```
;=======================================================================
; file          dci.mac
;
; program       dos65
;
; function      dos65 input/output, command interpreter
;
; by            ad brouwer
;
; date            -apr-84
;               11-nov-85       dos65 2.01 extended disk storage
;               26-nov-85       dos65 2.01 directories
;               28-dec-85       dos65 2.01 indirect command arguments
;               28-jun-86       dos65 2.01 Octopus hardware (ev)
;=======================================================================

        argl    SYSTEM
.argm   argm    argl
```

```
                ;=====================================================================
                ; file          syshavi.mac
                ;
                ; function      include file for system select
                ;
                ; by            ad brouwer
                ;
                ; date           7-oct-85
                ;               28-jun-86       version 2.01 Havisoft (ev)
                ;=====================================================================

                ;*********************************************************************
                ;               system
                ;*********************************************************************
                ;select the havisoft system

0004  max_dr  equ     4               maximum drives
0006  max_fd  equ     6               maximum file descriptors

                ;dosmode byte ($C012)
00E3  dosmod  equ     $e3             7 : inhibit lowercase input
                ;                     6 : inhibit lowercase on command
                ;                     5 : filename to uppercase
                ;                     2 : search free blocks
                ;                     1 : verify modified file

0001  brosys  equ     1               origin
0002  havisys equ     2               havisoft system
0003  junior  equ     3               standard junior

0002  system  equ     havisys

                ;interrupt and break vector
E75A  kbint   equ     $e75a           keyboard interrupt
E750  tmint   equ     $e750           timer interrupt
E7B3  brkvec  equ     $e7b3
E7B5  irqvec  equ     $e7b5

                ;*********************************************************************
                ;               console
                ;*********************************************************************

0001  vt100   equ     1               my own
0002  vt52    equ     2               standard vt72 (no invers)
0003  xvt52   equ     3               extended vt52
0004  graffix equ     4
0005  haviter equ     5               havisoft system

0005  terminal equ    haviter

                ;console i/o type

0001  standard equ    1               using standard M65 i/o
0002  polling equ     2               poll input
0003  aciaint equ     3               handle input interrupt
0004  havicon equ     4               havisoft system

0004  console equ     havicon

F000  vidout  equ     $f000           output
F003  vidchk  equ     $f003           check input
F003  vidin   equ     $f003           get
F006  outxdv  equ     $f006           output device x
F009  inpxdv  equ     $f009           input device x
F00C  devini  equ     $f00c           initialize device
```

```
                ;console hardware

0001  stdjun  equ     1               software serial interface
0002  si6850  equ     2               6850 acia
0003  si6551  equ     3               6551 acia
0004  pkeyb   equ     4               keyboard at pia (1900)
0005  vdujun  equ     5               vdu video output (e800)
0006  havicsl equ     6               havisoft i/o

0006  cslin   equ     havicsl
0006  cslout  equ     havicsl

E100  c.viab  equ     $e100
E101  c.vbpad equ     c.viab+1
E10D  c.vbifr equ     c.viab+13

E7B7  keycnt  equ     $e7b7
E7B8  keybuf  equ     $e7b8
0027  kbfmax  equ     39

                ;*********************************************************************
                ;               printer
                ;*********************************************************************

0001  piapri  equ     1               pia port a
0002  viapri  equ     2               via port a/b
0003  havipri equ     3               havisoft system

0003  printer equ     havipri

E100  p.viaa  equ     $e100
E100  p.vapbd equ     p.viaa
E10D  p.vaifr equ     p.viaa+13

0048  defpage equ     72              length page
0042  defpagl equ     66              lines / page

                ;*********************************************************************
                ;               timer
                ;*********************************************************************

0001  rtctim  equ     1
0002  viatim  equ     2
0003  havitim equ     3

0003  timer   equ     havitim

E100  t.via   equ     $e100

E77F  day     equ     $e77f
E780  month   equ     day+1
E781  year    equ     day+2
E782  hours   equ     $e782
E783  minutes equ     hours+1

                ;*********************************************************************
                ;               disk drivers
                ;*********************************************************************

E000  d.pia   equ     $e000
E004  d.fdc   equ     $e004

0001  delay2  equ     1               2 Mhz
0002  virtual equ     2               virtual drive
0004  winches equ     4               winchester drive
0008  fdsize  equ     8               8 inch drive
0010  fd2stp  equ     16              two step

0002  drivers equ     virtual

0003  maxdrive equ    3               number of drives
0000  trkaccess equ   0               0=6, 1=12, 2=20, 3=30 msec per step
```

```
                ;drive type definition

                ;bit 0,1
                ;drive conversion (2 drives or ls139 for drive decode pb0-1)
                ;1:     drive 0
                ;2:     drive 1
                ;3:     drive 2

                ;drive conversion (direct bit addressing pb0-2)
                ;0:     drive 0
                ;1:     drive 1
                ;2:     drive 2

                ;if enabled fdsize (pb2 connected to size select)
                ;bit 2  read/write 8 inch
                ;bit 3  seek 8 inch (3 ms step)
                ;bit 4  two step for 40 tracks diskettes

                ;bit 6,7
                ;if enabled for virtual and/or winchester drive
                ;80:    virtual drive
                ;c0:    winchester drive

                ; pseudo use 139
0000  dtype0  equ     $00             drive 0
0001  dtype1  equ     $01             drive 1
0080  dtype2  equ     $80             Virtual
0000  dtype3  equ     0

                ;virtual disk info in disk drivers
000E  vdoffset equ    $0e             offset block (1 block free)
0032  vdtracks equ    3*16+2          total tracks

                ;virtual disk addressing
                ;if address vdorgadr equals 0 vdorgbik is immediatly used
                ;to select original bank
0000  vdorgadr equ    0
000D  vdorgbik equ    $0d
FFFD  vddatadr equ    $fffd
D000  vdmemadr equ    $d000           Will be swapped out
```

```
                ;=====================================================================
                ; file          dvar.mac
                ;
                ; program       DOS65
                ;
                ; function      DOS65 constants / variables
                ;
                ; by            ad brouwer
                ;
                ; date            -apr-84
                ;               10-nov-85       dos65 2.1 extended disk storage
                ;               26-nov-85       dos65 2.1 directories
                ;               28-dec-85       dos65 2.1 indirect command substitution
                ;               28-jun-86       dos65 2.01 Octopus hardware (ev)
                ;=====================================================================

                ;*********************************************************************
                ;               file control block position
                ;*********************************************************************
0018  f.size  equ     24              every file f.size bytes
0000  mod     equ     0               fd
0001  fmod    equ     1               mode file
0002  tslc    equ     2               number of tsl sector
0003  tsln    equ     3               position in tsl sector
0004  tsltk   equ     4               track sector tsl sector
0005  tslsk   equ     5
0006  inpc    equ     6               position in data sector
0007  blkc    equ     7               number of sectors read or write (2)
0009  chrc    equ     9               file count (3)
                ;$c reserved
000D  f.drdi  equ     $d              drive - directory
000E  dirc    equ     $e              directory sector
000F  dirn    equ     $f              position in dir sector
0010  rdwrf   equ     $10             open mode
0011  f.rws   equ     $11             status data sector
0012  tslf    equ     $12             tsl modified
0013  filf    equ     $13             file modified
                ;$14-$17 reserved

                ;*********************************************************************
                ;               bytes in system sector
                ;*********************************************************************
0000  s.stab  equ     0               lookup table
0020  s.mode  equ     $20             bit 8 : lookup side 1 via side 0
0021  s.mtrk  equ     $21             tracks
0022  s.mcil  equ     $22             sectors/cilinder
0023  s.msec  equ     $23             sectors/cilinder/side
0024  s.bpat  equ     $24             bitmap pattern
0028  s.acnt  equ     $28             sectors each bit
0029  s.sht   equ     $29             shift factor map / physical track
002A  s.tbas  equ     $2a             tsl sector allocation start
002B  s.tbam  equ     $2b             tsl bit map first bitmap byte
002C  s.dbas  equ     $2c             data sector allocation start
                ;$2d-$2f reserved
0030  s.boot  equ     $30             boot tsl address
0032  s.dir   equ     $32             directory start ts address
0040  s.name  equ     $40             disk name
0058  s.cdat  equ     $58             creation date
005C  s.mdat  equ     $5c             modified date
0060  s.bmap  equ     $60             bitmap

                ;*********************************************************************
                ;               bytes in the TSL sector.
                ;*********************************************************************
0000  ftsl    equ     $00             forward link (in all TSL sectors)
0002  btsl    equ     $02             backward link (in all TSL sectors)
0004  flmo    equ     $04             file mode (r-w-d etc)
                ;$05 res
0006  dblk    equ     $06             sector of directory
0007  dpos    equ     $07             directory and position in sector
                ;$08-$0f res
0010  stad    equ     $10             start address
0012  rnad    equ     $12             run address
0014  flln    equ     $14             file lenght
0017  vers    equ     $17             version number
0018  credat  equ     $18             creation date
001C  moddat  equ     $1c             modified date
```

```
                        ;*********************************************************************
                        ;               function zero page addresses
                        ;*********************************************************************
          00E0          .org    $e0             saved by most applications
00E0      varb    .res    2               start file info block (f.size bytes)
00E2      sysb    .res    2               system sector (drive)
00E4      tslb    .res    2               tsl sector (file)
00E6      inpb    .res    2               date sector (file)
00E8      rwpoin  .res    2               read write pointer
00EA      poin    .res    2               pointer filename (open)
00EC      mpoin   .res    2               pointer for r/w functions
00EE      dirpp                           ;pointer in directory sector (open)
00EE      temp    .res    2               pointer to memory (r/w/s)

                        ;*********************************************************************
                        ;               page 0 used by command interpreter
                        ;*********************************************************************
          00F0          .org    $f0
00F0      t1      .res    2               parameters
00F2      t2      .res    2
00F4      t3      .res    2
00F6      t4      .res    2

00F8      tepo                            ;opt/param, text output (saved)
00F8      rbg     .res    2               start mem. r/w, mode seek
00FA      opp                             ;option param. pointer
00FA      rln     .res    2               no. of bytes
00FC      cbp                             ;command buffer pointer
00FC      tcn     .res    2               temp use (count)

00FE      opt     .res    1               options mask
00FF      mode    .res    1               param. mask

                        ;*********************************************************************
                        ;               DOS65 errors values
                        ;*********************************************************************
          0001  e1      .equ    1               mem. full (command LOAD RUN)
          0002  e2      .equ    2               disk full (write create)
          0003  e3      .equ    3               begin of file (seek)
          0004  e4      .equ    4               end of file (read)
          0010  e10     .equ    $10             file spec. (more then 14 char.)
          0011  e11     .equ    $11             device - directory select
          0012  e12     .equ    $12             no filename (open)
          0013  e13     .equ    $13             file not found (open)
          0015  e15     .equ    $15             option
          0016  e16     .equ    $16             hex/dec data
          0020  e20     .equ    $20             no entry to open a file
          0021  e21     .equ    $21             try to open opened file
          0025  e25     .equ    $25             delete prot. (delete)
          0026  e26     .equ    $26             write prot. (open w, w+)
          0027  e27     .equ    $27             read prot. (open r, r+)
          0028  e28     .equ    $28             permission (open ext. bits)
          0029  e29     .equ    $29             format file
          0030  e30     .equ    $30             file not opened
          0031  e31     .equ    $31             non existing fd
          0035  e35     .equ    $35             no delete allowed (delete)
          0036  e36     .equ    $36             not write opened (write)
          0037  e37     .equ    $37             not read, seek opened (read seek)
          0038  e38     .equ    $38             no seek r/w
          0039  e39     .equ    $39             seek function
          0041  e41     .equ    $41             existing file (rename)
          0042  e42     .equ    $42             not the same drive
          0046  e46     .equ    $46             status function
                        ;               $80             read error
                        ;               $90             write error
                        ;               $b0             invalid drive/track/sector address
                        ;               $c0             drive not ready
                        ;               $d0             disk protected
                        ;                               add $01 for system, $02 for directory
                        ;                               and $04 for tsl sector.
```

```
                        ;*****************************************************************
                        ;               constants
                        ;*****************************************************************
0000                    systk   equ     0               system sector track/sector
0001                    syssc   equ     1
0000                    dir0tk  equ     0               directory @ start
0003                    dir0sc  equ     3

0080                    prred   equ     $80             mode read
0040                    prwrt   equ     $40             mode write
0020                    prdel   equ     $20             mode delete

                        ;*****************************************************************
                        ;               DOS65 work memory
                        ;*****************************************************************
AA00                    wmem    equ     $aa00
AA00                            org     wmem

AA00                    inbuf   res     80              command input buffer (max 80 char)
                        ;50
AA50                    ifd     res     1               char input fd.
AA51                    ofd     res     1               char output fd.
AA52                    iocfl   res     1               flag not close i/o redirect
AA53                    lffl    res     1               line feed and terminal flags
AA54                    doser   res     1               error flag
AA55                    aio     res     1               A io save
AA56                    xio     res     1               X io save
AA57                    yio     res     1               Y io save

AA58                    ipoin   res     2               input
AA5A                    opoin   res     1               output
AA5B                    hdx     res     1               save X hex <> dec conv.

AA5C                    cjmp    res     2               start address
AA5E                    cmdvec  res     2               command line input and execute

                        ;60
AA60                    eopt    res     1               command/@ flags
AA61                    efd     res     1               command/@ fds
AA62                    efd2    res     1
001D                    eclm    equ     wmem+$80-$
AA63                    ecml    res     eclm            execute command line

                        ;-----------------------------------------------------------------
AA80                            org     wmem+$80        i/o redirect buffers

0020                    inm     equ     32
AA80                    ibf     res     inm             input file buffer
0020                    outm    equ     32
AAA0                    obf     res     outm            output file buffer

                        ;-----------------------------------------------------------------
AAC0                            org     wmem+$c0        console variables and input buffer

AAC0                    msio    res     1               Mode of interrupting (^S^C^Z etc)

                        ;-----------------------------------------------------------------
AAE0                            org     wmem+$e0        print

AAE0                    prfd    res     1               printer file descr.
AAE1                    prfd2   res     1               next spool file
AAE2                    propt   res     1               print options
AAE3                    prent   res     1               entry flag
AAE4                    prpage  res     1               lines on page
AAE5                    prpagl  res     1

                        ;-----------------------------------------------------------------
AAF0                            org     wmem+$f0        disk drivers

                        ;*****************************************************************
AB00                            org     wmem+$100
AB00                    varbst
AB00                    varo    res     6*f.size        in use for file info

0020                    zbmax   equ     $20             max pagezero stack
AB90                    zpbf    res     zbmax           save zero page
```

```
                        ;--------------------------------------------------------------------------
                        ;abb0
ABB0                    free    res     1               flag dos in use
ABB1                    svzpf   res     1               flag to save and restore zeropage
ABB2                    zbsp    res     1               stack pointer in zpbf
ABB3                    fd      res     1               file descriptor
ABB4                    drive   res     1               drive
ABB5                    direct  res     1               directory
ABB6                    xtrk    res     1               track/error count
ABB7                    ysec    res     1               sector
ABB8                    psec    res     1               last physical sector
                        ;9
ABB9                    modf    res     2               mode open/create
ABBB                    end     res     3               end pointer (read write seek)
ABBE                    sekf    res     1               seek flag
ABBF                    chrn    res     1               char count last data sector
ABC0                    blss    res     1               first free sector search
ABC1                    wralc   res     1               first time write allocate
ABC2                    alcnt   res     1               allocation count
ABC3                    altrk   res     1               last allocated ts
ABC4                    alsec   res     1
                        ;11 res

                ABD0            org     wmem+$1d0

ABD0                    csufl   res     1               command, system user drive
ABD1                    sdrive  res     1               system drive
ABD2                    udrive  res     1               user drive
ABD3                    wdrive  res     1               work drive
ABD4                    ypos    res     1               Y pos next filename
ABD5                    apos    res     1               A pos next filename

ABD6                    dirtk   res     1               current directory track and sector
ABD7                    dirsk   res     1
ABD8                    dirp    res     1               dir position
ABD9                    dirq    res     1               dir sector no
ABDA                    dirno   res     1               number of file

ABDB                    asrw    res     1               A r/w
ABDC                    iopoin  res     0               i/o rw fcb
ABDC                    psrw    res     4               point to single r/w
                        ;bbe0
ABE0                    namb    res     14              file spec. buffer
ABEE                    fdrive  res     1               file drive
ABEF                    opfnam  res     1               flags options name

                        ;12 res
                        ;--------------------------------------------------------------------------
                ABFC            org     wmem+$1fc
ABFC                    date    res     4               time and date

                        ;--------------------------------------------------------------------------
                AC00            org     wmem+$200

AC00                    dirb    res     $100            directory sector
AD00                    sysbst  res     max_dr*$100     system sectors (one for each drive)

B100                    tslbst  res     max_fd*$100     tsl sectors (one for each file)
B700                    inpbst  res     max_fd*$100     data sectors
```

```
                ;*********************************************************************
                ;               DOS65 entries
                ;*********************************************************************
C000  cold      equ     $c000           cold start
C003  warm      equ     cold+$03        warm
C006  command   equ     cold+$06        execute command
C009  comcont   equ     cold+$09        continue with command
C00C  intwarm   equ     cold+$0c        entry after ^C
C00F  sync      equ     cold+$0f        close all files

                ;constants

C012  dosmode   equ     cold+$12        flags for:
                ;                       7:      inhibit lowercase for input
                ;                       6:      inhibit lowercase on command
                ;                       5:      filename to uppercase
                ;                       2:      search empty sectors (write)
                ;                       1:      verify modified file
                ;
C015  dchr      equ     cold+$15        delete char. input

C016  wmemloc   equ     cold+$16        work memory loc.
C018  irqvloc   equ     cold+$18        irq vector location
C01A  brkvloc   equ     cold+$1a        break vector

C01C  lome      equ     cold+$1c        free memory
C01E  hime      equ     cold+$1e

                ;---------------------------------------------------------------------
                ;i/o entries
C020  in        equ     cold+$20        get char A => C=0
C023  out       equ     cold+$23        put char A => C=0
C026  inecho    equ     cold+$26        get and put char
C029  bufin     equ     cold+$29        get line to inputbuffer; length Y
C02C  getbuf    equ     cold+$2c        get from buffer,Y A; check eoln
C02F  crlf      equ     cold+$2f        print cr and lf
C032  spa       equ     cold+$32        print space
C035  hnout     equ     cold+$35        A nibble or hexout
C038  hexout    equ     cold+$38        A hexout
C03B  print     equ     cold+$3b        print string after call until null
C03E  aschex    equ     cold+$3e        A ascii > hex => C=0
C041  loupch    equ     cold+$41        A lower > upper; check eoln => Z=1
C044  hexdec    equ     cold+$44        A hex > dec; no overflow check
C047  dechex    equ     cold+$47        A dec > hex

C04A  inputx    equ     cold+$4a        get from device X
C04D  outputx   equ     cold+$4d        put to device X

C050  inch      equ     cold+$50        get character
C053  outch     equ     cold+$53        put character
C056  inopn     equ     cold+$56        define input
C059  outcre    equ     cold+$59        define output
C05C  inclo     equ     cold+$5c        close input
C05F  outclo    equ     cold+$5f        close output
C062  seter     equ     cold+$62        set error flag
C065  setlffl   equ     cold+$65        define console flag

C068  sopt      equ     cold+$68        scan options; opt => X
C06B  spar      equ     cold+$6b        scan parameters; mask => X
```

```
                ;*****************************************************************
                ;               console entries
                ;*****************************************************************
CA00  incon   equ     cold+$a00       console
CA03  outcon  equ     cold+$a03
CA06  inpri   equ     cold+$a06       printer
CA09  outpri  equ     cold+$a09
CA0C  inser   equ     cold+$a0c       serial
CA0F  outser  equ     cold+$a0f

CA12  chcon   equ     cold+$a12       check console input
CA15  statpri equ     cold+$a15       status printer
CA18  putpri  equ     cold+$a18       print always (ignore status)

CA1B  term    equ     cold+$a1b       inititialize console
CA1E  break   equ     cold+$a1e       user break check
CA21  brkint  equ     cold+$a21       break interrupt check

CA24  intsio  equ     cold+$a24       console interrupt
CA27  siosub  equ     cold+$a27       drive timer interrupt subr

CA2A  clstcon equ     cold+$a2a       (re)set consolebit
CA2D  brkset  equ     cold+$a2d       define breakvector

CA30  mon     equ     cold+$a30       monitor entry
CA33  pbel    equ     cold+$a33       ring bell
CA36  pdel    equ     cold+$a36       delete character

CA39  clear   equ     cold+$a39       attributes terminal.
