C87F  26    13 C894       BNE   GNNCR
C881  A6 81               LDA   ,X++
C883  80    30      HOP35 SUBA  #'0
C885 1023   FE8A C713     LBLS  ER.DNE
C889  81    04            CMPA  #4
C88B 1022   FE84 C713     LBHI  ER.DNE
C88F  B7    065B          STA   TDRV
C892  C0    02            SUBB  #2

                    *  GET AND CHECK CHARACTERS
C894  A6 80         GNNCR LDA   ,X+         GET CHARCTER
C896  5A                  DECB              COUNT DOWN LENGTH
C897  2B    64 C8FD       BMI   RTS0        DONE?
C899  81    2F            CMPA  #'/         EXTENSION?
C89B  27    04 C8A1       BEQ   SLASH
C89D  81    2E            CMPA  #'.         ALSO EXTENSION
C89F  26    1C C8BD       BNE   HOP37
C8A1 1183   0650    SLASH CMPU  #TNAME      MAIN PART?
C8A5  27    05 C8AC       BEQ   ILLNAM
C8A7  7D    0660          TST   TEMP1       TWO SLASHES?
C8AA  27    03 C8AF       BEQ   HOP36
C8AC  C6    96    ILLNAM  LDB   #ERR.FS
C8AE  39            HOME8 RTS
C8AF  7C    0660    HOP36 INC   TEMP1       REMEMBER SLASH
C8B2  CE    0658          LDU   #TNAME+8    EXTENSION
C8B5  6F C4               CLR   0,U         REMOVE DEFAULT
C8B7  6F 41               CLR   1,U
C8B9  6F 42               CLR   2,U
C8BB  20    D7 C894       BRA   GNNCR
C8BD  81    41      HOP37 CMPA  #'A
C8BF  25    10 C8D1       BLO   CKNMS       CHECK FOR NUMS
C8C1  81    5A            CMPA  #'Z
C8C3  23    18 C8DD       BLS   VALCHR      LETTERS VALID
C8C5  80    20            SUBA  #'z-'Z      MAKE CAPS
C8C7  81    41            CMPA  #'A
C8C9  25    E1 C8AC       BLO   ILLNAM
C8CB  81    5A            CMPA  #'Z         LOWERCASE VALID
C8CD  23    0E C8DD       BLS   VALCHR
C8CF  20    DB C8AC       BRA   ILLNAM
C8D1  81    2D    CKNMS   CMPA  #'-
C8D3  27    08 C8DD       BEQ   VALCHR      HYPHEN VALID
C8D5  81    30            CMPA  #'0
```

```
C8D7  25    D3 C8AC          BLO   ILLNAM
C8D9  81    39               CMPA  #'9
C8DB  22    CF C8AC          BHI   ILLNAM      BETWEEN 9 & A
                    * TO HERE IF A VALID CHARACTER FOUND:
C8DD  A7 C0         VALCHR   STA   ,U+         PUT INTO TNAME
C8DF 1183   065B             CMPU  #TNAME+11   AT END?
C8E3  26    05 C8EA          BNE   HOP38
C8E5  5D                     TSTB               HOPE INPUT ENDS, TOO
C8E6  26    C4 C8AC          BNE   ILLNAM
C8E8  20    13 C8FD          BRA   RTS0         GOOD, WE'RE DONE
C8EA 1183   0658     HOP38   CMPU  #TNAME+8
C8EE  26    A4 C894          BNE   GNNCR
C8F0  A6 80                  LDA   ,X+
C8F2  5A                     DECB
C8F3  2B    08 C8FD          BMI   RTS0
C8F5  81    2E               CMPA  #'.
C8F7  27    A8 C8A1          BEQ   SLASH
C8F9  81    2F               CMPA  #'/
C8FB  27    A4 C8A1          BEQ   SLASH
C8FD  5F             RTS0    CLRB               NO ERROR
C8FE  39                     RTS
                    ********************************
```

```
                        ********************************
                        *
                        *  SEARCH FOR NAME IN FIBS
                        *
                        *  THE NAME IN TNAME IS SEARCHED
                        *  FOR IN THE FIBS.  IF FOUND,
                        *  THE FIB # IS RETURNED.  ELSE
                        *  A FIB IS CREATED, AND THE
                        *  DISK IS SEARCHED FOR THE
                        *  FILE NAME.  IF NONE IS FOUND,
                        *  THIS IS RETURNED AS A "MINOR"
                        *  ERROR CODE IN B.
                        *
                        *  CALLED WITH:
                        *    TNAME = NAME (11 BYTES),
                        *    TDRV = DRIVE # (1-4).
                        *
                        *  RETURN WITH:
                        *    A = FIB NUMBER,
                        *    X-) READ POINTER (3 BYTES),
                        *
                        *  ERROR CODE IN B:
                        *    ERRNEF - NON-EXISTANT FILE
                        *    ERRMNY - TOO MANY FILES OPEN
                        *    ERRIVD - INVALID DIRECTORY
                        *    ERRTMO, IOE, SNF - I/O ERR/

C8FF   8E    06BD      SEARCH   LDX    #FIBS       FIRST SPOT TO LOOK
C902   0F      F1               CLR    FIBNO       EQUALS FIB 0
C904   FC    0650               LDD    TNAME       FIRST TWO BYTES OF NAME
C907 10A3 84            HOP39    CMPD   0,X
C90A   26      16 C922          BNE    HOP42
C90C   CE    0652               LDU    #TNAME+2
C90F   31 02                    LEAY   2,X         PAST THE FIRST 2
C911   C6      0A               LDB    #11+1-2
C913   A6 C0           HOP40    LDA    ,U+
C915   A1 A0                    CMPA   ,Y+
C917   26      06 C91F          BNE    HOP41
C919   5A                       DECB
C91A   26      F7 C913          BNE    HOP40
C91C   16    00AB C9CA          LBRA   LEAV
C91F   FC    0650       HOP41    LDD    TNAME
C922   30 88   1F       HOP42    LEAX   SIZFIB,X
C925   0C      F1               INC    FIBNO
C927   8C    07F3               CMPX   #ENDFBS
C92A   25      DB C907          BLO    HOP39
                        *      FALL  THRU

                        *  FIB NOT THERE - CREATE FIB
                        *  AND SEARCH DISK DIRECTORY
                        *
C92C   0F      F1               CLR    FIBNO
C92E   8E    06BD               LDX    #FIBS
C931   6D 84           HOP43    TST    0,X
```

```
C933  27    0D C942          BEQ   EMPTY
C935  30 88  1F              LEAX  SIZFIB,X
C938  0C     F1              INC   FIBNO
C93A  8C   07F3              CMPX  #ENDFBS
C93D  25     F2 C931         BLO   HOP43
C93F  C6     A2              LDB   #ERR.TF
C941  39               HOME9 RTS

C942  C6     0C      EMPTY   LDB   #11+1       NAME + DRIVE
C944  1F     12              TFR   X,Y
C946  CE   0650              LDU   #TNAME
C949  A6 C0            HOP44 LDA   ,U+
C94B  A7 A0                  STA   ,Y+
C94D  5A                     DECB
C94E  26     F9 C949         BNE   HOP44
                       *  NOTE A STILL = DRIVE #
C950  97     EB              STA   DD.UNT
C952  CE   061C              LDU   #DRVINF
C955  C6     06              LDB   #DRVISZ
C957  3D                     MUL
C958  33 CB                  LEAU  D,U
C95A  6C 45                  INC   NUMFIL,U    FILES OPEN

C95C  C6     13              LDB   #SIZFIB-FDRIV-1
C95E  6F A0            HOP45 CLR   ,Y+
C960  5A                     DECB
C961  26     FB C95E         BNE   HOP45
C963  86     80              LDA   #$80        FILE DOESN'T EXIST
C965  A7 0F                  STA   FFLAGS,X
C967  7F   0681              CLR   TEMP3

C96A  BD   D1F9              JSR   DKINF       GET DISK INFO
C96D  26     D2 C941         BNE   HOME9       ERROR
C96F 10AE 84                 LDY   DIRLSN,X    DIRECTORY
C972  31 22                  LEAY  2,Y         PAST FREE LIST
C974  86     10              LDA   #16         SECTORS IN DIR
C976  B7   0660              STA   TEMP1       COUNTER
C979 10BF  065C      READDR  STY   SRCLSN      SAVE SECTOR #
C97D  BD   D2C3              JSR   READSB      GET SECTOR
C980  26     BF C941         BNE   HOME9
C982  AE 05                  LDX   PABADR,X
C984  33 89 00FA             LEAU  DIRPS*DIRSIZ,X END
C988  FF   065E              STU   TBFADR
C98B  A6 84            HOP46 LDA   FLAGS,X
C98D  85     81              BITA  #$81        INVALID, EXTTS
C98F  26     1A C9AB         BNE   HOP48
C991  FC   0650              LDD   TNAME
C994 10A3 01                 CMPD  DNAME,X
C997  26     12 C9AB         BNE   HOP48
C999  CE   0652              LDU   #TNAME+2
C99C  31 03                  LEAY  DNAME+2,X
C99E  C6     09              LDB   #9
C9A0  A6 C0            HOP47 LDA   ,U+
C9A2  A1 A0                  CMPA  ,Y+
C9A4  26     05 C9AB         BNE   HOP48
```

```
C9A6  5A                      DECB
C9A7  26    F7 C9A0           BNE    HOP47
C9A9  20    2C C9D7           BRA    FOUNDK
C9AB  A6 84         HOP48     LDA    FLAGS,X
C9AD  85    08                BITA   #8          END OF DIR
C9AF  26    16 C9C7           BNE    XLEAV
C9B1  7C    0681              INC    TEMP3
C9B4  30 88  19               LEAX   DIRSIZ,X
C9B7  BC    065E              CMPX   TBFADR
C9BA  25    CF C98B           BLO    HOP46
C9BC 10BE   065C              LDY    SRCLSN
C9C0  31 21                   LEAY   1,Y
C9C2  7A    0660              DEC    TEMP1       COUNT SECTORS
C9C5  26    B2 C979           BNE    READDR

                    *  LEAVE WITH STATUS
                    *
C9C7  BD    CF29    XLEAV     JSR    FIBADR
C9CA  5F            LEAV      CLRB
C9CB  6D 0F                   TST    FFLAGS,X
C9CD  2A    02 C9D1           BPL    HOP49
C9CF  C6    A0                LDB    #ERR.NE     NON-EXISTANT
C9D1  30 0C         HOP49     LEAX   RDBYTE,X
C9D3  96    F1                LDA    FIBNO
C9D5  5D                      TSTB
C9D6  39                      RTS

                    *  YES, IN DIRECTORY
                    *
C9D7  34    10      FOUNDK    PSHS   X
C9D9  BD   CF29               JSR    FIBADR
C9DC  35    20                PULS   Y
C9DE  B6   0681               LDA    TEMP3
C9E1  A7 88  1D               STA    DIRN,X
C9E4  A6 A4                   LDA    FLAGS,Y
C9E6  A7 0F                   STA    FFLAGS,X
C9E8  EC 2C                   LDD    DXTXT,Y
C9EA  ED 88  15               STD    XTLSN1,X
C9ED  A6 2E                   LDA    DXTXT+2,Y
C9EF  A7 88  17               STA    XTLEN1,X
C9F2  A7 88  19               STA    XTNT2+1,X
C9F5  6F 88  18               CLR    XTNT2,X
C9F8  6F 88  13               CLR    XTNT1,X
C9FB  6F 88  14               CLR    XTNT1+1,X
C9FE  EC 2F                   LDD    DXTXT+3,Y
CA00  ED 88  1A               STD    XTLSN2,X
CA03  A6 A8  11               LDA    DXTXT+5,Y
CA06  A7 88  1C               STA    XTLEN2,X

CA09  CC   FFFF               LDD    #$FFFF      LENGTH UNKNOWN
CA0C  ED 88  10               STD    FILLEN,X
CA0F  A7 88  12               STA    FILLEN+2,X
CA12  20    B6 C9CA           BRA    LEAV

                    ********************************
```

```
                        END
                        LIB             RW
                        NAM             RW
```

```
                        ********************************
                        * REVISED SEPTEMBER 14, 1983
                        ********************************
                        *
                        *  READ FROM FILE
                        *
                        *  CALLED WITH:
                        *    A = FILE NUMBER,
                        *    X-) AREA IN USER RAM,
                        *    Y = # OF BYTES ((FF00),
                        *    U = FILE SECTOR NUMBER,
                        *    B = BYTE NUMBER IN FILE.
                        *
                        *  RETURN WITH:
                        *    X = # BYTES NOT READ (IF
                        *        B = ERREOF)
                        *
                        *  ERROR CODE IN B:
                        *    ERR.PE - READ PAST EOF
                        *             non-fatal
                        *    ERR.IV - INVALID DIRECTORY,
                        *
                        ********************************
                        ********************************
                        *                              *
                        *  SECONDARY ENTRY POINTS:     *
                        *    RWRITE - CALLED BY WRITE, *
                        *    VERFY  - CALLED BY WRITE. *
                        *                              *
                        ********************************

CA14   0F      F5       READ     CLR    RWFLAG
CA16   97      F1               STA    FIBNO
CA18   20      07 CA21          BRA    BOTHRW

CA1A   86      01       RWRITE   LDA    #1
CA1C                            SKIP2
CA1D   86      FF       VERFY    LDA    #$FF
CA1F   97      F5               STA    RWFLAG

CA21 10BF    0661       BOTHRW   STY    RLENG       LENGTH OF R/W
CA25 1027    00DE CB07           LBEQ   READLV      NOTHING IF
CA29   FF    0669               STU    RSECT       STARTING SECTOR
CA2C   F7    0663               STB    WBYTE       STARTING BYTE
                        *
                        *  BUG:  SHOULD CHECK HERE FOR
                        *  READ BEYOND EOF.
                        *
                        *
                        * WE NEED TO UP DATE THE READ
                        * POINTER HERE
                        *
CA2F   34      10               PSHS   X           BUFFER ADDRESS
                        ******
```

```
                        * SET THE CORRECT DRIVE NUMBER
CA31  BD   CF29             JSR   FIBADR
CA34  A6 0B                 LDA   FDRIV,X
CA36  97   EB               STA   DD.UNT

CA38  0D   F5               TST   RWFLAG
CA3A  26   0B CA47          BNE   HOP51
CA3C  FC   0661             LDD   RLENG
CA3F  E3 0D                 ADDD  RDBYTE+1,X
CA41  24   02 CA45          BCC   HOP50
CA43  6C 0C                 INC   RDBYTE,X
CA45  ED 0D           HOP50 STD   RDBYTE+1,X
                        *
CA47  35   10         HOP51 PULS  X           BUFFER ADDRES
CA49  F6   0663             LDB   WBYTE

CA4C  4F              READLP CLRA
CA4D  50                    NEGB              DISTANCE TO END OF SECTOR
CA4E  26   01 CA51          BNE   HOP52
CA50  4C                    INCA              EXTEND TO 16 BIT #
CA51 10B3  0661       HOP52 CMPD  RLENG       )TOTAL LENGTH?
CA55  23   03 CA5A          BLS   HOP53
CA57  FC   0661             LDD   RLENG       DON'T EXCEED THE ACTUAL REQUEST
CA5A  34   16         HOP53 PSHS  D,X
CA5C  FE   0669             LDU   RSECT
CA5F  BD   CB09             JSR   FSNLSN      CONVERT LSN
CA62  26   0E CA72          BNE   RTS4S       ERROR?
CA64  1F   02               TFR   D,Y         LSN

                        *  CHECK FOR FILE PROTECTED
CA66  A6 0F                 LDA   FFLAGS,X
CA68  85   02               BITA  #2          BIT1=PROTECTED
CA6A  27   09 CA75          BEQ   HOP54
CA6C  0D   F5               TST   RWFLAG
CA6E  27   05 CA75          BEQ   HOP54       READ IS OK
CA70  C6   98               LDB   #ERR.PT
CA72  32 64           RTS4S LEAS  4,S
CA74  39                    RTS
CA75  AE 62           HOP54 LDX   2,S
CA77  6D 61                 TST   1,S         MSB LENGTH
CA79  26   2A CAA5          BNE   USEBUF
                        *  READ/WRITE WHOLE SECTOR WITH USER'S BUFFER
CA7B  0D   F5               TST   RWFLAG
CA7D  27   07 CA86          BEQ   HOP55
CA7F  2A   0A CA8B          BPL   HOP56
CA81  BD   D37F             JSR   VLSN        VERIFY SECTOR
CA84  20   08 CA8E          BRA   HOP57
CA86  BD   D386       HOP55 JSR   RLSN        READ SECTOR
CA89  20   03 CA8E          BRA   HOP57
CA8B  BD   D376       HOP56 JSR   WLSN        WRITE SECTOR
CA8E  26   E2 CA72    HOP57 BNE   RTS4S       B () 0?
CA90  6C 62                 INC   2,S         MSB BUFFER
CA92  BE   0669             LDX   RSECT
CA95  30 01                 LEAX  1,X
CA97  BF   0669             STX   RSECT
```

```
CA9A  7A   0661            DEC   RLENG       MSB LENGTH
CA9D  FC   0661            LDD   RLENG
CAA0  35     16            PULS  D,X
CAA2  26     A8 CA4C       BNE   READLP      IF RLENG()0
CAA4  39                   RTS                B=0, .EQ.

CAA5  0D     F5    USEBUF  TST   RWFLAG      SKIP THIS PART
CAA7  2B     41 CAEA       BMI   NXTSCT      ON VERIFY
CAA9  BD   D2C3            JSR   READSB      READ TO A BUFF
CAAC  26     C4 CA72       BNE   RTS4S       LEAVE IF ERROR
CAAE  BF   0667            STX   PAGPLC
CAB1 10AE 62               LDY   2,S         USER AREA
CAB4  F6   0663            LDB   WBYTE
CAB7  0D     F5            TST   RWFLAG
CAB9  27     23 CADE       BEQ   RFBUFF      FROM BUFFER

CABB  86     FF            LDA   #$FF        DIRTY
CABD  A7 02                STA   PAGVLD,X
CABF  AE 05                LDX   PAGADR,X
CAC1  3A                   ABX
                      *
                      * REMOVED BY JP
                      *
                      *     TFR   X,D
                      *     DECB
                      *     BCC   HOP58
CAC2  E6 61           HOP58  LDB   1,S
CAC4  A6 A0           HOP59  LDA   ,Y+
CAC6  A7 80                  STA   ,X+
CAC8  5A                     DECB
CAC9  26     F9 CAC4         BNE   HOP59
CACB  1F     10              TFR   X,D
CACD  5D                     TSTB
CACE  26     1A CAEA         BNE   NXTSCT
CAD0  BE   0667              LDX   PAGPLC
CAD3  34     20              PSHS  Y
CAD5  BD   D330              JSR   CLEANP
CAD8  35     20              PULS  Y
CADA  26     96 CA72         BNE   RTS4S
CADC  20     0C CAEA         BRA   NXTSCT

CADE  AE 05           RFBUFF LDX   PAGADR,X
CAE0  3A                     ABX
CAE1  E6 61                  LDB   1,S         LSB LENGTH
CAE3  A6 80           HOP60  LDA   ,X+
CAE5  A7 A0                  STA   ,Y+
CAE7  5A                     DECB              LSB LENGTH
CAE8  26     F9 CAE3         BNE   HOP60

CAEA  BE   0669       NXTSCT LDX   RSECT
CAED  30 01                  LEAX  1,X
CAEF  BF   0669              STX   RSECT
CAF2  1F     21              TFR   Y,X
CAF4  FC   0661              LDD   RLENG
CAF7  A3 E1                  SUBD  ,S++
```

```
CAF9  FD   0661            STD   RLENG
CAFC  32 62                LEAS  2,S         OLD USER PTR
CAFE  27     07 CB07       BEQ   READLV      IF RLENG()0
CB00  7F   0663            CLR   WBYTE
CB03  5F                   CLRB               BYTE # IN SECTOR
CB04  7E   CA4C            JMP   READLP

CB07  5F             READLV CLRB
CB08  39                    RTS

                    ********************************
```

```
                        ********************************
                        *
                        *  CONVERT FILE SECTOR NUMBER
                        *  TO LOGICAL SECTOR NUMBER.
                        *
                        *  CALLED WITH:
                        *    U = FILE SECTOR # (FSN),
                        *    FIBNO = FIB NUMBER.
                        *
                        *  RETURN WITH:
                        *    D = LSN,
                        *    X-) FIB ENTRY,
                        *    U = FSN (PRESERVED).
                        *
CB09  BD    CF29        FSNLSN  JSR   FIBADR      GET FIB ADDR
CB0C  1F      30        TRY1    TFR   U,D
CB0E  A3 88   13                BSUB  XTNT1,X
CB11  25      0D CB20           BLO   NOT1
CB13  4D                        TSTA
CB14  26      0A CB20           BNE   NOT1
CB16  E1 88   17                CMPB  XTLEN1,X
CB19  24      05 CB20           BHS   NOT1
CB1B  E3 88   15                ADDD  XTLSN1,X
CB1E  20      12 CB32           BRA   FSNEXT

CB20  1F      30        NOT1    TFR   U,D
CB22  A3 88   18                SUBD  XTNT2,X
CB25  25      0E CB35           BLO   NOT2
CB27  4D                        TSTA
CB28  26      0B CB35           BNE   NOT2
CB2A  E1 88   1C                CMPB  XTLEN2,X
CB2D  24      06 CB35           BHS   NOT2
CB2F  E3 88   1A                ADDD  XTLSN2,X
CB32                  FSNEXT    EQ
CB34  39                        RTS

CB35  34      40        NOT2    PSHS  U
CB37  8D      3E CB77           BSR   SCANXT      SCAN EXTENTS
CB39  26      0B CB46           BNE   HOP61
CB3B  FC    066B                LDD   TOTAL       HOW MANY SCANNED
CB3E 10B3   066D                CMPD  TARGET      MADE IT?
CB42  22      04 CB48           BHI   HOP62       GOOD
CB44  C6      9A                LDB   #ERR.PE     PAST EOF
CB46  35      C0        HOP61   PULS  U,PC
CB48  E0 22             HOP62   SUBB  2,Y         LENGTH OF XTNT
CB4A  82      00                SBCA  #0
CB4C  ED 88   13                STD   XTNT1,X     START OF THIS EXTENT
CB4F  A6 22                     LDA   2,Y
CB51  A7 88   17                STA   XTLEN1,X    LENGTH OF EXTENT
CB54  EC A4                     LDD   0,Y
CB56  ED 88   15                STD   XTLSN1,X    LSN OF START OF XTNT
CB59  1F      20                TFR   Y,D
CB5B  34      40                PSHS  U
CB5D  A3 E1                     SUBD  ,S++
```

```
CB5F  35      40              PULS  U
CB61  C1      13              CMPB  #DIRSIZ-6   END OF ENTRY?
CB63  24      10 CB75         BHS   NO2ND       NO MORE EXTENTS
CB65  A6 25                   LDA   5,Y         LENGTH NEXT EXTENT
CB67  A7 88   1C              STA   XTLEN2,X
CB6A  EC 23                   LDD   3,Y         LSN OF NEXT ENTENT
CB6C  ED 88   1A              STD   XTLSN2,X
CB6F  FC    066B              LDD   TOTAL
CB72  ED 88   18              STD   XTNT2,X     THINK ABOUT IT
CB75  20      95 CB0C NO2ND   BRA   TRY1        NOW DO IT
```

****

```
                        ********************************
                        *
                        *  SCAN EXTENTS
                        *
                        *  CALLED WITH:
                        *    X-) FIB,
                        *    U = FSN TO STOP AT.
                        *
                        *  RETURN WITH:
                        *    TOTAL = # SECTORS SCANNED,
                        *    TARGET = ORIGINAL U,
                        *    SCTEXT = LSN OF DIR (OR EXT) ENTRY,
                        *    U-) START OF ENTRY.