CA3C  dircur  equ     cold+$a3c
CA3F  cleol   equ     cold+$a3f
CA42  cleod   equ     cold+$a42
CA45  attri   equ     cold+$a45       4: set normal
                ;                     5: set invers

                ;*****************************************************************
                ;               disk drivers entries
                ;*****************************************************************
CC00  drives  equ     cold+$c00       n drives
CC03  voffset equ     cold+$c03       virtual disk offset
CC04  vtracks equ     cold+$c04       # segments
CC05  inihw   equ     cold+$c05       init hardware
CC08  rdyhw   equ     cold+$c08       select drive
CC0B  wshw    equ     cold+$c0b       write sector
CC0E  rshw    equ     cold+$c0e       read sector
CC11  cshw    equ     cold+$c11       check sector

CC1A  sethmt  equ     cold+$c1a       define headload and motor on time
CC1D  settd   equ     cold+$c1d       set time and date from date

                ;*****************************************************************
                ;               DOS65 functions
                ;*****************************************************************
D000  sread   equ     cold+$1000      single char read
D003  sread1  equ     cold+$1003
D009  swrite  equ     cold+$1009      single char write
D00C  swrite1 equ     cold+$100c
D012  fseek   equ     cold+$1012      function seek
D015  fseek1  equ     cold+$1015
D01B  read    equ     cold+$101b      read
D01E  read1   equ     cold+$101e
D024  write   equ     cold+$1024      write
D027  write1  equ     cold+$1027
D02D  seek    equ     cold+$102d      seek
D030  seek1   equ     cold+$1030
D036  create  equ     cold+$1036      create
D039  create1 equ     cold+$1039
D03F  open    equ     cold+$103f      open
D042  open1   equ     cold+$1042
D048  close   equ     cold+$1048      close
D04B  close1  equ     cold+$104b
D051  delete  equ     cold+$1051      delete
D054  delete1 equ     cold+$1054
D05A  rename  equ     cold+$105a      rename
D05D  rename1 equ     cold+$105d
D063  status  equ     cold+$1063      status
D066  status1 equ     cold+$1066
```

```
      ;-------------------------------------------------------------------------
D09F  setfiln equ     cold+$109f      set filename (default *)
D0A2  dircomp equ     cold+$10a2      compare dir - filn
D0A5  dirinit equ     cold+$10a5      open dir drive/direct A
D0A8  dirnext equ     cold+$10a8      get next dir entry
D0AB  dirwrit equ     cold+$10ab      write sis / dir sector

D0AE  svzp    equ     cold+$10ae      save zeropage
D0B1  rszp    equ     cold+$10b1      restore zeropage

D0B4  setnex  equ     cold+$10b4      get address next filename
D0B7  ermes   equ     cold+$10b7      error message
D0BA  prname  equ     cold+$10ba      print filespec (C=1 with drivespec)
D0BD  prfile  equ     cold+$10bd      print filename (C=1 with drivespec)

D0C0  readsect equ    cold+$10c0      read sector
D0C3  writsect equ    cold+$10c3      write sector
D0C6  checsect equ    cold+$10c6      check sector
D0C9  wrchsect equ    cold+$10c9      write and check sector
```

Page
12
missing

```
C082 4C 9AC0            jmp     bi4
C085 20 6EC0    bi1     jsr     in1
C088 C0 4F      bi2     cpy     #79
C08A B0 0B              bcs     bi15
C08C 99 00AA            sta     inbuf,y         store and output
C08F 20 CCC0            jsr     outp_           output
C092 C8         bi16    iny
C093 C0 48              cpy     #72             > 72
C095 90 03              bcc     bi4
C097 20 33CA    bi15    jsr     pbel            bell
C09A 20 6EC0    bi4     jsr     in1
C09D C9 16              cmp     #'V'-'@'        ^V get next
C09F F0 E4              beq     bi1
C0A1 CD 15C0            cmp     dchr            delete
C0A4 F0 D6              beq     bi10
C0A6 C9 15              cmp     #'U'-'@'        ^U delete buffer
C0A8 F0 CD              beq     bi51
C0AA C9 19              cmp     #'Y'-'@'        ^Y print buffer
C0AC D0 0F              bne     2.f
C0AE C0 48      1       cpy     #72             max 72
C0B0 B0 E8              bcs     bi4
C0B2 20 DDC1            jsr     getbuf1
C0B5 F0 E3              beq     bi4
C0B7 20 CCC0            jsr     outp_
C0BA C8                 iny                     next
C0BB D0 F1              bne     1.b
C0BD C9 0D      2       cmp     #'\r'           check start command
C0BF D0 C7              bne     bi2
C0C1 99 00AA            sta     inbuf,y
C0C4 AD 00AA            lda     inbuf           get first char

                        ;---------------------------------------------------------------------
                        ;               print newline
                        ;---------------------------------------------------------------------
C0C7 48         crlf1   pha
C0C8 A9 0D              lda     #'\r'
C0CA D0 3A              bne     outp2

C0CC 20 DDC0    outp_   jsr     ctrchk
C0CF B0 49              bcs     out1
C0D1 48                 pha                     print it as ^ char
C0D2 A9 5E              lda     #'^'
C0D4 20 1AC1            jsr     out1
C0D7 68                 pla
C0D8 48                 pha
C0D9 49 40              eor     #$40
C0DB D0 3E              bne     out2

C0DD C9 7F      ctrchk  cmp     #$7f
C0DF F0 3E              beq     8.f
C0E1 C9 20              cmp     #' '
C0E3 60                 rts

C0E4 B9 00AA    delb_   lda     inbuf,y         get and delete char
C0E7 20 DDC0            jsr     ctrchk
C0EA B0 03              bcs     2.
C0EC 20 36CA            jsr     pdel            delete 2 char
C0EF 4C 36CA    2       jmp     pdel
                        ;
C0F2 99 00AA    outbf   sta     inbuf,y
C0F5 C0 4F              cpy     #79             max
C0F7 B0 01              bcs     1.
C0F9 C8                 iny
C0FA 2C 60AA    1       bit     eopt            command - indirect
C0FD 30 17              bmi     9.
C0FF 70 15              bvs     9.

C101 48         outanl  pha                     print with lf after cr
C102 C9 0D      outp1   cmp     #'\r'           check
C104 D0 15              bne     out2
C106 20 6CC3    outp2   jsr     outch1
C109 EE 53AA            inc     lff1            set flag lf after cr
C10C A9 0A              lda     #'\n'
C10E 20 6CC3            jsr     outch1
C111 CE 53AA            dec     lff1            reset
C114 18                 clc
C115 68                 pla
C116 60         9       rts

C117 20 6EC0    inechol jsr     in1
                        ;---------------------------------------------------------------------
                        ;               print character
                        ;---------------------------------------------------------------------
C11A 48         out1    pha                     output char
```

```
C11B 20 6CC3    out2    jsr     outch1
C11E 68                 pla
C11F 18         8       clc
C120 60                 rts                     return, no possible error in A

                ;---------------------------------------------------------------
                ;               print space
                ;---------------------------------------------------------------
C121 48         spa1    pha
C122 A9 20              lda     #' '
C124 D0 F5              bne     out2

C126 C9 00      prax    cmp     #0
C128 F0 06              beq     prax1
C12A 20 31C1            jsr     hnout1
C12D 8A                 txa
C12E 90 05              bcc     hexout1

C130 8A         prax1   txa
C131 C9 10      hnout1  cmp     #$10            or hex
C133 90 0B              bcc     nibout          inhibit leading 0

                ;---------------------------------------------------------------
                ;               print hexadecimal
                ;---------------------------------------------------------------
C135 48         hexout1 pha                     accu hex out
C136 4A                 lsra
C137 4A                 lsra
C138 4A                 lsra
C139 4A                 lsra
C13A 20 40C1            jsr     nibout
C13D 68                 pla
C13E 29 0F              and     #$f             nibble out
C140 C9 0A      nibout  cmp     #$a
C142 90 02              bcc     nibo1
C144 69 06              adc     #6
C146 69 30      nibo1   adc     #'0'
C148 90 D0              bcc     out1

C14A 20 80C1    priner  jsr     seter1

                ;---------------------------------------------------------------
                ;               print string
                ;---------------------------------------------------------------
C14D 48         print1  pha                     string behind JSR instr. until 00
C14E 8A                 txa                     save mem. and X
C14F 48                 pha
C150 A5 F8              lda     tepo
C152 48                 pha
C153 A5 F9              lda     tepo+1
C155 48                 pha
C156 BA                 tsx
C157 BD 0501            lda     $105,x          get return address
C15A 85 F8              sta     tepo
C15C BD 0601            lda     $106,x
C15F 85 F9              sta     tepo+1
C161 20 84C1    pria    jsr     tincr
C164 F0 05              beq     prib
C166 20 01C1            jsr     outan1          auto lf
C169 90 F6              bcc     pria
C16B BA         prib    tsx                     setup new return address
C16C A5 F8              lda     tepo
C16E 9D 0501            sta     $105,x
C171 A5 F9              lda     tepo+1
C173 9D 0601            sta     $106,x
C176 68                 pla                     get mem and X back
C177 85 F9              sta     tepo+1
C179 68                 pla
C17A 85 F8              sta     tepo
C17C 68                 pla
C17D AA                 tax
C17E 68                 pla
C17F 60                 rts

C180 CE 54AA    seter1  dec     doser           set error flag
C183 60                 rts

C184 A2 F8      tincr   ldx     #tepo
C186 F6 00      incr    inc     0,x             incr ,X
C188 D0 02              bne     1.
C18A F6 01              inc     1,x
C18C A1 00      1       lda     [0,x]
C18E 60                 rts
```

```
C18F 24 FF      ascdh   bit     mode            asc - dec/hex
C191 30 08              bmi     2.

                ;---------------------------------------------------------------
                ;               convert ascii to hex
                ;---------------------------------------------------------------
C193 C9 47      aschex1 cmp     #'F'+1          convert ascii hex C=0
C195 B0 0C              bcs     9.
C197 C9 41              cmp     #'A'
C199 B0 09              bcs     3.
C19B 49 30      2       eor     #'0'
C19D C9 0A              cmp     #9+1
C19F 90 02              bcc     9.
C1A1 49 30              eor     #'0'
C1A3 60         9       rts                     non ascii => C=1
C1A4 E9 37      3       sbc     #'A'-10
C1A6 18                 clc
C1A7 60                 rts

                ;---------------------------------------------------------------
                ;               convert hexadecimal to decimal
                ;---------------------------------------------------------------
C1A8 8E 5BAA    hexdec1 stx     hdx             A hex => A dec (no error)
C1AB 18                 clc
C1AC AA                 tax
C1AD F0 09              beq     dech4
C1AF A9 00              lda     #0
C1B1 F8                 sed
C1B2 69 01      hexd1   adc     #1
C1B4 CA                 dex
C1B5 D0 FB              bne     hexd1
C1B7 D8         dech3   cld
C1B8 AE 5BAA    dech4   ldx     hdx
C1BB 60                 rts

                ;---------------------------------------------------------------
                ;               convert decimal to hexadecimal
                ;---------------------------------------------------------------
C1BC 8E 5BAA    dechex1 stx     hdx             A dec => A hex
C1BF A2 FF              ldx     #$ff
C1C1 38                 sec
C1C2 F8                 sed
C1C3 E8         dech1   inx
C1C4 E9 01              sbc     #1
C1C6 B0 FB              bcs     dech1
C1C8 8A                 txa
C1C9 90 EC              bcc     dech3

C1CB C8         bufkey1 iny                     get next character
C1CC B1 FC      bufkey  lda     [cbp],y
C1CE C9 20              cmp     #' '            space terminator
C1D0 F0 14              beq     bk9

                ;---------------------------------------------------------------
                ;               convert lower to uppercase
                ;---------------------------------------------------------------
C1D2 C9 61      loupch1 cmp     #'a'
C1D4 90 0A              bcc     bk2
C1D6 C9 7B              cmp     #'z'+1
C1D8 B0 06              bcs     bk2             check eoln
C1DA 49 20              eor     #'a'-'A'
C1DC 60                 rts

                ;---------------------------------------------------------------
                ;               get character from commandline
                ;---------------------------------------------------------------
C1DD B9 00AA    getbuf1 lda     inbuf,y         inputbuffer,Y
C1E0 C9 00      bk2     cmp     #0              check eoln
C1E2 F0 02              beq     bk9
C1E4 C9 0D              cmp     #'\r'           test <cr>
C1E6 18         bk9     clc                     clear carry
C1E7 60                 rts

C1E8 84 FC      setcbp  sty     cbp             set string address
C1EA 85 FD              sta     cbp+1
C1EC A0 00              ldy     #0
C1EE 20 94C2            jsr     calposp         skip spaces
C1F1 A0 FF              ldy     #-1

C1F3 C8         sech1   iny                     next
C1F4 B1 FC      sechar  lda     [cbp],y         skip spaces
C1F6 C9 20              cmp     #' '
C1F8 F0 F9              beq     sech1
C1FA D0 E4              bne     bk2
```

```
                        ;************************************************************
                        ;               parameter option scanning
                        ;************************************************************
C1FC 86 FF      spar1   stx     mode            bit 7:decimal, 6:+option
C1FE 38                 sec                     5:no clear
C1FF B0 01              bcs     1.
C201 18         sopt1   clc                     get options => X
C202 08         1       php
C203 20 E8C1            jsr     setcbp          set line address
C206 28                 plp
C207 68                 pla
C208 85 FA              sta     opp             return address
C20A 85 F8              sta     tepo
C20C 68                 pla
C20D 85 FB              sta     opp+1
C20F 85 F9              sta     tepo+1
C211 20 84C1    sopt0   jsr     tincr           search new return address
C214 D0 FB              bne     sopt0
C216 A5 F9              lda     tepo+1
C218 48                 pha                     set return address
C219 A5 F8              lda     tepo
C21B 48                 pha
C21C B1 FC              lda     [cbp],y
C21E 90 4B              bcc     sopt10          -options
                        ;------------------------------------------------------------------------
                        ;               scan parameters
                        ;------------------------------------------------------------------------
C220 84 F8              sty     tepo
C222 84 F9              sty     tepo+1          clear param
C224 24 FF              bit     mode            mode
C226 50 08              bvc     spar22
C228 C9 2B              cmp     #'+'            check + option
C22A F0 03              beq     spar21
C22C E6 FF              inc     mode            flag no param
C22E 88                 dey
C22F C8         spar21  iny
C230 38         spar22  sec
C231 66 F9      spar2   ror     tepo+1          clear param
C233 A2 FA              ldx     #opp
C235 20 86C1            jsr     incr            set next
C238 F0 2B              beq     spar8           till last
C23A AA                 tax                     set page 0 X
C23B A9 20              lda     #$20
C23D 25 FF              and     mode            clear current par
C23F D0 04              bne     spar20          no
C241 95 00              sta     0,x
C243 95 01              sta     1,x
C245 A5 FF      spar20  lda     mode            scan enabled
C247 4A                 lsra
C248 90 09              bcc     spar40
C24A 88                 dey
C24B C8         spar26  iny
C24C 18         spar25  clc
C24D 90 E2              bcc     spar2           continue

C24F 20 A9C2    spar3   jsr     inshf
C252 C8                 iny
C253 20 CCC1    spar40  jsr     bufkey          while no terminator
C256 F0 D9              beq     spar2
C258 20 8FC1            jsr     ascdh           and hex
C25B 90 F2              bcc     spar3
C25D C9 2C              cmp     #','            parameter terminator
C25F F0 EA              beq     spar26

C261 A9 16      spar9   lda     #e16            error data
C263 38                 sec
C264 60                 rts

C265 A6 F8      spar8   ldx     tepo
C267 86 FF              stx     mode
C269 90 29              bcc     calposp         calc next

                        ;------------------------------------------------------------------------
                        ;               scan options
                        ;------------------------------------------------------------------------
C26B 84 FE      sopt10  sty     opt
C26D C9 2D              cmp     #'-'
C26F F0 1C              beq     sopt6
C271 D0 1F              bne     sopt8           no
C273 85 F9      sopt4   sta     tepo+1
C275 98                 tya                     save Y
C276 AA                 tax
C277 A0 00              ldy     #0
```

```
C279 C8          sopt3   iny
C27A B1 FA               lda     [opp],y
C27C F0 27               beq     sopt9           err
C27E 45 F9               eor     tepo+1
C280 D0 F7               bne     sopt3           test option table
C282 38                  sec
C283 6A          sopt5   rora                    shift bit in A
C284 88                  dey
C285 D0 FC               bne     sopt5           until Y equal start
C287 05 FE               ora     opt             set bit in OPT
C289 85 FE               sta     opt
C28B 8A                  txa
C28C A8                  tay
C28D 20 CBC1     sopt6   jsr     bufkey1         get next char
C290 D0 E1               bne     sopt4           check option
C292 A6 FE       sopt8   ldx     opt

                 ;---------------------------------------------------------------
                 ;               skip spaces and calculate new address
                 ;---------------------------------------------------------------
C294 20 F4C1     calposp jsr     sechar
C297 98          calpoin tya                     calc address
C298 18                  clc
C299 65 FC               adc     cbp
C29B A8                  tay
C29C A5 FD               lda     cbp+1
C29E 69 00               adc     #0
C2A0 84 FC               sty     cbp
C2A2 85 FD               sta     cbp+1
C2A4 60                  rts

C2A5 A9 15       sopt9   lda     #e15            option error
C2A7 38                  sec
C2A8 60                  rts

                 ;---------------------------------------------------------------
                 ;               shift and add to current parameter
                 ;---------------------------------------------------------------
C2A9 48          inshf   pha                     save
C2AA A5 F8               lda     tepo            test first number
C2AC 24 F9               bit     tepo+1
C2AE D0 0A               bne     inshf1
C2B0 05 F9               ora     tepo+1          set mask bit
C2B2 85 F8               sta     tepo
C2B4 A9 00               lda     #0
C2B6 95 00               sta     0,x
C2B8 95 01               sta     1,x
C2BA B5 01       inshf1  lda     1,x             get *1
C2BC 48                  pha
C2BD B5 00               lda     0,x
C2BF 16 00               asl     0,x             set *4
C2C1 36 01               rol     1,x
C2C3 16 00               asl     0,x
C2C5 36 01               rol     1,x
C2C7 24 FF               bit     mode
C2C9 30 06               bmi     inshf2
C2CB 68                  pla                     get *4
C2CC B5 01               lda     1,x
C2CE 48                  pha
C2CF B5 00               lda     0,x
C2D1 18          inshf2  clc                     calc *5 or *8
C2D2 75 00               adc     0,x
C2D4 95 00               sta     0,x
C2D6 68                  pla
C2D7 75 01               adc     1,x
C2D9 95 01               sta     1,x
C2DB 16 00               asl     0,x             set *10 or *16
C2DD 36 01               rol     1,x
C2DF 18                  clc
C2E0 68                  pla                     add
C2E1 75 00               adc     0,x
C2E3 95 00               sta     0,x
C2E5 90 02               bcc     inshf3
C2E7 F6 01               inc     1,x
C2E9 60          inshf3  rts

                 ;---------------------------------------------------------------
                 ;               input
                 ;---------------------------------------------------------------
C2EA 4C 00CA     rdfvid  jmp     incon
C2ED 4C 06CA     rdfpri  jmp     inpri
C2F0 4C 0CCA     rdfaci  jmp     inser

C2F3 2C 54AA     inch1   bit     doser           test error
```

```
C2F6 30 F2          bmi     rdfvid          console
C2F8 8E 56AA        stx     xio             save X
C2FB AE 50AA        ldx     ifd
C2FE 20 11C3        jsr     rdf15
C301 4C 7AC3        jmp     outch2

                    ;---------------------------------------------------------------
                    ;               device input
                    ;---------------------------------------------------------------
C304 E0 04      inputx1 cpx     #4
C306 90 09          bcc     rdf15
C308 8C 57AA        sty     yio
C30B 20 03D0        jsr     sread1
C30E 4C 53C3        jmp     rdf17

C311 E0 01      rdf15   cpx     #1
C313 90 D5          bcc     rdfvid          console
C315 F0 D6          beq     rdfpri          printer
C317 E0 03          cpx     #3
C319 90 D5          bcc     rdfaci          serial
C31B F0 4B          beq     rdfdum          dummy
C31D 8C 57AA        sty     yio             file
C320 AC 58AA        ldy     ipoin           end of input buffer
C323 C0 20          cpy     #inm
C325 D0 16          bne     rdf11
C327 A9 80          lda     #ibf&255        setup input buffer
C329 20 EBC3        jsr     setiop          read in temp input buffer
C32C 20 1ED0        jsr     read1
C32F 90 04          bcc     rdf12
C331 C9 04          cmp     #e4             test empty file
C333 D0 0F          bne     rdf13           no
C335 AC DEAB    rdf12   ldy     iopoin+2        save end
C338 8C 59AA        sty     ipoin+1
C33B A0 00          ldy     #0
C33D CC 59AA    rdf11   cpy     ipoin+1         test end
C340 90 0A          bcc     rdf18
C342 A9 04          lda     #e4
C344 48         rdf13   pha
C345 20 5FC4        jsr     cloi10          close input file
C348 68             pla
C349 38             sec                     set error flag
C34A B0 07          bcs     rdf17
C34C B9 80AA    rdf18   lda     ibf,y           get char
C34F C8             iny
C350 8C 58AA        sty     ipoin           set to next char
C353 AC 57AA    rdf17   ldy     yio
C356 60             rts

                    ;---------------------------------------------------------------
                    ;               output
                    ;---------------------------------------------------------------
C357 2C 53AA    wtfpri1 bit     1ff1            test -T flag after P command
C35A 10 03          bpl     wtfpri
C35C 20 62C3        jsr     wtfvid
C35F 4C 09CA    wtfpri  jmp     outpri
C362 4C 03CA    wtfvid  jmp     outcon
C365 4C 0FCA    wtfaci  jmp     outser
C368 A9 00      rdfdum  lda     #0
C36A 18         wtfdum  clc
C36B 60             rts

C36C 2C 54AA    outch1  bit     doser           test error
C36F 30 F1          bmi     wtfvid          console output
C371 8E 56AA        stx     xio
C374 AE 51AA        ldx     ofd             get fd
C377 20 8FC3        jsr     wtf15
C37A AE 56AA    outch2  ldx     xio             restore X
C37D 60             rts

                    ;---------------------------------------------------------------
                    ;               device output
                    ;---------------------------------------------------------------
C37E E0 01      outputx1 cpx    #1
C380 F0 DD          beq     wtfpri          printer
C382 E0 04          cpx     #4
C384 90 09          bcc     wtf15
C386 8C 57AA        sty     yio
C389 20 0CD0        jsr     swrite1
C38C 4C E5C3        jmp     wtf17

C38F E0 01      wtf15   cpx     #1
C391 90 CF          bcc     wtfvid          console
C393 F0 C2          beq     wtfpri1         printer/console
C395 E0 03          cpx     #3
```

```
C397 90 CC              bcc     wtfaci          serial
C399 F0 CF              beq     wtfdum          dummy
C39B 8C 57AA            sty     yio             file
C39E A8                 tay
C39F A9 41              lda     #$41            filter and newline
C3A1 2C 53AA            bit     lffl
C3A4 10 06              bpl     wtf13           no console
C3A6 08                 php
C3A7 98                 tya
C3A8 20 62C3            jsr     wtfvid
C3AB 28                 plp
C3AC F0 14      wtf13   beq     wtf10
C3AE C0 20              cpy     #$20            space - '~'
C3B0 90 04              bcc     wtf12
C3B2 C0 7F              cpy     #$7f
C3B4 90 0C              bcc     wtf10
C3B6 C0 09      wtf12   cpy     #'\t'           tab
C3B8 F0 08              beq     wtf10
C3BA C0 0D              cpy     #'\r'           ret
C3BC F0 04              beq     wtf10
C3BE 98                 tya                     not to file
C3BF 18                 clc
C3C0 90 23              bcc     wtf17
C3C2 98         wtf10   tya
C3C3 AC 5AAA            ldy     opoin
C3C6 99 A0AA            sta     obf,y           save in output buffer
C3C9 C8                 iny
C3CA C0 20              cpy     #outm           test end
C3CC 90 14              bcc     wtf9
C3CE 48                 pha
C3CF 20 E9C3            jsr     setop           save to file
C3D2 20 27D0            jsr     write1
C3D5 90 0B              bcc     wtf8            no error
C3D7 A8                 tay                     set error on stack
C3D8 68                 pla
C3D9 98                 tya
C3DA 48                 pha
C3DB 20 78C4            jsr     cloo10          close output
C3DE 38                 sec
C3DF A0 00      wtf8    ldy     #0
C3E1 68                 pla
C3E2 8C 5AAA    wtf9    sty     opoin           setup pos. outputbuffer
C3E5 AC 57AA    wtf17   ldy     yio
C3E8 60                 rts

C3E9 A9 A0      setop   lda     #obf&255        setup output buffer
C3EB 8D DCAB    setiop  sta     iopoin          setup begin buffer
C3EE A9 AA              lda     #obf>>8
C3F0 8D DDAB            sta     iopoin+1
C3F3 8C DEAB            sty     iopoin+2        set bufferlength
C3F6 A0 00              ldy     #0
C3F8 8C DFAB            sty     iopoin+3
C3FB A0 DC              ldy     #iopoin&255     setup A,Y
C3FD A9 AB              lda     #iopoin>>8
C3FF 60                 rts

                ;*********************************************************************
                ;               command < input redirect
                ;*********************************************************************
C400 A2 81      seifil  ldx     #$81            open input file for read

                ;---------------------------------------------------------------------
                ;               redirect input
                ;---------------------------------------------------------------------
C402 20 8CC5    inopn1  jsr     cloi50          close current input file
C405 E0 04              cpx     #4
C407 90 5B              bcc     cloi2
C409 20 42D0            jsr     open1
C40C B0 59              bcs     9.