CB77  34    10      SCANXT  PSHS   X
CB79  7F    066C            CLR    TOTAL+1
CB7C  7F    066B            CLR    TOTAL          CLEAR TOTAL
CB7F  FF    066D            STU    TARGET
CB82  E6 88  1D             LDB    DIRN,X
CB85  F7    0682            STB    LASDIR
CB88  BD    D250            JSR    GETDIR
CB8B  26      46 CBD3       BNE    RTS2S          ERROR?
CB8D  1F      13            TFR    X,U
CB8F  35      10            PULS   X              GET FIB ADDRESS
CB91  31 4C                 LEAY   DXTXT,U
CB93  C6      04            LDB    #4
CB95  A6 C4           CKNG  LDA    0,U
CB97  84      20            ANDA   #$20           MORE XTNTS?
CB99  27      03 CB9E       BEQ    HOP63
CB9B  A6 C8   18            LDA    DLSCT,U
CB9E  34      06      HOP63 PSHS   A,B
CBA0  FC    066B      CKNXTT LDD   TOTAL
CBA3  EB 22                 ADDB   2,Y
CBA5  89      00            ADCA   #0
CBA7  FD    066B            STD    TOTAL
CBAA 10B3   066D            CMPD   TARGET
CBAE  22      22 CBD2       BHI    FDXT           EXIT WITH B=0
CBB0  31 23                 LEAY   3,Y            TO NEXT EXTENT
CBB2  6A 61                 DEC    1,S            ARE ANY MORE?
CBB4  26      EA CBA0       BNE    CKNXTT
CBB6  E6 E4                 LDB    0,S
CBB8  27      16 CBD0       BEQ    PSTEOF         PAST EOF
CBBA  32 62                 LEAS   2,S
CBBC  F7    0682            STB    LASDIR
CBBF  34      10            PSHS   X
CBC1  BD    D250            JSR    GETDIR
CBC4  1F      13            TFR    X,U
CBC6  35      10            PULS   X
CBC8  26      0B CBD5       BNE    HOME10
CBCA  31 41                 LEAY   1,U
CBCC  C6      07            LDB    #7             EXTENT PER ENTRY
CBCE  20      C5 CB95       BRA    CKNG

CBD0  31 3D           PSTEOF LEAY  -3,Y
```

```
CBD2  5F          FDXT    CLRB
CBD3  32 62       RTS2S   LEAS  2,S
CBD5  39          HOME10  RTS
                  ********************************
```

```
                        ********************************
                        *
                        *  WRITE TO FILE.
                        *
                        *  CALLED WITH:
                        *    A = FILE NUMBER,
                        *    X-) USER RAM AREA,
                        *    U = # BYTES,
                        *    Y-) SECTOR IN FILE,
                        *    B = BYTE IN FILE.
                        *
                        *  ERROR CODE IN B:
                        *    ERRFUL - DISK FULL,
                        *    ERRDRF - DIRECTORY FULL,
                        *    ERRWEF - WRITE PAST EOF,
                        *    ERRIVD - INVALID DIRECTORY,
                        *    ERRPRT - FILE PROTECTED,
                        *    ERRWPT - DISK PROTECTED,
                        *    ERRTMO,IOE,SNF - I/O ERROR.
                        *                              *
                        ********************************

CBD6   97     F1       WRITE    STA    FIBNO
CBD8   BF   0671                STX    BUFAD
CBDB   FF   0673                STU    NMBYTE
CBDE 10BF   0675                STY    FLSCT
CBE2   F7   0677                STB    FLBYT
                        ******
                        * SET CORRECT DRIVE NUMBER
CBE5   BD   CF29                JSR    FIBADR
CBE8   E6 0B                    LDB    FDRIV,X
CBEA   D7     EB                STB    DD.UNT

CBEC   BD   CEF6        WRITE2  JSR    LENFIL
CBEF   27     0E CBFF           BEQ    ISTHR
CBF1   C1     9C                CMPB   #ERR.FF      NOT FOUND
CBF3   27     01 CBF6           BEQ    MAKIT
CBF5   39                       RTS

CBF6   96     F1       MAKIT    LDA    FIBNO
CBF8   BD   CF91                JSR    CREATE
CBFB   26     D8 CBD5           BNE    HOME10
CBFD   20     ED CBEC           BRA    WRITE2

CBFF 11B3   0675        ISTHR   CMPU   FLSCT        U=MSB OF LENGTH OF THE FILE
CC03   22     0A CC0F           BHI    NOTEOF
CC05   25     05 CC0C           BLO    ERWEOF
CC07   B1   0677                CMPA   FLBYT
CC0A   24     03 CC0F           BHS    NOTEOF
CC0C   C6     9A       ERWEOF  LDB    #ERR.PE      WRITE ) EOF
CC0E   39              HOME11   RTS

CC0F   34     02       NOTEOF  PSHS   A
CC11   FC   0673                LDD    NMBYTE       # OF BYTES TO ADD
```

```
CC14  FB   0677            ADDB  FLBYT      BYTE WITHIN THE SECTOR
CC17  B9   0676            ADCA  FLSCT+1    LSB OF FSN
CC1A  34     04            PSHS  B
CC1C  1F     89            TFR   A,B
CC1E  B6   0675            LDA   FLSCT      MSB OF FSN
CC21  89     00            ADCA  #0
CC23  34     40            PSHS  U          MSB OF LENGTH
CC25  A3 E1                SUBD  ,S++
CC27  1F     98            TFR   B,A
CC29  35     04            PULS  B
CC2B  22     08 CC35       BHI   TOEX
CC2D  25     0E CC3D       BLO   NOEX
CC2F  E0 E4                SUBB  ,S
CC31  24     06 CC39       BHS   TOEX2
CC33  20     08 CC3D       BRA   NOEX

CC35  E0 E4          TOEX  SUBB  ,S
CC37  82     00            SBCA  #0
CC39  8D     2E CC69 TOEX2 BSR   EXTEND
CC3B  26     29 CC66       BNE   RTS1S
CC3D  32 61          NOEX  LEAS  1,S

CC3F  F6   0677            LDB   FLBYT
CC42  BE   0671            LDX   BUFAD
CC45 10BE  0673            LDY   NMBYTE
CC49  FE   0675            LDU   FLSCT
CC4C  BD   CA1A            JSR   RWRITE
CC4F  26     BD CC0E       BNE   HOME11
CC51  7D   0608            TST   VERFLG
CC54  27     B8 CC0E       BEQ   HOME11
CC56  F6   0677            LDB   FLBYT
CC59  BE   0671            LDX   BUFAD
CC5C 10BE  0673            LDY   NMBYTE
CC60  FE   0675            LDU   FLSCT
CC63  16   FDB7 CA1D       LBRA  VERFY
CC66  32 61          RTS1S LEAS  1,S
CC68  39                   RTS
                   ********************************
```

```
                        ********************************
                        *
                        *  EXTEND FILE
                        *  (CALLED BY WRITE)
                        *
                        *  CALLED WITH:
                        *    FIBNO = FILE NUMBER,
                        *    DD.UNT = DRIVE #,
                        *    D = # OF BYTES TO ADD TO FILE.
                        *
                        *  ERROR CODE IN B:
                        *    ERRFNF - FILE NOT FOUND,
                        *    ERRFUL - DISK FULL,
                        *    ERRDRF - DIRECTORY FULL,
                        *    ERRIVD - INVALID DIRECTORY,
                        *    ERRPRT - FILE PROTECTED,
                        *    ERRWPT - DISK PROTECTED,
                        *    ERRTMO,IOE,SNF - I/O ERROR.
                        *
                        *  THE DATA ON DISK IS NOT PRESET.
                        *
                        *  PROCEDURE:
                        *  THE NUMBER OF BYTES WHICH CAN
                        *  BE ADDED TO THE LAST SECTOR
                        *  ALREADY ALLOCATED ARE DEDUCTED.
                        *  GET FREE EXTENT (GFSECT).
                        *  IF CONTIGUOUS, ADD TO FILE'S
                        *  EXTENT; ELSE MAKE NEW ONE.
                        *  DECREASE FREE EXTENT BY # OF
                        *  SECTORS NEEDED (OR ALL).
                        *  IF MORE SECTORS NEEDED, LOOP.
                        *  FIGURE NUMBER BYTES USED IN
                        *  LAST SECTOR AND STORE IN DIR.
                        *
                        *  UPDATE LENFIL AND LSDIRB

E CC69                  REORG   SET     *
* 0000                           ORG     0
D 0000                  NUMXTD  RMB     2           BYTES TO EXTEND
D 0002                  YTMP    RMB     2           EXTENT POINTER
D 0004                  LSENT   RMB     2           ENTRY POINTER
D 0006                  WANT    RMB     2           BEST LSN HOPE
D 0008                  DKPTR   RMB     2           PTR TO DRIVE TABLE
D 000A                  SAVEXD  RMB     2
E 000C                  FRAMSZ  EQU     *           SIZE OF THIS FRAME
* CC69                           ORG     REORG

  CC69  32 74           EXTEND  LEAS    -FRAMSZ,S
  CC6B  ED E4                   STD     NUMXTD,S
  CC6D  ED 6A                   STD     SAVEXD,S
  CC6F  86      01              LDA     #1
  CC71  97      F6              STA     LOCK

  CC73  BD    D1F9              JSR     DKINF
```

```
CC76 1026  00E5 CD5F        LBNE   RTSFS
CC7A   AF 68                STX    DKPTR,S
CC7C   BD  CF29             JSR    FIBADR
CC7F   E6 88   1E           LDB    LSDIRB,X
CC82   BD  D250             JSR    GETDIR      GET DIR ENTRY
CC85 1026  00D6 CD5F        LBNE   RTSFS       ERROR EXIT
CC89   AF 64                STX    LSENT,S     SAVE POINTER
CC8B   FE  067F             LDU    DIRBFP
CC8E   86    FF             LDA    #-1
CC90   A7 42                STA    PAGVLD,U    MARK DIRTY
CC92   4F                   CLRA
CC93   E6 88   18           LDB    DLSCT,X     NUMBER OF BYTES IN LAST SECT
CC96   26    01 CC99        BNE    HOP64
CC98   4C                   INCA
CC99   E3 E4          HOP64 ADDD   NUMXTD,S
CC9B   5D                   TSTB
CC9C   26    01 CC9F        BNE    HOP65
CC9E   4A                   DECA
CC9F   ED E4          HOP65 STD    NUMXTD,S
CCA1   4D                   TSTA
CCA2 1027  00A1 CD47        LBEQ   EXLV
CCA6   C6    04             LDB    #4          EXTENTS IN DIR ENTRY
CCA8   31 0C                LEAY   DXTXT,X     FIRST XTNT
CCAA   A6 84                LDA    FLAGS,X
CCAC   85    01             BITA   #1          IS IT EXTENTS?
CCAE   27    04 CCB4        BEQ    HOP66
CCB0   31 01                LEAY   1,X         FIRST XTXT
CCB2   C6    07             LDB    #7          EXTENTS PER ENTRY
CCB4   6D 22          HOP66 TST    2,Y         LENGTH EXTENT
CCB6   26    14 CCCC        BNE    FNLSX       GO FIND LAST 1
CCB8   CC  FFFF             LDD    #$FFFF      NEW FILE
CCBB   ED 66                STD    WANT,S      NO WANTED LSN
CCBD   31 3D                LEAY   -3,Y
CCBF 10AF 62                STY    YTMP,S
CCC2   AE 68                LDX    DKPTR,S
CCC4   6D 02                TST    FRELEN,X    HAVE AN EXTENT ALREADY?
CCC6 1026  009B CD65        LBNE   NXTXTT      YES GO ADD IT
CCCA   20    3E CD0A        BRA    GNXT        NO, GO GET ONE

CCCC   6D 22          FNLSX TST    2,Y
CCCE   27    05 CCD5        BEQ    HOP67
CCD0   31 23                LEAY   3,Y
CCD2   5A                   DECB
CCD3   26    F7 CCCC        BNE    FNLSX
CCD5   31 3D          HOP67 LEAY   -3,Y
CCD7   EC A4                LDD    0,Y
CCD9   EB 22                ADDB   2,Y
CCDB   89    00             ADCA   #0
CCDD   ED 66                STD    WANT,S      WHERE THE NEXT LSN WOULD BE NICE
CCDF 10AF 62                STY    YTMP,S
CCE2   AE 68                LDX    DKPTR,S
CCE4   6D 02                TST    FRELEN,X
CCE6   27    22 CD0A        BEQ    GNXT        NEED NEW EXTENT

CCE8   EC 66                LDD    WANT,S
```

```
CCEA  A3 03                    SUBD  FREXT,X     IS IT THERE?
CCEC  27     2C CD1A           BEQ   MKNXT
CCEE  BD   CDAE                JSR   GFEXNT      GET FREE XTNT
CCF1  27     06 CCF9           BEQ   REPLC
CCF3  C1     94                CMPB  #ERR.DF
CCF5  26     68 CD5F           BNE   RTSFS       ERROR EXIT
CCF7  20     6C CD65           BRA   NXTXTT      USE WHAT WE HAVE

                        *  REPLACE THE OLD EXTENT
CCF9  AE 68            REPLC   LDX   DKPTR,S
CCFB  34     42                PSHS  U,A
CCFD  E6 02                    LDB   FRELEN,X
CCFF  AE 03                    LDX   FREXT,X
CD01  BD   D08F                JSR   DELEXT      DELETE EXTENT
CD04  35     42                PULS  U,A
CD06  26     57 CD5F           BNE   RTSFS       ERROR EXIT
CD08  20     05 CD0F           BRA   INSTEX

CD0A  BD   CDAE      GNXT      JSR   GFEXNT
CD0D  26     50 CD5F           BNE   RTSFS
CD0F  AE 68            INSTEX  LDX   DKPTR,S
CD11  EF 03                    STU   FREXT,X
CD13  A7 02                    STA   FRELEN,X
CD15 11A3 66                   CMPU  WANT,S
CD18  26     4B CD65           BNE   NXTXTT      NEW EXTENT

CD1A  A6 E4            MKNXT   LDA   NUMXTD,S
CD1C  A1 02                    CMPA  FRELEN,X
CD1E  25     02 CD22           BLO   HOP68
CD20  A6 02                    LDA   FRELEN,X
CD22 10AE 62           HOP68   LDY   YTMP,S
CD25  34     02                PSHS  A
CD27  AB 22                    ADDA  2,Y
CD29  24     04 CD2F           BCC   HOP69
CD2B  35     02                PULS  A
CD2D  20     36 CD65           BRA   NXTXTT
CD2F  A7 22            HOP69   STA   2,Y
CD31  A6 02                    LDA   FRELEN,X
CD33  A0 E4                    SUBA  0,S
CD35  A7 02                    STA   FRELEN,X
CD37  EC 03                    LDD   FREXT,X
CD39  EB E4                    ADDB  0,S
CD3B  89     00                ADCA  #0
CD3D  ED 03                    STD   FREXT,X
CD3F  A6 61                    LDA   NUMXTD+1,S  (BECAUSE OF PSHS A)
CD41  A0 E0                    SUBA  ,S+
CD43  A7 E4                    STA   NUMXTD,S
CD45  26     C3 CD0A           BNE   GNXT

CD47  AE 64            EXLV    LDX   LSENT,S
CD49  1F     13                TFR   X,U
CD4B  BD   CF29                JSR   FIBADR
CD4E  EC 6A                    LDD   SAVEXD,S
CD50  E3 88  11                ADDD  FILLEN+1,X
CD53  24     03 CD58           BCC   HOP70
```

```
CD55   6C 88   10             INC    FILLEN,X
CD58   ED 88   11     HOP70   STD    FILLEN+1,X
CD5B   E7 C8   18             STB    DLSCT,U
CD5E   5F                     CLRB
CD5F   32 6C          RTSFS   LEAS   FRAMSZ,S
CD61   0F      F6             CLR    LOCK
CD63   5D                     TSTB
CD64   39                     RTS

CD65   EC 62          NXTXTT  LDD    YTMP,S
CD67   1F      02             TFR    D,Y
CD69   A3 64                  SUBD   LSENT,S
CD6B   C1      13             CMPB   #$13          ROOM FOR ANOTHER?
CD6D   24      0B CD7A        BHS    NEWDIR
CD6F   31 23                  LEAY   3,Y
CD71 10AF 62                  STY    YTMP,S
CD74   EC 03          PUTXT   LDD    FREXT,X
CD76   ED A4                  STD    0,Y
CD78   20      A0 CD1A        BRA    MKNXT

CD7A   86      01     NEWDIR  LDA    #1            EXTENT-TYPE ENTRY
CD7C   BD    D178             JSR    FEMPDR        FIND EMPTY
CD7F   26      DE CD5F        BNE    RTSFS
CD81 10AE 64                  LDY    LSENT,S
CD84   A7 A8   18             STA    DLSCT,Y
CD87   E6 A4                  LDB    FLAGS,Y
CD89   CA      20             ORB    #$20
CD8B   E7 A4                  STB    FLAGS,Y

CD8D   C6      FF             LDB    #$FF
CD8F   E7 02                  STB    PABVLD,X      NEW DIR DIRTY
CD91   C6      01             LDB    #1
CD93   E7 C4                  STB    FLAGS,U
CD95   EF 64                  STU    LSENT,S
CD97   BD    CF29             JSR    FIBADR
CD9A   A7 88   1E             STA    LSDIRB,X
CD9D   31 C8   19             LEAY   DIRSIZ,U
CDA0   C6      18             LDB    #DIRSIZ-1
CDA2   6F A2          HOP71   CLR    ,-Y
CDA4   5A                     DECB
CDA5   26      FB CDA2        BNE    HOP71
CDA7 10AF 62                  STY    YTMP,S
CDAA   AE 68                  LDX    DKPTR,S
CDAC   20      C6 CD74        BRA    PUTXT
```

****

```
                        ********************************
                        *
                        *  GET FREE EXTENT
                        *
                        *  THIS MUST BE CALLED ONLY BY
                        *  EXTEND AND ALLOCATES ONE
                        *  EXTENT FROM THE FREE LIST.
                        *
                        *  IN ORDER OF PREFERENCE, IT
                        *  WILL FIND A SECTOR:
                        *    AT THE LSN SPECIFIED;
                        *    AT START OF PAIR OF TRACKS;
                        *    OUTSIDE OF DIRECTORY;
                        *    AT START OF TRACK OUTSIDE
                        *    OF DIRECTORY;
                        *    AT START OF TRACK INSIDE
                        *    DIRECTORY;
                        *    ANY FREE SECTOR AT START OF
                        *    A CONTIGUOUS BLOCK.
                        *
                        *  CALLED WITH:
                        *    WANT+2,S = LSN PREFERED,
                        *             OR $FFFF;
                        *    DD.UNT = DRIVE #.
                        *
                        *  RETURN WITH:
                        *    U = LSN,
                        *    A = LENGTH OF EXTENT.
                        *
                        *  ERROR CODE IN B:
                        *    ERRFUL - DISK FULL.
                        *    ERRIVD - INVALID DIRECTORY,
                        *    ERRTMO,IOE,SNF - I/O ERROR.
                        *
E CDAE                  REORG   SET     *
* 0000                           ORG     0
D 0000                  GFTRY   RMB     2
D 0002                  GFPBF   RMB     2
D 0004                  GFLG    RMB     1
D 0005                  TMPLSN  RMB     2
D 0007                  TMPCNT  RMB     1
D 0008                  BTCNT   RMB     1
E 0009                  GFSSZ   EQU     *
* CDAE                           ORG     REORG

  CDAE 10FF    0601      GFEXNT  STS     TEMPS
  CDB2   32 77                   LEAS    -GFSSZ,S
  CDB4   6F 64                   CLR     GFLG,S
  CDB6   BD    D1F9              JSR     DKINF
  CDB9 1026    00F4 CEB1         LBNE    GFRTS
                        *  REMOVED:  PAGVLD,X:=-1
  CDBD 10AE 84                   LDY     DIRLSN,X
  CDC0   EE E8   11              LDU     WANT+GFSSZ+2,S
  CDC3   EF 65                   STU     TMPLSN,S
```

```
CDC5  30 41                    LEAX  1,U
CDC7  27     0F CDD8           BEQ   SERCIN
CDC9  8D     3E CE09           BSR   XGETB      =GETBIT
CDCB  26     39 CE06           BNE   TGOTIT     =GOTIT
                        *  THE FOLLOWING 2 LINES ADDED:
CDCD 1183   05A0               CMPU  #1440
CDD1  25     05 CDD8           BLO   SERCIN
CDD3  CE    058E               LDU   #1440-18
CDD6  20     21 CDF9           BRA   SEROUT

CDD8  33 A4           SERCIN   LEAU  0,Y        GET DIR LSN
CDDA  86     FF       HOP72    LDA   #-1
CDDC  A7 E4                    STA   GFTRY,S
CDDE  33 C8   EE      HOP73    LEAU  -18,U
CDE1  EF 7E                    STU   -2,S        TSTU
CDE3  2B     0C CDF1           BMI   SERCOT
CDE5  8D     22 CE09           BSR   XGETB      =GETBIT
CDE7  27     F1 CDDA           BEQ   HOP72
CDE9  6D E4                    TST   GFTRY,S
CDEB  2A     19 CE06           BPL   TGOTIT     =GOTIT
CDED  EF E4                    STU   GFTRY,S
CDEF  20     ED CDDE           BRA   HOP73

CDF1  EE E4           SERCOT   LDU   GFTRY,S
CDF3  EF 7E                    STU   -2,S        TSTU
CDF5  2A     0F CE06           BPL   TGOTIT
CDF7  33 A4                    LEAU  0,Y         GET DIR LSN
CDF9  33 C8   12      SEROUT   LEAU  18,U
CDFC 1183   0B40               CMPU  #2880
CE00  22     0A CE0C           BHI   SERCAL
CE02  8D     05 CE09           BSR   XGETB
CE04  27     F3 CDF9           BEQ   SEROUT
CE06  16    0082 CE8B TGOTIT   LBRA  GOTIT

CE09  16    00AB CEB7 XGETB    LBRA  GETBIT      ISLAND

CE0C  86     FF       SERCAL   LDA   #$FF        FLAG VALUE
CE0E  A7 67                    STA   TMPCNT,S
CE10  A6 64                    LDA   GFLG,S      WHICH BUF?
CE12  DE     8A       SCIT     LDU   DBLZER      =0
CE14  44                       LSRA               BOTTOM BUF?
CE15  25     03 CE1A           BCS   HOP74       YES
CE17  CE    05A0               LDU   #1440       OTHER BUF
CE1A  8D     ED CE09 HOP74     BSR   XGETB       GET TABLE

CE1C  C6     B4                LDB   #180
CE1E  A6 80           HOP75    LDA   ,X+
CE20  26     07 CE29           BNE   SMTHG
CE22  33 48                    LEAU  8,U
CE24  5A                       DECB
CE25  26     F7 CE1E           BNE   HOP75
CE27  20     41 CE6A           BRA   SCITB

CE29  5F              SMTHG    CLRB
CE2A  E7 68                    STB   BTCNT,S
```

```
CE2C  34      40             PSHS  U
CE2E  5C              SELSE  INCB
CE2F  A6 80                  LDA   ,X+
CE31  4C                    INCA
CE32  27      FA CE2E        BEQ   SELSE
CE34  E1 6A                  CMPB  BTCNT+2,S
CE36  23      04 CE3C        BLS   HOP76
CE38  E7 6A                  STB   BTCNT+2,S
CE3A  EF E4                  STU   0,S
CE3C  30 1F           HOP76  LEAX  -1,X
CE3E  86      08             LDA   #8
CE40  3D                     MUL
CE41  33 CB                  LEAU  D,U

CE43  5F              GMORE  CLRB
CE44  A6 80                  LDA   ,X+
CE46  26      E6 CE2E        BNE   SELSE
CE48  33 48                  LEAU  8,U
CE4A  1F      10             TFR   X,D
CE4C  C1      B4             CMPB  #180
CE4E  25      F3 CE43        BLO   GMORE

CE50  35      40             PULS  U          LSN OF 8 GROUP
CE52  8D      63 CEB7        BSR   GETBIT
CE54  33 48                  LEAU  8,U
CE56  A6 84                  LDA   0,X        GET 8 BITS
CE58  E6 68                  LDB   BTCNT,S
CE5A  C1      01             CMPB  #1
CE5C  22      05 CE63        BHI   HOP78
CE5E  33 5F           HOP77  LEAU  -1,U
CE60  48                     LSLA
CE61  24      FB CE5E        BCC   HOP77
CE63  48              HOP78  LSLA             CHECK 1 SECTOR
CE64  24      04 CE6A        BCC   SCITB      IF NG, DONE
CE66  33 5F                  LEAU  -1,U       BACK ONE
CE68  20      F9 CE63        BRA   HOP78      FIND NG SECTOR
CE6A  C1      02      SCITB  CMPB  #2         ARBITRARY NO.
CE6C  24      1D CE8B        BHS   GOTIT      ENOUGH
CE6E  E1 67                  CMPB  TMPCNT,S
CE70  22      19 CE8B        BHI   GOTIT
CE72  A6 67                  LDA   TMPCNT,S
CE74  4C                     INCA             =$FF?
CE75  26      0A CE81        BNE   HOP79
CE77  E7 67                  STB   TMPCNT,S   TRY FOR BETTER
CE79  EF 65                  STU   TMPLSN,S   LSN
CE7B  A6 64                  LDA   GFLG,S     WHICH BUF?
CE7D  88      03             EORA  #3         OTHER ONE
CE7F  20      91 CE12        BRA   SCIT       SCAN OTHER
CE81  EE 65           HOP79  LDU   TMPLSN,S   NO, OTHER U
CE83  8D      32 CEB7        BSR   GETBIT     ANY BEFORE
CE85  26      04 CE8B        BNE   GOTIT      YES
CE87  C6      94             LDB   #ERR.DF    NO: ERROR
CE89  20      26 CEB1        BRA   GFRTS

CE8B  8D      2A CEB7 GOTIT  BSR   GETBIT
```

```
CE8D   5F                    CLRB
CE8E   5C             GOTITL INCB
CE8F   27      17 CEA8       BEQ    OVFG
CE91   34      02            PSHS   A
CE93   43                    COMA
CE94   A4 84                 ANDA   0,X
CE96   A7 84                 STA    0,X
CE98   35      02            PULS   A
CE9A   48                    LSLA
CE9B   24      04 CEA1       BCC    GOTL2
CE9D   86      01            LDA    #1
CE9F   30 01                 LEAX   1,X
CEA1   A5 84          GOTL2  BITA   0,X
CEA3   26      E9 CE8E       BNE    GOTITL
CEA5   1F      98            TFR    B,A
CEA7                         SKIP2
CEA8   86      FF     OVFG   LDA    #$FF
CEAA   AE 62                 LDX    GFPBF,S
CEAC   C6      FF            LDB    #-1
CEAE   E7 02                 STB    PAGVLD,X
CEB0   5F                    CLRB
CEB1 10FE    0601     GFRTS  LDS    TEMPS
CEB5   5D                    TSTB             SET CC FOR ERROR
CEB6   39                    RTS

                      ********************************
```

```
                        ********************************
                        * GET BYTE NUMBER AND BIT MASK
                        * FOR A LSN.
                        *
                        * THE APPROPRIATE SECTOR IS READ
                        * IN IF NOT ALREADY PRESENT.
                        *
                        * ON ERROR, RETURNS FROM THE
                        * CALLING PROGRAM.
                        *
                        * CALL WITH:
                        *  U = LSN
                        *  Y = DIR LSN
                        *
                        * RETURN WITH:
                        *  A = BIT MASK
                        *  X = BYTE WITH IN TABLE
                        *  U = PRESERVED.
                        *  Y = PRESERVED
                        *
CEB7   34      60       GETBIT   PSHS   U,Y
CEB9   86      01                LDA    #1          BIT FOR 1ST SECT
CEBB 1183    05A0                CMPU   #1440
CEBF   25      07 CEC8           BLO    HOP80
CEC1   31 21                     LEAY   1,Y         2ND SAT TABLE
CEC3   33 C9 FA60                LEAU   -1440,U
CEC7   48                        LSLA               BIT FOR 2ND SECT
CEC8   34      40       HOP80    PSHS   U
CECA   A5 6C                     BITA   GFLG+8,S    ALREADY?
CECC   26      0D CEDB           BNE    ALRDY
CECE   A7 6C                     STA    GFLG+8,S
CED0   BD    D2C3                JSR    READSB
CED3   26      DC CEB1           BNE    GFRTS
CED5   AF 6A                     STX    GFPBF+8,S
CED7   86      FF                LDA    #-1
CED9   A7 02                     STA    PAGVLD,X
CEDB   AE 6A             ALRDY   LDX    GFPBF+8,S
CEDD   AE 05                     LDX    PAGADR,X
CEDF   35      06                PULS   A,B         GET LSN
CEE1                             LSRD
CEE3                             LSRD
CEE5                             LSRD
CEE7   A6 63                     LDA    3,S
CEE9   84      07                ANDA   #7
CEEB   CE    A672                LDU    #BITAB+7
CEEE   40                        NEGA
CEEF   A6 C6                     LDA    A,U
CEF1   3A                        ABX
CEF2   A5 84                     BITA   0,X
CEF4   35      E0                PULS   Y,U,PC
                        ********************************
                                 END

                                 LIB                MISC
```

```
                    NAM             MISC
```

```
                        ********************************
                        *
                        *  LENGTH OF FILE
                        *
                        *  CALLED WITH:
                        *    A = FILE NUMBER
                        *
                        *  RETURN WITH:
                        *    LENGTH IN UHI : ULO : A
                        *
                        *  ERROR CODE IN B:
                        *    ERRFNF - FILE NOT FOUND,
                        *    ERRIVD - INVALID DIRECTORY,
                        *    ERRTMO,IDE,SNF - I/O ERROR.

CEF6  97     F1      LENFIL  STA   FIBNO
CEF8  8D     2F CF29         BSR   FIBADR
CEFA  6D 0F                  TST   FFLAGS,X
CEFC  2A     03 CF01         BPL   HOP81
CEFE  C6     9C              LDB   #ERR.FF
CF00  39             HOME12  RTS
CF01  A6 88  12      HOP81   LDA   FILLEN+2,X
CF04  EE 88  10              LDU   FILLEN,X
CF07  31 41                  LEAY  1,U          U=$FFFF?
CF09  26     1C CF27         BNE   LVOKL
                        * NOTE: U MUST STILL BE $FFFF HERE SO
                        * THAT SCANXT WON'T STOP TOO SOON.
CF0B  BD   CB77              JSR   SCANXT       SCAN EXTENTS
CF0E  26     F0 CF00         BNE   HOME12       ?ERROR?
CF10  F6   0682              LDB   LASDIR
CF13  E7 88  1E              STB   LSDIRB,X
CF16  A6 C8  18      NUMFIX  LDA   DLSCT,U
CF19  A7 88  12              STA   FILLEN+2,X
CF1C  FE   066B              LDU   TOTAL
CF1F  4D                     TSTA
CF20  27     02 CF24         BEQ   HOP82
CF22  33 5F                  LEAU  -1,U
CF24  EF 88  10      HOP82   STU   FILLEN,X
CF27  5F             LVOKL   CLRB
CF28  39                     RTS
                        ********************************
```

```
                        ********************************
                        *
                        *  GET FIB ADDRESS
                        *
                        *  CALLED WITH:
                        *    FIBNO = FIB NUMBER.
                        *
                        *  RETURN WITH:
                        *    X = FIB ADDRESS,
                        *    D, U, Y UNCHANGED.
                        *
                        *  NO ERROR CONDITIONS.