C40E A9 20              lda     #inm            set charcount to end
C410 8D 58AA            sta     ipoin
C413 20 B4D0            jsr     setnex
C416 90 4C              bcc     cloi2           setup IFD

                ;*********************************************************************
                ;               command >, >> and >P output redirect
                ;*********************************************************************
C418 A2 01      seoprt  ldx     #1              printer
C41A D0 06              bne     seo2
C41C A2 91      seafil  ldx     #$91            append output file
C41E D0 02              bne     seo2
C420 A2 E1      seofil  ldx     #$e1            create output file
C422 86 FF      seo2    stx     mode
C424 20 68C0            jsr     sopt
```

```
C427 544600             fcc     'TF',0          tf terminal filter
C42A B0 3B              bcs     9.
C42C 8E 53AA            stx     1ff1
C42F A6 FF              ldx     mode

                        ;---------------------------------------------------------------
                        ;               redirect output
                        ;---------------------------------------------------------------
C431 20 7DC5    outcre1 jsr     cloo50          close current output file
C434 E0 04              cpx     #4              set i/o pointers
C436 90 45              bcc     cloo2
C438 E0 91              cpx     #$91
C43A F0 06              beq     pucr1
C43C 20 39D0            jsr     create1
C43F 4C 4CC4            jmp     creo1
C442 20 42D0    pucr1   jsr     open1           open r+
C445 B0 20              bcs     9.
C447 A9 04              lda     #4
C449 20 15D0            jsr     fseek1          seek end
C44C B0 19      creo1   bcs     9.
C44E A9 00              lda     #0              set charcount to 0
C450 8D 5AAA            sta     opoin
C453 20 B4D0            jsr     setnex
C456 90 25              bcc     cloo2           setup OFD

                        ;---------------------------------------------------------------
                        ;               close input redirect
                        ;---------------------------------------------------------------
C458 AE 50AA    inclo1  ldx     ifd             close input file
C45B E0 04              cpx     #4
C45D 90 03              bcc     cloi1           no file
C45F 20 4BD0    cloi10  jsr     close1
C462 A2 00      cloi1   ldx     #0
C464 8E 50AA    cloi2   stx     ifd             setup input fd IFD
C467 60         9       rts

                        ;---------------------------------------------------------------
                        ;               close output redirect
                        ;---------------------------------------------------------------
C468 AE 51AA    outclo1 ldx     ofd             close output file
C46B E0 04              cpx     #4
C46D 90 0C              bcc     cloo1           no file
C46F AC 5AAA            ldy     opoin
C472 20 E9C3            jsr     setop           save current buffer
C475 20 27D0            jsr     write1          write
C478 20 4BD0    cloo10  jsr     close1          close file
C47B A2 00      cloo1   ldx     #0
C47D 8E 51AA    cloo2   stx     ofd             setup output fd OFD
C480 60                 rts

C481 8D 53AA    setlff11 sta    1ff1            set output options
C484 60                 rts

                        ;*************************************************************
                        ;               main program
                        ;*************************************************************
C485 A2 00      colds   ldx     #0              clear buffers and init
C487 8A                 txa
C488 CA         ini1    dex
C489 9D 00AA            sta     wmem,x
C48C 9D FCAA            sta     wmem+$fc,x      not date
C48F D0 F7              bne     ini1
C491 20 05CC            jsr     inihw           hardware and terminal init
C494 20 3BC0            jsr     print
C497 444F533635         fcc     "DOS65 2.01\r",0
C49C 20322E3031
C4A1 0D00
C4A3 60                 rts
C4A4 A2 80      intwarm1 ldx    #$80            interrupt ^C
C4A6 20 94C6            jsr     sync10          no close spool files
C4A9 20 3BC0            jsr     print
C4AC 5E430D00           fcc     "^C\r",0
C4B0 A2 FF      warm1   ldx     #$ff            reset stack
C4B2 9A                 txs
C4B3 20 2AC5            jsr     exe7            continue with exec file
C4B6 20 BCC4            jsr     cmdin
C4B9 4C B0C4            jmp     warm1

C4BC AD 5FAA    cmdin   lda     cmdvec+1
C4BF F0 03              beq     1.
C4C1 6C 5EAA            jmp     [cmdvec]
C4C4 20 3BC0    1       jsr     print
C4C7 2400               fcc     "$",0
C4C9 EAEAEA             fcc     $ea,$ea,$ea     for new prompt
```

```
C4CC 20 29C0            jsr     bufin
C4CF 4C 23C5            jmp     cmnd0           command from inputbuffer

                        ;---------------------------------------------------------------------
                        ;               indirect / ascii commands
                        ;---------------------------------------------------------------------
C4D2 AD 62AA    exe8    lda     efd2            error reading file
C4D5 F0 6F              beq     exe9
C4D7 20 4CC5            jsr     cloe1           close level 2
C4DA A0 00      exe2    ldy     #0
C4DC AE 62AA    5       ldx     efd2            level 2
C4DF D0 05              bne     1.
C4E1 AE 61AA            ldx     efd             level 1
C4E4 F0 81              beq     9.              return
C4E6 20 4AC0    1       jsr     inputx          get char
C4E9 B0 E7              bcs     exe8            error eof
C4EB C9 26              cmp     #'&'            command line argument
C4ED D0 21              bne     4.
C4EF 20 4AC0            jsr     inputx          arg select
C4F2 B0 DE              bcs     exe8            error eof
C4F4 A2 FF              ldx     #-1
C4F6 C9 31              cmp     #'1'
C4F8 F0 0A              beq     2.
C4FA C9 32              cmp     #'2'
C4FC D0 12              bne     4.
C4FE E8         1       inx
C4FF BD 63AA            lda     ecml,x          skip 1st arg
C502 D0 FA              bne     1.
C504 E8         2       inx
C505 BD 63AA            lda     ecml,x          copy from exec cml
C508 F0 D2              beq     5.
C50A 20 F2C0            jsr     outbf
C50D 4C 04C5            jmp     2.
C510 20 F2C0    4       jsr     outbf
C513 C9 0D              cmp     #'\r'           eoln - start command
C515 D0 C5              bne     5.
C517 A0 00              ldy     #0              check end file
C519 20 66D0            jsr     status1
C51C C9 04              cmp     #e4
C51E D0 03              bne     cmnd0           not at end
C520 20 4CC5            jsr     cloe1           close file in advance
C523 A0 00      cmnd0   ldy     #inbuf&255
C525 A9 AA              lda     #inbuf>>8
C527 20 A1C5    cmnd1   jsr     cmnd2           exec command
C52A AD 60AA    exe7    lda     eopt            no i/o close
C52D D0 03              bne     1.f
C52F 20 75C5            jsr     cloiof          close i/o redirect
C532 20 62C5    1       jsr     cloerr          close i/o/e if error
C535 A9 E1              lda     #$e1            enable ^S^Q ^C and ^I
C537 20 2ACA            jsr     clstcon
C53A A0 0C              ldy     #intwarm&255    init ^C vector
C53C A9 C0              lda     #intwarm>>8
C53E 20 2DCA            jsr     brkset
C541 20 1ECA            jsr     break           user break
C544 90 94              bcc     exe2
C546 20 75C5    exe9    jsr     cloiof          close i/o

                        ;---------------------------------------------------------------------
                        ;               close command / indirect files
                        ;---------------------------------------------------------------------
C549 20 4CC5    cloe    jsr     cloe1
C54C A0 01      cloe1   ldy     #1              close execute file
C54E BE 61AA    1       ldx     efd,y
C551 D0 04              bne     2.
C553 88                 dey
C554 10 F8              bpl     1.
C556 60                 rts
C557 A9 00      2       lda     #0
C559 99 61AA            sta     efd,y           clear fd
C55C 0E 60AA            asl     eopt
C55F 4C 4BD0            jmp     close1

                        ;---------------------------------------------------------------------
                        ;               restore page 0 and close i/o/e files after error
                        ;---------------------------------------------------------------------
C562 20 B1D0    cloerr  jsr     rszp            restore zero page for sure
C565 2C 54AA            bit     doser           error
C568 10 21              bpl     9.              no
C56A A2 00      cloioe1 ldx     #0              sync entry
C56C 8E 52AA            stx     iocfl           reset redirect level
C56F 8E 54AA            stx     doser           reset error flag
C572 20 49C5            jsr     cloe            close ascii commands

C575 0E 52AA    cloiof  asl     iocfl           decr level i/o
```

```
C578 B0 11            bcs     9.              delay i/o close
C57A 20 8CC5    cloio   jsr     cloi50

                ;---------------------------------------------------------------
                ;               close > saving A,Y,X
                ;---------------------------------------------------------------
C57D 48         cloo50  pha                     close output file, no error in A
C57E 98                 tya
C57F 48                 pha
C580 8A                 txa
C581 48                 pha
C582 20 5FC0            jsr     outclo
C585 68         cloo51  pla
C586 AA                 tax
C587 68                 pla
C588 A8                 tay
C589 68                 pla
C58A 18                 clc
C58B 60         9       rts

                ;---------------------------------------------------------------
                ;               close < saving A,Y,X
                ;---------------------------------------------------------------
C58C 48         cloi50  pha                     close input file, no error in A
C58D 98                 tya
C58E 48                 pha
C58F 8A                 txa
C590 48                 pha
C591 20 5CC0            jsr     inclo
C594 4C 85C5            jmp     cloo51

                ;*******************************************************************
                ;               command <>  postphone reset i/o redirect
                ;*******************************************************************
C597 38         sgpf1   sec
C598 6E 52AA            ror     iocf1           increment level
C59B 60         comf9   rts                     C = 0 assumed level < 8

                ;---------------------------------------------------------------
                ;               < > >> >P commands
                ;---------------------------------------------------------------
C59C 20 FFC5    comf2   jsr     comf1           execute jump
C59F B0 07              bcs     ermese          continue if ok
C5A1 20 E8C1    cmnd2   jsr     setcbp          command interpretation
C5A4 A2 00              ldx     #0              set to first entry command table
C5A6 F0 28              beq     comnex

C5A8 20 62C0    ermese  jsr     seter           set error and print message
C5AB 4C B7D0            jmp     ermes

C5AE 20 F4C1    coma    jsr     sechar          delete spaces
C5B1 F0 E8              beq     comf9           no command
C5B3 C9 3B              cmp     #';'            comment line
C5B5 F0 E4              beq     comf9
C5B7 88                 dey
C5B8 E8         coma1   inx
C5B9 20 CBC1            jsr     bufkey1         get next char
C5BC DD 02C6            cmp     comtb,x
C5BF D0 0F              bne     comnex          not equal
C5C1 C6 FF              dec     mode            dec ness. length
C5C3 D0 F3              bne     coma1
C5C5 20 CBC1    comc    jsr     bufkey1         check other not ness. char
C5C8 F0 25              beq     comfou          eoln
C5CA E8                 inx
C5CB DD 02C6            cmp     comtb,x         compare other command char
C5CE F0 F5              beq     comc
C5D0 CA         comnex  dex                     search next command
C5D1 E8         com20   inx
C5D2 BD 02C6            lda     comtb,x         end mark
C5D5 10 FA              bpl     com20
C5D7 85 FE              sta     opt             command mode flags
C5D9 06 FE              asl     opt
C5DB E8                 inx
C5DC BC 02C6            ldy     comtb,x         set jump address
C5DF 84 F8              sty     tepo
C5E1 E8                 inx
C5E2 BC 02C6            ldy     comtb,x
C5E5 84 F9              sty     tepo+1
C5E7 A0 00              ldy     #0              reset line pointer
C5E9 29 0F              and     #$0f
C5EB 85 FF              sta     mode            length
C5ED D0 BF              bne     coma            if not last

C5EF 20 94C2    comfou  jsr     calposp
```

```
C5F2 A2 00              ldx     #0
C5F4 24 FE              bit     opt             comand flags
C5F6 86 FE              stx     opt             default option clear
C5F8 70 A2              bvs     comf2           i/o file command
C5FA 10 03              bpl     comf1           no zero page save
C5FC 20 AED0            jsr     svzp
C5FF 6C F800    comf1   jmp     [tepo]

C602 82         comtb   fcb     $82             length command
C603 73C8               fdb     goil            jump address
C605 4C434D44           fcc     'LCMD'          command
C609 E2                 fcb     $e2             ex i/o command
C60A 97C5               fdb     sgpfl
C60C 3C3E               fcc     '<>'
C60E E2                 fcb     $e2
C60F 8CC5               fdb     cloi50
C611 3C54               fcc     '<T'
C613 E2                 fcb     $e2
C614 7DC5               fdb     cloo50
C616 3E54               fcc     '>T'
C618 E1                 fcb     $e1
C619 00C4               fdb     seifil
C61B 3C                 fcc     '<'
C61C E1                 fcb     $e1
C61D 20C4               fdb     seofil
C61F 3E                 fcc     '>'
C620 E2                 fcb     $e2
C621 1CC4               fdb     seafil
C623 3E3E               fcc     '>>'
C625 E2                 fcb     $e2
C626 18C4               fdb     seoprt
C628 3E50               fcc     '>P'
                ;       fcb     $81
                ;       fdb     mon
                ;       fcc     'MON'
C62A 81                 fcb     $81
C62B 7BC8               fdb     go
C62D 474F               fcc     'GO'
C62F 84                 fcb     $84
C630 8DC6               fdb     syncc
C632 53594E43           fcc     'SYNC'
C636 84                 fcb     $84
C637 6DC6               fdb     stat
C639 53544154           fcc     'STAT'

C63D C3                 fcb     $c3             cx save zeropage command
C63E C5C6               fdb     type
C640 534545             fcc     'SEE'
C643 C5                 fcb     $c5
C644 E7C6               fdb     chmode
C646 43484D4F44         fcc     'CHMODE'
C64B 45
C64C C1                 fcb     $c1
C64D 07C7               fdb     dir
C64F 444952             fcc     'DIR'
C652 C1                 fcb     $c1
C653 2EC8               fdb     exec
C655 40                 fcc     '@'
C656 C1                 fcb     $c1
C657 3DC8               fdb     run
C659 52554E             fcc     'RUN'
C65C C2                 fcb     $c2
C65D 92C8               fdb     save
C65F 53415645           fcc     'SAVE'
C663 C2                 fcb     $c2
C664 F9C8               fdb     load
C666 4C4F4144           fcc     'LOAD'
C66A C0                 fcb     $c0             end
C66B 32C8               fdb     comrns          jump command

                ;*********************************************************************
                ;               command STAT : list opened files
                ;*********************************************************************
C66D A2 04      stat    ldx     #4
C66F A0 00      stat2   ldy     #0
C671 20 66D0            jsr     status1
C674 B0 0D              bcs     stat1
C676 20 30C1            jsr     prax1
C679 20 3BC0            jsr     print
C67C 206F70656E         fcc     " open\r",0
C681 0D00
C683 E8         stat1   inx
C684 E0 0A              cpx     #4+max_fd
C686 90 E7              bcc     stat2
```

```
C688 60                 rts

                        ;*********************************************************************
                        ;               command SYNC [-P]: close all files
                        ;*********************************************************************
C689 A2 00      sync1   ldx     #0              extern entry
C68B F0 07              beq     sync10
C68D 20 68C0    syncc   jsr     sopt
C690 5000               fcc     'P',0           -p printer spooling
C692 B0 4F              bcs     see9
C694 8E 5BAA    sync10  stx     hdx
C697 20 6AC5            jsr     cloioe1         close i/o/e files
C69A A2 04              ldx     #4
C69C A0 01      sync2   ldy     #1              2 print fds
C69E 8A         1       txa
C69F 59 E0AA            eor     prfd,y          equal print fd
C6A2 F0 05              beq     2.f
C6A4 88                 dey
C6A5 10 F7              bpl     1.b
C6A7 30 08              bmi     3.f
C6A9 2C 5BAA    2       bit     hdx             printfile close
C6AC 30 06              bmi     4.f
C6AE 99 E0AA            sta     prfd,y
C6B1 20 4BD0    3       jsr     close1
C6B4 E8         4       inx
C6B5 E0 0A              cpx     #4+max_fd
C6B7 90 E3              bcc     sync2
C6B9 60         sync9   rts

                        ;*********************************************************************
                        ;               command TYPE filespec ...
                        ;*********************************************************************
C6BA C9 04      see7    cmp     #e4             eof
C6BC D0 25              bne     see9
C6BE 20 48D0            jsr     close
C6C1 E6 FF              inc     mode
C6C3 A9 FA              lda     #$fa            next file
C6C5 A2 81      type    ldx     #$81
C6C7 20 3FD0            jsr     open            open
C6CA B0 0F              bcs     typ8
C6CC 20 00D0    see2    jsr     sread           read single
C6CF B0 E9              bcs     see7
C6D1 20 01C1            jsr     outanl          auto newline
C6D4 20 1ECA            jsr     break
C6D7 90 F3              bcc     see2
C6D9 B0 29              bcs     chm8
C6DB C9 12      typ8    cmp     #e12            no filespec
C6DD D0 04              bne     see9
C6DF A4 FF              ldy     mode            1st file
C6E1 D0 D6              bne     sync9
C6E3 38         see9    sec
C6E4 4C A8C5            jmp     ermese

                        ;*********************************************************************
                        ;               command CHMODE -mode file
                        ;*********************************************************************
C6E7 20 68C0    chmode  jsr     sopt
C6EA 52574400           fcc     'RWD',0         -RWD mode file
C6EE B0 F3              bcs     see9
C6F0 A2 10              ldx     #$10
C6F2 20 3FD0            jsr     open
C6F5 B0 EC              bcs     see9
C6F7 A0 01              ldy     #fmod
C6F9 B1 E0              lda     [varb],y
C6FB 29 1F              and     #$1f
C6FD 05 FE              ora     opt
C6FF A0 02              ldy     #2
C701 20 63D0            jsr     status          put and set file changed flag
C704 4C 48D0    chm8    jmp     close

                        ;*********************************************************************
                        ;               command DIR [-/] [filespec] : directory listing
                        ;*********************************************************************
C707 20 68C0    dir     jsr     sopt
C70A 2F00               fcc     '/',0           -/ all directories
C70C B0 D5              bcs     see9
C70E 20 9FD0            jsr     setfiln         set filename
C711 B0 D0              bcs     see9            error name
C713 24 FE              bit     opt
C715 10 05              bpl     1.
C717 29 E3              and     #$e3
C719 8D EEAB    dir10   sta     fdrive
C71C A9 80      1       lda     #$80
C71E 85 FC              sta     tcn             1st file
```

```
C720 A9 F8              lda     #$f8
C722 A2 01      dir11   ldx     #1
C724 20 3FD0            jsr     open            open
C727 AA                 tax                     ok
C728 F0 04              beq     dir2
C72A C9 13              cmp     #e13
C72C D0 B5              bne     see9
                ;---------------------------------------------------------------
                ;               disk name or directory
                ;---------------------------------------------------------------
C72E 06 FC      dir2    asl     tcn             1st entry
C730 90 48              bcc     dir4
C732 A5 FF              lda     mode            1st directory
C734 D0 1C              bne     4.f
C736 E6 FF              inc     mode
C738 20 3BC0            jsr     print
C73B 0D44697368         fcc     "\rDisk:",0
C740 3A00
C742 A9 20              lda     #' '
C744 A0 3F              ldy     #$3f
C746 C8         2       iny
C747 20 23C0            jsr     out
C74A B1 E2              lda     [sysb],y        floppy name
C74C D0 F8              bne     2.
C74E 24 FE              bit     opt
C750 10 03              bpl     3.
C752 8A         4       txa
C753 D0 45              bne     dir80           empty
C755 AD EEAB    3       lda     fdrive
C758 29 1C              and     #$1c
C75A F0 17              beq     6.f
C75C 20 3BC0            jsr     print
C75F 0D44697265         fcc     "\rDirectory: ",0
C764 63746F7279
C769 3A2000
C76C 4A                 lsra
C76D 4A                 lsra
C76E 09 40              ora     #'@'
C770 20 23C0            jsr     out
C773 A9 80      6       lda     #$80
C775 85 FD      5       sta     tcn+1
C777 20 2FC0            jsr     crlf
                ;---------------------------------------------------------------
                ;               print filename
                ;---------------------------------------------------------------
C77A 8A         dir4    txa
C77B D0 1A              bne     dir8
C77D A9 08              lda     #8
C77F 06 FD              asl     tcn+1
C781 B0 F2              bcs     5.
C783 20 BDD0            jsr     prfile          print from directory
C786 20 32C0    4       jsr     spa             fill until
C789 C8                 iny
C78A C0 13              cpy     #19
C78C 90 F8              bcc     4.
C78E 20 1ECA            jsr     break           test break key
C791 A9 F9              lda     #$f9            next file
C793 90 8D              bcc     dir11           continue
C795 46 FE              lsr     opt
C797 20 2FC0    dir8    jsr     crlf
C79A 24 FE      dir80   bit     opt
C79C 10 0C              bpl     dir90
C79E 06 FF              asl     mode
C7A0 B0 08              bcs     dir90
C7A2 AD EEAB            lda     fdrive
C7A5 69 04              adc     #4
C7A7 4C 19C7            jmp     dir10
                ;---------------------------------------------------------------
                ;               disk sectors left
                ;---------------------------------------------------------------
C7AA 20 2FC0    dir90   jsr     crlf
C7AD A0 28              ldy     #s.acnt
C7AF B1 E2              lda     [sysb],y        allocation sectors / bit
C7B1 AA                 tax
C7B2 F0 01              beq     1.
C7B4 CA                 dex
C7B5 86 FC      1       stx     tcn
C7B7 A2 00              ldx     #0              clear count
C7B9 86 FD              stx     tcn+1
C7BB A0 60              ldy     #$60
C7BD B1 E2      pre2    lda     [sysb],y        check if sector free
C7BF 0A         pre5    asla
C7C0 48                 pha
C7C1 90 0C              bcc     1.f
```

```
C7C3 F8                 sed
C7C4 8A                 txa
C7C5 65 FC              adc     tcn
C7C7 AA                 tax
C7C8 A5 FD              lda     tcn+1
C7CA 69 00              adc     #0
C7CC 85 FD              sta     tcn+1
C7CE D8                 cld
C7CF 68         1       pla
C7D0 D0 ED              bne     pre5
C7D2 C8                 iny
C7D3 D0 E8              bne     pre2
C7D5 A5 FD              lda     tcn+1           print
C7D7 20 26C1            jsr     prax
C7DA 20 3BC0            jsr     print
C7DD 20626C6F63         fcc     " blocks free.\r",0
C7E2 6B73206672
C7E7 65652E0D00
C7EC 60                 rts

                ;---------------------------------------------------------------------
                ;               ascii command level and argument scan
                ;---------------------------------------------------------------------
C7ED AD 62AA    ru2     lda     efd2
C7F0 F0 07              beq     4.
C7F2 8A                 txa
C7F3 48                 pha
C7F4 20 4CC5            jsr     cloe1           close second level
C7F7 68                 pla
C7F8 AA                 tax
C7F9 46 FE      4       lsr     opt
C7FB 6E 60AA            ror     eopt            set command bit
C7FE AD 61AA            lda     efd
C801 D0 27              bne     5.f
C803 8E 61AA            stx     efd             level 1
C806 20 B4D0            jsr     setnex          save argument in ecml
C809 20 E8C1            jsr     setcbp
C80C A2 00              ldx     #0
C80E 20 15C8            jsr     ru10            scan 2 arguments
C811 20 F4C1            jsr     sechar          skip spaces
C814 E8         11      inx
C815 20 CCC1    ru10    jsr     bufkey
C818 F0 0A              beq     3.              argument terminated
C81A B1 FC              lda     [cbp],y
C81C 9D 63AA            sta     ecml,x
C81F C8                 iny
C820 E0 1B              cpx     #eclm-2         max execbuffer
C822 90 F0              bcc     11.
C824 A9 00      3       lda     #0              nul terminate
C826 9D 63AA            sta     ecml,x
C829 60                 rts
C82A 8E 62AA    5       stx     efd2
C82D 60                 rts

                ;*********************************************************************
                ;command @ file : get input for command from file
                ;*********************************************************************
C82E A2 81      exec    ldx     #$81            open for read ascii
C830 D0 0D              bne     ru1

                ;*********************************************************************
                ;               COMMAND arguments
                ;*********************************************************************
C832 A2 81      comrns  ldx     #$81            command
C834 86 FE              stx     opt
C836 8E D0AB            stx     csuf1           set system drive
C839 A2 88              ldx     #$88            command
C83B D0 02              bne     ru1

                ;*********************************************************************
                ;               command RUN file
                ;*********************************************************************
C83D A2 84      run     ldx     #$84            open for read and execute
C83F 20 3FD0    ru1     jsr     open            open
C842 A0 00              ldy     #0
C844 8C D0AB            sty     csuf1           clear
C847 B0 4F              bcs     sv91
C849 A0 04              ldy     #flmo           get filemode
C84B B1 E4              lda     [tslb],y
C84D 29 03              and     #$03
C84F C9 01              cmp     #$01            ascii command
C851 F0 9A              beq     ru2
C853 20 07C9            jsr     ldi             load
C856 C9 04              cmp     #e4             end of file
```

```
C858 D0 37              bne     9.
C85A 20 B1D0            jsr     rszp            restore zp
C85D 20 B4D0            jsr     setnex
C860 A6 F5              ldx     t3+1            startaddress
C862 F0 23              beq     go8             trap address 0
C864 24 FE              bit     opt
C866 10 08              bpl     ru3
C868 8E 5DAA            stx     cjmp+1          save command startaddress
C86B A6 F4              ldx     t3
C86D 8E 5CAA            stx     cjmp
C870 6C F400    ru3     jmp     [t3]

                ;*********************************************************************
                ;               command LCMD
                ;*********************************************************************
C873 AE 5DAA    go11    ldx     cjmp+1          go to last utility
C876 F0 0F              beq     go8             not in mem
C878 6C 5CAA            jmp     [cjmp]

                ;*********************************************************************
                ;               command GO address
                ;*********************************************************************
C87B 20 F1C9    go      jsr     hexpar
C87E B0 07              bcs     go8
C880 E0 80              cpx     #$80
C882 90 03              bcc     go8
C884 6C F000            jmp     [t1]            A,Y set after address
C887 A9 16      go8     lda     #e16
C889 20 4AC1            jsr     priner
C88C 546F3F0D00         fcc     "To?\r",0
C891 60         9       rts

                ;*********************************************************************
                ;               command SAVE [-AB] file begin,end[,start]
                ;*********************************************************************
C892 20 68C0    save    jsr     sopt
C895 414200             fcc     'AB',0          -ab append binary
C898 B0 29      sv91    bcs     sv90
C89A 20 E3C9            jsr     filpar          get filename and addresses
C89D B0 24              bcs     sv90
C89F E0 C0              cpx     #$c0
C8A1 90 53              bcc     sv11
C8A3 20 C0C9            jsr     insvld          calc no bytes
C8A6 90 4E              bcc     sv11
C8A8 06 FE              asl     opt             test append
C8AA A9 F8              lda     #$f8
C8AC 90 10              bcc     sv1
C8AE A2 92              ldx     #$92            r+ b
C8B0 20 3FD0            jsr     open
C8B3 B0 0E              bcs     sv90
C8B5 A9 04              lda     #4              seek end
C8B7 20 12D0            jsr     fseek
C8BA B0 31              bcs     sv36            error check close
C8BC 90 14              bcc     sv40            save binary
C8BE A2 E0      sv1     ldx     #$e0            default mode
C8C0 20 36D0            jsr     create          create file
C8C3 B0 2B      sv90    bcs     sv99
C8C5 20 9CC9            jsr     svmset
C8C8 24 FE              bit     opt
C8CA 10 1B              bpl     sv34            not binary
                ;---------------------------------------------------------------------
                ;               write binary beginaddress, size and startaddress
                ;---------------------------------------------------------------------
C8CC A0 F4              ldy     #t3             save run address
C8CE A9 02              lda     #2
C8D0 D0 04              bne     sv41
C8D2 A0 F8      sv40    ldy     #rbg            save start and length
C8D4 A9 01              lda     #1
C8D6 20 4DC0    sv41    jsr     outputx         single char output
C8D9 B0 12              bcs     sv36
C8DB B9 0000            lda     0,y
C8DE C8                 iny
C8DF C0 F7              cpy     #t3+3
C8E1 F0 EF              beq     sv40
C8E3 C0 FD              cpy     #rln+3
C8E5 D0 EF              bne     sv41
C8E7 20 DEC9    sv34    jsr     isv4
C8EA 20 24D0            jsr     write
C8ED 4C 3CC9    sv36    jmp     ld90
C8F0 4C A8C5    sv99    jmp     ermese
C8F3 20 48D0    sv10    jsr     close           no loading
C8F6 4C 87C8    sv11    jmp     go8             to?

                ;*********************************************************************
```

```
                        ;               command LOAD file [begin[,endaddress]]
                        ;*****************************************************************
C8F9 20 E3C9    load    jsr     filpar          get name and stad
C8FC B0 F2              bcs     sv99
C8FE A2 80              ldx     #prred          open for read
C900 A9 F8              lda     #$f8
C902 20 3FD0            jsr     open            open
C905 B0 E9              bcs     sv99
C907 20 B9C9    ld1     jsr     sv35            get startaddress, runaddress
C90A 24 FF              bit     mode
C90C 30 10              bmi     ld3
C90E A0 01              ldy     #fmod
C910 B1 E0              lda     [varb],y
C912 29 03              and     #$03            binary
C914 F0 08              beq     ld3             memory save
C916 49 02              eor     #$02
C918 F0 2D              beq     ld11            yes
C91A A5 F1              lda     t1+1            page zero?
C91C F0 D5              beq     sv10            int. for sure
C91E 70 06      ld3     bvs     ld2             no endaddress given
C920 A9 FE              lda     #$fe
C922 85 F2              sta     t2
C924 85 F3              sta     t2+1            set max endaddress
C926 20 C0C9    ld2     jsr     insvld          setup readpointers
C929 90 C8              bcc     sv10            error mem
C92B 20 1BD0            jsr     read
C92E B0 07              bcs     ld95            error
C930 20 00D0            jsr     sread           test end file
C933 B0 02              bcs     ld95
C935 A9 01              lda     #e1
C937 C9 04      ld95    cmp     #e4             empty file
C939 F0 06              beq     ld93
C93B 38                 sec
C93C 90 03      ld90    bcc     ld93
C93E 20 A8C5            jsr     ermese
C941 48         ld93    pha
C942 20 48D0            jsr     close
C945 68                 pla
C946 60         ld92    rts

                        ;---------------------------------------------------------------
                        ;               binary load
                        ;---------------------------------------------------------------
C947 20 4AC0    ld11    jsr     inputx
C94A B0 EB              bcs     ld95            empty?
C94C A0 F8              ldy     #rbg
C94E C9 01              cmp     #1              memaddress
C950 F0 06              beq     ld12
C952 A0 F4              ldy     #t3
C954 C9 02              cmp     #2              startaddress
C956 D0 9B              bne     sv10            no
C958 20 4AC0    ld12    jsr     inputx
C95B B0 DA              bcs     ld95            err
C95D 99 0000            sta     0,y
C960 C8                 iny
C961 C0 F6              cpy     #t3+2           run address
C963 F0 E2              beq     ld11
C965 C0 FC              cpy     #rln+2          length
C967 D0 EF              bne     ld12
C969 20 DEC9            jsr     isv4
C96C 20 1BD0            jsr     read
C96F 90 D6              bcc     ld11            ok
C971 B0 C4              bcs     ld95

                        ;---------------------------------------------------------------
                        ;               get or store file addresses
                        ;---------------------------------------------------------------
C973 A0 10      getadr  ldy     #stad           put or get addresses
C975 A2 F0              ldx     #t1
C977 24 FF              bit     mode            test load address
C979 20 80C9            jsr     geadr
C97C A2 F4              ldx     #t3
C97E A9 00              lda     #0
C980 08         geadr   php
C981 20 85C9            jsr     gea1
C984 28                 plp
C985 90 0C      gea1    bcc     gea3
C987 A9 00              lda     #0              put to TSL
C989 24 FE              bit     opt             binary
C98B 30 02              bmi     1.
C98D B5 00              lda     0,x
C98F 91 E4      1       sta     [tslbl],y
C991 B0 06              bcs     gea9
C993 30 04      gea3    bmi     gea9            get from TSL
```

```
C995 B1 E4              lda     [tslb],y
C997 95 00              sta     0,x
C999 C8         gea9    iny
C99A E8                 inx
C99B 60                 rts

                ;---------------------------------------------------------------
                ;               store file mode
                ;---------------------------------------------------------------
C99C A5 FF      svmset  lda     mode            set execute
C99E C9 E0              cmp     #$e0
C9A0 B0 08              bcs     sv32
C9A2 A4 F0              ldy     t1              set default
C9A4 84 F4              sty     t3
C9A6 A4 F1              ldy     t1+1
C9A8 84 F5              sty     t3+1
C9AA A0 01      sv32    ldy     #fmod
C9AC B1 E0              lda     [varb],y
C9AE 29 FC              and     #$fc
C9B0 24 FE              bit     opt
C9B2 10 02              bpl     1.
C9B4 09 02              ora     #$02
C9B6 91 E0      1       sta     [varb],y
C9B8 38                 sec
C9B9 20 73C9    sv35    jsr     getadr          put addresses in TSL block
C9BC AE B3AB            ldx     fd
C9BF 60                 rts

                ;---------------------------------------------------------------
                ;               calculate memory range
                ;---------------------------------------------------------------
C9C0 A5 F0      insvld  lda     t1              setup read write pointers
C9C2 85 F8              sta     rbg
C9C4 A5 F1              lda     t1+1
C9C6 85 F9              sta     rbg+1
C9C8 38                 sec                     calc. bytes
C9C9 A5 F2              lda     t2
C9CB E5 F8              sbc     rbg
C9CD 85 FA              sta     rln
C9CF A5 F3              lda     t2+1
C9D1 E5 F9              sbc     rbg+1
C9D3 85 FB              sta     rln+1
C9D5 E6 FA              inc     rln             incr length
C9D7 D0 05              bne     isv4
C9D9 E6 FB              inc     rln+1
C9DB D0 01              bne     isv4
C9DD 38                 sec                     set C ok
C9DE A0 F8      isv4    ldy     #rbg&255        set A,Y and fd in X
C9E0 A9 00              lda     #rbg>>8
C9E2 60                 rts

                ;---------------------------------------------------------------
                ;               filename and parameters
                ;---------------------------------------------------------------
C9E3 20 9FD0    filpar  jsr     setfiln         get first file name
C9E6 B0 12              bcs     9.
C9E8 8A                 txa
C9E9 D0 03              bne     filp1
C9EB 8D E0AB            sta     namb            clear namebf
C9EE 20 B4D0    filp1   jsr     setnex

C9F1 A2 00      hexpar  ldx     #0
C9F3 20 6BC0            jsr     spar
C9F6 F0 F2 F4           fcb     t1,t2,t3,0
C9F9 00
C9FA 60         9       rts

Errors detected: 0
```

```
                        ;======================================================================
                        ; file          CONSOLE.MAC
                        ;
                        ; program       DOS65
                        ;
                        ; function      Dos65 terminal input/output
                        ;
                        ; by            Ad brouwer
                        ;
                        ; date          20-nov-84
                        ;                7-oct-85       one console file
                        ;               26-apr-86       version 2.01 for Havisoft (ev)
                        ;======================================================================
                        ;comment after end statement

                AA00    wmem    equ     $aa00           work memory
                C000    bast    equ     $c000           base address

                AAC0            org     wmem+$c0        console memory c0-e0

AAC0                    msio    res     1               mode ^S^Q ^C ^Z, lock ^C
AAC1                    ssio    res     1               state after ^S^Q, ^C, lock output
AAC2                    xsio    res     1               save X
AAC3                    sip     res     1               input put pointer
AAC4                    sig     res     1               input get
                001B    sim     equ     wmem+$e0-$
AAC5                    sib     res     sim             standard input buffer

                        ;printer spooler
                AAE4    prpage  equ     wmem+$e4        lines/page
                AAE5    prpagl  equ     wmem+$e5        lines to print

                ABB0    free    equ     wmem+$1b0       dos lock flag (neg if dos in use)


                C020    in      equ     bast+$20        character in/out
                C023    out     equ     bast+$23

                CA00            org     bast+$a00

CA00 4C D2CA                    jmp     incon1          input 0
CA03 4C E0CA                    jmp     outcon1         output 0
CA06 4C D2CA                    jmp     incon1          input 1
CA09 4C C3CA                    jmp     prout1          print 1
CA0C 4C D2CA                    jmp     incon1          input 2
CA0F 4C E0CA                    jmp     outcon1         output 2

CA12 4C CACA                    jmp     chcon1          input check
CA15 4C B5CA                    jmp     statpr1         printer status
CA18 4C BECA                    jmp     putpr1          printer out

CA1B 4C 9CCA                    jmp     term1           setup terminal (at cold start)

CA1E 4C 15CB                    jmp     break1          test break key => C=1 or not => C=0
CA21 4C 83CB                    jmp     brkint1         check and break

CA24 4C 2BCB                    jmp     conint          interrupt entry
CA27 4C 32CB                    jmp     siosub1         subr after drive int.

CA2A 4C F7CA                    jmp     msior1          reset set mode bit
CA2D 4C 0CCB                    jmp     brkset1         set ^C vector

CA30 60                         rts                     no monitor available
CA31 EA                         nop
CA32 EA                         nop

CA33 4C 85CA                    jmp     pbell           ring bell
CA36 4C 89CA                    jmp     pdel1           delete character
CA39 A9 00      clear   lda     #0
CA3B 2C                 fcb     $2c
CA3C A9 01      dircur  lda     #1
CA3E 2C                 fcb     $2c
CA3F A9 02      cleol   lda     #2
CA41 2C                 fcb     $2c
CA42 A9 03      cleos   lda     #3
CA44 EA                 nop
CA45 8E C2AA    attri   stx     xsio
CA48 AA                 tax
CA49 EE C1AA            inc     ssio
CA4C BD 6FCA            lda     atbof,x
CA4F AA                 tax
CA50 BD 75CA    1       lda     atb,x
CA53 F0 13              beq     2.f
```

```
CA55 10 0B              bpl     3.f
CA57 C9 FF              cmp     #$ff
CA59 98                 tya
CA5A B0 03              bcs     4.
CA5C AD C2AA            lda     xsio
CA5F 18         4       clc
CA60 69 01              adc     #1
CA62 20 23C0    3       jsr     out
CA65 E8                 inx
CA66 D0 E8              bne     1.b
CA68 AE C2AA    2       ldx     xsio
CA6B CE C1AA            dec     ssio
CA6E 60                 rts

CA6F 00 02 06   atbof   fcb     atb0-atb,atb1-atb,atb2-atb,atb3-atb
CA72 08
CA73 0A 0D              fcb     atb4-atb,atb5-atb
CA75            atb
CA75 0C00       atb0    fcc     '\f',0
CA77 14FEFF00   atb1    fcc     $14,254,255,0
CA7B 1A00       atb2    fcc     $1a,0
CA7D 1900       atb3    fcc     $19,0
CA7F 1B6E00     atb4    fcc     "\33n",0
CA82 1B6900     atb5    fcc     "\33i",0

CA85 A9 07      pbell   lda     #7              ring bell (07)
CA87 D0 0C              bne     out1

CA89 A9 08      pdell   lda     #8              delete char (08 20 08)
CA8B 20 95CA            jsr     out1
CA8E A9 20              lda     #$20
CA90 20 95CA            jsr     out1
CA93 A9 08              lda     #8
CA95 4C 23C0    out1    jmp     out


CA98 20C0       lumj    fdb     in              setup for monitor i/o
CA9A 23C0               fdb     out

CA9C A9 42      term1   lda     #defpagl        def printer lines/page
CA9E 8D E5AA            sta     prpagl
CAA1 A9 48              lda     #defpage
CAA3 8D E4AA            sta     prpage
CAA6 A9 2A              lda     #conint-1&255   setup keyboard interrupt vector
CAA8 8D 5AE7            sta     kbint
CAAB A9 CB              lda     #conint-1>>8
CAAD 8D 5BE7            sta     kbint+1

CAB0 A2 02              ldx     #$02
CAB2 4C 0CF0            jmp     devini          printer init

CAB5 AD 0DE1    statpr1 lda     p.vaifr         acknoledge
CAB8 29 10              and     #$10
CABA D0 05              bne     1.f             yes
CABC 38                 sec
CABD 60                 rts

CABE 8D 00E1    putpr1  sta     p.vapbd
CAC1 18         1       clc
CAC2 60                 rts

CAC3 A2 02      prout1  ldx     #2
CAC5 20 06F0            jsr     outxdv          select printer as output
CAC8 18                 clc
CAC9 60                 rts

                ;CHCON1 check if input received
                ;=> C=0 yes
                ;=> C=1 no

                ;INCON1 get char from input
                ;=> C=0

CACA 48         chcon1  pha                     test if char in buffer
CACB A9 00              lda     #0
CACD CD B7E7            cmp     keypnt
CAD0 68                 pla
CAD1 60                 rts

CAD2 EE C1AA    incon1  inc     ssio
CAD5 20 03F0            jsr     vidin           get char from buffer
CAD8 20 15CB            jsr     break1
CADB CE C1AA            dec     ssio
CADE 18                 clc
```

```
CADF 60                 rts

                        ;OUTCON1 print char to output
                        ;=> C=0

CAE0 2C C1AA    outcon1 bit     ssio            ^S^Q flag
CAE3 10 05              bpl     1.              no
CAE5 2C C0AA            bit     msio            ^S^Q en.
CAE8 30 F6              bmi     outcon1
CAEA EE C1AA    1       inc     ssio            output char
CAED 20 00F0            jsr     vidout
CAF0 18                 clc
CAF1 CE C1AA            dec     ssio
CAF4 4C 83CB            jmp     brkint1         check ^C

                        ;dis/enable special function input chars

CAF7 78         msior1  sei                     set reset sio mode bit
CAF8 4A                 lsra
CAF9 90 06              bcc     msios3          test bit 0 Accu
CAFB 0A                 asla
CAFC 0D C0AA            ora     msio            enable
CAFF 90 06              bcc     msios2
CB01 0A         msios3  asla
CB02 49 FF              eor     #$ff
CB04 2D C0AA            and     msio            disable
CB07 8D C0AA    msios2  sta     msio
CB0A 58                 cli
CB0B 60                 rts

                        ;set ^C entry vector

CB0C 78         brkset1 sei                     set ^C vector
CB0D 8C B3E7            sty     brkvec
CB10 8D B4E7            sta     brkvec+1
CB13 58                 cli
CB14 60                 rts

                        ;test if ^Z,^C or other break test

CB15 48         break1  pha
CB16 AD C1AA            lda     ssio
CB19 29 60              and     #$60
CB1B F0 0B              beq     brk1            no ^C or ^Z
CB1D AD C1AA            lda     ssio
CB20 29 9F              and     #$9f
CB22 8D C1AA            sta     ssio
CB25 38         brk2    sec
CB26 68                 pla
CB27 60                 rts
                brk1    if      console!=havicon
CB28 18                 clc
CB29 68                 pla
CB2A 60         9       rts

                        ;interrupt handler entered if not drivers interrupt

CB2B AD 01E1    conint  lda     c.vbpad         via input
CB2E 20 33CB            jsr     si0
CB31 60                 rts

                        ;drive timer subroutine entry if polling input
                        ;currently no used

CB32 60         siosub1 rts                     no function

                si0     if      console!=havicon
CB33 AE B7E7            ldx     keypnt
CB36 A8                 tay
CB37 A9 20              lda     #$20
CB39 2C C0AA            bit     msio
CB3C F0 04              beq     si6
CB3E C0 1A              cpy     #$1a            ^Z
CB40 F0 27              beq     si12
CB42 50 05      si6     bvc     si2             ^C
CB44 0A                 asla
CB45 C0 03              cpy     #3
CB47 F0 20              beq     si12
CB49 2C C1AA    si2     bit     ssio            locked output
CB4C 30 09              bmi     si13
CB4E 2C C0AA            bit     msio            control func.