CF29   34      06       FIBADR   PSHS   D
CF2B   96      F1                LDA    FIBNO
CF2D   C6      1F                LDB    #SIZFIB
CF2F   3D                        MUL
CF30   1F      01                TFR    D,X
CF32   30 89 06BD                LEAX   FIBS,X
CF36   35      86                PULS   D,PC
                        ********************************
```

```
                        ********************************
                        *
                        *  CLOSE ALL FILES AND CLEAN UP
                        *  BUFFERS FOR ONE DRIVE.
                        *
                        *  THIS SHOULD BE CALLED FOUR
                        *  TIMES WHENEVER BASIC RETURNS
                        *  TO THE "OK" PROMPT.
                        *
                        *  CALLED WITH:
                        *    DD.UNT = DRIVE # (1-4).
                        *
                        *  ERROR CODE IN B:
                        *    ERRWPT - DISK PROTECTED,
                        *    ERRTMO,IOE,SNF - I/O ERROR.
                        *
CF38  86      0A        CLOSAL  LDA     #10
CF3A  97      F1                STA     FIBND
CF3C  8D      EB CF29   HOP83   BSR     FIBADR
CF3E  A6 0B                     LDA     FDRIV,X
CF40  91      EB                CMPA    DD.UNT
CF42  26      04 CF48           BNE     HOP84
CF44  8D      0A CF50           BSR     CLOS1
CF46  26      B8 CF00           BNE     HOME12
CF48  0A      F1        HOP84   DEC     FIBND
CF4A  2A      F0 CF3C           BPL     HOP83
CF4C  5F                CEXIT   CLRB
CF4D  39                        RTS
                        ********************************
```

```
                        ********************************
                        *
                        *  CLOSE ONE FILE.
                        *
                        *  CALLED WITH:
                        *    A = FILE NUMBER.
                        *
                        *  ERROR CODE IN B:
                        *    ERRWPT - DISK PROTECTED,
                        *    ERRTMO,IOE,SNF - I/O ERROR.
                        *
                        *  IF THIS WAS THE LAST FILE
                        *  OPEN FOR THAT DRIVE, IT WILL
                        *  CALL CLEANP FOR ALL PAGES
                        *  ASSOCIATED WITH THAT DRIVE
                        *  AND THEN INVALIDATE THEM;
                        *  THE DIRECTORY SECTORS FOR
                        *  THAT DRIVE ARE BACKED UP.

CF4E  97     F1      CLOSE1  STA   FIBNO
CF50  8D     D7 CF29 CLOS1   BSR   FIBADR
CF52  6D 84                  TST   0,X
CF54  27     F6 CF4C         BEQ   CEXIT
CF56  96     EB              LDA   DD.UNT
CF58  34     02              PSHS  A
CF5A  A6 0B                  LDA   FDRIV,X
CF5C  97     EB              STA   DD.UNT
CF5E  6F 84                  CLR   0,X          MARK FIB UNUSED
CF60  BD    D1F9             JSR   DKINF
CF63  26     26 CF8B         BNE   CLOSX
CF65  6D 05                  TST   NUMFIL,X
CF67  27     07 CF70         BEQ   HOP85
CF69  6A 05                  DEC   NUMFIL,X
CF6B  27     03 CF70         BEQ   HOP85
CF6D  5F                     CLRB
CF6E  20     1B CF8B         BRA   CLOSX
                        * CLEAR UP THE DRIVE
CF70  E6 02             HOP85   LDB   FRELEN,X
CF72  27     0D CF81         BEQ   HOP86
CF74  34     10              PSHS  X
CF76  AE 03                  LDX   FREXT,X
CF78  BD    D08F             JSR   DELEXT
CF7B  35     10              PULS  X
CF7D  26     0C CF8B         BNE   CLOSX
CF7F  6F 02                  CLR   FRELEN,X
CF81  BD    C7B4     HOP86   JSR   BACKDR       BACKUP DIR

                        * INVALIDATE THE DRIVE TABLE
                        *  SO THAT THE USER CAN SWAP
                        *  DISKETTES NOW
CF84  CE    0696             LDU   #LUVALD
CF87  96     EB              LDA   DD.UNT
CF89  6F C6                  CLR   A,U          VALID FLAG
```

```
CF8B  35    02    CLOSX   PULS  A
CF8D  97    EB            STA   DD.UNT
CF8F  5D                  TSTB
CF90  39          HOME13  RTS
                  ********************************
```

```
                        ********************************
                        *
                        *  CREATE NEW FILE.
                        *
                        *  IF A FILE BY THAT NAME EXISTS
                        *  ON THE DRIVE ALREADY, IT IS
                        *  RENAMED WITH A ".BAK" EXTENSION.
                        *  IF THERE IS ALREADY A ".BAK"
                        *  FILE, IT IS DELETED.
                        *  "RCREAT" IS CALLED FOR THE
                        *  ACTUAL CREATION.
                        *
                        *  CALLED WITH:
                        *    A = FILE NUMBER.
                        *
                        *  RETURN WITH:
                        *    B = ERROR CODE.
                        *
                        *  ERROR CONDITIONS:
                        *    ERRMNY - TOO MANY FILES
                        *      WHEN .BAK IS OPENED.
                        *    ERRPRT - FILE OR .BAK FILE
                        *      IS PROTECTED.
                        *    ERRWPT - DISK PROTECTED.
                        *    ERRDRF - DIRECTORY FULL.
                        *    ERRIVD - INVALID DIRECTORY,
                        *    ERRTMO,IOE,SNF - I/O ERROR.

CF91  B7   067D     CREATE  STA   TFIBN      SAVE FIB #
CF94  BD   CF29             JSR   FIBADR
CF97  BF   0678             STX   SAVFIB
CF9A  C6     0C             LDB   #8+3+1     NAME, DRIVE
CF9C  CE   0650             LDU   #TNAME
CF9F  A6 80         HOP87   LDA   ,X+
CFA1  A7 C0                 STA   ,U+
CFA3  5A                    DECB
CFA4  26     F9 CF9F        BNE   HOP87
CFA6  EC 5C                 LDD   -4,U       EXTENSION
CFA8  FD   067A             STD   SAVEXT
CFAB  A6 5E                 LDA   -2,U
CFAD  B7   067C             STA   SAVEXT+2
CFB0  CC   4241             LDD   #$4241
CFB3  ED 5C                 STD   -4,U
CFB5  86     4B             LDA   #'K
CFB7  A7 5E                 STA   -2,U
CFB9  6D 0F                 TST   FFLAGS,X
CFBB  2B     29 CFE6        BMI   CRENEW
CFBD  BD   C8FF             JSR   SEARCH
CFC0  C1     A0             CMPB  #ERR.NE
CFC2  27     0B CFCC        BEQ   RENIT
CFC4  5D                    TSTB
CFC5  26     C9 CF90        BNE   HOME13     ERROR?
CFC7  BD   D033             JSR   DELETE     A=FIBNO
CFCA  26     C4 CF90        BNE   HOME13     ERROR?
```

```
CFCC  6D 9F 0678      RENIT   TST   [SAVFIB]
CFD0  27     14 CFE6          BEQ   CRENEW
CFD2  B6   067D               LDA   TFIBN
CFD5  BD   D14B               JSR   RNAME
CFD8  27     04 CFDE          BEQ   HOP88
CFDA  C1     9C               CMPB  #ERR.FF
CFDC  26     B2 CF90          BNE   HOME13
CFDE  B6   067D       HOP88   LDA   TFIBN
CFE1  BD   CF4E               JSR   CLOSE1
CFE4  26     AA CF90          BNE   HOME13

CFE6  FC   067A       CRENEW  LDD   SAVEXT
CFE9  FD   0658               STD   TNAME+8
CFEC  B6   067C               LDA   SAVEXT+2
CFEF  B7   065A               STA   TNAME+10
CFF2  BD   C8FF               JSR   SEARCH
CFF5  27     04 CFFB          BEQ   RCREAT
CFF7  C1     A0               CMPB  #ERR.NE
CFF9  26     95 CF90          BNE   HOME13
                      * FALL THRU TO RCREAT
                      ********************************
```

```
                        ********************************
                        *
                        *  RAW CREATE.
                        *
                        *  CALLED WITH:
                        *    A = FILE NUMBER
                        *
                        *  ERROR CODE IN B:
                        *    ERRALL - ALREADY EXISTS,
                        *    ERRDRF - DIRECTORY FULL,
                        *    ERRWPT - DISK PROTECTED,
                        *    ERRIVD - INVALID DIRECTORY,
                        *    ERRTMO,IOE,SNF - I/O ERROR.

CFFB  97     F1      RCREAT  STA   FIBNO
CFFD  BD   CF29              JSR   FIBADR
D000  6D 0F                  TST   FFLAGS,X
D002  2B     03 D007         BMI   HOP89
D004  C6     9E              LDB   #ERR.FE
D006  39             HOME14  RTS
D007  4F             HOP89   CLRA
D008  BD   D178              JSR   FEMPDR
D00B  26     F9 D006         BNE   HOME14
D00D  BD   CF29              JSR   FIBADR
D010  A7 88  1D              STA   DIRN,X      DIR NUMBER
D013  A7 88  1E              STA   LSDIRB,X    LAST DIR
D016  C6     1C              LDB   #XTLEN2
D018  6F 85          HOP90   CLR   B,X
D01A  5A                     DECB
D01B  C1     0C              CMPB  #RDBYTE
D01D  24     F9 D018         BHS   HOP90

D01F  C6     18              LDB   #DIRSIZ-1
D021  6F C5          HOP91   CLR   B,U
D023  5A                     DECB
D024  2A     FB D021         BPL   HOP91

D026  33 41                  LEAU  DNAME,U
D028  C6     0B              LDB   #11
D02A  A6 80          HOP92   LDA   ,X+
D02C  A7 C0                  STA   ,U+
D02E  5A                     DECB
D02F  26     F9 D02A         BNE   HOP92
D031  5F                     CLRB
D032  39                     RTS
                        ********************************
```

```
                        ******************************
                        *
                        *  DELETE FILE.
                        *
                        *  CALLED WITH:
                        *    A = FILE NUMBER.
                        *
                        *  ERROR CODE IN B:
                        *    ERRFNF - FILE NOT FOUND,
                        *    ERRPRT - FILE PROTECTED,
                        *    ERRIVD - INVALID DIRECTORY,
                        *    ERRWPT - DISK PROTECTED,
                        *    ERRTMO,IOE,SNF - I/O ERROR.

D033  BD   D138     DELETE  JSR   CKPROT     CHECK PROTECTION, AND GET FIB ADDRESS
D036  26     CE D006        BNE   HOME14     ?ERROR?
D038  34     10             PSHS  X
D03A  E6 88  1D             LDB   DIRN,X     DIR NUMBER
D03D  BD   D284             JSR   GETDE
D040  26     47 D089        BNE   DEL2S      ?ERROR?

D042  1F     13             TFR   X,U
D044  35     10             PULS  X          FIB ADDRESS
D046  31 4C                 LEAY  DXTXT,U
D048  C6     04             LDB   #4         # OF EXTENTS

D04A  A6 C4         DELOOP  LDA   ,U         GET FLAGS
D04C  84     20             ANDA  #$20       MORE EXTENTS?
D04E  27     03 D053        BEQ   HOP93      NO MORE EXTENTS
D050  A6 C8  18             LDA   DLSCT,U    GET DIR ENTRY NUMBER OF MORE EXTENTS
                        *
D053  34     06     HOP93   PSHS  D          SAVE EXTENT COUNTER AND MORE EXTENTS FLAG
D055  C6     81             LDB   #$81
D057  E7 C4                 STB   ,U         MAKE INVALID
D059  34     10             PSHS  X          SAVE FIB ADDRESS
                        *
D05B  AE A1         HOP94   LDX   ,Y++       GET LSN
D05D  E6 A0                 LDB   ,Y+        GET LENGHT OF LSN
D05F  27     0A D06B        BEQ   HOP95      IF NULL THEN LOOP
D061  34     20             PSHS  Y          SAVE POINTER TO DIRECTORY ENTRY IN BUFFER
D063  8D     2A D08F        BSR   DELEXT     DE-ALLOCATE SPACE
D065  35     20             PULS  Y          GET POINTER BACK
D067 1026    FA07 CA72      LBNE  RTS4S      ERRORS?
D06B  6A 63         HOP95   DEC   3,S        DEC EXTENTS IN ENTRY COUNTER
D06D  26     EC D05B        BNE   HOP94      NOT DONE WITH THIS ENTRY YET
D06F  35     10             PULS  X          GET FIB ADDRESS
D071  E6 E4                 LDB   ,S         GET FLAG FOR MORE EXTENTS
D073  27     13 D088        BEQ   DONE       NO MORE EXTENTS, LEAVE
D075  32 62                 LEAS  2,S        FLUSH COUNTER, FLAG
D077  34     10             PSHS  X          SAVE FIB ADDRESS
D079  BD   D284             JSR   GETDE      GET NEXT DIRECTORY ENTRY
D07C  1F     13             TFR   X,U        MOVE ENTRY POINTER INTO U
D07E  35     10             PULS  X          GET FIB ADDRESS
D080  26     84 D006        BNE   HOME14     ?ERRORS?
```

```
D082  31 41                    LEAY  1,U        POINT AT THE FIRST EXTENT IN ENTRY
D084  C6      07               LDB   #7         # OF EXTENTS
D086  20      C2 D04A          BRA   DELOOP     LOOP

D088  5F              DONE     CLRB             NO ?ERRORS?
D089  0F      F6      DEL2S    CLR   LOCK       TIME-OUT OK
D08B  32 62                    LEAS  2,S        FLUSH
D08D  5D                       TSTB             SET FLAGS
D08E  39                       RTS              RETURN TO CALLER
                      ********************************
```

```
                        ********************************
                        *
                        *  DELETE ONE EXTENT
                        *
                        *  CALLED BY DELETE.  MARKS ONE
                        *  EXTENT'S WORTH OF SECTORS
                        *  AS FREE.
                        *
                        *  CALLED WITH:
                        *    B = NUMBER OF SECTORS,
                        *    X = STARTING LSN.
                        *
                        *  ERROR CODES IN B:
                        *    ERRIVD - INVALID DIRECTORY,
                        *    ERRTMO,IOE,SNF - I/O ERROR.

D08F   4F              DELEXT   CLRA
D090   34      02               PSHS   A
D092   34      16               PSHS   D,X
D094   BD    D1F9               JSR    DKINF
D097   26      7D D116          BNE    RTS5S
D099 10AE 84                    LDY    DIRLSN,X
D09C   EC 62                    LDD    2,S
D09E   83    05A0               SUBD   #180*8
D0A1   25      0B D0AE          BLO    HOP96
D0A3   31 3F                    LEAY   -1,Y
D0A5   ED 62                    STD    2,S
D0A7   E3 E4                    ADDD   0,S
D0A9   83    05A0               SUBD   #180*8
D0AC   24      66 D114          BHS    DELIVD
D0AE   EC 62           HOP96    LDD    2,S
D0B0   E3 E4                    ADDD   0,S
D0B2   83    05A0               SUBD   #180*8
D0B5   25      07 D0BE          BLO    HOP97
D0B7   E7 64                    STB    4,S
D0B9   50                       NEGB
D0BA   EB 61                    ADDB   1,S
D0BC   E7 61                    STB    1,S
D0BE   BD    D2C3      HOP97    JSR    READSB
D0C1   26      53 D116          BNE    RTS5S
D0C3   86      FF               LDA    #-1
D0C5   A7 02                    STA    PAGVLD,X
D0C7   EC 62                    LDD    2,S
D0C9   44                       LSRA
D0CA   56                       RORB
D0CB   66 E4                    ROR    0,S
D0CD   44                       LSRA
D0CE   56                       RORB
D0CF   66 E4                    ROR    0,S
D0D1   44                       LSRA
D0D2   56                       RORB
D0D3   66 E4                    ROR    0,S
D0D5   AE 05                    LDX    PAGADR,X
D0D7   3A                       ABX
```

```
D0D8  C6      01              LDB   #$01
D0DA  A6 E4                   LDA   0,S
D0DC  27      05 D0E3         BEQ   HOP99
D0DE  58              HOP98   LSLB
D0DF  80      20              SUBA  #$20
D0E1  26      FB D0DE         BNE   HOP98
D0E3  E7 E4           HOP99   STB   0,S
D0E5  E6 61                   LDB   1,S

D0E7  A6 E4           HOP100  LDA   0,S
D0E9  AA 84                   ORA   0,X
D0EB  A7 84                   STA   0,X
D0ED  5A                      DECB
D0EE  27      18 D108         BEQ   DEL5S
D0F0  68 E4                   LSL   0,S
D0F2  24      F3 D0E7         BCC   HOP100
D0F4  86      01              LDA   #1
D0F6  A7 E4                   STA   0,S
D0F8  30 01                   LEAX  1,X
D0FA  C1      10      HOP101  CMPB  #16
D0FC  25      E9 D0E7         BLO   HOP100
D0FE  86      FF              LDA   #$FF
D100  A7 80                   STA   ,X+
D102  A7 80                   STA   ,X+
D104  C0      10              SUBB  #16
D106  26      F2 D0FA         BNE   HOP101
D108  8E    05A0      DEL5S   LDX   #180*8
D10B  32 64                   LEAS  4,S
D10D  E6 E0                   LDB   ,S+
D10F  27      07 D118         BEQ   HOME15
D111  16    FF7B D08F         LBRA  DELEXT

D114  C6      90      DELIVD  LDB   #ERR.IV
D116  32 65           RTS5S   LEAS  5,S
D118  39              HOME15  RTS

                      ********************************
```

```
                        ********************************
                        *
                        *  SET PROTECT STATUS.
                        *
                        *  CALLED WITH:
                        *    A = FILE NUMBER,
                        *    B = 1 (PROT), 0 (UNPROT).
                        *
                        *  ERROR CODES IN B:
                        *    ERRFNF - FILE NOT FOUND,
                        *    ERRWPT - DISK PROTECTED,
                        *    ERRTMO,IOE,SNF - I/O ERROR.

D119  97     F1      PROTEC  STA    FIBNO
D11B  BD   CF29              JSR    FIBADR
D11E  A6 0F                  LDA    FFLAGS,X
D120  2B     26 D148         BMI    FNF
D122  5D                     TSTB
D123  27     03 D128         BEQ    HOP102
D125  8A     02              ORA    #2          PROTECTED NOW
D127                         SKIP2
D128  84     FD      HOP102  ANDA   #$FD
D12A  A7 0F                  STA    FFLAGS,X
D12C  E6 88  1D              LDB    DIRN,X
D12F  BD   D284              JSR    GETDE
D132  26     E4 D118         BNE    HOME15
D134  A7 84                  STA    FLAGS,X
D136  5F                     CLRB
D137  39                     RTS

                        ********************************
                        *  CHECK FILE PROTECTION
D138  97     F1      CKPROT  STA    FIBNO
D13A  BD   CF29              JSR    FIBADR
D13D  A6 0F                  LDA    FFLAGS,X
D13F  2B     07 D148         BMI    FNF
D141  85     02              BITA   #2          FILE PROT BIT
D143  27     32 D177         BEQ    HOME16
D145  C6     98              LDB    #ERR.PT
D147  39                     RTS
D148  C6     9C      FNF     LDB    #ERR.FF     NOT FOUND
D14A  39                     RTS

                        ********************************
```

```
                        ********************************
                        *
                        *  RENAME FILE
                        *
                        *  CALLED WITH:
                        *    TNAME = NEW NAME,
                        *    A = FIB #,
                        *    (TDRV IGNORED).
                        *
                        *  ERROR CODE IN B:
                        *    ERRPRT - FILE PROTECTED,
                        *    ERRIVD - INVALID DIRECTORY,
                        *    ERRWPT - DISK PROTECTED,
                        *    ERRTMO,IOE,SNF - I/O ERROR.

D14B  BD   D138      RNAME   JSR   CKPROT      FILE PROTECT
D14E  26     27 D177         BNE   HOME16
D150  C6     0B              LDB   #11
D152  CE   0650              LDU   #TNAME

                        * COPY NAME INTO FIB
D155  A6 C0          HOP103  LDA   ,U+
D157  A7 80                  STA   ,X+
D159  5A                     DECB
D15A  26     F9 D155         BNE   HOP103

                        * GET DIR ENTRY
D15C  30 15                  LEAX  -11,X
D15E  E6 88   1D             LDB   DIRN,X
D161  BD    D284             JSR   GETDE
D164  26     11 D177         BNE   HOME16
D166  CE   0650              LDU   #TNAME
D169  C6     0B              LDB   #11

                        * COPY NAME INTO THE DIRECTORY
D16B  30 01                  LEAX  1,X
D16D  A6 C0          HOP104  LDA   ,U+
D16F  A7 80                  STA   ,X+
D171  5A                     DECB
D172  26     F9 D16D         BNE   HOP104
D174  0F     F6              CLR   LOCK
D176  5F                     CLRB
D177  39             HOME16  RTS
                        ********************************
```

```
                        ********************************
                        *
                        *  FIND EMPTY DIRECTORY SLOT
                        *
                        *  CALLED WITH:
                        *    A = 0 (NORMAL), OR
                        *    A = 1 (EXTENT BLOCKS CAN'T
                        *          START IN SLOT 0).
                        *  RETURN WITH:
                        *    A = DIR ENTRY NUMBER FOUND
                        *    X-) PAGE TABLE FOR BUFFER,
                        *    U-) ENTRY WITHIN BUFFER.
                        *
                        *  ERROR CODE IN B:
                        *    ERRDRF - DIRECTORY FULL,
                        *    ERRIVD - INVALID DIRECTORY,
                        *    ERRTMO,IOE,SNF - I/O ERROR.
                        *
                        *  NOTE THAT THE BUFFER IS
                        *  ALWAYS MARKED DIRTY.

D178  40              FEMPDR  NEGA
D179  B7   067E               STA   TEMP2
D17C  BD   D1F9               JSR   DKINF
D17F  26     F6 D177          BNE   HOME16      ERROR?
D181  AE 84                   LDX   DIRLSN,X
D183  34     10               PSHS  X
D185  30 02                   LEAX  2,X
D187  BF   066F       FRNS    STX   SCTEXT
D18A  1F     12               TFR   X,Y
D18C  BD   D2C3               JSR   READSB
D18F  26     35 D1C6          BNE   FEMERR      ERROR?
D191  EE 05                   LDU   PAGADR,X
D193  C6     0A               LDB   #DIRPS
D195  7D   067E               TST   TEMP2
D198  2A     05 D19F          BPL   HOP105
D19A  70   067E               NEG   TEMP2
D19D  20     07 D1A6          BRA   HOP106
D19F  A6 C4           HOP105  LDA   0,U
D1A1  2B     1B D1BE          BMI   FEMP        FOUND EMPTY
D1A3  7C   067E               INC   TEMP2
D1A6  33 C8  19       HOP106  LEAU  DIRSIZ,U
D1A9  5A                      DECB
D1AA  26     F3 D19F          BNE   HOP105
D1AC  BE   066F               LDX   SCTEXT
D1AF  30 01                   LEAX  1,X
D1B1  1F     10               TFR   X,D
D1B3  A3 E4                   SUBD  0,S
D1B5  C1     12               CMPB  #18
D1B7  25     CE D187          BLO   FRNS
D1B9  32 62                   LEAS  2,S
D1BB  C6     92               LDB   #ERR.FD
D1BD  39                      RTS
```

```
D1BE  86    FF    FEMP    LDA   #-1
D1C0  A7 02               STA   PAGVLD,X
D1C2  B6    067E          LDA   TEMP2
D1C5  5F                  CLRB
D1C6  32 62         FEMERR LEAS  2,S
D1C8  39                  RTS
                    ********************************
```

```
                        ********************************
                        *
                        *  GET FREE SPACE ON DRIVE.
                        *
                        *  CALLED WITH:
                        *    DD.UNT = DRIVE #
                        *
                        *  RETURN WITH:
                        *    X=#FREE SECTORS.
                        *
                        *  ERROR CODE IN B:
                        *    ERRIVD - INVALID DIRECTORY,
                        *    ERRTMO,IOE,SNF - I/O ERROR.
                        *
D1C9   8D      2E D1F9 GETFRE   BSR    DKINF       GET DIR LSN
D1CB   26      10 D1DD          BNE    HOP107
D1CD 10AE 84                    LDY    DIRLSN,X
D1D0   9E      8A               LDX    DBLZER
D1D2   8D      0A D1DE          BSR    COUNT
D1D4   26      07 D1DD          BNE    HOP107
D1D6   31 21                    LEAY   1,Y
D1D8   8D      04 D1DE          BSR    COUNT
D1DA   26      01 D1DD          BNE    HOP107
D1DC   5F                       CLRB
D1DD   39              HOP107   RTS
                        *  COUNT THE 1 BITS IN SECTOR
                        *

D1DE   34      10       COUNT    PSHS   X
D1E0   BD    D2C3               JSR    READSB
D1E3   26      E1 D1C6          BNE    FEMERR
D1E5   EE 05                    LDU    PAGADR,X
D1E7   35      10               PULS   X
D1E9   C6      B4               LDB    #180        BYTES IN TABLE
D1EB   A6 C0           HOP108   LDA    ,U+
                        *
D1ED   44              HOP109   LSRA
D1EE   24      02 D1F2          BCC    HOP110
D1F0   30 01                    LEAX   1,X
                        *
D1F2   4D              HOP110   TSTA
D1F3   26      F8 D1ED          BNE    HOP109
                        *
D1F5   5A                       DECB
D1F6   26      F3 D1EB          BNE    HOP108
D1F8   39                       RTS

                        ********************************
```

```
                        ********************************
                        *  GET DISK INFO TABLE.
                        *  READ FROM DISK IF NOT VALID
                        *  IN RAM.
                        *
                        *  DD.UNT = DRIVE # (1-4)
                        *
                        *  RETURN WITH:
                        *    X-) TABLE1,
                        *    U-) TABLE2;
                        *    B = ERROR CODE
                        *
D1F9   8E    061C       DKINF   LDX    #DRVINF
D1FC   CE    0696               LDU    #LUVALD
D1FF   C6      06               LDB    #DRVISZ
D201   96      EB               LDA    DD.UNT
D203   33 C6                    LEAU   A,U
D205   4A                       DECA
D206   3D                       MUL
D207   30 8B                    LEAX   D,X
D209   6D C4                    TST    ,U          TABLE VALID ?
D20B   26      3E D24B          BNE    LVOK
D20D 108E    0168               LDY    #DIRTRK*18
D211   86      12               LDA    #18
D213   A7 C8   10               STA    LUMSEC-LUVALD,U SECTORS/TRACK
D216   34      10               PSHS   X
D218   BD    D2C3               JSR    READSB
D21B 1026    F9B4 CBD3          LBNE   RTS2S
D21F   AE 05                    LDX    PAGADR,X
D221   EC 89 00FE               LDD    254,X
D225   43                       COMA
D226   53                       COMB
D227 10A3 89 00FC               CMPD   252,X
D22C   35      10               PULS   X
D22E   26      1D D24D          BNE    ERINVD
D230   E7 C8   10               STB    LUMSEC-LUVALD,U SECTORS/TRACK
D233   A7 4C                    STA    LUMTRK-LUVALD,U TRACKS/SIDE
D235   6A C4                    DEC    0,U         VALID FLAG
D237   C1      12               CMPB   #18
D239   27      01 D23C          BEQ    HOP111
D23B   5F                       CLRB
D23C   34      04       HOP111  PSHS   B
D23E   6F 02                    CLR    FRELEN,X
D240   CC    0168               LDD    #DIRTRK*18
D243   6D E0                    TST    ,S+
D245   26      02 D249          BNE    HOP112
D247                            ASLD
D249   ED 84            HOP112  STD    DIRLSN,X
D24B   5F                LVOK    CLRB
D24C   39                       RTS

D24D   C6      90       ERINVD  LDB    #ERR.IV     INVALID DIR
D24F   39                       RTS
```

****

```
                        ********************************
                        *
                        *  GET DIRECTORY ENTRY.
                        *
                        *  CALLED WITH:
                        *    DD.UNT = DRIVE # (1-4)
                        *    B = WHICH ENTRY (0-N).
                        *
                        *  RETURN WITH:
                        *    X-) ENTRY (IN A BUFFER),
                        *    SCTEXT = LSN.
                        *    DIRBFP = BUFFER POINTER.
                        *
                        *  ERROR CODE IN B:
                        *    ERRIVD - INVALID DIRECTORY,
                        *    ERRTMO,IOE,SNF - I/O ERROR.
                        *
                        *  THIS DOES NOT CHECK FOR:
                        *    BEYOND END OF DIRECTORY,
                        *    ENTRIES WHICH ARE EXTENTS,
                        *    BEYOND LAST USED ENTRY.
                        *
D250  86    FF      GETDIR  LDA   #-1
D252  4C            HOP113  INCA
D253  C0    0A              SUBB  #DIRPS      DIRS/SECTOR
D255  2C    FB D252         BGE   HOP113
D257  CB    0A              ADDB  #DIRPS
D259  34    06              PSHS  D
D25B  8D    9C D1F9         BSR   DKINF       GET DISK INFO
D25D 1026   F972 CBD3       LBNE  RTS2S       ERROR?
D261  EC 84                 LDD   DIRLSN,X
D263  C3    0002            ADDD  #2          PAST BIT TABLE
D266  EB E0                 ADDB  ,S+
D268  89    00              ADCA  #0          = LSN TO READ
D26A  FD    066F            STD   SCTEXT
D26D  1F    02              TFR   D,Y
D26F  BD    D2C3            JSR   READSB
D272  35    02              PULS  A           WHICH ENTRY
D274  26    0D D283         BNE   HOME27
D276  1F    13              TFR   X,U
D278  C6    19              LDB   #DIRSIZ
D27A  3D                    MUL               FORM OFFSET
D27B  BF    067F            STX   DIRBFP
D27E  AE 05                 LDX   PAGADR,X
D280  30 8B                 LEAX  D,X
D282  5F                    CLRB
D283  39            HOME27  RTS
                        ********************************
```

```
                        ********************************
                        *  GET DIRECTORY ENTRY
                        *
                        * ENTRY:
                        *   B =DIRECTORY ENTRY #
                        *
                        * EXIT:
                        *  MARKS THE BUFFER DIRTY
                        *  [X]=DIRECTORY ENTRY
                        *  [Y]=BUFFER TABLE
                        *   D IS PRESERVED
                        *
D284   34      06       GETDE    PSHS   D
D286   8D      C8 D250           BSR    GETDIR
D288 1026    F947 CBD3           LBNE   RTS2S
D28C 10BE    067F                LDY    DIRBFP
D290   86      FE                LDA    #-2
D292   A7 22                     STA    PAGVLD,Y
D294   5F                        CLRB
D295   35      86                PULS   D,PC

                        ********************************
```

```
                        ********************************
                        *
                        *  FIND BUFFER WITH SAME SECTOR
                        *
                        *  CALLED WITH:
                        *    Y = LSN,
                        *    DD.UNT = DRIVE # (1-4).
                        *
                        *  RETURN WITH:
                        *    X-) BUFFER TABLE,
                        *    U-) EMPTY BUFFER TABLE (OR =0)
                        *    B = 0 (FOUND), OR
                        *    B = $FF (NOT FOUND).
                        *
                        *  THE PAGE IS SET TO MOST
                        *  RECENTLY USED STATUS.
D297   8E    0634      FMBUF   LDX    #PBFINF
D29A   DE      8A              LDU    DBLZER      NO EMPTIES YET
D29C   C6      04              LDB    #4          NUMBER OF BUFFERS

D29E   A6 02           HOP114  LDA    PAGVLD,X
D2A0   27      17 D2B9         BEQ    HOP116
D2A2   81      55              CMPA   #$55
D2A4   27      15 D2BB         BEQ    HOP117
D2A6 10AC 84                   CMPY   PAGLSN,X
D2A9   26      0A D2B5         BNE    HOP115
D2AB   96      EB              LDA    DD.UNT
D2AD   A1 03                   CMPA   PAGDRV,X
D2AF   26      04 D2B5         BNE    HOP115
D2B1   8D      66 D319         BSR    THSLRU      REFERENCED!
D2B3   5F                      CLRB               VALID FIND
D2B4   39                      RTS                WE'RE DONE
D2B5   6D 02           HOP115  TST    PAGVLD,X
D2B7   26      02 D2BB         BNE    HOP117
D2B9   1F      13      HOP116  TFR    X,U
D2BB   30 07           HOP117  LEAX   LENPBF,X
D2BD   5A                      DECB
D2BE   26      DE D29E         BNE    HOP114
D2C0   C6      FF              LDB    #-1
D2C2   39                      RTS
                        ********************************
```

```
                        ********************************
                        *
                        *  READ SECTOR TO BUFFER.
                        *
                        *  CALLED WITH:
                        *    Y = LSN,
                        *    DD.UNT = DRIVE # (1-4).
                        *
                        *  RETURN WITH:
                        *     X-> BUFFER INFO TABLE.
                        *     U PRESERVED
                        *
                        *  ERROR CODE IN B:
                        *    ERRIVD - INVALID DIRECTORY,
                        *    ERRTMO,IOE,SNF - I/O ERROR.
                        *
                        *  A BUFFER IS ALLOCATED FROM
                        *  THE 4-BUFFER POOL.  PREFERENCE:
                        *     (1) SECTOR ALREADY IN A BUFFER
                        *     (2) AN EMPTY BUFFER
                        *     (3) CLEAN OUT THE LRU BUFFER

D2C3   34      40       READSB  PSHS    U
D2C5   BD    D297               JSR     FMBUF           DO WE HAVE IT
D2C8 1027    F907 CBD3          LBEQ    RTS2S
D2CC   30 C4                    LEAX    0,U             TEST U
D2CE   35      40               PULS    U
D2D0   26      04 D2D6          BNE     HAVMT           HAVE AN EMPTY
D2D2   8D      1C D2F0          BSR     FINDMT
D2D4   26      19 D2EF          BNE     HOME17
D2D6   6F 02            HAVMT   CLR     PAGVLD,X
D2D8 10AF 84                    STY     PAGLSN,X
D2DB   96      EB               LDA     DD.UNT
D2DD   A7 03                    STA     PAGDRV,X
D2DF   34      10               PSHS    X
D2E1   AE 05                    LDX     PAGADR,X
D2E3   BD    D386               JSR     RLSN            READ SECTOR
D2E6   35      10               PULS    X
D2E8   26      05 D2EF          BNE     HOME17
D2EA   86      01               LDA     #1
D2EC   A7 02                    STA     PAGVLD,X
D2EE   5F                       CLRB
D2EF   39               HOME17  RTS

                        ********************************
D2F0   34      66       FINDMT  PSHS    D,U,Y
D2F2   8E    0634       HOP118  LDX     #PBFINF
D2F5   C6      04               LDB     #4
D2F7   A6 04            HOP119  LDA     PAGLRU,X
D2F9   81      01               CMPA    #1
D2FB   27      05 D302          BEQ     HOP120
D2FD   30 07                    LEAX    LENPBF,X
D2FF   5A                       DECB
D300   26      F5 D2F7          BNE     HOP119
```

```
D302   8D      15 D319 HOP120   BSR    THSLRU      REFERENCED!
D304   A6 02                    LDA    PAGVLD,X
D306   81      55               CMPA   #$55        NON-USABLE?
D308   27      E8 D2F2          BEQ    HOP118
D30A   4C                       INCA               GRACE TO WRITES
D30B   26      04 D311          BNE    HOP121
D30D   6A 02                    DEC    PAGVLD,X    GRACE EXPIRED
D30F   20      E1 D2F2          BRA    HOP118
D311   8D      1D D330 HOP121   BSR    CLEANP
D313 1026    F75B CA72          LBNE   RTS4S
D317   35      E6               PULS   D,U,Y,PC
```

****

```
D302   8D      15 D319 HOP120   BSR    THSLRU       REFERENCED!
D304   A6 02                    LDA    PAGVLD,X
D306   81      55               CMPA   #$55         NON-USABLE?
D308   27      E8 D2F2          BEQ    HOP118
D30A   4C                       INCA                GRACE TO WRITES
D30B   26      04 D311          BNE    HOP121
D30D   6A 02                    DEC    PAGVLD,X     GRACE EXPIRED
D30F   20      E1 D2F2          BRA    HOP118
D311   8D      1D D330 HOP121   BSR    CLEANP
D313 1026    F75B CA72          LBNE   RTS4S
D317   35      E6               PULS   D,U,Y,PC
```

****

```
                        ********************************
                        *  UPDATE LRU COUNTS OF BUFFERS
                        *
                        *  CALLED WITH:
                        *    X-) PAGE TABLE ENTRY
                        *
                        *  NO ERROR CONDITIONS.
                        *
                        *  CALL "THSLRU" MAKES AN ENTRY
                        *  THE MOST RECENTLY USED.
                        *
D319   C6      04       THSLRU   LDB    #4
D31B   A6 04                     LDA    PAGLRU,X
D31D   CE    0634                LDU    #PBFINF
D320   A1 44            HOP122   CMPA   PAGLRU,U
D322   22      02 D326           BHI    HOP123
D324   6A 44                     DEC    PAGLRU,U
D326   33 47            HOP123   LEAU   LENPBF,U
D328   5A                       DECB
D329   26      F5 D320           BNE    HOP122
D32B   86      04                LDA    #4
D32D   A7 04                     STA    PAGLRU,X
D32F   39                       RTS
                        ********************************
```

```
                        ********************************
                        *  UPDATE LRU COUNTS OF BUFFERS
                        *
                        *  CALLED WITH:
                        *    X-) PAGE TABLE ENTRY
                        *
                        *  NO ERROR CONDITIONS.
                        *
                        *  CALL "THSLRU" MAKES AN ENTRY
                        *  THE MOST RECENTLY USED.
                        *
D319  C6      04        THSLRU  LDB     #4
D31B  A6 04                     LDA     PAGLRU,X
D31D  CE    0634                LDU     #PBFINF
D320  A1 44             HOP122  CMPA    PAGLRU,U
D322  22      02 D326           BHI     HOP123
D324  6A 44                     DEC     PAGLRU,U
D326  33 47             HOP123  LEAU    LENPBF,U
D328  5A                        DECB
D329  26      F5 D320           BNE     HOP122
D32B  86      04                LDA     #4
D32D  A7 04                     STA     PAGLRU,X
D32F  39                        RTS
                        ********************************
```

```
                        ********************************
                        *
                        *  CLEAN UP A BUFFER
                        *
                        *  IF DIRTY, WRITE IT OUT AND
                        *  (IF VERIFY ON OR TO DIRECTORY
                        *  TRACKS) VERIFY IT.
                        *
                        *  MAINTAIN A LIST OF DIRECTORY
                        *  SECTORS WHICH NEED UPDATING.
                        *
                        *  CALLED WITH:
                        *    X-) BUFFER TABLE
                        *
                        *  RETURN WITH:
                        *    X PRESERVED
                        *
                        *  ERROR CODES IN B:
                        *    ERRWPT - DISK PROTECTED,
                        *    ERRTMO,IOE,SNF - I/O ERROR.

D330  6D 02           CLEANP  TST   PAGVLD,X
D332  2B    02 D336           BMI   HOP124
D334  5F                      CLRB
D335  39                      RTS
D336  96    EB        HOP124  LDA   DD.UNT      SAVE DRIVE
D338  34    02                PSHS  A
D33A  34    10                PSHS  X
D33C  86    FF                LDA   #-1
D33E  97    F6                STA   LOCK
D340  6F 02                   CLR   PAGVLD,X
D342  A6 03                   LDA   PAGDRV,X
D344  97    EB                STA   DD.UNT
D346 10AE 84                  LDY   PAGLSN,X
D349  AE 05                   LDX   PAGADR,X

D34B  BD    D376              JSR   WLSN
D34E  35    10                PULS  X
D350  26    1C D36E           BNE   HOP126
                        ******
                        * MARK DTYDIR SO WE CAN BACKUP
                        * DIRECTORY LATER ON
D352  96    EC                LDA   DD.TRK
D354  81    14                CMPA  #DIRTRK
D356  26    11 D369           BNE   HOP125
D358  CE    A673              LDU   #BITAB+8
D35B  96    EB                LDA   DD.UNT
D35D  40                      NEGA
D35E  A6 C6                   LDA   A,U
D360  CE    06AA              LDU   #DTYDIR-1
D363  D6    ED                LDB   DD.SEC
D365  AA C5                   ORA   B,U
D367  A7 C5                   STA   B,U
D369  86    01        HOP125  LDA   #1
```

```
D36B  A7 02              STA   PAGVLD,X
D36D  5F                 CLRB
D36E  35    02   HOP126  PULS  A
D370  97    EB           STA   DD.UNT      RESTORE DRIVE NUMBER
D372  0F    F6           CLR   LOCK
D374  5D                 TSTB
D375  39                 RTS
                         ********************************
```

```
                        ********************************
                        *  READ   LSN
                        *  VERIFY LSN
                        *  WRITE  LSN
                        *
                        *  CALLED WITH:
                        *    DD.UNT = DRIVE # (1-4)
                        *    X-) AREA OF RAM TO WRITE,
                        *    Y = LSN.
                        *
                        *  RETURN WITH:
                        *    X PRESERVED.
                        *
                        *  ERROR CODE IN B:
                        *    ERRWPT - DISK PROTECTED,
                        *    ERRIVD - INVALID DIRECTORY,
                        *    ERRTMO,IOE,SNF - I/O ERROR.
                        ********************************
                        *  NOTE:  IF THE LSN IS ALSO IN
                        *  A BUFFER BUT THE DATA IS NOT
                        *  COMING FROM THE BUFFER, THE
                        *  BUFFER IS INVALIDATED.
D376   8D      15 D38D WLSN     BSR    DODIV
D378   BD    C0FE              JSR    GO.WRS
D37B   9E      EE      HOP127   LDX    DD.BUF
D37D   5D                       TSTB
D37E   39                       RTS
                        ********************************
                        *  NOTE:  THIS WILL TRY TO READ
                        *  THE SECTOR TO CHECK FOR AN
                        *  I/O ERROR, BUT WILL NOT
                        *  COMPARE THE DATA.
D37F   8D      0C D38D VLSN     BSR    DODIV
D381   BD    C15F              JSR    GO.VER
D384   20      F5 D37B         BRA    HOP127
                        ********************************
D386   8D      05 D38D RLSN     BSR    DODIV
D388   BD    C104              JSR    GO.RDS
D38B   20      EE D37B         BRA    HOP127

                        ********************************
```

```
                        ********************************
                        *
                        * DIVIDE SUB-ROUTINE
                        *
                        * CALL :
                        *      Y = DIVIDEND [LSN]
                        *
                        * RETURN WITH :
                        *      A = QUOTIENT  [TRACK]
                        *      B = REMAINDER [SECTOR]
                        *  DD.TRK AND DD.SEC ARE SET
                        *  FOR IMPENDING DISK I/O
                        *
D38D  9F      EE        DODIV   STX   DD.BUF
D38F  8E    06A6                LDX   #LUMSEC
D392  D6      EB                LDB   DD.UNT
D394  E6 85                     LDB   B,X
D396  4F                        CLRA
D397  34      06                PSHS  A,B
D399  6F E2                     CLR   ,-S
D39B  1F      20                TFR   Y,D
D39D  6C E4             HOP128  INC   0,S
D39F  A3 61                     SUBD  1,S
D3A1  2A      FA D39D           BPL   HOP128
D3A3  EB 62                     ADDB  2,S
D3A5  A6 E4                     LDA   0,S
D3A7  4A                        DECA
D3A8  5C                        INCB
D3A9  32 63                     LEAS  3,S
D3AB  DD      EC                STD   DD.TRK
D3AD  39                        RTS

                        ********************************
                                END
                                LIB             DISKLOAD
                                NAM             DISKLOAD
```

```
                                ********************************
                                *
                                *
E D3AE                          REORG   SET   *
* 0000                                   ORG   0
D 0000                          SFIBNO  RMB   1
D 0001                          DFIBNO  RMB   1
D 0002                          BUFSIZ  RMB   2
D 0004                          SUM     RMB   3
E 0007                          COPSTK  EQU   *
* D3AE                                   ORG   REORG
  D3AE   17    FB87 CF38 COPY            LBSR  CLOSAL
  D3B1   26      3B D3EE                 BNE   COPERR
  D3B3   32 79                           LEAS  -COPSTK,S
  D3B5   1F      40                      TFR   S,D
  D3B7   83    0100                      SUBD  #$100
  D3BA   93      1F                      SUBD  ARRTOP
  D3BC 102B    AF82 8342                 LBMI  OMERR
  D3C0   5F                              CLRB
  D3C1   4D                              TSTA
  D3C2 1027    AF7C 8342                 LBEQ  OMERR
  D3C6   ED 62                           STD   BUFSIZ,S
  D3C8   17    0353 D71E                 LBSR  GNMNUL
  D3CB   26      21 D3EE                 BNE   COPERR
  D3CD   A7 E4                           STA   SFIBNO,S
  D3CF   17    FB24 CEF6                 LBSR  LENFIL
  D3D2   26      1A D3EE                 BNE   COPERR
  D3D4   9D      A5                      JSR   CHRGOT
  D3D6   81      BC                      CMPA  #TOTOK
  D3D8 1026    B5D8 89B4                 LBNE  SNERR
  D3DC   9D      9F                      JSR   GETCHR
  D3DE   17    033D D71E                 LBSR  GNMNUL
  D3E1   27      04 D3E7                 BEQ   HOP129
  D3E3   C1      A0                      CMPB  #ERR.NE
  D3E5   26      07 D3EE                 BNE   COPERR
  D3E7   A7 61           HOP129          STA   DFIBNO,S
  D3E9   17    FBA5 CF91                 LBSR  CREATE
  D3EC   27      03 D3F1                 BEQ   COPLOP
  D3EE   7E    C718      COPERR          JMP   NEWERR
  D3F1   A6 E4           COPLOP          LDA   SFIBNO,S
  D3F3   97      F1                      STA   FIBNO
  D3F5   17    FB31 CF29                 LBSR  FIBADR
  D3F8   EC 0C                           LDD   RDBYTE,X
  D3FA 10A3 88   10                      CMPD  FILLEN,X
  D3FE   25      07 D407                 BLO   HOP130
  D400   A6 0E                           LDA   RDBYTE+2,X
  D402   A1 88   12                      CMPA  FILLEN+2,X
  D405   27      50 D457                 BEQ   COPDON
  D407   EE 0C           HOP130          LDU   RDBYTE,X
  D409   A6 0E                           LDA   RDBYTE+2,X
  D40B   A7 66                           STA   SUM+2,S
  D40D   EF 64                           STU   SUM,S
  D40F   EC 62                           LDD   BUFSIZ,S
  D411   E3 65                           ADDD  SUM+1,S
```

```
D413   ED 65                    STD    SUM+1,S
D415   24    02 D419            BCC    HOP131
D417   6C 64                    INC    SUM,S
D419   A6 64          HOP131    LDA    SUM,S
D41B   A0 88   10               SUBA   FILLEN,X
D41E   25    0E D42E            BCS    HOP132
D420   EC 65                    LDD    SUM+1,S
D422   A3 88   11               SUBD   FILLEN+1,X
D425   23    07 D42E            BLS    HOP132
D427   EC 88   11               LDD    FILLEN+1,X
D42A   A3 0D                    SUBD   RDBYTE+1,X
D42C   ED 62                    STD    BUFSIZ,S
D42E   A6 E4          HOP132    LDA    SFIBNO,S
D430   EE 0C                    LDU    RDBYTE,X
D432   E6 0E                    LDB    RDBYTE+2,X
D434 10AE 62                    LDY    BUFSIZ,S
D437   9E     1F               LDX    ARRTOP
D439   17   F5D8 CA14           LBSR   READ
D43C   26     B0 D3EE           BNE    COPERR
D43E   A6 61                    LDA    DFIBNO,S
D440   97     F1               STA    FIBNO
D442   17   FAE4 CF29           LBSR   FIBADR
D445 10AE 88   10               LDY    FILLEN,X
D449   E6 88   12               LDB    FILLEN+2,X
D44C   EE 62                    LDU    BUFSIZ,S
D44E   9E     1F               LDX    ARRTOP
D450   17   F783 CBD6           LBSR   WRITE
D453   26     99 D3EE           BNE    COPERR
D455   20     9A D3F1           BRA    COPLOP
D457   17   FADE CF38 COPDON    LBSR   CLOSAL
D45A   26     92 D3EE           BNE    COPERR
D45C   32 67                    LEAS   COPSTK,S
D45E   39                       RTS
```

****

```
                        ********************************
                        *
                        * MERGE A BASIC PROGRAM ON
                        * DISK WITH THE ONE IN MEMORY
                        *
                        * SYNTAX :
                        *  MERGE "filespec"
D45F   17    02B6 D718 MERGE    LBSR   GNMBAS
D462   26      8A D3EE          BNE    COPERR
D464   8D      7D D4E3          BSR    CHKHDR
D466   26      86 D3EE          BNE    COPERR
D468   81      01               CMPA   #1          BASIC FILE TYPE
D46A   26      3E D4AA          BNE    BFMERR
D46C   DE      1B               LDU    VARTAB
D46E 109E      19               LDY    TXTTAB
D471   34      60               PSHS   Y,U
D473   33 41                    LEAU   1,U
D475   DF      19               STU    TXTTAB
D477   17    00DA D554          LBSR   LOADBS
D47A   35      50               PULS   X,U
D47C   DF      1B               STU    VARTAB
D47E   9F      19               STX    TXTTAB
D480   33 41                    LEAU   1,U

D482   EC C1           MERGLP   LDD    ,U++        SKIP LINK
D484   27      1E D4A4          BEQ    MRGXIT      DONE
D486   EC C1                    LDD    ,U++        GRAB LINE #
D488   FD    02DA               STD    BUFLNM      BUFFER LINE#
D48B   DD      2B               STD    LINNUM
D48D   5F                       CLRB
D48E   8E    02DC               LDX    #BUF-1      KEYBOARD BUFFER

D491   5C             HOP133    INCB
D492   A6 C0                    LDA    ,U+
D494   A7 80                    STA    ,X+
D496   26      F9 D491          BNE    HOP133

D498   CB      04               ADDB   #$04
D49A   D7      03               STB    BCOUNT
D49C   34      40               PSHS   U
D49E   8D      0D D4AD          BSR    ADDLIN
D4A0   35      40               PULS   U
D4A2   20      DE D482          BRA    MERGLP
D4A4   7F    0611       MRGXIT   CLR    RLFLAG
D4A7   16    0087 D531          LBRA   LOADFX
D4AA   7E    B848       BFMERR   JMP    FMERR       BAD FILE MODE
                        ********************************
```

```
                        ********************************
                        *
                        * ADD A LINE TO BASIC PROGRAM
                        *
D4AD  BD   83FF     ADDLIN  JSR   FINLIN      FIND LINE #
D4B0  25    12 D4C4         BCS   INSLIN
D4B2  DC    47              LDD   LOWTR
D4B4  A3 84                 SUBD  ,X
D4B6  D3    1B              ADDD  VARTAB
D4B8  DD    1B              STD   VARTAB
D4BA  EE 84                 LDU   ,X
D4BC  A6 C0         HOP134  LDA   ,U+
D4BE  A7 80                 STA   ,X+
D4C0  9C    1B              CMPX  VARTAB
D4C2  26    F8 D4BC         BNE   HOP134
D4C4  DC    1B      INSLIN  LDD   VARTAB
D4C6  DD    43              STD   HIGHTR
D4C8  DB    03              ADDB  BCOUNT
D4CA  89    00              ADCA  #0
D4CC  DD    41              STD   HIGHDS
D4CE  BD   831C             JSR   BLTU        MAKE ROOM
D4D1  CE   02D8             LDU   #BUFLNM-2

D4D4  A6 C0         HOP135  LDA   ,U+
D4D6  A7 80                 STA   ,X+
D4D8  9C    45              CMPX  LOWDS
D4DA  26    F8 D4D4         BNE   HOP135
D4DC  9E    41              LDX   HIGHDS
D4DE  9F    1B              STX   VARTAB

D4E0  7E   83ED     INS03   JMP   CHEAD       FIX LINKS
                        ********************************
```

```
                        ********************************
                        *
                        * CHECK FILE HEADER BLOCK
                        * WILL NOT RETURN IF AN ERROR
                        * OCCURES IN THE DISK ROUTINES
                        * OF IF THE HEADER BLOCK IS
                        * INVALID.
                        *
                        * CALL WITH
                        *  FIBNO = FILE TO READ
                        *
                        *  RETURN WITH
                        *   A = FILE TYPE
                        *  [X]= FILE HEADER BLOCK
                        *
D4E3   8E    0650       CHKHDR   LDX    #TNAME
D4E6 108E    0009                LDY    #9
D4EA   DE      8A                LDU    DBLZER
D4EC   5F                        CLRB
D4ED   96      F1                LDA    FIBND
D4EF   17    F522 CA14           LBSR   READ
D4F2   26      11 D505           BNE    HOME18
D4F4   86      55                LDA    #$55
D4F6   8E    0650                LDX    #TNAME
D4F9   A1 84                     CMPA   0,X
D4FB   26      AD D4AA           BNE    BFMERR
D4FD   43                        COMA
D4FE   A1 08                     CMPA   8,X
D500   26      A8 D4AA           BNE    BFMERR
D502   A6 01                     LDA    1,X
D504   5F                        CLRB
D505   39                HOME18   RTS
                        ********************************
```