CB51 10 24              bpl     si3
CB53 C0 11              cpy     #$11            ^Q
```

```
CB55 D0 08              bne     si10
CB57 AD C1AA    si13    lda     ssio
CB5A 29 7F              and     #$7f            enable output
CB5C 4C 6CCB            jmp     si14
                si10    if      console==havicon
CB5F C0 80              cpy     #$80            start/stop key
CB61 F0 04              beq     21.
CB63 C0 13              cpy     #$13            ^S
CB65 D0 10              bne     si3
CB67 A9 80      21      lda     #$80
CB69 0D C1AA    si12    ora     ssio            disable output
CB6C 8D C1AA    si14    sta     ssio
CB6F A2 00              ldx     #0
CB71 8E B7E7            stx     keypnt          reset pointer
CB74 4C 83CB            jmp     si4
CB77 98         si3     tya
CB78 9D B8E7            sta     keybuf,x
CB7B E8                 inx
CB7C E0 27              cpx     #kbfmax
CB7E B0 03              bcs     1.
CB80 8E B7E7            stx     keypnt
CB83                    1
CB83            si4                             ;walk through break interrupt check

                ;check to stop current program

CB83 08         brkint1 php
CB84 2C C1AA            bit     ssio            ^C
CB87 50 1C              bvc     7.              no
CB89 2C B0AB            bit     free            free
CB8C 30 17              bmi     7.              no
CB8E 48                 pha
CB8F AD C0AA            lda     msio
CB92 29 02              and     #$02            locked
CB94 0D C1AA            ora     ssio            output
CB97 29 0F              and     #$0f
CB99 D0 09              bne     8.
CB9B 8D C1AA            sta     ssio            reset state
CB9E 68                 pla
CB9F 28                 plp
CBA0 58                 cli
CBA1 6C B3E7            jmp     [brkvec]        might be a subroutine
CBA4 68         8       pla
CBA5 28         7       plp
CBA6 60         9       rts

Errors detected: 0
```

```
                ;=========================================================
                ; file          drivers.mac
                ;
                ; program       dos65
                ;
                ; function      dos65 disk i/o drivers
                ;               time update
                ;               printer spooler
                ;
                ; by            ad brouwer
                ;
                ; date          20-nov-84
                ;               21-sep-85       write protect bug repaired
                ;                7-oct-85       one driver file
                ;               13-nov-85       winchester drive
                ;               24-apr-86       version 2.01 Havisoft (ev)
                ;=========================================================

          E000  iopag   equ     d.pia&$ff00


                ;fdc controller and pia

          E002  pia.b   equ     d.pia+2         select drives and test DRQ, INTR
          E003  pia.cb  equ     d.pia+3         port A not used

          E004  fdc.cr  equ     d.fdc           command reg WD 1793
          E004  fdc.sr  equ     d.fdc           status reg
          E005  fdc.tkr equ     d.fdc+1         track reg
          E006  fdc.scr equ     d.fdc+2         sector reg
          E007  fdc.dtr equ     d.fdc+3         data reg

                ;fdc commands

          0008  rest    equ     $08!trkaccess   restore
          0018  seekt   equ     $18!trkaccess   seek
          0080  rdskt   equ     $80             read sector
          00A0  wrskt   equ     $a0             write sector

                ;floppy disk select bits

          0007  fdecode equ     7

          AA00  wmem    equ     $aa00           work memory
          C000  bast    equ     $c000           base address dos65

                ;memory for diskdrivers
                ;wmem+$f0 - wmem+$ff

          AAF0          org     wmem+$f0

AAF0              dtype   res     4               drive type (7:virtual 6:winches)
AAF4              ftrack  res     4               track / density floppy drive
AAF8              ctype   res     1               current drive type
AAF9              drive   res     1               last drive accessed
AAFA              fdrive  res     1               last floppy drive
AAFB              ctrack  res     1               current head position
AAFC              ercnt   res     1               error / change density
AAFD              ytmp    res     1
AAFE              tmcnt   res     2               timer count 20 ms. 1 s.

                ;memory for printer spooler
                ;wmem+$e0 - wmem+$ef
                ;the last 8 bytes are for scratch

          AAE0          org     wmem+$e0        printer spooler

AAE0              prfd    res     1               file descr print file
AAE1              prfd2   res     1               next file
AAE2              propt   res     1               options
AAE3              prent   res     1               entry and flags
AAE4              prpage  res     1               lines/page
AAE5              prpagl  res     1               lines to print
AAE6                      res     2               res
AAE8              prcc    res     1               pos on line
AAE9              prlc    res     1               pos on page
AAEA              prcoun  res     1               print char count
AAEB              prtmp   res     1               print char. buffer
                  ;4 res

                ;dos65 entries for print
```

```
                D000  sread   equ     bast+$1000
                D048  close   equ     bast+$1048
                D063  status  equ     bast+$1063
                D0AE  svzp    equ     bast+$10ae
                D0B1  rszp    equ     bast+$10b1

                      ;predefined memory

                00E8  rwpoin  equ     $e8             pointer to mem for r/w sector
                ABB0  free    equ     wmem+$1b0       doslock flag
                ABFC  date    equ     wmem+$1fc       time and date (4 bytes)

                      ;entries in console part

                CA15  statpri equ     bast+$a15       printer status (busy)
                CA18  putpri  equ     bast+$a18       printer output

                CA1B  term    equ     bast+$a1b       init console and printer
                CA24  intsio  equ     bast+$a24       standard i/o int.
                CA27  siosub  equ     bast+$a27       subroutine drivers int.

                CC00          org     bast+$c00       drivers object in dos

CC00 03                       fcb     maxdrive        number of drives
CC01 05         hddrof        fcb     5               drive-select time
CC02 08         mtdrof        fcb     8               motor-on time

CC03 0E         voffset       fcb     vdoffset        Vdisk offset
CC04 32         vtracks       fcb     vdtracks        Vdisk tracks

CC05 4C AFCC                  jmp     inihw1          init timers and fdc pia
CC08 4C 82CE                  jmp     rdyhw1          select drive, test ready
CC0B 4C DACD                  jmp     wshw1           write track X sector Y
CC0E 4C 06CE                  jmp     rshw1           read tX sY
CC11 4C 37CE                  jmp     cshw1           check tX sY
CC14 00 00 00                 fcb     0,0,0,0,0,0     reserved
CC17 00 00 00
CC1A 4C D1CD                  jmp     sethmt1         set select, motor time
CC1D 4C A7CC                  jmp     settd1          set time and date

CC20 D8         intr          cld
CC21 48                       pha
CC22 2C 04E1                  bit     t.via+4         clear interrupt
CC25 8A                       txa                     deselect drive
CC26 48                       pha
CC27 98                       tya
CC28 48                       pha
CC29 20 27CA                  jsr     siosub          subroutine in console
CC2C 2C B0AB                  bit     free            locked dos
CC2F 30 44                    bmi     20.f

CC31 CE FEAA                  dec     tmcnt           dec part of second
CC34 10 28                    bpl     10.f            not yet zero
CC36 A9 13                    lda     #20-1
CC38 8D FEAA                  sta     tmcnt
CC3B A9 17                    lda     #$10+fdecode    motor off
CC3D AE FFAA                  ldx     tmcnt+1
CC40 EC 02CC                  cpx     mtdrof          motor off time
CC43 F0 10                    beq     7.              yes
CC45 B0 14                    bcs     5.
CC47 EC 01CC                  cpx     hddrof          head off?
CC4A F0 07                    beq     4.
CC4C B0 0A                    bcs     6.
CC4E AD 04E0                  lda     fdc.sr          drive ready
CC51 10 05                    bpl     6.              yes
CC53 A9 07      4             lda     #fdecode        deselect drives
CC55 20 6BCF    7             jsr     pbor
CC58 EE FFAA    6             inc     tmcnt+1         inc seconds
                5             if      timer==rtctim
CC5B 20 7DCC                  jsr     tupdate

CC5E AE E0AA    10            ldx     prfd            file spooling?
CC61 F0 12                    beq     20.f
CC63 38                       sec
CC64 6E B0AB                  ror     free            lock
CC67 20 AED0                  jsr     svzp            save zeropage
CC6A 58                       cli
CC6B 20 FECC                  jsr     prsp1           print
CC6E 78                       sei
CC6F 20 B1D0                  jsr     rszp            restore zpage
CC72 0E B0AB                  asl     free            unlock
CC75 68         20            pla
CC76 A8                       tay
CC77 68                       pla
```

```
CC78 AA                 tax
CC79 68                 pla
CC7A 4C 0000    intre   jmp     $0000           update time

CC7D AD 82E7    tupdate lda     hours
CC80 8D FCAB            sta     date
CC83 AD 83E7            lda     minutes
CC86 8D FDAB            sta     date+1
CC89 AD 7FE7            lda     day
CC8C 0A                 asla
CC8D 0A                 asla
CC8E 0A                 asla
CC8F 8D FEAB            sta     date+2
CC92 AD 80E7            lda     month
CC95 4A                 lsra
CC96 08                 php
CC97 0D FEAB            ora     date+2
CC9A 8D FEAB            sta     date+2
CC9D AD 81E7            lda     year
CCA0 0A                 asla
CCA1 28                 plp
CCA2 6A                 rora
CCA3 8D FFAB            sta     date+3
CCA6 60                 rts

CCA7 18         settd1  clc                     set time and date
CCA8 A0 00              ldy     #0              not used here
CCAA 60                 rts

                ;drive type initialization table

CCAB 00 01 80   ddeft   fcb     dtype0,dtype1,dtype2,dtype3
CCAE 00

CCAF A0 03      inihw1  ldy     #3
CCB1 B9 ABCC    1       lda     ddeft,y         define drive type
CCB4 99 F0AA            sta     dtype,y
CCB7 88                 dey
CCB8 10 F7              bpl     1.b
CCBA A9 04              lda     #4
CCBC 2C 03E0            bit     pia.cb          pia initialized
CCBF F0 02              beq     2.
CCC1 A0 04              ldy     #itb1-initb-1   past init fdc pia
CCC3 78         2       sei
CCC4 C8         2       iny                     init via table
CCC5 B9 F8CC            lda     initc,y         get data
CCC8 BE F2CC            ldx     initb,y         get address
CCCB 9D 00E0    3       sta     iopag,x         init on io page
CCCE E0 04              cpx     #fdc.cr         end
CCD0 D0 F2              bne     2.
                ;       lda     tmint+1
                ;       cmp     #intr-1>>8      already set?
                ;       beq     1.f
CCD2 18                 clc
CCD3 AD 50E7            lda     tmint
CCD6 69 01              adc     #$01
CCD8 8D 78CC            sta     intre+1
CCDB AD 51E7            lda     tmint+1
CCDE 69 00              adc     #$00
CCE0 8D 7CCC            sta     intre+2
CCE3 A9 1F              lda     #intr-1&255     set IRQ vector
CCE5 8D 50E7            sta     tmint
CCE8 A9 CC              lda     #intr-1>>8
CCEA 8D 51E7            sta     tmint+1
CCED 20 1BCA            jsr     term            init terminal driver
CCF0 58         1       cli

CCF1 60                 rts                     if no via

CCF2 03 02 03   initb   fcb     pia.cb,pia.b,pia.cb,pia.b,pia.cb ;fdc pia
CCF5 02 03
CCF7 04         itb1    fcb     fdc.cr          init fdc (end)

CCF8 04 1F 00   initc   fcb     4,$1f,0,$3f,4   init fdc pia
CCFB 3F 04
CCFD D0                 fcb     $d0             forced int. fdc

CCFE AC EAAA    prspl   ldy     prcoun          routine interrupt printing
CD01 A9 19              lda     #25             for non buffered printers
CD03 8D EAAA            sta     prcoun
CD06 2C E3AA            bit     prent           test start spooling
CD09 10 0C              bpl     1.f
CD0B A0 01              ldy     #1              init variables
CD0D 8C E8AA            sty     prcc
```

```
CD10 88                 dey
CD11 8C E3AA            sty     prent
CD14 8C E9AA            sty     prlc
CD17 98         1       tya
CD18 F0 03              beq     10.
CD1A 4C A1CD            jmp     51.f

CD1D AD E3AA    10      lda     prent           tabspaces
CD20 0A                 asla
CD21 10 42              bpl     20.f            no extra
CD23 0A                 asla
CD24 10 6C              bpl     45.f            tab
CD26 CE E8AA            dec     prcc
CD29 EE E9AA            inc     prlc
CD2C A0 0A              ldy     #'\n'           line feed
CD2E 0A                 asla
CD2F 30 1A              bmi     2.f             form feed
CD31 A9 20              lda     #$20
CD33 2C E2AA            bit     propt
CD36 F0 63              beq     49.f            no page opt
CD38 A0 00              ldy     #0
CD3A 20 63D0            jsr     status
CD3D A0 0A              ldy     #'\n'
CD3F C9 04              cmp     #4
CD41 F0 08              beq     2.f
CD43 AD E9AA            lda     prlc
CD46 CD E5AA            cmp     prpagl          test max lines
CD49 90 50              bcc     49.f
CD4B A9 10      2       lda     #$10
CD4D 2C E2AA            bit     propt
CD50 D0 0A              bne     1.f
CD52 AD E9AA            lda     prlc
CD55 CD E4AA            cmp     prpage          test end page
CD58 90 44              bcc     50.f
CD5A F0 02              beq     2.f
CD5C A0 0C      1       ldy     #'\f'           form feed
CD5E A9 00      2       lda     #0
CD60 8D E9AA            sta     prlc
CD63 F0 36              beq     49.f

CD65 20 00D0    20      jsr     sread           read single char
CD68 B0 52              bcs     80.f            error
CD6A A8                 tay
CD6B 2C E2AA            bit     propt           auto lf
CD6E 30 2E              bmi     50.f
CD70 49 0D              eor     #'\r'
CD72 D0 08              bne     2.f
CD74 8D E8AA            sta     prcc
CD77 A9 60              lda     #$60            set lf
CD79 8D E3AA            sta     prent
CD7C C0 0C      2       cpy     #'\f'
CD7E D0 07              bne     2.f
CD80 A9 70              lda     #$70            set ff
CD82 8D E3AA            sta     prent
CD85 D0 2D              bne     52.f
CD87 70 15      2       bvs     50.f            no tabexpand
CD89 C0 09              cpy     #'\t'
CD8B D0 11              bne     50.f
CD8D A9 40              lda     #$40
CD8F 8D E3AA            sta     prent
CD92 A0 20      45      ldy     #' '            tab expand
CD94 AD E8AA            lda     prcc
CD97 29 07              and     #$07
CD99 D0 03              bne     50.f
CD9B 4E E3AA    49      lsr     prent

CD9E 8C EBAA    50      sty     prtmp
CDA1 A0 0A      51      ldy     #10             test max 10 *
CDA3 88         1       dey
CDA4 F0 2A              beq     90.f
CDA6 20 15CA            jsr     statpri
CDA9 B0 F8              bcs     1.b
CDAB AD EBAA            lda     prtmp
CDAE 20 18CA            jsr     putpri          print
CDB1 EE E8AA            inc     prcc
CDB4 CE EAAA    52      dec     prcoun
CDB7 F0 17              beq     90.f
CDB9 4C 1DCD            jmp     10.b

CDBC A8         80      tay                     test r/w err
CDBD 30 03              bmi     1.f
CDBF 20 48D0            jsr     close
CDC2 AD E1AA    1       lda     prfd2           next file
CDC5 8D E0AA            sta     prfd
```

```
CDC8 A9 00              lda     #0
CDCA 8D E1AA            sta     prfd2
CDCD 8D EAAA            sta     prcoun
CDD0 60         90      rts

                ;set drive select and motor on time

CDD1 8C 01CC    sethmt1 sty     hddrof          set time
CDD4 8D 02CC            sta     mtdrof
CDD7 4C E0CE            jmp     rdy9            reset counter

                ;write sector on track X sector Y

CDDA 20 EACE    wshw1   jsr     setup           select drive/side/track/sector
CDDD B0 54              bcs     rs9             not ready
CDDF 30 70              bmi     wvirt

CDE1 08         ws1     php                     floppy write sector
CDE2 78                 sei
CDE3 A0 A0              ldy     #wrskt          write sector command
CDE5 20 8FCF            jsr     stcm            return Y=0
CDE8 90 04              bcc     ws4             always
CDEA 8D 07E0    ws3     sta     fdc.dtr         store in data reg.
CDED C8                 iny
CDEE B1 E8      ws4     lda     [rwpoin],y      get byte
CDF0 2C 02E0    ws5     bit     pia.b           test DRQ, INTR
CDF3 30 F5              bmi     ws3             DRQ
CDF5 50 F9              bvc     ws5             no INTR, continue loop
CDF7 AD 04E0            lda     fdc.sr          get status
CDFA 28                 plp
CDFB 29 DC              and     #$dc            ready, wp, rnf, crc and data lost
CDFD F0 31              beq     rs8             ok, get Y
CDFF 20 23CF            jsr     resdens         change density or restore
CE02 90 DD              bcc     ws1             try again
CE04 B0 2A              bcs     rs8             error, get result

                ;read sector track X sector Y

CE06 20 EACE    rshw1   jsr     setup           select drive/track/sector
CE09 B0 28              bcs     rs9             not ready
CE0B 30 45              bmi     rvirt

CE0D 08         rs1     php                     floppy read sector
CE0E 78                 sei
CE0F A0 80              ldy     #rdskt          read sector command
CE11 20 8FCF            jsr     stcm            Y=0
CE14 90 06              bcc     4.              always
CE16 AD 07E0    3       lda     fdc.dtr         get data
CE19 91 E8              sta     [rwpoin],y
CE1B C8                 iny
CE1C 2C 02E0    4       bit     pia.b           test DRQ and INTR
CE1F 30 F5              bmi     3.              DRQ
CE21 50 F9              bvc     4.              no INTR, continue looking
CE23 AD 04E0            lda     fdc.sr
CE26 28                 plp
CE27 29 9C              and     #$9c            ready, rnf, crc and data lost
CE29 F0 08              beq     rs9             ok
CE2B 20 23CF            jsr     resdens         change density or restore
CE2E 90 DD              bcc     rs1             try again
CE30 2C 04E0    rs8     bit     fdc.sr          result if error
CE33 AC FDAA    rs9     ldy     ytmp            restore Y
CE36 60         rs10    rts                     C = error (write V = write prot)

                ;check sector track X sector Y

CE37 20 EACE    cshw1   jsr     setup
CE3A B0 F7              bcs     rs9
CE3C 30 F5              bmi     rs9

                ;floppy check sector

CE3E A0 80      cs1     ldy     #rdskt          read command
CE40 20 4FCF            jsr     comst           start and wait till finished
CE43 AD 04E0            lda     fdc.sr          status
CE46 29 98              and     #$98            no test on data lost
CE48 F0 E9              beq     rs9             ok
CE4A 20 23CF            jsr     resdens
CE4D 90 EF              bcc     cs1             try again
CE4F B0 E2              bcs     rs9             error


                ; read write virtual sector
```

```
                00FE  mempnt  equ     $fe             memory pointer

CE51 38               wvirt   sec
CE52 08               rvirt   php                     no interrupt allowed
CE53 78                       sei
CE54 8D FDFF                  sta     vddatadr        select ram segment
CE57 A5 FF                    lda     mempnt+1        save memory pointer
CE59 48                       pha
CE5A A5 FE                    lda     mempnt
CE5C 48                       pha
CE5D 84 FF                    sty     mempnt+1        set high address
CE5F A0 00                    ldy     #0
CE61 84 FE                    sty     mempnt          clear low
CE63 90 06            4       bcc     5.
CE65 B1 E8                    lda     [rwpoin],y      write to virtual
CE67 91 FE                    sta     [mempnt],y
CE69 B0 04                    bcs     6.
CE6B B1 FE            5       lda     [mempnt],y      read from virtual
CE6D 91 E8                    sta     [rwpoin],y
CE6F C8               6       iny
CE70 D0 F1                    bne     4.
CE72 68                       pla                     restore memory pointer
CE73 85 FE                    sta     mempnt
CE75 68                       pla
CE76 85 FF                    sta     mempnt+1
CE78 A9 02                    lda     #vdorgblk^$0f
CE7A 8D FDFF                  sta     vddatadr
CE7D 28                       plp
CE7E 18                       clc                     result ok
CE7F 4C 33CE                  jmp     rs9

                      ;select drive A

CE82 8C FDAA    rdyhw1  sty     ytmp
CE85 8D F9AA            sta     drive
CE88 A8                 tay
CE89 B9 F0AA            lda     dtype,y         drive type
CE8C 8D F8AA            sta     ctype
CE8F 30 47              bmi     rdy8            virtual or winchester
CE91 29 03              and     #3
CE93 CD FAAA            cmp     fdrive          same floppy drive?