```
                        ********************************
                        *
                        * LOAD A BASIC OR BINARY PROGRAM
                        * INTO MEMORY
                        *
                        * IF BASIC :
                        *  ANY PROGRAM ALREADY IN MEMORY
                        *  WILL BE OVER WRITTEN AND ALL
                        *  VARIABLES CLEARED.
                        *
                        * IF BINARY :
                        *  THE PROGRAM WILL BE READ INTO
                        *  ANY MEMORY LOCATION BELOW
                        *  THE BASIC ROM.
                        *
                        * SYNTAX :
                        *  RUN "filespec"
                        *  LOAD "filespec"
                        *
D506  7F  0614      RUN       CLR   ONERRF      ERROR TRAP
D509  7F  0619                CLR   ERRNUM      NO ERROR
D50C  7F  0617                CLR   ERLINE      NO LINE #
D50F  7F  0618                CLR   ERLINE+1
D512  81    22                CMPA  #$22        QUOTE
D514  27    02 D518           BEQ   HOP136
D516  4D                      TSTA
D517  39                      RTS
D518  C6    01      HOP136    LDB   #1
D51A                          SKIP1
D51B  5F            LOAD      CLRB
D51C  F7  0611                STB   RLFLAG      RUN/LOAD FLAG
D51F  17  01F6 D718           LBSR  GNMBAS
D522  26    04 D528           BNE   NEWER9

                        * GET HEADER BLOCK
D524  8D    BD D4E3           BSR   CHKHDR
D526  27    03 D52B           BEQ   HOP137
D528  7E  C718      NEWER9    JMP   NEWERR
D52B  81    01      HOP137    CMPA  #$01        BASIC PROGRAM
D52D  26    47 D576           BNE   LDBIN
D52F  8D    23 D554           BSR   LOADBS
D531  0F    F6      LOADFX    CLR   LOCK
D533  9E    19                LDX   TXTTAB
D535  BD  85EE                JSR   DEXPTR
D538  9E    27                LDX   MEMSIZ
D53A  9F    23                STX   FRETOP
D53C  9E    1B                LDX   VARTAB
D53E  9F    1D                STX   ARYTAB
D540  9F    1F                STX   STREND
D542  BD  8514      CHAINX    JSR   RESTOR
D545  BD  8434                JSR   STKINI
D548  7D  0611                TST   RLFLAG
D54B  27    03 D550           BEQ   GOIDLE
D54D  7E  849F                JMP   NEWSTT       RUN IT
```

```
D550  4F            GOIDLE  CLRA
D551  7E  8371              JMP   READY

                    ********************************
                    * READ A BASIC PROGRAM
D554  EC 04         LOADBS  LDD   4,X         GET LENGTH
D556  1F    02              TFR   D,Y
D558  D3    19              ADDD  TXTTAB      +WHERE LOAD
D55A  DD    1F              STD   ARRTOP
D55C  C6    40              LDB   #$40        # OF WORDS
D55E  BD  8331              JSR   GETSTK      ?OM?
D561  96    F1      LDBAS   LDA   FIBNO
D563  C6    09              LDB   #9          BYTE # IN FSN
D565  9E    19              LDX   TXTTAB      READ BUFFER
D567  DE    8A              LDU   DBLZER
D569  17  F4A8 CA14          LBSR  READ
D56C  26    BA D528         BNE   NEWER9

D56E  BD  83ED              JSR   CHEAD       FIX LINKS
D571  30 02                 LEAX  2,X
D573  9F    1B              STX   VARTAB
D575  39                    RTS

                    ********************************
                    * LOAD A BINARY FILE
D576  81    02      LDBIN   CMPA  #$02        BIN FILE TYPE
D578 1026  FF2E D4AA        LBNE  BFMERR
D57C  EE 06                 LDU   6,X         GET XFER ADDRESS
D57E  DF    9D              STU   EXECAD
D580  9D    A5              JSR   CHRGOT
D582  27    12 D596         BEQ   HOP138
D584  34    10              PSHS  X
D586  8D    6F D5F7         BSR   GETADR
D588  1F    13              TFR   X,U
D58A  35    10              PULS  X
D58C  EC 06                 LDD   6,X
D58E  A3 02                 SUBD  2,X
D590  EF 02                 STU   2,X
D592  E3 02                 ADDD  2,X
D594  DD    9D              STD   EXECAD
D596 10AE 04        HOP138  LDY   4,X
D599  96    F1              LDA   FIBNO
D59B  C6    09              LDB   #9
D59D  DE    8A              LDU   DBLZER
D59F  AE 02                 LDX   2,X         GET LOAD ADDRESS
D5A1  17  F470 CA14         LBSR  READ
D5A4  26    48 D5EE         BNE   NEWER7
D5A6  7D  0611              TST   RLFLAG
D5A9  27    4B D5F6         BEQ   HOME19
D5AB  6E 9F 009D            JMP   [EXECAD]

                    ********************************
```

```
                        ********************************
                        *
                        * SAVE A BASIC OR BINARY PROGRAM
                        * TO DISK.
                        *
                        * IF BASIC:
                        *  SYNTAX:
                        *   SAVE "filespec"
                        *
                        * IF BINARY:
                        *  SYNTAX:
                        *   SAVE "FILESPEC",ba,ea,xa
                        *    ba = BEGINNING ADDRESS
                        *    ea = ENDING ADDRESS
                        *    xa = TRANSFER ADDRESS
                        *
D5AF   BD    8887       SAVE     JSR    FRMEVL
D5B2   BD    8877                JSR    CHKSTR
D5B5   9D      A5                JSR    CHRGOT
D5B7   27      44 D5FD           BEQ    SAVBSC
                        * GET FILE NAME & FIB
D5B9 108E    DFA6                LDY    #XNTBIN
D5BD   8D      26 D5E5           BSR    SAVNAM
                        * GET STARTING ADDRESS
D5BF   8D      36 D5F7           BSR    GETADR
D5C1   BF    0652                STX    TNAME+2

                        * GET THE ENDING ADDRESS
D5C4   8D      31 D5F7           BSR    GETADR
D5C6   1F      10                TFR    X,D
D5C8   BC    0652                CMPX   TNAME+2
D5CB 1025    F147 C716           LBLO   ER.PRM

                        * MAKE IT A LENGTH
D5CF   B3    0652                SUBD   TNAME+2
D5D2 102B    B5B7 8B8D           LBMI   FCERR
D5D6   C3    0001                ADDD   #$1
D5D9   FD    0654                STD    TNAME+4

                        * GET THE ENTRY POINT
D5DC   8D      19 D5F7           BSR    GETADR
D5DE   BF    0656                STX    TNAME+6
D5E1   C6      02                LDB    #$02        BIN FILE TYPE
D5E3   20      32 D617           BRA    BLDHDR

D5E5   17    0144 D72C SAVNAM    LBSR   GNMKLD
D5E8   27      07 D5F1           BEQ    HOP139
D5EA   C1      A0                CMPB   #ERR.NE
D5EC   27      03 D5F1           BEQ    HOP139
D5EE   7E    C718       NEWER7   JMP    NEWERR
D5F1   17    F99D CF91  HOP139   LBSR   CREATE
D5F4   26      F8 D5EE           BNE    NEWER7
D5F6   39               HOME19   RTS
D5F7   BD    89AA       GETADR   JSR    CHKCOM
```

```
D5FA   7E    8E83            JMP    GETWRD

                    ********************************
                    * GET ADDRESS'S FOR SAVE BASIC
                    *  TNAME+2 = START OF PROGRAM
                    *  TNAME+4 = LENGTH OF PROGRAM
                    *  TNAME+6 = ENTRY POINT FOR BIN
                    *
D5FD 108E    DFA0   SAVBSC  LDY    #XNTBAS
D601   8D      E2 D5E5       BSR    SAVNAM
                    *
D603   9E      19            LDX    TXTTAB      START
D605   BF    0652            STX    TNAME+2

                    * GET LENGHT
D608   DC      1B            LDD    VARTAB
D60A   93      19            SUBD   TXTTAB
D60C   FD    0654            STD    TNAME+4

D60F   8E    8B8D            LDX    #FCERR
D612   BF    0656            STX    TNAME+6
D615   C6      01            LDB    #$01

                    * BUILD THE HEADER BLOCK
D617   8E    0650   BLDHDR   LDX    #TNAME      GET A BUFFER
D61A   86      55            LDA    #$55
D61C   A7 84                 STA    0,X         HEADER FLAG
D61E   43                    COMA
D61F   A7 08                 STA    8,X         HEADER FLAG
D621   E7 01                 STB    1,X         FILE TYPE

                    * WRITE THE HEADER BLOCK
D623   96      F1            LDA    FIBNO
D625   5F                   CLRB
D626 109E      8A            LDY    DBLZER
D629   CE    0009            LDU    #9
D62C   17    F5A7 CBD6       LBSR   WRITE
D62F   26      BD D5EE       BNE    NEWER7
D631   96      F1            LDA    FIBNO
D633   C6      09            LDB    #9
D635   BE    0652            LDX    TNAME+2     START
D638   FE    0654            LDU    TNAME+4     LENGTH
D63B 109E      8A            LDY    DBLZER
                    *  A  = FILE NUMBER
                    * [X] = BUFFER ADDRESS
                    *  U  = NUMBER OF BYTES
                    *  Y  = FILE SECTOR NUMBER
                    *  B  = BYTE NUMBER IN FILE
D63E   17    F595 CBD6       LBSR   WRITE
D641   26      AB D5EE       BNE    NEWER7
D643   0F      F6    SAVDON   CLR    LOCK
D645   39            SAVRTS   RTS

                    ********************************
```

```
                        ********************************
                        *
                        * BASIC CHAIN COMMAD
                        *
                        * SYNTAX:
                        *  CHAIN "filespec"
                        *  CHAIN "filespec",line number
                        *
                        *
                        * THIS COMMAND WILL SAVE ALL
                        * OF THE USERS SIMPLE VARIABLES,
                        * ARRAYS, AND STRING VARIABLES
                        * REINIT THE STACK SO TO FLUSH
                        * ANY PENDING RETURNS OR
                        * FOR-NEXT LOOPS.
                        *
                        * IT WILL ALSO RESET THE
                        * READ-DATA POINTER BACK TO THE
                        * BEGININNG OF THE PROGRAM.
                        *
D646   8D      75 D6BD CHAIN    BSR    GARBAG
D648   17    00CD D718          LBSR   GNMBAS
D64B   26      A1 D5EE          BNE    NEWER7
D64D   17    FE93 D4E3          LBSR   CHKHDR
D650   26      9C D5EE          BNE    NEWER7
D652   81      01               CMPA   #$01
D654 1026    FE52 D4AA          LBNE   BFMERR
D658   9D      A5               JSR    CHRGOT
D65A   27      0F D66B          BEQ    HOP140
D65C   BD    89AA               JSR    CHKCOM
D65F   BD    869A               JSR    LINGET
D662   8D      12 D676          BSR    CAIN
D664   DC      2B               LDD    LINNUM
D666   BD    85E7               JSR    LUKALL
D669   20      08 D673          BRA    HOP141
                        *
D66B   8D      09 D676 HOP140   BSR    CAIN
D66D   DE      19               LDU    TXTTAB
D66F   33 5F                    LEAU   -1,U
D671   DF      A6               STU    TXTPTR
D673   7E    D542      HOP141   JMP    CHAINX

                        *******
                        * DO THE ACTUAL CHAIN
D676   FC    0654      CAIN     LDD    TNAME+04
D679   1F      02               TFR    D,Y
D67B   D3      19               ADDD   TXTTAB
D67D   93      1B               SUBD   VARTAB
D67F   34      06               PSHS   D
D681   D3      1F               ADDD   ARRTOP
D683   DD      1F               STD    ARRTOP

                        ******
                        * CHECK FOR OUT OF MEMORY
```

```
D685  C6    40          LDB   #$40       CHECK FOR $40 WORDS OF SPACE
D687  F7  0611          STB   RLFLAG     RUN IT TOO
D68A  BD  8331          JSR   GETSTK
                  ******
                  *
D68D  EC E4             LDD   ,S
D68F  2A    0F D6A0     BPL   CHNUP

                  ******
                  * THIS IS TO MOVE THE VARS DOWN
D691  9E    1B    CHNDWN  LDX   VARTAB
D693  33 8B             LEAU  D,X
D695  A6 80       HOP142  LDA   ,X+
D697  A7 C0             STA   ,U+
D699 1193    1F         CMPU  ARRTOP
D69C  23    F7 D695     BLS   HOP142
D69E  20    0E D6AE     BRA   DOCHN

                  ******
                  * THIS IS TO MOVE THE VARS UP
D6A0  9E    1F    CHNUP   LDX   ARRTOP
D6A2  30 01             LEAX  1,X
D6A4  33 8B             LEAU  D,X
D6A6  A6 82       HOP143  LDA   ,-X
D6A8  A7 C2             STA   ,-U
D6AA  9C    1B          CMPX  VARTAB
D6AC  24    F8 D6A6     BHS   HOP143
D6AE  EC E4       DOCHN   LDD   ,S
D6B0  D3    1B          ADDD  VARTAB
D6B2  DD    1B          STD   VARTAB
D6B4  35    06          PULS  D
D6B6  D3    1D          ADDD  ARRTAB
D6B8  DD    1D          STD   ARRTAB
D6BA  7E  D561          JMP   LDBAS      LOAD PROGRAM

                  *********************************
```

```
                        ********************************
                        *
                        * CHAIN GARBAGE COLLECTOR
                        *
D6BD   9E      1B       GARBAG   LDX    VARTAB
D6BF   30 02            VARLOP   LEAX   2,X
D6C1   9C      1D       VARLOP   CMPX   ARYTAB
D6C3   24      0A D6CF           BHS    DO.ARY
D6C5   6D 1F                     TST    -1,X
D6C7   2A      02 D6CB           BPL    VARSKP
D6C9   8D      2C D6F7           BSR    FIXKLD
D6CB   30 07            VARSKP   LEAX   7,X
D6CD   20      F2 D6C1           BRA    VARLOP
                        * NOW FOR THE ARRAYS
D6CF   DE      1D       DO.ARY   LDU    ARYTAB
D6D1 1193      1F       ARYLOP   CMPU   STREND
D6D4   24      1E D6F4           BHS    DO.GON
D6D6   EC 42                     LDD    2,U
D6D8   30 CB                     LEAX   D,U
D6DA   34      10                PSHS   X
D6DC   6D 41                     TST    1,U
D6DE   2A      10 D6F0           BPL    ARYSKP
D6E0   E6 44                     LDB    4,U
D6E2   4F                        CLRA
D6E3   58                        ASLB
D6E4   30 CB                     LEAX   D,U
D6E6   30 05                     LEAX   5,X
D6E8   8D      0D D6F7 ARYLP1    BSR    FIXKLD
D6EA   30 05                     LEAX   5,X
D6EC   AC E4                     CMPX   0,S
D6EE   25      F8 D6E8           BLO    ARYLP1
D6F0   35      40       ARYSKP   PULS   U
D6F2   20      DD D6D1           BRA    ARYLOP
D6F4   7E    8CD7       DO.GON   JMP    GARBA2

D6F7   34      50       FIXKLD   PSHS   X,U
D6F9   9C      21                CMPX   STKTOP
D6FB   24      13 D710           BHS    HOP144
D6FD   E6 84                     LDB    ,X
D6FF   27      0F D710           BEQ    HOP144
D701   BD    8CB3               JSR    GETSPA
D704   1F      13               TFR    X,U
D706 10AE E4                     LDY    0,S
D709   AE 22                     LDX    2,Y
D70B   EF 22                     STU    2,Y
D70D   BD    B7CC               JSR    BLKMOV
D710   35      D0       HOP144   PULS   X,U,PC
                        ********************************
                                 END
```

```
                    NAM             DISKFILE
```

```
                        ********************************
D712 108E   DFA3        GNMDAT  LDY     #XNTDAT
D716   20     0A D722          BRA     HOP145
D718 108E   DFA0        GNMBAS  LDY     #XNTBAS
D71C   20     04 D722          BRA     HOP145
D71E 108E   DFA9        GNMNUL  LDY     #XNTNUL
D722   34     20        HOP145  PSHS    Y
D724   BD   8887                JSR     FRMEVL
D727   BD   8877                JSR     CHKSTR
D72A   35     20                PULS    Y
D72C   9E     52        GNMKLD  LDX     FAC
D72E   34     10                PSHS    X
D730   E6 84                    LDB     ,X
D732   AE 02                    LDX     2,X
D734   17   F100 C837           LBSR    LOOKUP
D737   35     10                PULS    X
D739   34     06                PSHS    D
D73B   BD   8D9F                JSR     FRETMP
D73E   35     06                PULS    D
D740   5D                      TSTB
D741   39                      RTS

                        ********************************
```

```
                        ********************************
                        *
                        * CLOSE 1 OR ALL OPEN FILES
                        *
                        * SYNTAX:
                        *  CLOSE
                        *  CLOSE d
                        *
D742  86     0A         HKCLOS  LDA   #10
D744  97     F1                 STA   FIBNO
D746  17   F807 CF50    HOP146  LBSR  CLOS1
D749  26     19 D764            BNE   NEWER2
D74B  0A     F1                 DEC   FIBNO
D74D  2A     F7 D746            BPL   HOP146
D74F  0F     F6         CLOSEX  CLR   LOCK
D751  39                HOME20  RTS

D752  2F     FD D751    HKCLO1  BLE   HOME20
D754  AE E4                     LDX   ,S
D756  8C   B7EE                 CMPX  #CLOSRT
D759  26     F6 D751            BNE   HOME20
D75B  32 62                     LEAS  2,S
D75D  D7     EB                 STB   DD.UNT
D75F  17   F7D6 CF38            LBSR  CLOSAL
D762  27     EB D74F            BEQ   CLOSEX
D764  7E   C718         NEWER2  JMP   NEWERR
D767  8D     D9 D742    DOCLOS  BSR   HKCLOS
D769  7E   DB62                 JMP   AUTHCK

                        ********************************
```

```
                        ********************************
                        * CREATE
                        *
                        * SYNTAX :
                        *  CREATE "filespec",LENGTH
                        *
D76C  8D     A4 D712 DCREAT  BSR    GNMDAT
D76E  27     08 D778         BEQ    HOP147
D770  C1     9E              CMPB   #ERR.FE
D772  27     04 D778         BEQ    HOP147
D774  C1     A0              CMPB   #ERR.NE
D776  26     EC D764         BNE    NEWER2
                        *
                        * CREATE THE FILE
D778  96     F1      HOP147  LDA    FIBNO
D77A  17   F814 CF91         LBSR   CREATE
D77D  26     E5 D764         BNE    NEWER2
                        *
                        * UNFLOAT THE NUMBER IN FAC
D77F  9D     A5              JSR    CHRGOT
D781  27     36 D7B9         BEQ    DCEXT
D783  BD   89AA              JSR    CHKCOM
D786  BD   8887              JSR    FRMEVL
D789  17   013F D8CB         LBSR   UNFLOT

                        * ALLOCAT SPACE TO IT
D78C  0D     51      HOP148  TST    FAC-1
D78E  27     12 D7A2         BEQ    HOP149
D790  CC   FF00              LDD    #$FF00
D793  8D     1F D7B4         BSR    DOEXTN
D795  DC     52              LDD    FAC
D797  83   FF00              SUBD   #$FF00
D79A  DD     52              STD    FAC
D79C  24     EE D78C         BCC    HOP148
D79E  0A     51              DEC    FAC-1
D7A0  26     EA D78C         BNE    HOP148
D7A2  DC     52      HOP149  LDD    FAC
D7A4  27     13 D7B9         BEQ    DCEXT
D7A6 1083  FF00              CMPD   #$FF00
D7AA  23     08 D7B4         BLS    DOEXTN
D7AC  83   FF00              SUBD   #$FF00
D7AF  8D     03 D7B4         BSR    DOEXTN
D7B1  CC   FF00              LDD    #$FF00
D7B4  17   F4B2 CC69 DOEXTN  LBSR   EXTEND
D7B7  26     AB D764         BNE    NEWER2
D7B9  39                DCEXT   RTS
                        ********************************
```

```
                        ********************************
                        *
                        * REMOVE A FILE FROM THE DISK
                        * DIRECTORY. DE-ALLOCATE ANY
                        * SPACE ON THE DISK FOR THIS
                        * FILE
                        *
                        * SYNTAX:
                        *  KILL "FILESPEC"
                        *
D7BA   17    FF61 D71E KILL     LBSR   GNMNUL      GET NAME
D7BD   26      05 D7C4          BNE    NEWER4
D7BF   17    F871 D033          LBSR   DELETE      DELETE IT
D7C2   27      8B D74F          BEQ    CLOSEX
D7C4   7E    C718       NEWER4  JMP    NEWERR
                        ********************************
```

```
                        ********************************
                        *
                        * PROTECT
                        *
                        * SYNTAX
                        *  PROTECT "filespec" (TURNS ON PROTECTION)
                        *  PROTECT ON "filespec"
                        *  PROTECT OFF "filespec"
                        *
D7C7   4D               PROTCT   TSTA
D7C8   2A      0A D7D4           BPL    PROTON      DEFAULT ON
D7CA   81      C2                CMPA   #OFFTOK
D7CC   27      19 D7E7           BEQ    PROTOF
D7CE   81      88                CMPA   #ONTOK
D7D0 1026    B1E0 89B4           LBNE   SNERR

D7D4   8D      08 D7DE PROTON    BSR    PROTIV
D7D6   C6      01                LDB    #1
D7D8   17    F93E D119 SPROT     LBSR   PROTEC
D7DB   26      E7 D7C4           BNE    NEWER4
D7DD   39                        RTS

D7DE   17    28BE 009F PROTIV    LBSR   GETCHR
D7E1   17    FF3A D71E           LBSR   GNMNUL      FILE NAME
D7E4   26      25 D80B           BNE    RENERR
D7E6   39                        RTS

D7E7   8D      F5 D7DE PROTOF    BSR    PROTIV
D7E9   5F                        CLRB
D7EA   20      EC D7D8           BRA    SPROT

                        ********************************
```

```
                        ********************************
                        *
                        * RENAME A DISK FILE
                        *
                        * SYNTAX
                        *  RENAME "filespec" TO "filepec"
                        *
D7EC  17  FF2F D71E RENAME  LBSR  GNMNUL      GET NAME
D7EF  26    1A D80B         BNE   RENERR
D7F1  34    02              PSHS  A          SAVE FIB NUNBER

                        * LOOK FOR A TO TOKEN
D7F3  C6    BC              LDB   #TOTOK
D7F5  BD  89AC              JSR   SYNCHK

                        * GET SECOND NAME
D7F8  17  FF23 D71E         LBSR  GNMNUL
D7FB  27    0C D809         BEQ   ALREDY
D7FD  C1    A0              CMPB  #ERR.NE     NON-EXISTENT
D7FF  26    0A D80B         BNE   RENERR

                        * LETS RENAME IT
D801  35    02              PULS  A          GET FIRST FIB
D803  17  F945 D14B         LBSR  RNAME
D806  26    03 D80B         BNE   RENERR

                        * LETS GO AWAY HAPPY
D808  39                    RTS
                        * GO AWAY MAD
D809  C6    9E      ALREDY  LDB   #ERR.FE
D80B  7E  C718      RENERR  JMP   NEWERR
                        ********************************
```

```
                        ********************************
                        *
                        *
                        *
                        *
D80E  8D     0D D81D FLREAD  BSR   FRDSUP      GET FOR READ
D810  86     FF              LDA   #-1
D812  B7   0612              STA   RDTYPE
D815  17   0197 D9AF         LBSR  HKITM2      DO THE READ
D818  8D     30 D84A         BSR   UNREAD      FIX POINTERS
D81A  7E   9DD9              JMP   FILINE      FINISH OFF THE INPUT, STUFF IT INTO A VARIABLE
                        ********************************
D81D  17   FEF2 D712 FRDSUP  LBSR  GNMDAT
D820  26     52 D874         BNE   FRERR
D822  8D     53 D877         BSR   GPARM
D824  17   F702 CF29         LBSR  FIBADR
D827  7D   067E              TST   TEMP2
D82A  27     0A D836         BEQ   HOP150
D82C  FE   0664              LDU   WRBYTE
D82F  F6   0666              LDB   WRBYTE+2
D832  EF 0C                  STU   RDBYTE,X
D834  E7 0E                  STB   RDBYTE+2,X
D836  17   01E9 DA22 HOP150  LBSR  FREDRD
D839  8E   02DC              LDX   #BUF-1
D83C  6F 84                  CLR   ,X
D83E  17   285E 009F         LBSR  GETCHR
D841  39                     RTS
                        ********************************
```

```
                        ********************************
                        *
                        * FREAD "filespec",AT xx,FOR yy; varible list
                        *
                        *  0 ) xx ) 350,208
                        *  0 ) yy ) 255
                        *
D842  8D    D9 D81D FREAD   BSR   FRDSUP
D844  7F    0612            CLR   RDTYPE

                        ******
                        * GO SUB INPUT
D847  BD    877A            JSR   READAT      IN THE INPUT
                        ******
                        * GIVE WHAT WE DID'NT USE BACK
D84A  17    F6DC CF29 UNREAD  LBSR  FIBADR
                        * IS THERE ANYTHING LEFT ?
D84D  F6    0604            LDB   TEMP.L
D850  27      0E D860       BEQ   HOP151      -NO-
                        *
D852  7F    0603            CLR   TEMP.L-1    BYTES NOT USED
D855  EC 0D                 LDD   RDBYTE+1,X
D857  B3    0603            SUBD  TEMP.L-1
D85A  ED 0D                 STD   RDBYTE+1,X
D85C  24      02 D860       BCC   HOP151
D85E  6A 0C                 DEC   RDBYTE,X
D860  0D      F4      HOP151  TST   FWFLAG
D862  27      0D D871       BEQ   HOP152
D864  D6      F3            LDB   FWFOR
D866  27      09 D871       BEQ   HOP152
D868  4F                    CLRA
D869  E3 0D                 ADDD  RDBYTE+1,X
D86B  ED 0D                 STD   RDBYTE+1,X
D86D  24      02 D871       BCC   HOP152
D86F  6C 0C                 INC   RDBYTE,X
D871  0F      6F      HOP152  CLR   DEVNUM
D873  39                    RTS

D874  7E    C718      FRERR   JMP   NEWERR
                        ********************************
D877  BD    9C76      GPARM   JSR   ERRDIR      ARE WE IN DIRECT MODE
D87A  96      F1            LDA   FIBNO
D87C  17    F677 CEF6       LBSR  LENFIL
D87F  26      F3 D874       BNE   FRERR       ?ERROR?
D881  FF    0664            STU   WRBYTE
D884  B7    0666            STA   WRBYTE+2
D887  0F      F4            CLR   FWFLAG
D889  7F    067E            CLR   TEMP2
D88C  86      01            LDA   #1
D88E  97      6F            STA   DEVNUM

D890  9D      A5      HOP153  JSR   CHRGOT
D892  81      3B            CMPA  #';
D894  27      52 D8E8       BEQ   HOME21
```

```
D896  BD   89AA            JSR   CHKCOM
D899  81     E5            CMPA  #ATTOK
D89B  26     14 D8B1       BNE   FORLEN
                    ******
                    * WHERE TO WRITE
D89D  9D     9F            JSR   GETCHR
D89F  BD   8887            JSR   FRMEVL
D8A2  8D     27 D8CB       BSR   UNFLOT
D8A4  FF   0664            STU   WRBYTE
D8A7  B7   0666            STA   WRBYTE+2
D8AA  86     FF            LDA   #-1         SAY WE WILL USE THIS
D8AC  B7   067E            STA   TEMP2
D8AF  20     DF D890       BRA   HOP153
                    ******
                    * GET THE LENGTH
D8B1  81     80    FORLEN  CMPA  #FORTOK
D8B3  26     0D D8C2       BNE   UNSNER
D8B5  9D     9F            JSR   GETCHR
D8B7  17   EDFF C6B9       LBSR  NZRBYT
D8BA  D7     F3            STB   FWFOR
D8BC  C6     FF            LDB   #-1
D8BE  D7     F4            STB   FWFLAG
D8C0  20     CE D890       BRA   HOP153

D8C2  7E   89B4    UNSNER  JMP   SNERR
D8C5  7E   8B8D    UNFCER  JMP   FCERR
D8C8  7E   8882    UNTMER  JMP   TMERR
                   ********************************
D8CB  0D     54    UNFLOT  TST   FAC+2
D8CD  2B     F6 D8C5       BMI   UNFCER
D8CF  0D     06            TST   STRFLG
D8D1  26     F5 D8C8       BNE   UNTMER
D8D3  86     A0            LDA   #$A0
D8D5  90     4F            SUBA  $4F
D8D7  27     0B D8E4       BEQ   HOP154
D8D9  04     50    UNFLT   LSR   $50
D8DB  06     51            ROR   $51
D8DD  06     52            ROR   $52
D8DF  06     53            ROR   $53
D8E1  4A                   DECA
D8E2  26     F5 D8D9       BNE   UNFLT
D8E4  DE     51    HOP154  LDU   $51
D8E6  96     53            LDA   $53
D8E8  39           HOME21  RTS

                   ********************************
```

```
                        ********************************
                        *
                        * FWRITE "filespec",AT xx,FOR yy; varaible list
                        *
                        *
                        *  0 ) xx ) 350,208
                        *  0 ) yy ) 255
                        *
D8E9  17   FE26 D712 FWRITE  LBSR  GNMDAT
D8EC  27     09 D8F7         BEQ   HOP155
D8EE  C1     A0              CMPB  #ERR.NE
D8F0  26     68 D95A         BNE   FNWERR
D8F2  17   F69C CF91         LBSR  CREATE
D8F5  26     63 D95A         BNE   FNWERR
D8F7  17   FF7D D877 HOP155  LBSR  GPARM
D8FA  17   F9F3 D2F0         LBSR  FINDMT
D8FD  26     5B D95A         BNE   FNWERR
D8FF  BF   060B              STX   FWBUF       POINTER
                        * MAKE BUFFER UN-USABLE.
D902  86     55              LDA   #$55
D904  A7 02                  STA   PAGVLD,X
D906  0F     F2              CLR   FWBYTE      BUFFER EMPTY
                        ******
D908  9D     9F              JSR   GETCHR
D90A  BD   903D              JSR   PRINT       WRITE IT
D90D  0D     F4              TST   FWFLAG
D90F  27     0B D91C         BEQ   HOP157
D911  D6     F3              LDB   FWFOR       # LEFT
D913  27     07 D91C         BEQ   HOP157
D915  86     20              LDA   #$20        SPACE
D917  8D     4A D963 HOP156  BSR   FINHOK
D919  5A                     DECB
D91A  26     FB D917         BNE   HOP156
                        *
D91C  0D     F2      HOP157  TST   FWBYTE
D91E  27     18 D938         BEQ   FWRIXT
D920  BE   060B              LDX   FWBUF
D923  AE 05                  LDX   PAGADR,X    X=BUFFER
D925  4F                     CLRA
D926  D6     F2              LDB   FWBYTE
D928  1F     03              TFR   D,U         U=# OF BYTES
D92A  96     F1              LDA   FIBNO       A=FIB NUBER
D92C 10BE  0664              LDY   WRBYTE      Y=FSN
D930  F6   0666              LDB   WRBYTE+2    B=BYTE WITHIN FSN
D933  17   F2A0 CBD6         LBSR  WRITE
D936  26     22 D95A         BNE   FNWERR
D938  BE   060B      FWRIXT  LDX   FWBUF
D93B  6F 02                  CLR   PAGVLD,X
D93D  39             HOME44  RTS

                        ********************************
                                END
                                LIB             FILEHOOK
                                NAM             FILEHOOK
```

```
                        ********************************
                        * REVISED SEPTEMBER 14, 1983
                        ********************************
                        * CHECK VALIDITY OF DEVICE NUMBER
                        *
D93E  2F      FD D93D HKDNCK  BLE   HOME44      NOT DISK
D940  C1      04              CMPB  #4          WE ONLY USE #1
D942  22      14 D958         BHI   HKDNER
D944  35      90              PULS  X,PC        SLEW HOOK RETURN ADDRESS
                        ********************************
                        *
                        * CHECK FOR DEVICE OPEN
                        *  IF DEVICE NUMBER IS GREATER
                        *  THEN ZERO THEN AN DN ERROR
                        *  IS CALLED FOR.
                        *
D946  32 62             HKOPEN  LEAS  2,S
D948  BD    8887                JSR   FRMEVL
D94B  BD    8DEA                JSR   ASC2
D94E  34      04                PSHS  B
D950  BD    B7D4                JSR   GETCON
D953  5D                        TSTB
D954 102F   DEDD B835           LBLE  OPNDEV
D958  C6      28        HKDNER  LDB   #ERR.DN
D95A  7E    C718        FNWERR  JMP   NEWERR
                        ********************************
```

```
                        ********************************
                        * CHARACTER OUTPUT HOOK.
                        *
                        * THIS IS THE HOOK ROUTINE FOR
                        * WRITEING A SINGLE BYTE TO A
                        * DATA FILE.
                        *
D95D  0D      6F        HKCHRO  TST     DEVNUM
D95F  2F      DC D93D           BLE     HOME44      BRA IF (=0
D961  32 62                     LEAS    2,S         SLEW RETURN ADDRESS
D963  34      76        FINHOK  PSHS    D,X,U,Y     SAVE THIS
D965  D6      F4                LDB     FWFLAG
D967  27      04 D96D           BEQ     HOP158      NO LENGTH
D969  D6      F3                LDB     FWFOR
D96B  27      37 D9A4           BEQ     HKOUT       WRITE NO MORE
D96D  BE    060B        HOP158  LDX     FWBUF
D970  AE 05                     LDX     PAGADR,X
D972  D6      F2                LDB     FWBYTE
D974  3A                        ABX
D975  A7 84                     STA     ,X
D977  0A      F3                DEC     FWFOR       COUNT DOWN
D979  0C      F2                INC     FWBYTE
D97B  26      27 D9A4           BNE     HKOUT

                        ******
                        * WRITE THE BUFFER, ITS FULL
D97D  96      F1                LDA     FIBNO
D97F  BE    060B                LDX     FWBUF
D982  AE 05                     LDX     PAGADR,X
D984  CE    0100                LDU     #$100
D987 10BE   0664                LDY     WRBYTE
D98B  F6    0666                LDB     WRBYTE+2
D98E  BD    CBD6                JSR     WRITE
D991  27      03 D996           BEQ     HOP159
D993  7E    C718                JMP     NEWERR
                        ******
                        * UPDATE WRITE POINTER
D996  FC    0665        HOP159  LDD     WRBYTE+1
D999  C3    0100                ADDD    #$100
D99C  24      03 D9A1           BCC     HOP160
D99E  7C    0664                INC     WRBYTE
D9A1  FD    0665        HOP160  STD     WRBYTE+1
D9A4  35      F6        HKOUT   PULS    D,X,U,Y,PC

                        ********************************
```

```
                        ********************************
                        * HOOK TO SCAN FOR INPUT FROM
                        * DISK.
                        *
                        * SHOULD DO:
                        *  IF STRING:
                        *   DO NOT SLEW LEADING SPACES
                        *   TERMINATE INPUT ON:
                        *      $0D - CR
                        *      $2C - COMMA
                        *      EOF
                        *
                        *  IF NUMBERIC:
                        *   SLEW LENDING SPACES
                        *   TERMINATE ON:
                        *      ANY CHARACTER LESS THAN
                        *      OR EQUAL TO A SPACE
                        *
D9A6  0D      6F        HKITMS  TST   DEVNUM
D9A8  2F      63 DA0D         BLE   HOME40
D9AA  8E    879A              LDX   #READRT     RETURN ADDR
D9AD  AF E4                   STX   ,S

D9AF  96      F4        HKITM2  LDA   FWFLAG
D9B1  27      04 D9B7         BEQ   HOP161
D9B3  96      F3              LDA   FWFOR
D9B5  27      4D DA04         BEQ   HKEXT       READ NO MORE!
D9B7  8E    02DD        HOP161  LDX   #BUF
                        * POINT AT NEXT BYTE IN BUFFER
D9BA  D6      F2              LDB   FWBYTE
D9BC  F7    0603              STB   TEMP.C
D9BF  3A                      ABX
                        * SAVE # OF BYTES IN BUFFER
D9C0  F6    0604              LDB   TEMP.L
D9C3  F7    0611              STB   RLFLAG
                        *
D9C6  5F                      CLRB
D9C7  E7 1F                   STB   -1,X
D9C9  D7      00              STB   BRKFLG      CLEAR FLAG

D9CB  0A      F3        HKLP    DEC   FWFOR
D9CD  7A    0604              DEC   TEMP.L
D9D0  0C      F2              INC   FWBYTE
D9D2  A6 80                   LDA   ,X+
D9D4  27      2E DA04         BEQ   HKEXT       ?DONE IF EOF
D9D6  81      0D              CMPA  #$0D        CR
D9D8  27      2A DA04         BEQ   HKEXT       ?DONE IF CR
D9DA  7D    0612              TST   RDTYPE
D9DD  26      08 D9E7         BNE   HOP162
D9DF  81      2C              CMPA  #',         COMMA
D9E1  27      21 DA04         BEQ   HKEXT       ?DONE IF COMMA
D9E3  81      3A              CMPA  #':         COLON
D9E5  27      1D DA04         BEQ   HKEXT       ?DONE IF COLON
D9E7  96      F4        HOP162  LDA   FWFLAG
```

```
D9E9  27    04 D9EF         BEQ   HOP163
D9EB  96    F3              LDA   FWFOR
D9ED  27    15 DA04         BEQ   HKEXT       READ NO MORE!
D9EF  5C            HOP163  INCB              # OF BYTES IN STRING
D9F0  C1    FF              CMPB  #$FF
D9F2  27    0E DA02         BEQ   HKEX2       ?DONE IF STRING FULL
D9F4  B6  0604              LDA   TEMP.L
D9F7  26    D2 D9CB         BNE   HKLP
                    ******
                    * READ A NEW BUFFER OF DATA
D9F9  D6    00              LDB   BRKFLG
D9FB  26    05 DA02         BNE   HKEX2
D9FD  8D    0F DA0E         BSR   HKREAD
D9FF  5F                    CLRB
DA00  20    C9 D9CB         BRA   HKLP

                    ******
                    * RETURN TO INPUT
DA02  30    01      HKEX2   INX
DA04  F6  0603      HKEXT   LDB   TEMP.C
DA07  6F 1F                 CLR   -1,X
DA09  8E  02DC              LDX   #BUF-1
DA0C  3A                    ABX
DA0D  39            HOME40  RTS

                    ********************************
```

```
                        ********************************
                        *
                        * READ IN A NEW BUFFER FOR FREAD
                        *
DA0E  BD   CF29   HKREAD  JSR   FIBADR
DA11  4F                  CLRA
DA12  F6   0611           LDB   RLFLAG
DA15  12                  NOP
DA16  34     06           PSHS  D
DA18  EC 0D               LDD   RDBYTE+1,X
DA1A  A3 E1               SUBD  ,S++        # OF BYTES
DA1C  ED 0D               STD   RDBYTE+1,X
DA1E  24     02 DA22      BCC   FREDRD
DA20  6A 0C               DEC   RDBYTE,X
DA22  34     60   FREDRD  PSHS  U,Y
DA24  CE   00FF           LDU   #$FF
DA27  8D     19 DA42      BSR   BYTADJ
DA29  CE   02DD           LDU   #BUF
DA2C  6F CB               CLR   D,U
DA2E  96     F1           LDA   FIBNO
DA30  E6 0E               LDB   RDBYTE+2,X
DA32  AE 0C               LDX   RDBYTE,X
DA34  1E     13           EXG   X,U
DA36  BD   CA14           JSR   READ
DA39  26     3D DA78      BNE   HKRDER
DA3B  8E   02DD           LDX   #BUF
DA3E  6F 1F               CLR   -1,X
DA40  35     E0           PULS  U,Y,PC

                        ********************************
```

```
                        ********************************
                        *
                        * ADJUST THE READ LENGTH SO
                        * THAT WE WILL NOT READ PAST
                        * THE END OF THE FILE.
                        *
                        * IF RDBYTE+READ LEN)FILLEN
                        *    THEN READ LEN=FILLEN-RDBYTE
                        *
                        * CALL WITH
                        *  U=READ LENGTH
                        *  X=FIB
                        *
                        * RETURN WITH
                        *  Y=ADJUSTED LENGTH
                        *A:B=ADJUSTED LENGTH
                        *  X=FIB
                        *
DA42  34      40        BYTADJ  PSHS   U
DA44  EC 0D                     LDD    RDBYTE+1,X
DA46  E3 E4                     ADDD   0,S
DA48  34      06                PSHS   D
DA4A  A6 0C                     LDA    RDBYTE,X
DA4C  89      00                ADCA   #0
DA4E  A0 88   10                SUBA   FILLEN,X
DA51  35      06                PULS   D
DA53  25      12 DA67           BLO    HOP164
DA55  22      1F DA76           BHI    HKPEER
DA57  A3 88   11                SUBD   FILLEN+1,X
DA5A  23      0B DA67           BLS    HOP164
DA5C  EC 88   11                LDD    FILLEN+1,X
DA5F  A3 0D                     SUBD   RDBYTE+1,X
DA61  27      13 DA76           BEQ    HKPEER
DA63  ED E4                     STD    0,S
DA65  03      00                COM    BRKFLG
DA67  EC E4           HOP164    LDD    0,S
DA69  F7    0604               STB    TEMP.L
DA6C  F7    0611               STB    RLFLAG
DA6F  7F    0603               CLR    TEMP.C
DA72  0F      F2                CLR    FWBYTE
DA74  35      A0                PULS   Y,PC

DA76  C6      9A      HKPEER    LDB    #ERR.PE
DA78  7E    C718      HKRDER    JMP    NEWERR
                        ********************************
                                END
                                LIB                DIRCMMD
                                NAM                DIRCMMD
```

```
                        ********************************
                        * DIRECTORY COMMAND
                        *
                        * DIR [UNIT]
                        *
DA7B  27    05 DA82 DIR     BEQ   DIRDF
DA7D  17  EC89 C709         LBSR  DRVNUM
DA80  20    03 DA85         BRA   DIRUOK

DA82  F6  060A      DIRDF   LDB   DEFDRV      DEFAULT
DA85  D7    EB      DIRUOK  STB   DD.UNT      SET DRIVE #
DA87  6F E2                 CLR   ,-S
DA89  5F                    CLRB
DA8A  BD  851B      DIRLOP  JSR   ISCNTC      USER INPUT
DA8D  17  F7C0 D250         LBSR  GETDIR
DA90  26    E6 DA78         BNE   HKRDER
DA92  A6 84                 LDA   ,X
DA94  85    08              BITA  #8
DA96  26    41 DAD9         BNE   DIRDDN
DA98  85    81              BITA  #$81
DA9A  26    35 DAD1         BNE   DIRNXT
DA9C  30    01              INX
                        ******
                        * PRINT NAME
DA9E  C6    08              LDB   #8
DAA0  8D    47 DAE9         BSR   DIRPR
                        ******
                        * PRINT EXTENSION
DAA2  C6    04              LDB   #4
DAA4  86    2E              LDA   #'.
DAA6  8D    47 DAEF         BSR   DIRPR2

                        ******
                        * PRINT PROTECTION STATUS
DAA8  A6 14                 LDA   -12,X
DAAA  85    02              BITA  #$02
DAAC  27    03 DAB1         BEQ   HOP165
DAAE  86    70              LDA   #'p
DAB0                        SKIP2
DAB1  86    20      HOP165  LDA   #$20       SPACE
DAB3  BD  B54A              JSR   CHROUT
DAB6  BD  90F5              JSR   VDSPAC

                        ******
                        * PRINT FILE SIZE
DAB9  CE  FFFF              LDU   #$FFFF
DABC  8E  06BD              LDX   #FIBS       X=NULL FIB
DABF  E6 E4                 LDB   ,S
DAC1  E7 88  1D             STB   DIRN,X      DIRECTORY ENTRY TO SCAN
DAC4  17  F0B0 CB77         LBSR  SCANXT
DAC7  26    AF DA78         BNE   HKRDER
DAC9  17  F44A CF16         LBSR  NUMFIX
DACC  8D    28 DAF6         BSR   PRFAC       PRINT THIS
```

```
                        ******
                        * NEXT LINE PLEASE
DACE   BD    90A1               JSR    VDCRLF

                        ******
                        * DONE YET ?
DAD1   6C E4            DIRNXT   INC    ,S
DAD3   E6 E4                     LDB    ,S
DAD5   C1      B4                CMPB   #180
DAD7   25      B1 DA8A           BLO    DIRLOP
DAD9   35      02       DIRDON   PULS   A
DADB   17    F6EB D1C9           LBSR   GETFRE
DADE   4F                        CLRA
DADF   1F      13                TFR    X,U
DAE1   8D      13 DAF6           BSR    PRFAC
DAE3   8E    DB0C                LDX    #FREMES-1
DAE6   7E    90E5                JMP    STROUT

DAE9   A6 80            DIRPR    LDA    ,X+
DAEB   26      02 DAEF           BNE    DIRPR2
DAED   86      20                LDA    #$20          SPACE
DAEF   BD    B54A       DIRPR2   JSR    CHROUT
DAF2   5A                        DECB
DAF3   26      F4 DAE9           BNE    DIRPR
DAF5   39                        RTS

                        ********************************
DAF6   DF      51        PRFAC    STU    $51
DAF8   97      53                STA    $53
DAFA   0F      54                CLR    $54
DAFC   0F      50                CLR    $50
DAFE   0F      63                CLR    $63
DB00   86      A0                LDA    #$A0
DB02   97      4F                STA    $4F           EXPONENT
DB04   BD    9165                JSR    NORMFC
DB07   BD    9587                JSR    UNPAC         UNPACK FAC
DB0A   7E    90E5                JMP    STROUT
                        ********************************
C DB0D 20 46 52 45      FREMES   FCC    ' FREE BYTES',$D,$D,0
                        ********************************
                                 END
                                 LIB                  DISKCMMD.A
                                 NAM                  DISKCMMD
```

```
                        ********************************
                        *
                        * BASIC COMMANDS
                        *
                        *       BACKUP
                        *       BOOT
                        *       CHAIN
                        *       CLOSE
                        *       COPY
                        *       CREATE
                        *       DIR
                        *       DORK
                        *       DRIVE
                        *       DSKINIT
                        *       KILL
                        *       LOAD
                        *       MERGE
                        *       PROTECT
                        *       RENAME
                        *       RUN
                        *       SAVE
                        *       SREAD
                        *       SWAP
                        *       SWRITE
                        *       VERIFY
                        *       XFER
                        *
                        ********************************
DB1B   9E      68       AUTO     LDX    CURLIN      IMMED MODE?
DB1D   30 01                     LEAX   1,X
DB1F 1026    B06A 8B8D           LBNE   FCERR       MUST BE $FFFF
DB23   8E    0064       L1       LDX    #100        SET UP DEFLTS
DB26   CC    000A                LDD    #10         ...INC AMT
DB29   34      16                PSHS   D,X
DB2B   9D      A5                JSR    CHRGOT      USER ARGS?
DB2D   27      1E DB4D           BEQ    L3          NO, PROCEED
DB2F   BD    8E83                JSR    GETWRD      GET ARG1
DB32   DC      52                LDD    FAC         ...
DB34   ED 62                     STD    2,S         SAVE IT
DB36   9D      A5                JSR    CHRGOT      EOL YET?
DB38   27      13 DB4D           BEQ    L3          YES, PROCEED
DB3A   BD    89AA                JSR    CHKCOM      NO-COMMA REQD
DB3D   BD    8E83                JSR    GETWRD      GET ARG2
DB40   DC      52                LDD    FAC         NULL?
DB42   27      06 DB4A           BEQ    L2          NO, SKIP
DB44   ED E4                     STD    0,S         SAVE IT
DB46   9D      A5                JSR    CHRGOT      EOL YET?
DB48   27      03 DB4D           BEQ    L3          OK IF TRUE
DB4A   7E    89B4       L2       JMP    SNERR       ELSE ERROR
DB4D   1A      50       L3       ORCC   #$50        NO INTRRPTS
DB4F   EC E1                     LDD    ,S++        GET ARG2
DB51   FD    060F                STD    AUTINC      STASH IT
DB54   EC E1                     LDD    ,S++        GET ARG1
DB56   B3    060F                SUBD   AUTINC      GET 1ST #
```

```
DB59  FD  060D            STD   AUTLIN     SET IT

DB5C  86    FF            LDA   #$FF
DB5E  B7  0613            STA   AUTFLG     TURN AUTO ON

DB61  39                  RTS              GO TO BASIC
                    ********************************
DB62  34    56    AUTHCK  PSHS  D,X,U
DB64  7D  0613            TST   AUTFLG
DB67  26    02 DB6B       BNE   M2
DB69  35    D6    AT.RTS  PULS  D,X,U,PC

DB6B  FC  060D    M2      LDD   AUTLIN     GET LINE NUMBER
DB6E  F3  060F            ADDD  AUTINC     YES. GET
DB71 1083 F9FF            CMPD  #MAXLIN    NEXT AND
DB75  22    F2 DB69       BHI   AT.RTS     CHECK IT.
DB77  ED E4               STD   0,S        ITS OK, SAVE
DB79  BD  957A            JSR   PRTLIN     AND PRINT
DB7C  86    20            LDA   #$20       IT AND (SP)
DB7E  BD  B54A            JSR   CHROUT     TO CRT
DB81  CE  03DA            LDU   #FACBUF    MOVE #
DB84  EC E4               LDD   0,S
DB86  FD  060D            STD   AUTLIN
DB89  8E  02DD            LDX   #LINBUF    INTO BUFF
DB8C  C6    00            LDB   #0         AND COUNT
DB8E  A6 C0       M3      LDA   ,U+        GET NEXT
DB90  27    05 DB97       BEQ   M4         UNTIL DONE..
DB92  A7 80               STA   ,X+        (NOT DONE)
DB94  5C                  INCB             COUNT CHR
DB95  20    F7 DB8E       BRA   M3         GET MORE
DB97  86    20    M4      LDA   #$20       STUFF A (SP)
DB99  A7 80               STA   ,X+        INTO BUFFER
DB9B  5C                  INCB             COUNT IT TOO
DB9C  BD  B505    M5      JSR   KBCHAR     GET FM KBD
DB9F  81    0D            CMPA  #$D        (CR)?
DBA1  27    04 DBA7       BEQ   M6         YES. SKIP
DBA3  81    03            CMPA  #$3        BREAK?
DBA5  26    16 DBBD       BNE   M7         NO, SKIP.
DBA7  7F  0613    M6      CLR   AUTFLG     NO "AUTO"
DBAA  86    0D            LDA   #$D        PRINT A (CR)
DBAC  BD  B54A            JSR   CHROUT     ...
DBAF  32 68               LEAS  8,S        FIX STACK
DBB1  0F    85            CLR   VALTYP
DBB3  8E  02DD            LDX   #LINBUF
DBB6  86    0D            LDA   #$D
DBB8  C6    01            LDB   #1
DBBA  7E  B5D3            JMP   GOBACK     GO GETLIN
DBBD  81    20    M7      CMPA  #$20       LESS THAN SP
DBBF  25    DB DB9C       BLO   M5         YES.  SLEW.
DBC1  81    7B            CMPA  #$7B       MORE THAN 7B?
DBC3  24    D7 DB9C       BHS   M5         YES.  SLEW.
DBC5  32 68               LEAS  8,S        FIX STACK
DBC7  0F    85            CLR   VALTYP
DBC9  7E  B5D3            JMP   GOBACK     REENTER.
```

****

```
                        ********************************
DBCC   4F               DOBEEP   CLRA
DBCD   8E    0008               LDX    #$8
DBD0   BD    BE12       DOBEP2   JSR    DCSND
DBD3   30 1F                    LEAX   -1,X
DBD5   26      F9 DBD0          BNE    DOBEP2
DBD7   39                       RTS

                        ********************************
                        *
                        * SYNTAX:
                        *  BEEP nn
                        *  00 ) nn ) 256
                        *
DBD8   27      04 DBDE BEEP     BEQ    BEEP1
DBDA   17    EADC C6B9          LBSR   NZRBYT
DBDD                            SKIP2
DBDE   C6      01       BEEP1   LDB    #$1
DBE0   34      04               PSHS   B
DBE2   5F                       CLRB
DBE3   BD    BAED               JSR    SNDON       TURN ON SOUND
DBE6   8D      E4 DBCC BEPLOP   BSR    DOBEEP
DBE8   BD    851B               JSR    ISCNTC      LOOK FOR USERS INPUT
DBEB   6A E4                    DEC    ,S
DBED   27      0A DBF9          BEQ    BEPDON
DBEF 108E    6000               LDY    #$6000
DBF3   31 3F            BEPWIT  LEAY   -1,Y
DBF5   27      EF DBE6          BEQ    BEPLOP
DBF7   20      FA DBF3          BRA    BEPWIT
DBF9   35      84       BEPDON  PULS   B,PC

                        ********************************
                        *
                        * SYNTAX:
                        *  WAIT mmmm
                        *  00 ) mmmm ) 65535
                        *
DBFB   27      11 DC0E WAIT     BEQ    HOME23
DBFD   BD    8E83               JSR    GETWRD
DC00   9E      52               LDX    (FAC
DC02   BD    851B       WAIT1   JSR    ISCNTC
DC05   C6      A0               LDB    #160
DC07   5A               WAIT2   DECB
DC08   26      FD DC07          BNE    WAIT2
DC0A   30 1F                    LEAX   -1,X
DC0C   26      F4 DC02          BNE    WAIT1
DC0E   39               HOME23  RTS

                        ********************************
```

```
                        ********************************
                        *
                        * SWAP
                        *
                        *  SWAP v1,v2
                        *
DC0F   BD    8A94       SWAP    JSR    PTRGET
DC12   9E      39               LDX    VARPNT
DC14   96      06               LDA    STRFLG      GET VAL TYPE
DC16   34      12               PSHS   X,A
DC18   BD    89AA               JSR    CHKCOM
DC1B   BD    8A94               JSR    PTRGET
DC1E   35      22               PULS   Y,A
DC20   91      06               CMPA   STRFLG      SAME VAL TYPES ??
DC22 1026    AC5C 8882          LBNE   TMERR       ?ERROR?
DC26   DE      39               LDU    VARPNT
DC28   8E    0005               LDX    #5
DC2B   A6 C4           HOP166   LDA    ,U
DC2D   E6 A4                    LDB    ,Y
DC2F   E7 C0                    STB    ,U+
DC31   A7 A0                    STA    ,Y+
DC33   30 1F                    LEAX   -1,X
DC35   26      F4 DC2B          BNE    HOP166
DC37   39                       RTS