CE96 F0 29              beq     rdy3
CE98 AC FAAA            ldy     fdrive
CE9B 8D FAAA            sta     fdrive
CE9E A9 07              lda     #fdecode        deselect drives
CEA0 20 6BCF            jsr     pbor
CEA3 0A                 asla
CEA4 0A                 asla
CEA5 29 80              and     #$80            density
CEA7 49 80              eor     #$80
CEA9 0D F8AA            ora     ctrack          last track
CEAC 99 F4AA            sta     ftrack,y
CEAF AC FAAA            ldy     fdrive          new drive
CEB2 B9 F4AA            lda     ftrack,y
CEB5 C9 80              cmp     #$80
CEB7 29 7F              and     #$7f
CEB9 8D F8AA            sta     ctrack          new track
CEBC A9 20              lda     #$20
CEBE 20 69CF            jsr     pbaor           set dens
CEC1 A9 FF      rdy3    lda     #$ff
CEC3 8D FFAA            sta     tmcnt+1         stop drive time
CEC6 20 70CF            jsr     fdsel
CEC9 2C 04E0    2       bit     fdc.sr          test ready
CECC 10 0A              bpl     rdy8
CECE A9 10              lda     #100/6
CED0 20 87CF            jsr     wait
CED3 CE FFAA            dec     tmcnt+1         max 128
CED6 30 F1              bmi     2.b
CED8 AC FDAA    rdy8    ldy     ytmp
CEDB 2C F8AA            bit     ctype
CEDE 30 41              bmi     8.f
CEE0 A9 00      rdy9    lda     #0
CEE2 8D FFAA            sta     tmcnt+1         reset drive timer
CEE5 AD 04E0            lda     fdc.sr
CEE8 0A                 asla
CEE9 60                 rts                     C: not ready

CEEA 8C FDAA    setup   sty     ytmp
CEED 2C F8AA            bit     ctype
CEF0 10 17              bpl     10.f            floppy disk
CEF2 88                 dey
CEF3 C0 10              cpy     #16             max sector
CEF5 B0 2B              bcs     9.f
```

```
CEF7 EC 04CC            cpx     vtracks         max track
CEFA B0 26              bcs     9.f
CEFC 98                 tya
CEFD 09 D0              ora     #vdmemadr>>8    set page address
CEFF A8                 tay
CF00 8A                 txa
CF01 6D 03CC            adc     voffset         first block
CF04 49 0F              eor     #$0f            toggle for S189
CF06 90 16              bcc     7.f
CF08 60                 rts                     wrong size

CF09 A9 0D      10      lda     #$d             error and change density
CF0B 8D FCAA            sta     ercnt           ch - cdrs - cd
CF0E 98                 tya
CF0F 29 7F              and     #$7f
CF11 8D 06E0            sta     fdc.scr
CF14 C0 80              cpy     #$80
CF16 A9 08              lda     #$08            side select
CF18 20 69CF            jsr     pbaor
CF1B 20 33CF            jsr     tseek
CF1E 2C F8AA    7       bit     ctype
CF21 18         8       clc
CF22 60         9       rts

CF23 4E FCAA    resdens lsr     ercnt           max 3 errors
CF26 F0 FA              beq     9.
CF28 20 48CF            jsr     restor          restore and change dens.

CF2B AD 02E0    chdens  lda     pia.b           switch density
CF2E 49 20              eor     #$20
CF30 8D 02E0            sta     pia.b

                tseek   if      drivers&fd2stp
CF33 8A                 txa
CF34 29 7F      1       and     #$7f
CF36 CD FBAA            cmp     ctrack
CF39 F0 1C              beq     3.f             head on same track
CF3B 8D 07E0            sta     fdc.dtr         desired track
CF3E AC FBAA            ldy     ctrack
CF41 8C 05E0            sty     fdc.tkr         current head position
CF44 A0 18              ldy     #seekt
CF46 D0 04              bne     1.f
CF48 A0 0B      restor  ldy     #rest           restore
CF4A A9 00              lda     #0
CF4C 8D FBAA    1       sta     ctrack

CF4F 20 8FCF    comst   jsr     stcm
CF52 2C 02E0    2       bit     pia.b           wait till command finished
CF55 50 FB              bvc     2.

CF57 8A         3       txa
CF58 29 7F              and     #$7f
CF5A 8D 05E0            sta     fdc.tkr         actual r/w track

CF5D 18                 clc
CF5E 60                 rts

CF5F 49 FF      pband   eor     #$ff            set bit zero
CF61 2D 02E0            and     pia.b
CF64 8D 02E0    2       sta     pia.b
CF67 18                 clc
CF68 60                 rts

CF69 B0 F4      pbaor   bcs     pband
CF6B 0D 02E0    pbor    ora     pia.b           set bit to one
CF6E D0 F4              bne     2.b

CF70 A9 10      fdsel   lda     #$10            motor on
CF72 20 7DCF            jsr     fdsl1
CF75 AD FAAA            lda     fdrive          drive select
CF78 0A                 asla
CF79 D0 02              bne     1.
CF7B A9 01              lda     #1
CF7D            1
CF7D 2C 02E0    fdsl1   bit     pia.b           test on
CF80 F0 0C              beq     9.              yes
CF82 20 5FCF            jsr     pband           select
CF85 A9 C8              lda     #1200/6

CF87 20 94CF    wait    jsr     waity
CF8A E9 01              sbc     #1
CF8C B0 F9              bcs     wait
CF8E 60         9       rts
```

```
CF8F 8C 04E0    stcm    sty     fdc.cr          start command
CF92 A0 06              ldy     #6              wait 40 usec
CF94 88         waity   dey
CF95 D0 FD              bne     waity
CF97 60                 rts

Errors detected: 0
```

```
                        ;=====================================================================
                        ; file          dfn.mac
                        ;
                        ; program       dos65
                        ;
                        ; function      dos65 file management
                        ;
                        ; by            ad brouwer
                        ;
                        ; date            -apr-84
                        ;               10-nov-85       2.1 extented disk storage
                        ;               26-nov-85       2.1 directories
                        ;               28-jun-86       dos65 2.01 Octopus hardware
                        ;=====================================================================
                        ;*********************************************************************
                        ;               function entries
                        ;*********************************************************************
C000                    bast    equ     $c000
D000                            org     bast+$1000

                        func    macro   ofs
                                clc                     no save zero page
                                bcc     1.
                                sec                     save zero page
                        1       pha
                                lda     #ofs
                                if      ofs!=22
                                bpl     jdfun
                                endif
                                endm

                                func    0               single read
D000 18                         clc                     no save zero page
D001 90 01                      bcc     1.
D003 38                         sec                     save zero page
D004 48                 1       pha
D005 A9 00                      lda     #0
D007 10 61                      bpl     jdfun
                                func    2               single write
D009 18                         clc                     no save zero page
D00A 90 01                      bcc     1.
D00C 38                         sec                     save zero page
D00D 48                 1       pha
D00E A9 02                      lda     #2
D010 10 58                      bpl     jdfun
                                func    4               seek function
D012 18                         clc                     no save zero page
D013 90 01                      bcc     1.
D015 38                         sec                     save zero page
D016 48                 1       pha
D017 A9 04                      lda     #4
D019 10 4F                      bpl     jdfun
                                func    6               read
D01B 18                         clc                     no save zero page
D01C 90 01                      bcc     1.
D01E 38                         sec                     save zero page
D01F 48                 1       pha
D020 A9 06                      lda     #6
D022 10 46                      bpl     jdfun
                                func    8               write
D024 18                         clc                     no save zero page
D025 90 01                      bcc     1.
D027 38                         sec                     save zero page
D028 48                 1       pha
D029 A9 08                      lda     #8
D02B 10 3D                      bpl     jdfun
                                func    10              seek
D02D 18                         clc                     no save zero page
D02E 90 01                      bcc     1.
D030 38                         sec                     save zero page
D031 48                 1       pha
D032 A9 0A                      lda     #10
D034 10 34                      bpl     jdfun
                                func    12              create
D036 18                         clc                     no save zero page
D037 90 01                      bcc     1.
D039 38                         sec                     save zero page
D03A 48                 1       pha
D03B A9 0C                      lda     #12
D03D 10 2B                      bpl     jdfun
                                func    14              open
D03F 18                         clc                     no save zero page
D040 90 01                      bcc     1.
D042 38                         sec                     save zero page
```

```
D043 48         1       pha
D044 A9 0E              lda     #14
D046 10 22              bpl     jdfun
                        func    16              close
D048 18                 clc                     no save zero page
D049 90 01              bcc     1.
D04B 38                 sec                     save zero page
D04C 48         1       pha
D04D A9 10              lda     #16
D04F 10 19              bpl     jdfun
                        func    18              delete
D051 18                 clc                     no save zero page
D052 90 01              bcc     1.
D054 38                 sec                     save zero page
D055 48         1       pha
D056 A9 12              lda     #18
D058 10 10              bpl     jdfun
                        func    20              rename
D05A 18                 clc                     no save zero page
D05B 90 01              bcc     1.
D05D 38                 sec                     save zero page
D05E 48         1       pha
D05F A9 14              lda     #20
D061 10 07              bpl     jdfun
                        func    22              status
D063 18                 clc                     no save zero page
D064 90 01              bcc     1.
D066 38                 sec                     save zero page
D067 48         1       pha
D068 A9 16              lda     #22
D06A 08         jdfun   php
D06B 38                 sec
D06C 6E B0AB            ror     free            lock dos65
D06F 28                 plp
D070 6E B1AB            ror     svzpf           page 0 save flag
D073 10 03              bpl     1.
D075 20 FDD0            jsr     svzp1
D078 84 E9      1       sty     rwpoin+1
D07A A8                 tay
D07B B9 CCD0            lda     funlu,y         jumpaddress
D07E 85 E8              sta     rwpoin
D080 B9 CDD0            lda     funlu+1,y
D083 A4 E9              ldy     rwpoin+1
D085 85 E9              sta     rwpoin+1
D087 68                 pla
D088 20 9BD0            jsr     jufun
D08B 08                 php
D08C 2C B1AB            bit     svzpf
D08F 10 03              bpl     1.
D091 20 E4D0            jsr     rszp1
D094 0E B0AB    1       asl     free            unlock
D097 28                 plp
D098 4C 21CA            jmp     brkint          check ^C int
D09B 6C E800    jufun   jmp     [rwpoin]

                ;---------------------------------------------------------------
                ;               special entries
                ;---------------------------------------------------------------
        D09F            org     bast+$109f

D09F 4C 2BD2            jmp     namdef          define name (default *)
D0A2 4C F0D2            jmp     namcom          compare with name
D0A5 4C 09DA            jmp     inidir          open directory read (if ness) SIS
D0A8 4C 1EDA            jmp     nexdir          next dir entry
D0AB 4C 8BDA            jmp     wrtsdi          write sis/dir
D0AE 4C FDD0            jmp     svzp1           save zp
D0B1 4C E4D0            jmp     rszp1           restore zp
D0B4 4C 3AD2            jmp     setnex1         get pos. next filename
D0B7 4C 39DE            jmp     ermes1
D0BA 4C F9DE            jmp     prname1
D0BD 4C 09DF            jmp     prfile1
D0C0 4C C3DD            jmp     readsect1
D0C3 4C A8DD            jmp     writsect1
D0C6 4C B8DD            jmp     checsect1
D0C9 4C 8DDD            jmp     wrchsect1

D0CC B0D6       funlu   fdb     sread2          jump addresses
D0CE A4D6               fdb     swrite2
D0D0 BCD6               fdb     fseek2
D0D2 4DD7               fdb     read2
D0D4 E5D6               fdb     write2
D0D6 99D7               fdb     seek2
D0D8 8CD3               fdb     create2
D0DA 39D4               fdb     open2
```

```
D0DC 58D5               fdb     close2
D0DE 1CD5               fdb     delete2
D0E0 B6D5               fdb     rename2
D0E2 0BD6               fdb     status2

                ;*************************************************************************
                ;               restore / save page zero addresses e0-ef
                ;*************************************************************************
D0E4 08         rszp1   php                     restore zeropage
D0E5 78                 sei
D0E6 48                 pha
D0E7 8A                 txa
D0E8 48                 pha
D0E9 98                 tya
D0EA 48                 pha
D0EB A2 0F              ldx     #$f
D0ED AC B2AB            ldy     zbsp            nothing left
D0F0 F0 26              beq     sz2
D0F2 B9 8FAB    rz1     lda     zpbf-1,y
D0F5 95 E0              sta     varb,x
D0F7 88                 dey
D0F8 CA                 dex
D0F9 10 F7              bpl     rz1
D0FB 30 1B              bmi     sz2

D0FD 08         svzp1   php                     save zero page
D0FE 78                 sei
D0FF 48                 pha
D100 8A                 txa
D101 48                 pha
D102 98                 tya
D103 48                 pha
D104 A2 00              ldx     #0              copy zeropage mem.
D106 AC B2AB            ldy     zbsp
D109 C0 20              cpy     #zbmax          check max
D10B B0 0B              bcs     sz2             full
D10D B5 E0      sz1     lda     varb,x
D10F 99 90AB            sta     zpbf,y
D112 C8                 iny
D113 E8                 inx
D114 E0 10              cpx     #$10
D116 D0 F5              bne     sz1
D118 8C B2AB    sz2     sty     zbsp            save stackpointer
D11B 68                 pla
D11C A8                 tay
D11D 68                 pla
D11E AA                 tax
D11F 68                 pla
D120 28                 plp
D121 60                 rts

                ;*************************************************************************
                ;               filespec to filenamebuffer
                ;*************************************************************************
D122 18         name1   clc
D123 84 EA      name3   sty     poin            put filename in filenamebuffer
D125 85 EB              sta     poin+1
D127 A0 00              ldy     #0
D129 8C EFAB            sty     opfnam          reset name flags
D12C 88                 dey
D12D C8         nam13   iny
D12E 30 18              bmi     63.f            overflow
D130 B1 EA              lda     [poin],y        skip spaces
D132 49 20              eor     #$20
D134 F0 F7              beq     nam13
D136 90 06              bcc     nam14
D138 49 0C              eor     #',^' '         ',' file on same device
D13A D0 02              bne     nam14
D13C F0 53              beq     dse14           set selected
D13E 88         nam14   dey
D13F AD D0AB            lda     csuf1           test system drive
D142 4A                 lsra
D143 A9 00              lda     #0
D145 AA                 tax
D146 10 52              bpl     dse10           select drive
D148 4C 10D2    63      jmp     nam63

D14B CA         dse     dex                     one char drive
D14C D0 24              bne     dverr
D14E A9 C0              lda     #$c0
D150 2C EEAB            bit     fdrive          specified
D153 D0 1D              bne     dverr
D155 AD E0AB            lda     namb            check device spec.
D158 20 41C0            jsr     loupch
```

```
D15B C9 53              cmp     #'S             system
D15D F0 39              beq     dse13
D15F E8                 inx
D160 C9 55              cmp     #'U             user
D162 F0 34              beq     dse13
D164 E8                 inx
D165 C9 57              cmp     #'W'            work
D167 F0 2F              beq     dse13
D169 49 30              eor     #'0'            check drive select
D16B CD 00CC            cmp     drives
D16E 09 80              ora     #$80
D170 90 2E              bcc     dse12           ok, set drive
D172 A9 11      dverr   lda     #e11
D174 38                 sec
D175 60                 rts

D176 CA         disel   dex                     one char drive
D177 D0 F9              bne     dverr
D179 2C EEAB            bit     fdrive          specified
D17C 70 F4              bvs     dverr
D17E AD E0AB            lda     namb            check device spec.
D181 20 41C0            jsr     loupch
D184 49 40              eor     #'@'            check directory select
D186 C9 08              cmp     #8
D188 B0 E8              bcs     dverr
D18A 0A                 asla
D18B 0A                 asla
D18C 4D EEAB            eor     fdrive          set directory
D18F 29 1C              and     #$1c
D191 4D EEAB    dse14   eor     fdrive
D194 09 40              ora     #$40
D196 D0 08              bne     dse12
D198 A9 80      dse13   lda     #$80
D19A B0 01      dse10   bcs     1.
D19C E8                 inx
D19D 1D D1AB    1       ora     sdrive,x        drive,x
D1A0 8D EEAB    dse12   sta     fdrive
D1A3 A2 0E              ldx     #14             clear name buffer
D1A5 A9 00              lda     #0
D1A7 9D DFAB    nam3    sta     namb-1,x
D1AA CA                 dex
D1AB D0 FA              bne     nam3
D1AD 86 E8      nam4    stx     rwpoin
D1AF A9 00              lda     #0
D1B1 A6 E8      nam31   ldx     rwpoin
D1B3 0D EFAB            ora     opfnam          set filename flag
D1B6 8D EFAB            sta     opfnam
D1B9 2C EFAB            bit     opfnam
D1BC 30 29              bmi     nam30           no special chars
D1BE C8                 iny
D1BF B1 EA              lda     [poin],y
D1C1 30 4D              bmi     nam63           no neg chars
D1C3 C9 3A              cmp     #':             drive select
D1C5 F0 84              beq     dse
D1C7 C9 2F              cmp     #'/             directory select
D1C9 F0 AB              beq     disel
D1CB C9 5C              cmp     #'\\            escape
D1CD F0 18              beq     nam30
D1CF AA                 tax
D1D0 A9 80              lda     #$80
D1D2 E0 25              cpx     #'%             no wildcards
D1D4 F0 DB              beq     nam31
D1D6 4A                 lsra
D1D7 E0 5E              cpx     #'^             make uppercase
D1D9 F0 D6              beq     nam31
D1DB 4A                 lsra
D1DC E0 7E              cpx     #'~             not file
D1DE F0 D1              beq     nam31
D1E0 A9 01              lda     #1
D1E2 E0 24              cpx     #'$             set negative name
D1E4 F0 CB              beq     nam31
D1E6 88                 dey
D1E7 C8         nam30   iny
D1E8 A6 E8              ldx     rwpoin
D1EA B1 EA              lda     [poin],y
D1EC F0 26              beq     nam40
D1EE C9 0D              cmp     #$d
D1F0 F0 22              beq     nam40
D1F2 C9 20              cmp     #' '
D1F4 F0 1E              beq     nam40
D1F6 90 18              bcc     nam63           lower than space
D1F8 C9 2C              cmp     #',
D1FA F0 18              beq     nam40
D1FC E0 0E              cpx     #14             check max 14 char
```

```
D1FE B0 10              bcs     nam63
D200 0A                 asla
D201 B0 06              bcs     nam32
D203 48                 pha
D204 AD EFAB            lda     opfnam          set bit 7
D207 4A                 lsra
D208 68                 pla
D209 6A         nam32   rora
D20A 9D E0AB            sta     namb,x          save char in namebuf
D20D E8                 inx
D20E D0 9D              bne     nam4
D210 A9 10      nam63   lda     #e10            error filename
D212 38                 sec
D213 60                 rts

D214 AD E0AB    nam40   lda     namb            1st char
D217 30 F7              bmi     nam63
D219 18                 clc                     save pos. file
D21A 98                 tya
D21B 65 EA              adc     poin
D21D 8D D4AB            sta     ypos
D220 A9 00              lda     #0
D222 65 EB              adc     poin+1
D224 8D D5AB            sta     apos
D227 AD EEAB    nam41   lda     fdrive          get selected drive
D22A 60         9       rts

D22B 20 22D1    namdef  jsr     name1
D22E B0 FA              bcs     9.
D230 8A                 txa
D231 D0 F4              bne     nam41
D233 A9 2A              lda     #'*'            default '*'
D235 8D E0AB            sta     namb
D238 D0 ED              bne     nam41

D23A AC D4AB    setnex1 ldy     ypos            next filespec address
D23D AD D5AB            lda     apos
D240 60                 rts

                ;---------------------------------------------------------------
                ;               set file pointers
                ;---------------------------------------------------------------
D241 8E B3AB    setpoin stx     fd
D244 8A                 txa                     check fd and set pointers
D245 C9 0A              cmp     #max_fd+4       4 <= fd < 4+max_fd
D247 B0 16              bcs     sep1            error fd
D249 E9 03              sbc     #3
D24B 90 12              bcc     sep1            error fd
D24D 0A                 asla                    calc. pos. (fd-4)*24
D24E 0A                 asla
D24F 0A                 asla
D250 85 E0              sta     varb
D252 0A                 asla
D253 65 E0              adc     varb
D255 A8                 tay
D256 B9 00AB            lda     varo,y
D259 D0 34              bne     sep2
D25B A9 30              lda     #e30            not open
D25D 38                 sec
D25E 60                 rts
D25F A9 31      sep1    lda     #e31            error fd
D261 38                 sec
D262 60                 rts

                ;---------------------------------------------------------------
                ;               search free file control block
                ;---------------------------------------------------------------
D263 A2 04      oppoin  ldx     #4
D265 A0 00              ldy     #0
D267 B9 00AB    opp1    lda     varo,y          check open fd.
D26A F0 09              beq     opp2            not in use
D26C E8                 inx
D26D 20 E0D2            jsr     nextv
D270 90 F5              bcc     opp1            max fd
D272 A9 20              lda     #e20            not available
D274 60                 rts
D275 AD EEAB    opp2    lda     fdrive
D278 29 1F              and     #$1f
D27A 99 0DAB            sta     varo+f.drdi,y   drive directory
D27D 29 1C              and     #$1c
D27F 4A                 lsra
D280 F0 02              beq     1.
D282 69 30              adc     #s.dir-2
D284 8D B5AB    1       sta     direct
```

```
D287 A9 00            lda     #0
D289 99 06AB          sta     varo+inpc,y
D28C 8E B3AB          stx     fd
D28F 84 E0      sep2  sty     varb            set var. varb (fd)
D291 A9 AB            lda     #varbst>>8
D293 85 E1            sta     varb+1
D295 8A               txa
D296 69 AD            adc     #(tslbst>>8)-4  set tslb (fd)
D298 85 E5            sta     tslb+1
D29A 69 06            adc     #max_fd         set inpb after tslb
D29C 85 E7            sta     inpb+1
D29E B9 06AB          lda     varo+inpc,y     set pos in data blok
D2A1 85 E6            sta     inpb
D2A3 B9 0DAB          lda     varo+f.drdi,y   set drive
D2A6 29 03            and     #3
D2A8 8D B4AB          sta     drive
D2AB 69 AD            adc     #sysbst>>8      set sysb (drive)
D2AD 85 E3            sta     sysb+1
D2AF A9 00            lda     #0
D2B1 85 E4            sta     tslb
D2B3 85 E2            sta     sysb
D2B5 60               rts

                ;---------------------------------------------------------------------
                ;               check if file is already open
                ;---------------------------------------------------------------------
D2B6 A0 00      chopen  ldy     #0
D2B8 A6 E0              ldx     varb
D2BA B9 00AB    cho1    lda     varo,y
D2BD F0 18              beq     cho3
D2BF B9 0DAB            lda     varo+f.drdi,y   check drive - directory
D2C2 DD 0DAB            cmp     varo+f.drdi,x
D2C5 D0 10              bne     cho3
D2C7 B9 0EAB            lda     varo+dirc,y     check dir sector
D2CA DD 0EAB            cmp     varo+dirc,x
D2CD D0 08              bne     cho3
D2CF B9 0FAB            lda     varo+dirn,y     check pos in dir sector
D2D2 DD 0FAB            cmp     varo+dirn,x
D2D5 F0 06              beq     cho4
D2D7 20 E0D2    cho3    jsr     nextv
D2DA 90 DE              bcc     cho1
D2DC 18                 clc
D2DD A9 21      cho4    lda     #e21            file open
D2DF 60                 rts                     C=0 ok

D2E0 18         nextv   clc                     next fcb
D2E1 98                 tya
D2E2 69 18              adc     #f.size
D2E4 A8                 tay
D2E5 C0 90              cpy     #max_fd*f.size
D2E7 60                 rts

                ;---------------------------------------------------------------------
                ;               close file control block
                ;---------------------------------------------------------------------
D2E8 48         clpoin  pha                     set closed fd
D2E9 A0 00              ldy     #0
D2EB 98                 tya
D2EC 91 E0              sta     [varb],y
D2EE 68                 pla
D2EF 60                 rts                     saved C and Accu

                ;*********************************************************************
                ;               compare filespec with current directory entry
                ;*********************************************************************
D2F0 84 EE      namcom  sty     dirpp           current position
D2F2 85 EF              sta     dirpp+1
D2F4 A0 FF              ldy     #$ff            compare dirfile with filenamebuff.