                        ********************************
DC38   34      15       SELDRV  PSHS   X,CC,B      save
DC3A   C6      0E               LDB    #14         point at B data register
DC3C   8E    FF24               LDX    #A.DSKSLT   point at base address of register
DC3F   1A      50               ORCC   #$50        mask all interrupts
DC41   E7 02                    STB    2,X         put address out
DC43   E6 84                    LDB    ,X          get port a
DC45   CA      03               ORB    #3          set for latch address
DC47   E7 84                    STB    ,X
DC49   C4      FC               ANDB   #$FC        restore control lines
DC4B   E7 84                    STB    ,X
DC4D   A7 02                    STA    2,X         output data to write
DC4F   6C 84                    INC    ,X          latch it
DC51   6A 84                    DEC    ,X
DC53   35      95               PULS   X,CC,B,PC   recover registers
                        *
E DC55                  BOOT    EQU    *
                        ********************************
```

```
                        ********************************
                        *
                        * DRIVE [UNIT]
                        *       1-4
                        *
                        * SETS THE DEFAULT DRIVE NUMBER TO 1-4
                        * OTHER NUMERIC ARGS GET DN ERR:ELSE SYNERR

DC55   27     07 DC5E DRIVE    BEQ    SNERR0
DC57   17   EAAF C709 DRIV1    LBSR   DRVNUM
DC5A   F7   060A               STB    DEFDRV
DC5D   39                      RTS
DC5E   7E   89B4       SNERR0  JMP    SNERR
                        ********************************
                        *
                        * BASIC COMMAND ERROR
                        *
                        * SYNTAX:
                        *  ERROR GOTO xxxxx
                        *
                        * LOOK FOR A 'GO'
DC61   81     81       ERROR   CMPA   #GOTOK
DC63   26     F9 DC5E          BNE    SNERR0
                        * LOOK FOR A 'TO'
DC65   9D     9F               JSR    GETCHR
DC67   81     BC               CMPA   #TOTOK
DC69   26     F3 DC5E          BNE    SNERR0
DC6B   9D     9F               JSR    GETCHR
DC6D   BD   869A               JSR    LINGET
DC70   9E     2B               LDX    LINNUM
DC72   8C   F9FF               CMPX   #MAXLIN
DC75 1022   AF14 8B8D          LBHI   FCERR
DC79   BF   0615               STX    ONERRL
DC7C   27     06 DC84          BEQ    ONEROF      TURN OFF
DC7E   86     FF               LDA    #$FF        TURN TRAP
DC80   B7   0614               STA    ONERRF
DC83   39                      RTS

DC84   7F   0614       ONEROF  CLR    ONERRF
DC87   39                      RTS

                        ********************************
                        * GET A STRING POINTER
DC88   BD   89AA       GETSTO  JSR    CHKCOM      FLUSH COMMA
DC8B   BD   8A94               JSR    PTRGET      GET DISCRIPTOR
DC8E   7E   8877               JMP    CHKSTR      CHECK STRING

                        ********************************
```

```
                        ********************************
                        * SREAD UNIT,TRACK,SECTOR,STRING1,STRING2
                        *       1-4  0-39  1-36   DESTINATION STRINGS (2)
                        *
                        *      SINGLE SECTOR READ
                        *
DC91  17   009C DD30 SREAD    LBSR  DVINFO     GET DRIVE INFO
DC94  8D     F2 DC88          BSR   GETSTO     GET FIRST STRING POINTER
DC96  34     10               PSHS  X          SAVE POINTER
DC98  8D     EE DC88          BSR   GETSTO     GET SECOND STRING POINTER
DC9A  34     10               PSHS  X          SAVE POINTER

                        * DONT TIME OUT
DC9C  C6     FF               LDB   #-1
DC9E  D7     F6               STB   LOCK
DCA0  17   F64D D2F0          LBSR  FINDMT     FIND BUFFER
DCA3  26     24 DCC9          BNE   NEWER8     ?ERROR?
DCA5  6F 02                   CLR   PAGVLD,X   NOT VALID
DCA7  AE 05                   LDX   PAGADR,X
DCA9  9F     EE               STX   DD.BUF
DCAB  17   E456 C104          LBSR  GO.RDS
DCAE  F7   0603               STB   TEMP.C

DCB1  DE     EE               LDU   DD.BUF
DCB3  33 C9 0080              LEAU  128,U      POINT STRING
DCB7  35     10               PULS  X          SECOND POINTER
DCB9  8D     11 DCCC          BSR   HOP167     CREATE STRING
DCBB  DE     EE               LDU   DD.BUF     POINT STRING
DCBD  35     10               PULS  X          GET FIRST POINTER
DCBF  8D     0B DCCC          BSR   HOP167
                        *
                        * NOW YOU CAN TIME-OUT
DCC1  0F     F6               CLR   LOCK
DCC3  F6   0603               LDB   TEMP.C
DCC6  26     01 DCC9          BNE   NEWER8
DCC8  39                      RTS
                        *
DCC9  7E   C718       NEWER8  JMP   NEWERR
DCCC  34     50       HOP167  PSHS  X,U        SAVE POINTERS
DCCE  C6     80               LDB   #128       STRING LENGTH
DCD0  BD   8C52               JSR   STR$       CREATE A STRING [B] BYTES LONG, [X] POINTS AT THE DISCRIPTOR
DCD3  33 84                   LEAU  ,X         MOVE THE ACTUALSTRING POINTER INTO U
DCD5  35     10               PULS  X          GET THE DISCRIPTOR POINTER
DCD7  E7 84                   STB   ,X         SAVE THE STRINGLENGTH IN THE DISCRIPTOR
DCD9  EF 02                   STU   2,X        SAVE THE POINTER TO THE STRING IN THE DISCRIPTOR
DCDB  35     10               PULS  X          GET THE POINTER TO THE STRING IN THE BUFFER
DCDD  7E   B7CC               JMP   BLKMOV     MOVE THE BUFFER INTO THE STRING

                        ********************************
```

```
                        ********************************
                        * SWRITE UNIT,TRACK,SECTOR,STRING1,STRING2
                        *       1-4  0-39  1-36   SOURCE STRINGS (2)
                        *
                        *     SINGLE SECTOR WRITE
                        *

DCE0  8D    4E DD30 SWRITE  BSR   DVINFO     GO GET THE DRIVE INFO
DCE2  8D    46 DD2A         BSR   GETST1     GO FIND THESTRING TO WRITE
DCE4  BD  8877              JSR   CHKSTR     MAKE SURE IT IS A STRING
DCE7  9E    52              LDX   FAC
DCE9  34    10              PSHS  X          SAVE THE STRING POINTER
DCEB  8D    3D DD2A         BSR   GETST1     GET THE SECOND STRING POINTER
DCED  BD  8D9A              JSR   FRESTR
DCF0  34    14              PSHS  B,X        PRESERVE LENGTH AND DISCRIPTOR POINTER

                        *     CLEAR THE WRITE BUFFER OF ANY GARBAGE

                        * DONT TIME OUT
DCF2  C6    FF              LDB   #-1
DCF4  D7    F6              STB   LOCK
DCF6  17  F5F7 D2F0         LBSR  FINDMT     POINT AT BUFFER
DCF9  26    CE DCC9         BNE   NEWER8     ?ERROR?
DCFB  6F 02                 CLR   PAGVLD,X
DCFD  AE 05                 LDX   PAGADR,X
DCFF  9F    EE              STX   DD.BUF
DD01  5F                    CLRB
DD02  6F 80           HOP168  CLR   ,X+      CLEAR BUFFER
DD04  5A                    DECB             DECREMENT COUNTER
DD05  26    FB DD02         BNE   HOP168     LOOP UNTIL DONE
DD07  35    14              PULS  B,X        GET LENGTH ANDPOINTER BACK
DD09  DE    EE              LDU   DD.BUF
DD0B  33 C9 0080            LEAU  )128,U     POINT AT THE SECOND HALF OF THE BUFFER
DD0F  5D                    TSTB             IS STRING A NULL
DD10  27    03 DD15         BEQ   HOP169     IF SO THEN DON'T MOVE IT
DD12  BD  B7CC              JSR   BLKMOV     MOVE THE STRING INTO THE BUFFER
DD15  35    10      HOP169  PULS  X          GET THE SECOND STRING POINTER
DD17  BD  8D9F              JSR   FRETMP     GET THE LENGTH AND MAINTAIN STRING SPACE
DD1A  DE    EE              LDU   DD.BUF     POINT AT BUFFER
DD1C  5D                    TSTB             IS STRING ANULL
DD1D  27    03 DD22         BEQ   HOP170     IF SO THEN DON'T MOVE IT
DD1F  BD  B7CC              JSR   BLKMOV     MOVE THE STRING INTO THE BUFFER
DD22  17  E3DC C101 HOP170  LBSR  GO.WRV     WRITE SECTOR OPT. VERIFY
DD25  25    A2 DCC9         BCS   NEWER8

                        * NOW YOU CAN TIME OUT
DD27  0F    F6              CLR   LOCK
DD29  39                    RTS

                        ********************************
DD2A  BD  89AA      GETST1  JSR   CHKCOM     EAT THE COMMA, ERROR IF NO COMMA
DD2D  7E  8887              JMP   FRMEVL     GO GET THE STRING POINTER

                        ********************************
```

```
                        *
                        * GENERAL PURPOSE DRIVE INFORMATION FETCHER
                        * GENERATES FUNC CALL ERRORS IF USER SCREWS UP
                        *
DD30  26    03 DD35 DVINFO  BNE   DRVDA1     MUST ARGUE
DD32  7E  89B4              JMP   SNERR
DD35  17  E9D1 C709 DRVDA1  LBSR  DRVNUM     GET UNIT #
DD38  D7    EB              STB   DD.UNT     SET UNIT #
DD3A  BD  8E7E              JSR   COMBYT     GET TRACK #
DD3D  C1    4F              CMPB  #79
DD3F  22    08 DD49         BHI   INFERR     )79 ERROR
DD41  D7    EC              STB   DD.TRK     SET TRACK
DD43  BD  8E7E              JSR   COMBYT     GET SECTOR #
DD46  D7    ED              STB   DD.SEC     SET SECTOR #
DD48  39            HOME22  RTS
DD49  7E  8B8D      INFERR  JMP   FCERR

                        ********************************
                        * VERIFY [ON/OFF]
                        *
                        * VERIFY FLAG: SET OR CLEAR
                        *
                        * VERIFY IS NORMALLY ON. USER MAY TURN OFF IF DESIRED
                        *
                        * VERIFY WITH NO ARGUMENTS MEANS 'ON'
                        * VERIFY 'ON' OR 'OFF' DO THE OBVIOUS. OTHER YIELDS SNERR

DD4C  27    0E DD5C VERIFY  BEQ   VERONN
DD4E  81    C2              CMPA  #OFFTOK
DD50  26    03 DD55         BNE   NOTOFF
DD52  5F                    CLRB
DD53  20    09 DD5E         BRA   VERSET
DD55  81    88      NOTOFF  CMPA  #ONTOK
DD57  27    03 DD5C         BEQ   VERONN
DD59  7E  89B4              JMP   SNERR
DD5C  C6    FF      VERONN  LDB   #$FF
DD5E  F7  0608      VERSET  STB   VERFLG
DD61  0E    9F              JMP   GETCHR

                        ********************************
```

```
                        ********************************
                        *
                        * BASIC FUNCTIONS
                        *
                        *       EOF
                        *       LOF
                        *       FREE
                        *       STATUS
                        *       ERL
                        *       ERR
                        *
                        ********************************
DD63   0D      06       EOF     TST     STRFLG
DD65   27      E1 DD48          BEQ     HOME22
DD67   0F      06               CLR     STRFLG
DD69   32 62                    LEAS    2,S
DD6B 108E    DFA3               LDY     #XNTDAT
DD6F   17    F9BA D72C          LBSR    GNMKLD
DD72   26      19 DD8D          BNE     NEWER3
DD74   17    F17F CEF6          LBSR    LENFIL
DD77   26      14 DD8D          BNE     NEWER3
DD79   17    F1AD CF29          LBSR    FIBADR
DD7C 11A3 0C                    CMPU    RDBYTE,X
DD7F   22      04 DD85          BHI     HOP171
DD81   A1 0E                    CMPA    RDBYTE+2,X
DD83   23      02 DD87          BLS     EOF0
DD85   4F              HOP171   CLRA
DD86                            SKIP2
DD87   86      01       EOF0    LDA     #1
DD89   DE      8A               LDU     DBLZER      CLEAR U
DD8B   20      1B DDA8          BRA     LOFLOT

DD8D   7E    C718       NEWER3  JMP     NEWERR
                        ********************************
                        *
                        * LOC COMMAND
                        *  GIVE THE CURRENT LOCATION
                        *  WITH IN THE FILE THAT WE
                        *  ARE READING AT
                        *
                        * SYNTAX:
                        *  xx = LOC "filespec"
                        *
DD90   17    F97F D712 LOC      LBSR    GNMDAT
DD93   26      F8 DD8D          BNE     NEWER3
DD95   17    F191 CF29          LBSR    FIBADR
DD98   EE 0C                    LDU     RDBYTE,X
DD9A   A6 0E                    LDA     RDBYTE+2,X
DD9C   20      0A DDA8          BRA     LOFLOT

                        ********************************
                        *
                        * SYNTAX:
                        *  xx = LOF "filespec"
```

```
                        *
DD9E  17   F971 D712 LOF     LBSR  GNMDAT
DDA1  26     EA DD8D         BNE   NEWER3
DDA3  17   F150 CEF6         LBSR  LENFIL
DDA6  26     E5 DD8D         BNE   NEWER3
DDA8  0F     06      LOFLOT  CLR   STRFLG     NOT A STRING
                        * AT THIS POINT THE LENGTH OF
                        * THE FILE IS IN U:A
DDAA  97     53              STA   FAC+1      24 BIT MATISSA
DDAC  DF     51              STU   FAC-1
DDAE  0F     50              CLR   FAC-2      CLEAR MSBYTE
DDB0  0F     63              CLR   FAC2       ROUNDING ERROR
DDB2  86     A0              LDA   #$A0       EXPONENT
DDB4  97     4F              STA   FAC-3
DDB6  7E   9165              JMP   NORMFC     MAKE IT FLOAT
                        ********************************
                        *
                        * FREE
                        *
                        * SYNTAX:
                        *  FREE d
                        *  FREE
                        *
DDB9  9D     A5      FREE    JSR   CHRGOT
DDBB  27     05 DDC2         BEQ   FREED
DDBD  17   E949 C709         LBSR  DRVNUM     GET DRIVE #
DDC0  20     03 DDC5         BRA   HOP172
DDC2  F6   060A      FREED   LDB   DEFDRV
DDC5  D7     EB      HOP172  STB   DD.UNT
DDC7  17   F3FF D1C9         LBSR  GETFRE
DDCA  26     C1 DD8D         BNE   NEWER3
DDCC  1F     13              TFR   X,U
DDCE  4F                     CLRA
DDCF  20     D7 DDA8         BRA   LOFLOT

                        ********************************

DDD1  FC   0617      ERL     LDD   ERLINE     GET LINE NUMBER
DDD4  7E   8C37      GIVAL   JMP   GIVABF

                        ********************************

DDD7  4F             ERR     CLRA
DDD8  F6   0619              LDB   ERRNUM     GET THE ERROR NUMBER
DDDB  20     F7 DDD4         BRA   GIVAL

                        ********************************

DDDD  DC     27      HIMEM   LDD   $27        HIGHEST LOCATION USED BY BASIC
DDDF  20     F3 DDD4         BRA   GIVAL

                        ********************************
                        *
                        * FRE(0)
                        *  RETURN THE AMOUNT OF FREE
```

```
                        *  STRING SPACE, AND FORCE A
                        *  STRING GARBAGE COLLECTION
                        *
DDE1  17    AEF3 8CD7 FRE$     LBSR   GARBA2      * GARBAGE
DDE4  DC      23               LDD    FRETOP
DDE6  93      21               SUBD   STKTOP
DDE8  20      EA DDD4          BRA    GIVAL

                       ********************************
                              END
                       ********************************
                              NAM               SUBMIT
```

```
                        ********************************
                        *  DISK DISPATCH TABLE
C DDEA C2CB             DSKTBL   FDB    RSTHED
C DDEC C2CE                      FDB    SEEK
C DDEE C2FA                      FDB    RDSEC
C DDF0 C365                      FDB    WRTVER
C DDF2 C362                      FDB    WRTSEC
C DDF4 C388                      FDB    WRTRAK
C DDF6 C2F7                      FDB    RDADR
C DDF8 C322                      FDB    VERF
E DDFA                  DTBEND   EQU    *

                        ********************************
                        * SECTOR INTERLEAVE TABLE
C DDFA 01 0A 02 0B      INTLEV   FCB    01,10,02,11,03,12
C DE00 04 0D 05 0E               FCB    04,13,05,14,06,15
C DE06 07 10 08 11               FCB    07,16,08,17,09,18
C DE0C 00                        FCB    0           TABLE TERMINATOR

                        ********************************
                        * TRACK FORMAT [PART 1]
C DE0D 35 4E 4E         FM.FMT   FCB    $35,$4E,$4E
C DE10 08 00 00                  FCB    $08,$00,$00
C DE13 03 F6 FC                  FCB    $03,$F6,$FC
C DE16 1F 4E 4E                  FCB    $1F,$4E,$4E

                        ********************************
                        * TRACKE FORMAT [PART 2]
C DE19 07 00 00         FM.FMS   FCB    $07,$00,$00
C DE1C 03 F5 FE                  FCB    $03,$F5,$FE

                        ********************************
                        * TRACK FORMAT [PART 3]
                        *
                        *      1 BYTE OF 1
                        *      1 BYTE TRACK NUMBER
                        *      1 BYTE OF 1
                        *      1 BYTE SIDE NUMBER
                        *      1 BYTE OF 1
                        *      1 BYTE SECTOR NUMBER

                        ********************************
                        * TRACK FORMAT [PART 4]
C DE1F 01 F7 4E                  FCB    $01,$F7,$4E
C DE22 14 4E 4E                  FCB    $14,$4E,$4E
C DE25 0B 00 00                  FCB    $0B,$00,$00
C DE28 03 F5 FB                  FCB    $03,$F5,$FB
C DE2B 00 E5 F7                  FCB    $00,$E5,$F7
C DE2E 17 4E 4E                  FCB    $17,$4E,$4E

                        ********************************
                        * TRACK FORMAT [PART 5]
C DE31 00 4E 4E                  FCB    $00,$4E,$4E
```

```
                        ********************************
                        * INTERRUPT JUMP TABLE
C DE34 09               INTABL  FCB     9
C DE35 0109                     FDB     NMIVEC
                        *
  DE37  7E   C774               JMP     NMIHLR
  DE3A  7E   C77D               JMP     IRQHLR
  DE3D  7E   C830               JMP     FIRQHD

                        ********************************
                        * DEFAULT DRIVE TABLE
C DE40 04                       FCB     4
C DE41 069F                     FDB     LUSTPR+1
                        *
C DE43 0202 0202                FDB     $0202,$0202

                        ********************************
                        * INFO TO MOVE TO $012A
C DE47 1E                       FCB     30
C DE48 012A                     FDB     $012A
                        *
E DE4A                  CMDPNT  EQU     *
C DE4A 1A                       FCB     (CMDEND-CMDADD)/2
C DE4B DEEA                     FDB     CMDTBL
C DE4D C6C6                     FDB     CMDITP
C DE4F 07                       FCB     (FNCEND-FNCADD)/2
C DE50 DED1                     FDB     FNCTBL
C DE52 C6E1                     FDB     FNCITP
C DE54 00                       FCB     0
C DE55 0000                     FDB     0
C DE57 89B4                     FDB     SNERR
C DE59 00                       FCB     0
C DE5A 0000                     FDB     0
C DE5C 89B4                     FDB     SNERR
C DE5E 0000 0000                FDB     $0,$0,$0,$0,$0
C DE68 00                       FCB     0
                        ********************************
                        * COMMAND ADDRESS TABLE
E DE69                  CMDADD  EQU     *
C DE69 DB1B                     FDB     AUTO
C DE6B C59A                     FDB     BACKUP
C DE6D DBD8                     FDB     BEEP
C DE6F C3BF                     FDB     BOOTDSK     NEW COMMAND FORMAT SYSTEM DISK
C DE71 D646                     FDB     CHAIN
C DE73 D3AE                     FDB     COPY
C DE75 D76C                     FDB     DCREAT
C DE77 DA7B                     FDB     DIR
C DE79 DC55                     FDB     DRIVE
C DE7B C3C5                     FDB     FORMAT      DSKINIT FORMAT DATA DISK
C DE7D D842                     FDB     FREAD
C DE7F D8E9                     FDB     FWRITE
C DE81 DC61                     FDB     ERROR
C DE83 D7BA                     FDB     KILL
C DE85 D51B                     FDB     LOAD
C DE87 D45F                     FDB     MERGE
```

```
C DE89 D7C7                     FDB    PROTCT      PROTECT
C DE8B DBFB                     FDB    WAIT
C DE8D D7EC                     FDB    RENAME
C DE8F D5AF                     FDB    SAVE
C DE91 DC91                     FDB    SREAD
C DE93 DCE0                     FDB    SWRITE
C DE95 DD4C                     FDB    VERIFY
C DE97 89B4                     FDB    SNERR       AT TOKEN
C DE99 D80E                     FDB    FLREAD
C DE9B DC0F                     FDB    SWAP
E DE9D                  CMDEND  EQU    *

                        ********************************
                        * FUNCTION ADDRESS TABLE
E DE9D                  FNCADD  EQU    *
C DE9D DD9E                     FDB    LOF
C DE9F DDB9                     FDB    FREE
C DEA1 DDD1                     FDB    ERL
C DEA3 DDD7                     FDB    ERR
C DEA5 DDDD                     FDB    HIMEM
C DEA7 DD90                     FDB    LOC
C DEA9 DDE1                     FDB    FRE$
E DEAB                  FNCEND  EQU    *

                        ********************************
                        * HOOK TABLE FOR BASIC
C DEAB D946             HKTABL  FDB    HKOPEN      $015E
C DEAD D93E                     FDB    HKDNCK      $0161
C DEAF C296                     FDB    FAKEHK      $0164
C DEB1 D95D                     FDB    HKCHRO      $0167
C DEB3 C296                     FDB    FAKEHK      $016A
C DEB5 C296                     FDB    FAKEHK      $016D
C DEB7 C296                     FDB    FAKEHK      $0170
C DEB9 C296                     FDB    FAKEHK      $0173
C DEBB D742                     FDB    HKCLOS      $0176
C DEBD C296                     FDB    FAKEHK      $0179
C DEBF D9A6                     FDB    HKITMS      $017C
C DEC1 C296                     FDB    FAKEHK      $017F
C DEC3 D767                     FDB    DOCLOS      $0182
C DEC5 C296                     FDB    FAKEHK      $0185
C DEC7 DD63                     FDB    EOF         $0188
C DEC9 C296                     FDB    FAKEHK      $018B
C DECB C296                     FDB    FAKEHK      $018E
C DECD C718                     FDB    NEWERR      $0191
C DECF D506                     FDB    RUN         $0194
                        ********************************
```

```
                        ********************************
                        * DISK BASIC FUNCTION TABLE
C DED1 4C 4F            FNCTBL  FCC     'LO'
C DED3 C6                       FCB     $80+'F
C DED4 46 52 45                 FCC     'FRE'
C DED7 C5                       FCB     $80+'E
C DED8 45 52                    FCC     'ER'
C DEDA CC                       FCB     $80+'L
C DEDB 45 52                    FCC     'ER'
C DEDD D2                       FCB     $80+'R
C DEDE 48 49 4D 45              FCC     'HIME'
C DEE2 CD                       FCB     $80+'M
C DEE3 4C 4F                    FCC     'LO'
C DEE5 C3                       FCB     $80+'C
C DEE6 46 52 45                 FCC     'FRE'
C DEE9 A4                       FCB     $80+'$
                        ********************************
                        * DISK BASIC COMMAND TABLE
E DEEA                  CMDTBL  EQU     *
C DEEA 41 55 54                 FCC     'AUT'
C DEED CF                       FCB     $80+'O
C DEEE 42 41 43 4B              FCC     'BACKU'
C DEF3 D0                       FCB     $80+'P
C DEF4 42 45 45                 FCC     'BEE'
C DEF7 D0                       FCB     $80+'P
C DEF8 42 4F 4F 54              FCC     'BOOTDS'
C DEFE CB                       FCB     $80+'K
C DEFF 43 48 41 49              FCC     'CHAI'
C DF03 CE                       FCB     $80+'N
C DF04 43 4F 50                 FCC     'COP'
C DF07 D9                       FCB     $80+'Y
C DF08 43 52 45 41              FCC     'CREAT'
C DF0D C5                       FCB     $80+'E
C DF0E 44 49                    FCC     'DI'
C DF10 D2                       FCB     $80+'R
C DF11 44 52 49 56              FCC     'DRIV'
C DF15 C5                       FCB     $80+'E
C DF16 44 53 4B 49              FCC     'DSKINI'
C DF1C D4                       FCB     $80+'T
C DF1D 46 52 45 41              FCC     'FREA'
C DF21 C4                       FCB     $80+'D
C DF22 46 57 52 49              FCC     'FWRIT'
C DF27 C5                       FCB     $80+'E
C DF28 45 52 52 4F              FCC     'ERRO'
C DF2C D2                       FCB     $80+'R
C DF2D 4B 49 4C                 FCC     'KIL'
C DF30 CC                       FCB     $80+'L
C DF31 4C 4F 41                 FCC     'LOA'
C DF34 C4                       FCB     $80+'D
C DF35 4D 45 52 47              FCC     'MERG'
C DF39 C5                       FCB     $80+'E
C DF3A 50 52 4F 54              FCC     'PROTEC'
C DF40 D4                       FCB     $80+'T
C DF41 57 41 49                 FCC     'WAI'
```

```
C DF44 D4                       FCB    $80+'T
C DF45 52 45 4E 41              FCC    'RENAM'
C DF4A C5                       FCB    $80+'E
C DF4B 53 41 56                 FCC    'SAV'
C DF4E C5                       FCB    $80+'E
C DF4F 53 52 45 41              FCC    'SREA'
C DF53 C4                       FCB    $80+'D
C DF54 53 57 52 49              FCC    'SWRIT'
C DF59 C5                       FCB    $80+'E
C DF5A 56 45 52 49              FCC    'VERIF'
C DF5F D9                       FCB    $80+'Y
C DF60 46 52 4F                 FCC    'FRO'
C DF63 CD                       FCB    $80+'M
C DF64 46 4C 52 45              FCC    'FLREA'
C DF69 C4                       FCB    $80+'D
C DF6A 53 57 41                 FCC    'SWA'
C DF6D D0                       FCB    $80+'P
                        ********************************
                        * BACKUP MESSAGES
C DF6E 49 4E 53 45      SSTRG   FCC    /INSERT SOURCE/,$D,0
C DF7D 49 4E 53 45      DESTRG  FCC    /INSERT DESTINATION/,$D,0
C DF91 50 52 45 53      DIRSTR  FCC    /PRESS ANY KEY/,$D,0

                        ********************************
                        * DEFAULT DRIVE NUBERS
C DFA0 42 41 53         XNTBAS  FCC    'BAS'
C DFA3 44 41 54         XNTDAT  FCC    'DAT'
C DFA6 42 49 4E         XNTBIN  FCC    'BIN'
C DFA9 20 20 20         XNTNUL  FCC    '   '