D2F6 A2 FF              ldx     #$ff            dir. pos (DIRPP)
D2F8 86 E8              stx     rwpoin          set * fl
D2FA E8         nc1     inx                     next
D2FB C8         nc10    iny
D2FC A9 00              lda     #0              set 0 for end namebuf
D2FE E0 0E              cpx     #14             check end namebuff
D300 B0 03              bcs     nc11
D302 BD E0AB            lda     namb,x
D305 C0 0E      nc11    cpy     #14             check dir end
D307 B0 3C              bcs     nc4             check A
D309 2C EFAB            bit     opfnam          no wildcards
D30C 30 2A              bmi     nc3
D30E C9 3F              cmp     #'?             single char n = 1
D310 D0 06              bne     nc20            comp next
D312 B1 EE              lda     [dirpp],y
D314 D0 E4              bne     nc1             ok
```

```
D316 F0 30              beq     nc7             not, must be a char

D318 C9 2A      nc20    cmp     #'*             match string n >= 0
D31A D0 1C              bne     nc3
D31C 86 E8              stx     rwpoin          save X
D31E 84 E9              sty     rwpoin+1
D320 E8                 inx                     next char in namebuf
D321 A9 00              lda     #0
D323 E0 0E              cpx     #14             end
D325 F0 23              beq     nc8
D327 BD E0AB    nc22    lda     namb,x          char after *
D32A F0 1E              beq     nc8             no => compare ok
D32C 20 53D3            jsr     chkfn           comp dir sector
D32F 90 C9              bcc     nc1
D331 C8                 iny
D332 C0 0E              cpy     #14             max
D334 90 F1              bcc     nc22
D336 B0 10              bcs     nc7

D338 20 53D3    nc3     jsr     chkfn
D33B 90 BD              bcc     nc1
D33D A4 E9              ldy     rwpoin+1
D33F A6 E8              ldx     rwpoin          test *
D341 10 B8              bpl     nc10            next pattern search
D343 30 03              bmi     nc7

D345 AA         nc4     tax                     check accu
D346 F0 02              beq     nc8
D348 A9 20      nc7     lda     #$20
D34A A6 EE      nc8     ldx     dirpp           restore X
D34C 4D EFAB            eor     opfnam          ^ opt: test not
D34F 29 20              and     #$20
D351 18                 clc
D352 60                 rts                     Z=1 if ok

D353 38         chkfn   sec
D354 29 7F              and     #$7f
D356 D0 0B              bne     chkfn1          not NUL
D358 B1 EE              lda     (dirpp),y       check neg fname
D35A 10 2C              bpl     chkfn3
D35C 2C D0AB            bit     csuf1           command
D35F 30 29              bmi     chkfn8          ok
D361 10 28              bpl     chkfn9
D363 2C EFAB    chkfn1  bit     opfnam          ^ opt: convert lo to up
D366 70 0C              bvs     chkfn5
D368 2C 12C0            bit     dosmode
D36B 30 07              bmi     chkfn5          case inhibit
D36D 50 09              bvc     chkfn6          on command
D36F 2C D0AB            bit     csuf1           command
D372 10 04              bpl     chkfn6
D374 20 41C0    chkfn5  jsr     loupch          convert lo-up
D377 18                 clc
D378 8D C0AB    chkfn6  sta     blss
D37B B1 EE              lda     (dirpp),y       get dirfn
D37D 29 7F              and     #$7f
D37F B0 04              bcs     chkfn7
D381 20 41C0            jsr     loupch
D384 38                 sec
D385 4D C0AB    chkfn7  eor     blss            compare
D388 D0 01      chkfn3  bne     chkfn9
D38A 18         chkfn8  clc
D38B 60         chkfn9  rts                     C=0 if equal

                ;*********************************************************************
                ;               function create file
                ;*********************************************************************
D38C 8E BAAB    create2 stx     modf+1          save mode for create
D38F 48                 pha
D390 8A                 txa
D391 29 0F              and     #$0f
D393 09 40              ora     #$40
D395 AA                 tax
D396 68                 pla
D397 20 39D4            jsr     open2           open for at least write
D39A 90 05              bcc     cr99            ok
D39C C9 13              cmp     #e13            not found?
D39E F0 02              beq     cr1
D3A0 38                 sec
D3A1 60         cr99    rts

D3A2 20 63D2    cr1     jsr     oppoin          open again
D3A5 20 15DA    cr20    jsr     firdir          search deleted file
D3A8 20 1EDA    cr21    jsr     nexdir
D3AB B0 F4              bcs     cr99
```

```
D3AD 30 42              bmi     cr3             found
D3AF D0 F7              bne     cr21
D3B1 C0 F0              cpy     #$f0            end of directory sector
D3B3 90 3C              bcc     cr3
D3B5 20 92DC            jsr     dirsrc          search new dir sector
D3B8 B0 E7              bcs     cr99            disk full
D3BA AD D9AB    cr16    lda     dirq            1st dir sector
D3BD D0 0D              bne     cr17
D3BF 98                 tya                     create new directory
D3C0 AC B5AB            ldy     direct
D3C3 C8                 iny
D3C4 91 E2              sta     [sysb],y
D3C6 88                 dey
D3C7 8A                 txa
D3C8 91 E2              sta     [sysb],y
D3CA 90 0B              bcc     cr18
D3CC 8E F0AC    cr17    stx     dirb+$f0        save in last dir sector
D3CF 8C F1AC            sty     dirb+$f1
D3D2 20 90DA            jsr     wrtdir          save dir sector
D3D5 B0 CA              bcs     cr99
D3D7 20 4DDA    cr18    jsr     reddir          get new dir sector
D3DA B0 C5              bcs     cr99
D3DC A9 00              lda     #0              clear dir sector
D3DE A8                 tay
D3DF 99 00AC    cr15    sta     dirb,y
D3E2 C8                 iny
D3E3 D0 FA              bne     cr15
D3E5 20 02DD            jsr     nexsec          last directory sector
D3E8 90 D0              bcc     cr16
D3EA 20 90DA            jsr     wrtdir          save last dir sector
D3ED 90 B6              bcc     cr20            start again
D3EF B0 B0              bcs     cr99            error

D3F1 84 EE      cr3     sty     dirpp           set in temp
D3F3 20 7EDC            jsr     tslsrc          search new tsl sector
D3F6 B0 A9      cr999   bcs     cr99            disk full
D3F8 98                 tya                     save track and sector
D3F9 A0 05              ldy     #tslsk
D3FB 91 E0              sta     [varb],y
D3FD 48                 pha
D3FE 8A                 txa
D3FF 88                 dey
D400 91 E0              sta     [varb],y
D402 A6 EE              ldx     dirpp           and in directory
D404 9D 0EAC            sta     dirb+$e,x
D407 68                 pla
D408 9D 0FAC            sta     dirb+$f,x
D40B 20 02DB            jsr     cltsl           clear tsl sector
D40E A0 04              ldy     #flmo           set filemode
D410 AD BAAB            lda     modf+1
D413 91 E4              sta     [tslb],y
D415 A0 06              ldy     #dblk           put dir, pos and sectorno in tsl
D417 AE D9AB            ldx     dirq
D41A AD B5AB            lda     direct
D41D 29 0F              and     #$0f
D41F 4A                 lsra
D420 05 EE              ora     dirpp
D422 20 15DB            jsr     stotsl1
D425 A0 18              ldy     #credat         set time as create
D427 A5 E5              lda     tslb+1
D429 20 95D6            jsr     movdat
D42C 20 F3DB            jsr     wrttsl          save tsl for sure
D42F B0 C5              bcs     cr999
D431 20 F6D5            jsr     ren2            set name and save directory
D434 B0 C0              bcs     cr999           set init
D436 4C B8D4            jmp     op61            set varbuffer

                ;*********************************************************************
                ;               function open file
                ;*********************************************************************
D439 8E B9AB    open2   stx     modf            save mode
D43C C9 F8              cmp     #$f8            test param A
D43E 90 18              bcc     open3           normal open
D440 F0 1B              beq     opn1            A=F8 filename defined
D442 C9 FA              cmp     #$fa
D444 F0 06              beq     nexsnopn1       A=FA next open (new filename)

D446 20 63D2    nexopn1 jsr     oppoin          A=F9 next open (same filename)
D449 4C 67D4            jmp     opn2

D44C 20 46D4    nexsnopn1 jsr   nexopn1         open next
D44F 90 7D              bcc     op99
D451 C9 13              cmp     #e13            not found
D453 D0 6E              bne     op90
```

```
D455 20 3AD2            jsr     setnex1         next filename

D458 20 23D1    open3   jsr     name3           def name
D45B B0 71              bcs     op99
D45D A9 12      opn1    lda     #e12            check filename
D45F AE E0AB            ldx     namb            check first char
D462 F0 5F              beq     op90
D464 20 09DA            jsr     inidir          open fd sis and dir
D467 B0 65      opn2    bcs     op99            not available
D469 20 1EDA    1       jsr     nexdir          get next directory pos.
D46C B0 60              bcs     op99
D46E F0 50              beq     opnf            end
D470 30 F7              bmi     1.b             deleted file
D472 20 F0D2            jsr     namcom          compare given filename
D475 D0 F2              bne     1.b             not equal => next file
D477 AC B9AB            ldy     modf            open for only dir. or check
D47A C0 02              cpy     #2
D47C 90 08              bcc     2.f
D47E 20 CEDB            jsr     redtsl1         read first tsl
D481 B0 40              bcs     op90
D483 AC B9AB            ldy     modf
D486 A9 00      2       lda     #0
D488 C0 03              cpy     #3              test get only tsl
D48A 90 37              bcc     op90
D48C 20 9FDA            jsr     invar
D48F 20 B6D2            jsr     chopen          if file already opened
D492 B0 2F              bcs     op90
D494 20 CFD4            jsr     tstmod
D497 B0 29              bcs     op92
D499 8A                 txa                     check mode write
D49A 29 40              and     #$40
D49C F0 26              beq     op95            not write
D49E 20 89D6            jsr     del             delete filecontents w w+
D4A1 B0 20              bcs     op90            err
D4A3 20 CEDB            jsr     redtsl1         read first tsl
D4A6 B0 1B              bcs     op90            err
D4A8 A9 00              lda     #0
D4AA A0 14              ldy     #fl1n
D4AC A2 03              ldx     #3
D4AE 20 06DB            jsr     mtsla           clear char count
D4B1 A0 20              ldy     #$20
D4B3 A2 E4              ldx     #$e4
D4B5 20 06DB            jsr     mtsla           clear tsa overlap till 04 (links)
D4B8 20 9FDA    op61    jsr     invar           init var. in fcb
D4BB 20 E2D9            jsr     filwset
D4BE 90 04              bcc     op95            get pos in directory and fd
                ;
D4C0 A0 13      opnf    ldy     #e13            not found
D4C2 98         op92    tya
D4C3 38         op90    sec
D4C4 AE B3AB    op95    ldx     fd              get fd
D4C7 B0 05              bcs     op99

D4C9 8A         mopoin  txa                     mark open file
D4CA A0 00              ldy     #0
D4CC 91 E0              sta     [varbl],y
D4CE 60         op99    rts

                ;---------------------------------------------------------------
                ;               compare filemode with open mode
                ;---------------------------------------------------------------
D4CF AD B9AB    tstmod  lda     modf            get mode to open
D4D2 AA                 tax                     save in X
D4D3 29 E0              and     #$e0
D4D5 F0 F7              beq     op99            opened for check (1-)
D4D7 48                 pha                     set test on stack
D4D8 A0 04              ldy     #flmo
D4DA 31 E4              and     [tslbl],y       test mode file r w d
D4DC F0 31              beq     op8             not ok => print prot.
D4DE 68                 pla                     adjust
D4DF A9 10              lda     #$10
D4E1 2C B9AB            bit     modf
D4E4 F0 0B              beq     op4             no +
D4E6 A9 80              lda     #$80            test r+ w+
D4E8 50 01              bvc     1.              read +
D4EA 4A                 lsra
D4EB 48         1       pha                     set test on stack
D4EC 31 E4              and     [tslbl],y       test r w
D4EE F0 1F              beq     op8             protected
D4F0 68                 pla                     adjust
D4F1 8A         op4     txa
D4F2 29 0C              and     #$0c            test bit 2,3
D4F4 F0 06              beq     op41            not used
D4F6 51 E4              eor     [tslbl],y       match mode file
```

```
D4F8 29 0C              and     #$0c
D4FA D0 0F              bne     op83            not allowed
D4FC 8A         op41    txa
D4FD 29 03              and     #$03            test bit 0,1
D4FF F0 CD              beq     op99
D501 51 E4              eor     [tslb],y
D503 29 03              and     #$03
D505 F0 C7              beq     op99
D507 A0 29      op81    ldy     #e29            permission
D509 D0 0F              bne     op82
D50B A0 28      op83    ldy     #e28
D50D D0 0B              bne     op82
D50F 68         op8     pla                     get test pattern
D510 A0 27              ldy     #e27
D512 0A                 asla
D513 B0 05              bcs     op82            not read
D515 88                 dey
D516 0A                 asla
D517 B0 01              bcs     op82            not write
D519 88                 dey                     delete
D51A 38         op82    sec
D51B 60                 rts

                ;*********************************************************************
                ;               function delete file
                ;*********************************************************************
D51C 20 39D4    delete2 jsr     open2           open file for delete
D51F B0 36              bcs     90.f
D521 AD B9AB            lda     modf
D524 29 20              and     #$20            wrong open
D526 D0 04              bne     2.f
D528 A9 35              lda     #e35
D52A 38                 sec
D52B 60                 rts
D52C AE D8AB    2       ldx     dirp
D52F BD 00AC            lda     dirb,x          set bit 8 in name file
D532 09 80              ora     #$80
D534 9D 00AC            sta     dirb,x
D537 BC 0FAC            ldy     dirb+$f,x       delete first tsl
D53A BD 0EAC            lda     dirb+$e,x
D53D AA                 tax
D53E 20 44DD            jsr     blkfre          first tsl ts
D541 20 90DA            jsr     wrtdir          save dir
D544 B0 0B              bcs     91.
D546 20 B9D6            jsr     del             free sectors, not 1st tsl
D549 B0 06              bcs     91.             err
D54B 20 91D6    del0    jsr     movsyda         copy date changed disk
D54E 20 7EDA            jsr     wrtsys          save SIS
D551 20 E8D2    91      jsr     clpoin          close filedescr
D554 AE B3AB    gfd     ldx     fd              get filedescriptor
D557 60         90      rts

                ;*********************************************************************
                ;               function close file
                ;*********************************************************************
D558 20 A1D8    close2  jsr     movpol          setup var
D55B B0 FA              bcs     90.b            fd
D55D A0 13              ldy     #filf
D55F B1 E0              lda     [varb],y        test file changed
D561 F0 EE              beq     91.b            no
D563 85 EE              sta     temp
D565 20 65DB    1       jsr     bchwtsl         get first tsl (check last data)
D568 B0 E7              bcs     91.b
D56A D0 F9              bne     1.              not at begin
D56C A5 EE              lda     temp
D56E 10 19              bpl     12.             set mode
D570 A0 09              ldy     #chrc           store char count
D572 A2 14              ldx     #flln
D574 20 E3DA            jsr     movtsl          move char count to tsl
D577 20 E0DA            jsr     movtsl2
D57A A0 17              ldy     #vers           next version file
D57C B1 E4              lda     [tslb],y
D57E 69 01              adc     #1
D580 91 E4              sta     [tslb],y
D582 A0 1C              ldy     #moddat         set modified date
D584 A5 E5              lda     tslb+1
D586 20 95D6            jsr     movdat
D589 A0 01      12      ldy     #fmod
D58B A2 04              ldx     #flmo
D58D 20 E3DA            jsr     movtsl          move filemode
D590 20 F3DB            jsr     wrttsl          save first tsl
D593 B0 BC              bcs     91.b
D595 20 4BD5            jsr     del0            update SIS and close fd
D598 B0 BA              bcs     gfd
```

```
D59A A5 EE              lda     temp
D59C 10 B6              bpl     gfd             only modechange
D59E AD 12C0            lda     dosmode         mode check sectors
D5A1 29 01              and     #$01
D5A3 F0 AF              beq     gfd             no
D5A5 20 FFDB            jsr     fslts
D5A8 20 E3DB            jsr     rdtsl
D5AB B0 A7              bcs     gfd
D5AD 20 12D9    cl11    jsr     nexblk          check sectors
D5B0 B0 A2              bcs     gfd
D5B2 D0 F9              bne     cl11
D5B4 90 9E              bcc     gfd

                ;*********************************************************************
                ;               function rename file
                ;*********************************************************************
D5B6 A2 00      rename2 ldx     #0              only dir
D5B8 20 39D4            jsr     open2           open
D5BB AA                 tax
D5BC D0 4C              bne     ren9            error
D5BE AD EEAB            lda     fdrive
D5C1 8D BEAB            sta     sekf
D5C4 A5 EE              lda     dirpp           position
D5C6 85 EC              sta     mpoin
D5C8 AD D9AB            lda     dirq
D5CB 85 ED              sta     mpoin+1
D5CD 20 3AD2            jsr     setnex1         get pos. file2
D5D0 A2 00              ldx     #0
D5D2 20 39D4            jsr     open2           open for check
D5D5 AA                 tax
D5D6 F0 2F              beq     ren7            error : existing file2
D5D8 C9 13              cmp     #e13            not found?
D5DA D0 2D              bne     ren99
D5DC A9 42              lda     #e42
D5DE AC EEAB            ldy     fdrive          check drives
D5E1 CC BEAB            cpy     sekf
D5E4 D0 23              bne     ren99           not equal
D5E6 20 15DA            jsr     firdir
D5E9 20 4DDA    2       jsr     reddir
D5EC B0 1C              bcs     ren9
D5EE C6 ED              dec     mpoin+1
D5F0 D0 F7              bne     2.b
D5F2 A5 EC              lda     mpoin
D5F4 85 EE              sta     dirpp
D5F6 A9 AC      ren2    lda     #dirb>>8        copy filename
D5F8 85 EF              sta     dirpp+1
D5FA A0 0D              ldy     #13
D5FC B9 E0AB    ren1    lda     namb,y
D5FF 91 EE              sta     [dirpp],y
D601 88                 dey
D602 10 F8              bpl     ren1
D604 4C 90DA            jmp     wrtdir          save changed directory

D607 A9 41      ren7    lda     #e41            error rename
D609 38         ren99   sec
D60A 60         ren9    rts

                ;*********************************************************************
                ;               function status file
                ;*********************************************************************
D60B 84 EC      status2 sty     mpoin           save info
D60D 85 ED              sta     mpoin+1         and mode
D60F 20 A1D8            jsr     movpo1          test and set open
D612 B0 35              bcs     9.
D614 A5 EC              lda     mpoin           status begin/end file
D616 D0 19              bne     10.f
D618 20 B5D8            jsr     mpv1            test end
D61B D0 09              bne     1.
D61D A5 E6              lda     inpb
D61F 51 E0              eor     [varb],y
D621 D0 03              bne     1.
D623 A9 04              lda     #e4
D625 60                 rts
D626 20 97DB    1       jsr     chbeg           check begin
D629 D0 03              bne     2.
D62B A9 03              lda     #e3
D62D 60                 rts
D62E A9 00      2       lda     #0
D630 60                 rts

D631 A0 01      10      ldy     #fmod           get filemode
D633 B1 E0              lda     [varb],y
D635 C6 EC              dec     mpoin
D637 F0 10              beq     9.