                        ********************************
                        *
                        * ERROR MESSAGE TEXT STRINGS
                        *
                        ********************************
E DFAC                  ERRTAB  EQU    *
E 0080                  ERR.NR  EQU    128
C DFAC 4E 52                    FCC    /NR/ NOT READY
E 0082                  ERR.SK  EQU    130
C DFAE 53 4B                    FCC    /SK/ SEEK
E 0084                  ERR.WP  EQU    132
C DFB0 57 50                    FCC    /WP/ WRITE PROTECT
E 0086                  ERR.RT  EQU    134
C DFB2 52 54                    FCC    /RT/ RECORD TYPE
E 0088                  ERR.RF  EQU    136
C DFB4 52 46                    FCC    /RF/ RNF
E 008A                  ERR.CC  EQU    138
C DFB6 43 43                    FCC    /CC/ CRC
E 008C                  ERR.LD  EQU    140
C DFB8 4C 44                    FCC    /LD/ LOST DATA
E 008E                  ERR.BT  EQU    142
C DFBA 42 54                    FCC    /BT/ BOOT
E 0090                  ERR.IV  EQU    144
C DFBC 49 56                    FCC    /IV/ INVALID DIR
E 0092                  ERR.FD  EQU    146
```

```
C DFBE 46 44                    FCC   /FD/ DIRECTOY FULL
E 0094                ERR.DF    EQU   148
C DFC0 44 46                    FCC   /DF/ DISK FULL
E 0096                ERR.FS    EQU   150
C DFC2 46 53                    FCC   /FS/ FILE SPEC
E 0098                ERR.PT    EQU   152
C DFC4 50 54                    FCC   /PT/ PROTECTION
E 009A                ERR.PE    EQU   154
C DFC6 50 45                    FCC   /PE/ READ PAST EOF
E 009C                ERR.FF    EQU   156
C DFC8 46 46                    FCC   /FF/ FILE NOT FOUND
E 009E                ERR.FE    EQU   158
C DFCA 46 45                    FCC   /FE/ FILE EXISTS
E 00A0                ERR.NE    EQU   160
C DFCC 4E 45                    FCC   /NE/ NON-EXISTENT
E 00A2                ERR.TF    EQU   162
C DFCE 54 46                    FCC   /TF/ TOO MANY OPEN
E 00A4                ERR.PR    EQU   164
C DFD0 50 52                    FCC   /PR/ PARAMETER
E 00A6                ERR.UN    EQU   166
C DFD2 3F 3F                    FCC   /??/ UNDEFINED ERR
                      ********************************
E 0028                ERR.DN    EQU   $28        BASICS ERROR
                      ********************************
                      * DISK ROM TITLE MESSAGE
C DFD4 44 52 41 47    TITLE     FCC   'DRAGONDOS 2.C'
C DFE1 0D 00                    FCB   $0D,0
E DFE3                ENDTTL    EQU   *
                      ********************************
                              END
```

```
            ERRORS :     1
          WARNINGS :  NONE
     PHASING OFFSET :  NONE
            LABELS :   987
LAST PROGRAM ADDRESS : 57314  $DFE2
LAST STORAGE ADDRESS :     6  $0006
      PROGRAM BYTES :  8163  $1FE3
      STORAGE BYTES :  1645  $066D
```

SYMBOL TABLE:

```
FF24 E A.DSKSLT   00D0 E ABTCMD     D4AD L ADDLIN     CEDB L ALRDY      D809 L ALREDY     0010 E ALTTRK
001D E ARRTAB     001F E ARRTOP     D6D1 L ARYLOP     D6E8 L ARYLP1     D6F0 L ARYSKP     001D E ARYTAB
8DEA E ASC2       DB69 L AT.RTS     00E5 E ATTOK      0613 D AUTFLG     DB62 L AUTHCK     060F E AUTINC
060D E AUTLIN     DB1B L AUTO       C7B4 L BACKDR     C59A E BACKUP     9D3D E BASIRQ     B44F E BASRST
84DA E BASRTS     0003 E BCOUNT     0000 D BD.BIT     0002 D BD.CBN     0003 D BD.CSN     0004 D BD.SCT
0001 D BD.UNT     0008 E BD.ZIS     DBD8 L BEEP       DBDE L BEEP1      DBF9 L BEPDON     DBE6 L BEPLOP
DBF3 L BEPWIT     82A9 E BERRLS     D4AA L BFMERR     060D D BGNBUF     C5D0 L BISNER     A66B E BITAB
C275 L BITERR     D617 L BLDHDR     B7CC E BLKMOV     831C E BLTU       DC55 E BOOT       2600 E BOOTBF
C3BF L BOOTDSK    CA21 L BOTHRW     0000 E BRKFLG     C56C L BSECTR     0008 D BTCNT      B4B2 E BTITLE
02DD E BUF        0800 D BUF.01     0671 D BUFAD      02DA E BUFLNM     0002 D BUFSIZ     C62C L BULOOP
C181 L BUMCMD     C294 L BUMUNT     C65E L BUOPRS     C626 L BUPRMT     C609 L BUSNER     0001 E BUSY
DA42 L BYTADJ     D676 L CAIN       C7A4 L CALCBF     BDDC E CASOFF     0137 E CERAML     CF4C L CEXIT
D646 L CHAIN      D542 L CHAINX     83ED E CHEAD      89AA E CHKCOM     D4E3 L CHKHDR     8877 E CHKSTR
D691 L CHNDWN     D6A0 L CHNUP      00A5 E CHRGOT     B54A E CHROUT     CB95 L CKNG       C8D1 L CKNMS
CBA0 L CKNXTT     D138 L CKPROT     C6F7 L CLEAN4     D330 L CLEANP     8424 E CLEARC     CF50 L CLOS1
CF38 L CLOSAL     CF4E L CLOSE1     D74F L CLOSEX     B7EE E CLOSRT     CF8B L CLOSX      DE69 E CMDADD
DE9D E CMDEND     00FC E CMDERR     C6DD L CMDEXP     C6C6 L CMDITP     C179 L CMDOK      DE4A E CMDPNT
DEEA E CMDTBL     8E7E E COMBYT     C3BB L COMPEX     D457 L COPDON     D3EE L COPERR     D3F1 L COPLOP
0007 E COPSTK     D3AE L COPY       D1DE L COUNT      CF91 L CREATE     CFE6 L CRENEW     0607 D CTLSAV
C6D3 L CTOKHI     0068 E CURLIN     0600 E DBLOCK     008A E DBLZER     D7B9 L DCEXT      D76C L DCREAT
00EE D DD.BUF     00EA D DD.CMD     0606 D DD.OPT     00ED D DD.SEC     00F0 D DD.STA     0605 D DD.TMR
00EC D DD.TRK     00EB D DD.UNT     060A D DEFDRV     D089 L DEL2S      D108 L DEL5S      D033 L DELETE
D08F L DELEXT     D114 L DELIVD     D04A L DELOOP     DF7D C DESTRG     006F E DEVNUM     C210 L DEXIT
85EE E DEXPTR     0001 D DFIBNO     C24A L DGONE      0008 E DIO.DK     0006 E DIO.RA     0000 E DIO.RE
0002 E DIO.RS     0001 E DIO.SE     0007 E DIO.VE     0004 E DIO.WS     0005 E DIO.WT     0003 E DIO.WV
DA7B L DIR        067F D DIRBFP     DA82 L DIRDF      DAD9 L DIRDON     DA8A L DIRLOP     0000 D DIRLSN
001D D DIRN       DAD1 L DIRNXT     DAE9 L DIRPR      DAEF L DIRPR2     000A E DIRPS      0019 E DIRSIZ
DF91 C DIRSTR     0014 E DIRTRK     DA85 L DIRUOK     C169 L DISK       C75C L DK.RST     C718 E DKERR
D1F9 L DKINF      0008 D DKPTR      0018 D DLSCT      0001 D DNAME      B851 E DNERR      D6CF L DO.ARY
D6F4 L DO.GON     C11E L DO.TRY     DBCC L DOBEEP     DBD0 L DOBEP2     D6AE L DOCHN      D767 L DOCLOS
D38D L DODIV      D7B4 L DOEXTN     84ED E DOJUMP     D088 L DONE       C74C L DONERR     0050 E DORERR
C2DC L DOSEEK     BE12 E DOSND      C2EF L DOTP11     C2E0 L DOTYP1     0000 E DRAGON     DC57 L DRIV1
DC55 L DRIVE      DD35 L DRVDA1     061C D DRVINF     0006 E DRVISZ     C709 L DRVNUM     DDEA C DSKTBL
C282 L DSKUND     DDFA E DTBEND     06AB D DTYDIR     C817 L DUMPN      DD30 L DVINFO     0009 D DXTNSN
000C D DXTXT      C13B L ED.ERP     C149 L ED.ERR     C15A L ED.RTS     C10D L ED.THR     C133 L ED.TKO
C106 L EDISK      E000 E EILOC      C658 L EMLOOP     C942 L EMPTY      C64F L EMPTYY     060F D ENDBUF
07F3 E ENDFBS     DFE3 E ENDTTL     0697 E ENDVAR     0040 E ENPRE      DD63 L EOF        DD87 L EOFO
C713 L ER.DNE     C716 L ER.PRM     D24D L ERINVD     DDD1 L ERL        0617 D ERLINE     DDD7 L ERR
008E E ERR.BT     008A E ERR.CC     0094 E ERR.DF     0028 E ERR.DN     0092 E ERR.FD     009E E ERR.FE
009C E ERR.FF     0096 E ERR.FS     0090 E ERR.IV     008C E ERR.LD     00A0 E ERR.NE     0080 E ERR.NR
009A E ERR.PE     00A4 E ERR.PR     0098 E ERR.PT     0088 E ERR.RF     0086 E ERR.RT     0082 E ERR.SK
00A2 E ERR.TF     00A6 E ERR.UN     0084 E ERR.WP     061A D ERRBEG     9C76 E ERRDIR     0609 D ERRMSK
0619 D ERRNUM     DC61 L ERROR      DFAC E ERRTAB     CC0C L ERWEOF     009D E EXECAD     CD47 L EXLV
CC69 L EXTEND     0052 E FAC        0063 E FAC2       03DA E FACBUF     C296 L FAKEHK     8B8D E FCERR
000B D FDRIV      CBD2 L FDXT       D1C6 L FEMERR     D1BE L FEMP       D178 L FEMPDR     013C E FERAML
000F D FFLAGS     CF29 L FIBADR     00F1 D FIBNO      06BD E FIBS       9DD9 E FILINE     C53E L FILL
0010 D FILLEN     D2F0 L FINDMT     D963 L FINHOK     83FF E FINLIN     C830 L FIRQHD     0037 E FIRQOK
D6F7 L FIXKLD     0000 D FLAGS      0677 D FLBYT      D80E L FLREAD     0675 D FLSCT      C55F L FM.BLD
0800 E FM.BUF     C482 L FM.ERO     DE19 C FM.FMS     DE0D C FM.FMT     C411 L FM.LOP     C430 L FM.ONE
00F4 D FM.SDE     00F3 D FM.SID     00F2 D FM.TRK     C550 L FM.VER     C43D L FM.VLP     D297 L FMBUF
B848 E FMERR      C3F3 L FMRGOT     0000 D FNAME      DE9D E FNCADD     DEAB E FNCEND     C6F3 L FNCEXP
```

C6E1 L FNCITP DED1 C FNCTBL D148 L FNF CCCC L FNLSX D95A L FNWERR D8B1 L FORLEN
C3C5 L FORMAT 0080 E FORTOK C9D7 L FOUNDK 000C E FRAMSZ D81D L FRDSUP DDE1 L FRE$
D842 L FREAD DA22 L FREDRD DDB9 L FREE DDC2 L FREED 0002 D FRELEN DB0D C FREMES
D874 L FRERR 8D9A E FRESTR 8D9F E FRETMP 0023 E FRETOP 0003 D FREXT 8887 E FRMEVL
D187 L FRNS CB32 L FSNEXT CB09 L FSNLSN C6E7 L FTOKHI 8874 E FUNRET 060B D FWBUF
00F2 D FWBYTE 00F4 D FWFLAG 00F3 D FWFOR D8E9 L FWRITE D938 L FWRIXT 8CD7 E GARBA2
D6BD L GARBAG D5F7 L GETADR CEB7 L GETBIT 8E51 E GETBYT 009F E GETCHR B7D4 E GETCON
D284 L GETDE D250 L GETDIR D1C9 L GETFRE C18C L GETSER 8CB3 E GETSPA DC88 L GETSTO
DD2A L GETST1 8331 E GETSTK 8E83 E GETWRD CDAE L GFEXNT 0004 D GFLG 0002 D GFPBF
CEB1 L GFRTS 0009 E GFSSZ 0000 D GFTRY 8C37 E GIVABF DDD4 L GIVAL CE43 L GMORE
D718 L GNMBAS D712 L GNMDAT D72C L GNMKLD D71E L GNMNUL C894 L GNNCR CD0A L GNXT
C0F8 L GO.RDA C104 L GO.RDS C165 L GO.RST C162 L GO.SEK C15F L GO.VER C0FE L GO.WRS
C0FB L GO.WRT C101 L GO.WRV B5D3 E GOBACK D550 L GOIDLE C297 L GOODUN C625 L GOON
CE8B L GOTIT CE8E L GOTITL CEA1 L GOTL2 85E7 E GOTO 0081 E GOTOK D877 L GPARM
BA77 E GRNCLS D2D6 L HAVMT 0041 E HIGHDS 0043 E HIGHTR DDDD L HIMEM D95D L HKCHRO
D752 L HKCLO1 D742 L HKCLOS D93E L HKDNCK D958 L HKDNER DA02 L HKEX2 DA04 L HKEXT
D9AF L HKITM2 D9A6 L HKITMS D9CB L HKLP D946 L HKOPEN D9A4 L HKOUT DA76 L HKPEER
DA78 L HKRDER DA0E L HKREAD DEAB C HKTABL C286 L HOME1 CBD5 L HOME10 CC0E L HOME11
CF00 L HOME12 CF90 L HOME13 D006 L HOME14 D118 L HOME15 D177 L HOME16 D2EF L HOME17
D505 L HOME18 D5F6 L HOME19 C547 L HOME2 D751 L HOME20 D8E8 L HOME21 DD48 L HOME22
DC0E L HOME23 D283 L HOME27 C55E L HOME3 C635 L HOME4 DA0D L HOME40 D93D L HOME44
C65D L HOME5 C708 L HOME6 C77C L HOME7 C816 L HOME77 C8AE L HOME8 C941 L HOME9
015E E HOOKS C25D L HOP1 C471 L HOP10 D0E7 L HOP100 D0FA L HOP101 D128 L HOP102
D155 L HOP103 D16D L HOP104 D19F L HOP105 D1A6 L HOP106 D1DD L HOP107 D1EB L HOP108
D1ED L HOP109 C488 L HOP11 D1F2 L HOP110 D23C L HOP111 D249 L HOP112 D252 L HOP113
D29E L HOP114 D2B5 L HOP115 D2B9 L HOP116 D2BB L HOP117 D2F2 L HOP118 D2F7 L HOP119
C48F L HOP12 D302 L HOP120 D311 L HOP121 D320 L HOP122 D326 L HOP123 D336 L HOP124
D369 L HOP125 D36E L HOP126 D37B L HOP127 D39D L HOP128 D3E7 L HOP129 C4A2 L HOP13
D407 L HOP130 D419 L HOP131 D42E L HOP132 D491 L HOP133 D4BC L HOP134 D4D4 L HOP135
D518 L HOP136 D52B L HOP137 D596 L HOP138 D5F1 L HOP139 C4EB L HOP14 D66B L HOP140
D673 L HOP141 D695 L HOP142 D6A6 L HOP143 D710 L HOP144 D722 L HOP145 D746 L HOP146
D778 L HOP147 D78C L HOP148 D7A2 L HOP149 C51D L HOP15 D836 L HOP150 D860 L HOP151
D871 L HOP152 D890 L HOP153 D8E4 L HOP154 D8F7 L HOP155 D917 L HOP156 D91C L HOP157
D96D L HOP158 D996 L HOP159 C52E L HOP16 D9A1 L HOP160 D9B7 L HOP161 D9E7 L HOP162
D9EF L HOP163 DA67 L HOP164 DAB1 L HOP165 DC2B L HOP166 DCCC L HOP167 DD02 L HOP168
DD15 L HOP169 C533 L HOP17 DD22 L HOP170 DD85 L HOP171 DDC5 L HOP172 C536 L HOP18
C552 L HOP19 C265 L HOP2 C58D L HOP20 C60C L HOP21 C612 L HOP22 C621 L HOP23
C681 L HOP24 C687 L HOP25 C6C4 L HOP26 C6FA L HOP27 C749 L HOP28 C7B1 L HOP29
C26D L HOP3 C807 L HOP30 C81B L HOP31 C82B L HOP32 C84C L HOP33 C879 L HOP34
C883 L HOP35 C8AF L HOP36 C8BD L HOP37 C8EA L HOP38 C907 L HOP39 C284 L HOP4
C913 L HOP40 C500 L HOP400 C91F L HOP41 C922 L HOP42 C931 L HOP43 C949 L HOP44
C95E L HOP45 C98B L HOP46 C9A0 L HOP47 C9AB L HOP48 C9D1 L HOP49 C396 L HOP5
CA45 L HOP50 CA47 L HOP51 CA51 L HOP52 CA5A L HOP53 CA75 L HOP54 CA86 L HOP55
CA8B L HOP56 CA8E L HOP57 CAC2 L HOP58 CAC4 L HOP59 C398 L HOP6 CAE3 L HOP60
CB46 L HOP61 CB48 L HOP62 CB9E L HOP63 CC99 L HOP64 CC9F L HOP65 CCB4 L HOP66
CCD5 L HOP67 CD22 L HOP68 CD2F L HOP69 C3E7 L HOP7 CD58 L HOP70 CDA2 L HOP71
CDDA L HOP72 CDDE L HOP73 CE1A L HOP74 CE1E L HOP75 CE3C L HOP76 CE5E L HOP77
CE63 L HOP78 CE81 L HOP79 C452 L HOP8 CEC8 L HOP80 CF01 L HOP81 CF24 L HOP82
CF3C L HOP83 CF48 L HOP84 CF70 L HOP85 CF81 L HOP86 CF9F L HOP87 CFDE L HOP88
D007 L HOP89 C46B L HOP9 D018 L HOP90 D021 L HOP91 D02A L HOP92 D053 L HOP93
D05B L HOP94 D068 L HOP95 D0AE L HOP96 D0BE L HOP97 D0DE L HOP98 D0E3 L HOP99
8371 E IDLE C8AC L ILLNAM 835E E INER1 8358 E INER2 DD49 L INFERR C02A L INIT
C02F L INIT1 C153 L INIT10 C049 L INIT2 C054 L INIT3 C066 L INIT4 C07A L INIT5
C0A4 L INIT6 C0D9 L INIT7 C0E8 L INIT8 C157 L INIT9 C0B8 L INITVR D4E0 L INS03
D4C4 L INSLIN CD0F L INSTEX DE34 C INTABL DDFA C INTLEV C7A1 L IRQERR C79E L IRQEXT

C77D L IRQHLR 0035 E IRQOK 010C E IRQVEC 851B E ISCNTC CBFF L ISTHR B505 E KBCHAR
D7BA L KILL DB23 L L1 DB4A L L2 DB4D L L3 0682 D LASDIR D561 L LDBAS
D576 L LDBIN C9CA L LEAV CEF6 L LENFIL 0007 E LENPBF 02DD E LINBUF 869A E LINGET
002B E LINNUM D51B L LOAD D554 L LOADBS D531 L LOADFX DD90 L LOC 00F6 D LOCK
DD9E L LOF DDA8 L LOFLOT C837 L LOOKUP C7D7 L LOOP C7D1 L LOOP2 0045 E LOWDS
0044 E LOWFUN 00CE E LOWTOK 0047 E LOWTR 001E D LSDIRB 0004 D LSENT 85E7 E LUKALL
06A6 E LUMSEC 06A2 E LUMTRK 069E E LUSTPR 069A E LUTRAK 0696 E LUVALD D24B L LVOK
CF27 L LVOKL DB6B L M2 DB8E L M3 DB97 L M4 DB9C L M5 DBA7 L M6
DBBD L M7 C4BA L MAKBOOT CBF6 L MAKIT C4DF L MAKSAT F9FF E MAXLIN 0027 E MEMSIZ
D45F L MERGE D482 L MERGLP CD1A L MKNXT C648 L MORA 0000 E MOTOFF 0000 E MOTRON
D4A4 L MRGXIT CD7A L NEWDIR C73B L NEWER1 D764 L NEWER2 DD8D L NEWER3 D7C4 L NEWER4
D5EE L NEWER7 DCC9 L NEWER8 D528 L NEWER9 C718 L NEWERR 849F E NEWSTT 0683 D NEWUSR
C7FD L NEX C27A L NEXTER C3EA L NFMARG C3EF L NFMSID C3F1 L NFMTRK 0673 D NMBYTE
FF25 E NMICNT C774 L NMIHLR 0020 E NMIOK 0109 E NMIVEC CB75 L NO2ND C4D9 L NOBOOT
C3B6 L NOCOMP CC3D L NOEX 000A E NOFIB 00BF E NOPRE 9165 E NORMFC C1F3 L NORTRY
CB20 L NOT1 CB35 L NOT2 CC0F L NOTEOF DD55 L NOTOFF C51B L NOTSYS C232 L NSEEKR
0005 D NUMFIL CF16 L NUMFIX 0000 D NUMXTD CAEA L NXTSCT CD65 L NXTXTT C6B9 L NZRBYT
00C2 E OFFTOK 8342 E OMERR DC84 L ONEROF 0614 D ONERRF 0615 D ONERRL C1C3 L ONESID
0088 E ONTOK B835 E OPNDEV C695 L OPRS1 C590 L OPTCHR 004F E OSLOC CEA8 L OVFG
0005 D PAGADR 0003 D PAGDRV 0004 D PAGLRU 0000 D PAGLSN 0667 D PAGPLC 0002 D PAGVLD
C83E L PARSE FF21 E PB2CRA FF27 E PB2CRB FF26 E PB2DAT 0634 D PBFINF FF1D E PIABEG
FF25 E PIAEND DAF6 L PRFAC 903D E PRINT C000 E PROBEG C699 L PROMPT D7C7 L PROTCT
D119 L PROTEC D7DE L PROTIV D7E7 L PROTOF D7D4 L PROTON 957A E PRTLIN CBD0 L PSTEOF
8A94 E PTRGET CD74 L PUTXT 00C0 E RACMD CFFB L RCREAT C2F7 L RDADR 000C D RDBYTE
C31A L RDFIRQ C319 L RDLOOP C2FA L RDSEC C308 L RDTIME 0612 D RDTYPE CA14 L READ
877A E READAT C979 L READDR CA4C L READLP CB07 L READLV 879A E READRT D2C3 L READSB
8371 E READY D7EC L RENAME D80B L RENERR CFCC L RENIT D3AE E REORG CCF9 L REPLC
0000 E RESCMD 8514 E RESTOR C111 L RETRY2 C1E2 L RETRYS CADE L RFBUFF 0661 D RLENG
0611 D RLFLAG D386 L RLSN D14B L RNAME 0000 E ROM 0088 E RSCMD 0669 D RSECT
003F E RSERR 0071 E RSTFLG 0072 E RSTHCK C2C8 L RSTHED C8FD L RTS0 CC66 L RTS1S
CBD3 L RTS2S CA72 L RTS4S D116 L RTS5S CD5F L RTSFS D506 L RUN 841F E RUNC
842B E RUNCC 00F5 D RWFLAG CA1A L RWRITE D5FD L SAVBSC D643 L SAVDON D5AF L SAVE
000A D SAVEXD 067A D SAVEXT 0678 D SAVFIB D5E5 L SAVNAM D645 L SAVRTS CB77 L SCANXT
00BA E SCBASE CE12 L SCIT CE6A L SCITB 8417 E SCRTCH 066F D SCTEXT 00B7 E SCTOP
C8FF L SEARCH C2CE L SEEK 0010 E SEKCMD DC38 L SELDRV CE2E L SELSE CE0C L SERCAL
CDD8 L SERCIN CDF1 L SERCOT CDF9 L SEROUT AA87 E SETGRA 0000 D SFIBNO 001F E SIZFIB
C8A1 L SLASH CE29 L SMTHG BAC3 E SNDENB 008C E SNDFRQ BAED E SNDON 89B4 E SNERR
DC5E L SNERR0 D7D8 L SPROT 065C D SRCLSN DC91 L SREAD DF6E C SSTRG C18F L STAKKR
8434 E STKINI 0021 E STKTOP 8C52 E STR$ 001F E STREND 0006 E STRFLG 90E5 E STROUT
C287 L STUNIT 0004 D SUM DC0F L SWAP 0106 E SWIVEC DCE0 L SWRITE 89AC E SYNCHK
00F7 D SYSDSK 066D D TARGET 065E D TBFADR 065B D TDRV 0603 D TEMP.C 0604 D TEMP.L
0660 D TEMP1 067E D TEMP2 0681 D TEMP3 0601 D TEMPS C636 L TESTA 067D D TFIBN
CE06 L TGOTIT 0600 D THRASH D319 L THSLRU 00FE E TIMERR 00D2 E TIMOUT DFD4 C TITLE
8882 E TMERR 0007 D TMPCNT 0005 D TMPLSN 0650 D TNAME CC35 L TOEX CC39 L TOEX2
066B D TOTAL 00BC E TOTOK C342 L TOUTER 0004 E TR00 C1C6 L TRY.IT CB0C L TRY1
C548 L TVER 002F E TXTBEG 00A6 E TXTPTR 0019 E TXTTAB 0019 E TY1ERR D8C5 L UNFCER
D8CB L UNFLOT D8D9 L UNFLT C299 L UNLOOP 9587 E UNPAC D84A L UNREAD D8C2 L UNSNER
C237 L UNSTKR 00FD E UNTERR D8C8 L UNTMER CAA5 L USEBUF 00B0 E USRPTR C8DD L VALCHR
0085 E VALTYP C20D L VANYWY D6C1 L VARLOP 0039 E VARPNT D6CB L VARSKP 001B E VARTAB
90A1 E VDCRLF 90F8 E VDQUES 90F5 E VDSPAC 003F E VERERR C322 L VERF 0608 D VERFLG
CA1D L VERFY DD4C L VERIFY DD5C L VERONN DD5E L VERSET 0004 E VFY C35B L VLOOP
D37F L VLSN C34E L VRFIRQ C206 L VRFWRT C333 L VRTIME DBFB L WAIT DC02 L WAIT1
DC07 L WAIT2 0006 D WANT 0663 D WBYTE C3A9 L WCOMP FF2F E WD.CSR FF2C E WD.DAT
FF2D E WD.SEC FF2E E WD.TRK D376 L WLSN 0664 D WRBYTE C382 L WRFIRQ CBD6 L WRITE
CBEC L WRITE2 C37F L WRLOOP C49D L WRTDLP C375 L WRTIME C388 L WRTRAK C362 L WRTSEC

```
C365 L WRTVER   C1F7 L WRTVRF   00A8 E WSCMD    005F E WSERR    00F4 E WTCMD    0047 E WTERR
00DF E WVERR    CE09 L XGETB    C9C7 L XLEAV    DFA0 C XNTBAS   DFA6 C XNTBIN   DFA3 C XNTDAT
DFA9 C XNTNUL   0017 D XTLEN1   001C D XTLEN2   0015 D XTLSN1   001A D XTLSN2   0008 D XTNSN
0013 D XTNT1    0018 D XTNT2    0002 D YTMP
```