```

```
D639 C6 EC              dec     mpoin
D63B D0 0D              bne     10.f
D63D A5 ED              lda     mpoin+1
D63F 91 E0              sta     [varb],y
D641 A0 13              ldy     #filf           set fileflag
D643 B1 E0              lda     [varb],y
D645 09 40              ora     #$40
D647 91 E0              sta     [varb],y
D649 60         9       rts
D64A A4 E0      10      ldy     varb            get filecontrolblock
D64C A5 E1              lda     varb+1
D64E A6 EC              ldx     mpoin
D650 CA                 dex
D651 F0 20              beq     8.f
D653 A5 E3              lda     sysb+1          get sis
D655 CA                 dex
D656 F0 19              beq     7.f
D658 A0 0F              ldy     #dirn           directory sector
D65A B1 E0              lda     [varb],y
D65C A8                 tay
D65D A9 AC              lda     #dirb>>8
D65F CA                 dex
D660 F0 11              beq     8.f
D662 A5 E5              lda     tslb+1          tsl sector
D664 CA                 dex
D665 F0 0A              beq     7.f
D667 A4 E6              ldy     inpb            data sector
D669 A5 E7              lda     inpb+1
D66B CA                 dex
D66C F0 05              beq     8.f
D66E A9 46              lda     #e46            no status function
D670 38                 sec
D671 A0 00      7       ldy     #0
D673 4C 54D5    8       jmp     gfd

                ;---------------------------------------------------------------------
                ;               free file sectors
                ;---------------------------------------------------------------------
D676 A0 03      de2     ldy     #tsln           get next data tsa
D678 B1 E0              lda     [varb],y
D67A C9 FE              cmp     #$fe            end tsl sector
D67C D0 07              bne     de4
D67E A0 00              ldy     #0              free sector of next tsl
D680 20 3BDD            jsr     tslfre
D683 A9 FE      de3     lda     #$fe
D685 A8         de4     tay
D686 20 3BDD            jsr     tslfre          free sector
D689 20 33DB    de1     jsr     nextsl          entry
D68C B0 02              bcs     de9
D68E D0 E6              bne     de2             not at end
D690 60         de9     rts

                ;---------------------------------------------------------------------
                ;               copy time and date info
                ;---------------------------------------------------------------------
D691 A0 5C      movsyda ldy     #s.mdat         system modified date
D693 A5 E3              lda     sysb+1
D695 84 E8      movdat  sty     rwpoin
D697 85 E9              sta     rwpoin+1
D699 A0 03              ldy     #3
D69B B9 FCAB    1       lda     date,y
D69E 91 E8              sta     [rwpoin],y
D6A0 88                 dey
D6A1 10 F8              bpl     1.b
D6A3 60                 rts

                ;*********************************************************************
                ;               function single write file
                ;*********************************************************************
D6A4 8D DBAB    swrite2 sta     asrw
D6A7 20 CAD6            jsr     setrwp          set
D6AA 20 E5D6            jsr     write2
D6AD 4C B6D6            jmp     wts             check

                ;*********************************************************************
                ;               function single read file
                ;*********************************************************************
D6B0 20 CAD6    sread2  jsr     setrwp          set pointer
D6B3 20 4DD7            jsr     read2
D6B6 B0 03      wts     bcs     1.              error
D6B8 AD DBAB            lda     asrw
D6BB 60         1       rts

                ;*********************************************************************
```

```
                        ;               function function seek file
                        ;*******************************************************************
D6BC A8         fseek2  tay                     seek fnc
D6BD 4A                 lsra
D6BE 4A                 lsra                    from current position (2,3)
D6BF A9 00              lda     #0
D6C1 20 CFD6            jsr     strw1
D6C4 20 99D7            jsr     seek2
D6C7 4C B6D6            jmp     wts

                        ;-------------------------------------------------------------------
                        ;               define r/w/s control block for single r/w/s
                        ;-------------------------------------------------------------------
D6CA A0 DB      setrwp  ldy     #asrw&255       point mem
D6CC A9 AB              lda     #asrw>>8
D6CE 38                 sec
D6CF 8C DCAB    strw1   sty     psrw            set address
D6D2 8D DDAB            sta     psrw+1
D6D5 A0 00              ldy     #0
D6D7 8C DFAB            sty     psrw+3
D6DA 90 01              bcc     1.              seek begin end
D6DC CB                 iny
D6DD 8C DEAB    1       sty     psrw+2
D6E0 A0 DC              ldy     #psrw&255       setup A,Y
D6E2 A9 AB              lda     #psrw>>8
D6E4 60                 rts

                        ;*******************************************************************
                        ;               function write file
                        ;*******************************************************************
D6E5 20 6CD8    write2  jsr     movpova         setup variables, check open
D6E8 B0 51              bcs     wr3             file not opened
D6EA A9 10              lda     #$10            test opened w or r+
D6EC 2C BAAB            bit     modf+1
D6EF 70 08              bvs     5.              w
D6F1 10 02              bpl     1.              not r
D6F3 D0 04              bne     5.              r+
D6F5 A9 36      1       lda     #e36
D6F7 D0 51              bne     wr31            set char written

D6F9 20 E2D9    5       jsr     filwset
D6FC 8D C1AB            sta     wralc           allocate flag
D6FF 20 53D8            jsr     endchk          no bytes
D702 F0 27              beq     wr4
D704 2C B9AB            bit     modf            sector allocated
D707 70 05              bvs     13.
D709 20 3AD9    10      jsr     wrss            search next sector
D70C B0 2D              bcs     wr3
D70E A9 C0      13      lda     #$c0
D710 8D B9AB            sta     modf            sector read and changed
D713 A0 00              ldy     #0
D715 B1 EE      20      lda     [temp],y
D717 91 E6              sta     [inpb],y
D719 E6 E6              inc     inpb
D71B F0 07              beq     30.             buffer full
D71D 20 3ED8            jsr     incchk          check end
D720 D0 F3              bne     20.             not at end bytes to write
D722 F0 07              beq     wr4
D724 20 29D8    30      jsr     chwren1
D727 B0 12              bcs     wr3
D729 D0 DE              bne     10.b
D72B 20 1DD8    wr4     jsr     chki            last sector position
D72E D0 0A              bne     wr41            not last sector
D730 A5 E6              lda     inpb
D732 DD 09AB            cmp     varo+chrc,x
D735 90 04              bcc     wr3
D737 9D 09AB            sta     varo+chrc,x     new length
D73A 18         wr41    clc
D73B 20 D0D8    wr3     jsr     movvapo         save vars to filecontrolblock
D73E 90 05              bcc     90.             no error
D740 10 03              bpl     90.             no r/w err
D742 20 EBD2            jsr     clpoin          close file
D745 4C 54D5    90      jmp     gfd             get fd

D748 A9 04      rd4     lda     #e4             file empty flag
D74A 38         wr31    sec
D74B B0 EE              bcs     wr3

                        ;*******************************************************************
                        ;               function read file
                        ;*******************************************************************
D74D 20 6CD8    read2   jsr     movpova         setup variables, check open
D750 B0 E9              bcs     wr3
D752 A9 10              lda     #$10
```

```
D754 2C BAAB            bit     modf+1
D757 30 0B              bmi     rd5             r
D759 50 02              bvc     rder            not w
D75B D0 04              bne     rd5             w+
D75D A9 37      rder    lda     #e37            flag not allowed
D75F D0 E9              bne     wr31
                ;
D761 20 53D8    rd5     jsr     endchk          check end
D764 F0 30              beq     rd9
D766 2C B9AB            bit     modf
D769 70 0A              bvs     13.             buffer not empty
D76B 20 08D9    10      jsr     cnblk           get next data sector
D76E B0 26              bcs     rd9             err
D770 F0 D6              beq     rd4             end of file
D772 20 B5D8            jsr     mpv1            check last sector => set CHRN
D775 A0 00      13      ldy     #0
D777 AD BFAB    20      lda     chrn            end? (0 is no end; no read)
D77A F0 04              beq     24.
D77C 45 E6              eor     inpb            end?
D77E F0 C8              beq     rd4
D780 B1 E6      24      lda     [inpb],y
D782 91 EE              sta     [temp],y
D784 E6 E6              inc     inpb
D786 F0 07              beq     30.             buffer end
D788 20 3ED8            jsr     incchk          check end
D78B D0 EA              bne     20.b            not at end bytes to write
D78D F0 07              beq     rd9
D78F 20 39D8    30      jsr     chwren          check write and end
D792 B0 02              bcs     rd9
D794 D0 D5              bne     10.b
D796 4C 3BD7    rd9     jmp     wr3

                ;*********************************************************************
                ;               function seek file
                ;*********************************************************************
D799 38         seek2   sec
D79A 20 6DD8            jsr     movpovb         setup variables, check open
D79D B0 29              bcs     se92
D79F A9 38              lda     #e38            mess
D7A1 2C BAAB            bit     modf+1          test r/w
D7A4 30 02              bmi     1.              r
D7A6 50 20              bvc     se92
D7A8 A9 39      1       lda     #e39
D7AA A4 EE              ldy     temp
D7AC 88                 dey
D7AD F0 1D              beq     se10            mode 1 seek from beginning
D7AF C0 03              cpy     #3              mode 1 <= mode <= 4 => ok
D7B1 90 27              bcc     se22            mode 2,3
D7B3 D0 13              bne     se92            error

                ;seek from end
D7B5 20 2EDB    se12    jsr     nchwts1         last ts (write last sector)
D7B8 B0 0F              bcs     se9
D7BA D0 F9              bne     se12            not at end => get next
D7BC A0 09              ldy     #chrc           set end to pointer
D7BE B1 E0              lda     [varb],y
D7C0 90 13              bcc     se21

D7C2 A9 04      se94    lda     #e4             end file
D7C4 D0 02              bne     se92
D7C6 A9 03      se93    lda     #e3             begin file
D7C8 38         se92    sec
D7C9 4C 3BD7    se9     jmp     wr3             result seek

                ;seek from begin
D7CC 20 65DB    se10    jsr     bchwts1         save if ness. last data sector
D7CF B0 F8              bcs     se9
D7D1 D0 F9              bne     se10
D7D3 A9 00              lda     #0
D7D5 85 E6      se21    sta     inpb            data buffer position
D7D7 20 B5D8            jsr     mpv1

                ;seek count
D7DA 20 53D8    se22    jsr     endchk
D7DD F0 EA              beq     se9
D7DF A5 EE      se30    lda     temp            1,2 forward
D7E1 C9 03              cmp     #3              3,4 backward
D7E3 A6 E6              ldx     inpb
D7E5 B0 1F              bcs     se40
D7E7 D0 0A              bne     3.f             buffer not empty
D7E9 20 33DB            jsr     nexts1          get next data sector
D7EC B0 DB              bcs     se9             err
D7EE F0 D2              beq     se94            end of file
D7F0 20 B5D8            jsr     mpv1            check last sector => set CHRN
```

```
D7F3 AD BFAB    3       lda     chrn            end? (0 is no end; no read)
D7F6 F0 04              beq     4.
D7F8 45 E6              eor     inpb            end?
D7FA F0 C6              beq     se94
D7FC E6 E6      4       inc     inpb
D7FE D0 16              bne     5.f             buffer end
D800 20 27D9            jsr     chwrbl          write sector
D803 4C 14D8            jmp     6.f
D806 D0 05      se40    bne     3.f
D808 20 97DB            jsr     chbeg           test begin file
D80B F0 B9              beq     se93
D80D C6 E6      3       dec     inpb
D80F D0 05              bne     5.f
D811 20 65DB            jsr     bchwtsl         check write, set sector before
D814 B0 B3      6       bcs     se9
D816 20 44D8    5       jsr     endch1
D819 D0 C4              bne     se30
D81B F0 AC              beq     se9

                ;---------------------------------------------------------------
                ;               adjust filesize and save data sector
                ;---------------------------------------------------------------
D81D A6 E0      chki    ldx     varb            last tsl sector?
D81F BD 03AB            lda     varo+tsln,x
D822 18                 clc
D823 69 03              adc     #3
D825 A8                 tay
D826 B1 E4              lda     [tslb],y        get sector address
D828 60                 rts
D829 20 1DD8    chwren1 jsr     chki            adjust file size
D82C D0 0B              bne     chwren
D82E 9D 09AB            sta     varo+chrc,x     current size
D831 FE 0AAB            inc     varo+chrc+1,x
D834 D0 03              bne     chwren
D836 FE 0BAB            inc     varo+chrc+2,x
D839 20 27D9    chwren  jsr     chwrbl          check to write
D83C B0 1F              bcs     end9

                ;---------------------------------------------------------------
                ;               next memory byte, inc count and check ready
                ;---------------------------------------------------------------
D83E E6 EE      incchk  inc     temp            next byte
D840 D0 02              bne     endch1
D842 E6 EF              inc     temp+1
D844 EE BBAB    endch1  inc     end             check
D847 D0 14              bne     end9
D849 EE BCAB    endch2  inc     end+1
D84C D0 0F              bne     end9
D84E EE BDAB            inc     end+2
D851 D0 0A              bne     end9
D853 AD BBAB    endchk  lda     end
D856 0D BCAB            ora     end+1
D859 0D BDAB            ora     end+2
D85C 18         end8    clc
D85D 60         end9    rts                     Z=1 => A=0, end

D85E AD BBAB    secles  lda     end             check almost at end count
D861 F0 06              beq     2.              more equal 256 byte
D863 AD BCAB            lda     end+1
D866 2D BDAB            and     end+2
D869 49 FF      2       eor     #$ff            last data sector?
D86B 60                 rts

                ;---------------------------------------------------------------
                ;               set memory address and count for r/w/s
                ;---------------------------------------------------------------
D86C 18         movpova clc
D86D 84 EC      movpovb sty     mpoin           seek entry
D86F 85 ED              sta     mpoin+1
D871 A9 03              lda     #3
D873 6A                 rora
D874 8D BEAB            sta     sekf            set seek flag
D877 A0 00              ldy     #0
D879 B1 EC              lda     [mpoin],y       set begin
D87B 85 EE              sta     temp
D87D C8                 iny
D87E B1 EC              lda     [mpoin],y
D880 85 EF              sta     temp+1
D882 C8                 iny
D883 A9 00              lda     #0              calc. number of byte
D885 F1 EC              sbc     [mpoin],y
D887 8D BBAB            sta     end
D88A C8                 iny
D88B A9 00              lda     #0
```

```
D88D F1 EC              sbc     [mpoin],y
D88F 8D BCAB            sta     end+1
D892 A9 00              lda     #0
D894 2C BEAB            bit     sekf            seek => 24 bit bytecount
D897 10 03              bpl     1.
D899 E5 EF              sbc     temp+1
D89B 38                 sec
D89C E9 00      1       sbc     #0
D89E 8D BDAB            sta     end+2
                ;---------------------------------------------------------------
                ;               set pointers and variables for fcb X
                ;---------------------------------------------------------------
D8A1 20 41D2    movpo1  jsr     setpoin
D8A4 B0 29              bcs     mv2             error fd
D8A6 CE BEAB            dec     sekf
D8A9 B9 10AB            lda     varo+rdwrf,y    open mode
D8AC 8D BAAB            sta     modf+1
D8AF B9 11AB            lda     varo+f.rws,y    sector read flag
D8B2 8D B9AB            sta     modf

                ;---------------------------------------------------------------
                ;               set count if last sector
                ;---------------------------------------------------------------
D8B5 A9 00      mpv1    lda     #0              test last data sector
D8B7 8D BFAB            sta     chrn
D8BA A0 03              ldy     #tsln
D8BC B1 E0              lda     [varb],y
D8BE 69 03              adc     #3
D8C0 18                 clc
D8C1 A8                 tay
D8C2 B1 E4              lda     [tslb],y        get sector address
D8C4 D0 09              bne     mv2
D8C6 A0 09              ldy     #chrc           set CHRN to value of A
D8C8 B1 E0              lda     [varb],y
D8CA 8D BFAB            sta     chrn            = 0 if not last sector
D8CD A9 00              lda     #0
D8CF 60         mv2     rts

                ;---------------------------------------------------------------
                ;               set characters r/w/s
                ;---------------------------------------------------------------
D8D0 48         movvapo pha
D8D1 90 26              bcc     10.f            all done
D8D3 A0 02              ldy     #2              set no of bytes r/w/s
D8D5 18                 clc
D8D6 B1 EC              lda     [mpoin],y
D8D8 6D BBAB            adc     end
D8DB 91 EC              sta     [mpoin],y
D8DD C8                 iny
D8DE B1 EC              lda     [mpoin],y
D8E0 6D BCAB            adc     end+1
D8E3 91 EC              sta     [mpoin],y
D8E5 2C BEAB            bit     sekf
D8E8 10 09              bpl     2.
D8EA A0 01              ldy     #1              seek
D8EC B1 EC              lda     [mpoin],y
D8EE 6D BDAB            adc     end+2
D8F1 91 EC              sta     [mpoin],y
D8F3 4E BEAB    2       lsr     sekf            invalid fd
D8F6 B0 0E              bcs     3.f
D8F8 38                 sec
D8F9 A4 E0      10      ldy     varb            current pos in data sector
D8FB A5 E6              lda     inpb
D8FD 99 06AB            sta     varo+inpc,y
D900 AD B9AB            lda     modf
D903 99 11AB            sta     varo+f.rws,y    data sector status flag
D906 68         3       pla
D907 60                 rts

                ;---------------------------------------------------------------
                ;               read current or next data sector
                ;---------------------------------------------------------------
D908 A5 E6      cnblk   lda     inpb            sector not at beginning
D90A F0 06              beq     nexblk
D90C 20 89DB    rdblts  jsr     tslget          read current sector
D90F 4C 17D9            jmp     1.
D912 20 33DB    nexblk  jsr     nextsl          next sector
D915 B0 0F              bcs     9.f
D917 F0 0D      1       beq     9.f             eof
D919 20 81DD    rdblk   jsr     stinpb          data memory
D91C 20 C3DD            jsr     readsect1
D91F B0 05              bcs     9.
D921 A9 40              lda     #$40            data sector read
D923 8D B9AB            sta     modf
```

```
D926 60         9       rts

                ;---------------------------------------------------------------
                ;               write current data sector
                ;---------------------------------------------------------------
D927 0E B9AB    chwrbl  asl     modf            check if written in sector => write
D92A A9 00              lda     #0
D92C 8D B9AB            sta     modf            clear status flag
D92F 90 F5              bcc     9.b
D931 20 89DB            jsr     tslget          get tsa
D934 20 81DD            jsr     stinpb
D937 4C A8DD            jmp     writsect1       write data sector

                ;---------------------------------------------------------------
                ;               get next data sector for write
                ;---------------------------------------------------------------
D93A A5 E6      wrss    lda     inpb            test pos in data sector
D93C D0 CE              bne     rdblts          read last sector
D93E 20 33DB            jsr     nextsl          check if available
D941 B0 E3              bcs     9.
D943 F0 06              beq     10.f
D945 20 5ED8            jsr     secles
D948 F0 CF              beq     rdblk           read existing sector
D94A 60                 rts
D94B 0E C1AB    10      asl     wralc           first time for current write
D94E 90 46              bcc     10.f
D950 20 12DD            jsr     smap            allocation count
D953 B0 02              bcs     2.f
D955 A9 00              lda     #0              default no left
D957 85 E8      2       sta     rwpoin          temp
D959 20 97DB            jsr     chbeg           no data sectors
D95C F0 0D              beq     1.f
D95E A0 03              ldy     #tsln
D960 B1 E0              lda     [varb],y        if false tsl
D962 C9 04              cmp     #4
D964 90 05              bcc     1.f
D966 20 89DB            jsr     tslget          last data sector
D969 D0 03              bne     2.f
D96B 20 FFDB    1       jsr     tslts
D96E 8E C3AB    2       stx     altrk
D971 8C C4AB            sty     alsec
D974 88                 dey
D975 98                 tya
D976 25 E8              and     rwpoin          map significant bits
D978 45 E8              eor     rwpoin          subtract used
D97A 8D C2AB            sta     alcnt           sectors left
D97D 18                 clc
D97E 6D BCAB            adc     end+1           sectors to go
D981 85 E9              sta     rwpoin+1
D983 B0 11              bcs     10.f            no sectors needed
D985 20 26DD            jsr     trtomap         last data track
D988 84 E8              sty     rwpoin
D98A 20 1ADD            jsr     defdba          starting data map
D98D C4 E8              cpy     rwpoin
D98F B0 02              bcs     1.
D991 A4 E8              ldy     rwpoin          last data map
D993 20 09DC    1       jsr     frblks          search free sector
D996 A0 03      10      ldy     #tsln
D998 B1 E0              lda     [varb],y        check tsl full
D99A C9 FE              cmp     #$fe
D99C D0 30              bne     of1
D99E 20 69DC            jsr     blksrc          new tsl sector
D9A1 B0 3E              bcs     of19            disk full
D9A3 98                 tya
D9A4 A0 00              ldy     #0              forward link
D9A6 20 15DB            jsr     stotsl1
D9A9 20 F3DB            jsr     wrttsl          save full tsl
D9AC B0 33      9       bcs     of19
D9AE 98                 tya                     save last tsa tsa
D9AF 48                 pha
D9B0 8A                 txa
D9B1 48                 pha
D9B2 A0 00              ldy     #0              put new tsl tsa in ts var
D9B4 A2 04              ldx     #tsltk
D9B6 20 EDDA            jsr     movvar2
D9B9 20 02DB            jsr     cltsl           clear new tsl sector
D9BC 68                 pla                     full tsl tsa for bw link
D9BD AA                 tax
D9BE 68                 pla
D9BF A0 02              ldy     #2
D9C1 20 15DB            jsr     stotsl1
D9C4 A6 E0              ldx     varb            incr tsl count
D9C6 FE 02AB            inc     varo+tslc,x
D9C9 A9 02              lda     #2              set cur pos to beg of new tsl
```

```
D9CB 9D 03AB            sta     varo+tsln,x
D9CE 20 69DC    of1     jsr     blksrc          search data sector
D9D1 B0 0E              bcs     of19            disk full
D9D3 8A                 txa                     save X
D9D4 20 1DDB            jsr     nextsl          nex pos, inc data bl. count
D9D7 AA                 tax
D9D8 20 0DDB            jsr     stotsl          set data tsa in tsl sector

                ;---------------------------------------------------------------
                ;               set tsl/file changed
                ;---------------------------------------------------------------
D9DB A0 12      tslwset ldy     #tslf           tsl changed
D9DD A9 80      5       lda     #$80
D9DF 91 E0              sta     [varb],y
D9E1 60         of19    rts

D9E2 A0 13      filwset ldy     #filf           file changed
D9E4 D0 F7              bne     5.b

                ;*****************************************************************
                ;               disk, directory, tsl and data tsa subroutines
                ;*****************************************************************

                ;---------------------------------------------------------------
                ;               open disk by reading system sector
                ;---------------------------------------------------------------
D9E6 A0 00      chrdsys ldy     #0              read system if no open file on drive
D9E8 B9 00AB    4       lda     varo,y          not active fcb
D9EB F0 0A              beq     3.f
D9ED B9 0DAB            lda     varo+f.drdi,y   drive/dir
D9F0 29 03              and     #3              mask drive
D9F2 CD B4AB            cmp     drive           equal
D9F5 F0 54              beq     8.
D9F7 20 E0D2    3       jsr     nextv
D9FA 90 EC              bcc     4.b
D9FC A2 00      redsys  ldx     #systk          read system sector
D9FE A0 01              ldy     #syssc
DA00 20 79DD            jsr     stsysb
DA03 20 C3DD            jsr     readsect1
DA06 09 01              ora     #$01            flag if error
DA08 60                 rts

                ;---------------------------------------------------------------
                ;               open directory
                ;---------------------------------------------------------------
DA09 20 63D2    inidir  jsr     oppoin          define fcb
DA0C B0 3E              bcs     9.
DA0E 20 E6D9            jsr     chrdsys         open drive
DA11 B0 39              bcs     9.
DA13 A5 E3              lda     sysb+1
DA15 A0 00      firdir  ldy     #0              init dir
DA17 8C D9AB            sty     dirq
DA1A 8C DAAB            sty     dirno
DA1D 60                 rts

                ;---------------------------------------------------------------
                ;               next directory entry
                ;---------------------------------------------------------------
DA1E AD D9AB    nexdir  lda     dirq            directory opened
DA21 F0 11              beq     nd1
DA23 AC D8AB            ldy     dirp
DA26 BE 00AC            ldx     dirb,y          end directory
DA29 F0 20              beq     8.
DA2B 98                 tya
DA2C 18                 clc
DA2D 69 10              adc     #$10            set to next filename
DA2F A8                 tay
DA30 C0 F0              cpy     #$f0            end sector?
DA32 90 0C              bcc     nd2             no
DA34 20 4DDA    nd1     jsr     reddir          read next dir sector
DA37 B0 13              bcs     9.
DA39 A0 F0              ldy     #$f0
DA3B AA                 tax
DA3C F0 0D              beq     8.
DA3E A0 00              ldy     #0
DA40 8C D8AB    nd2     sty     dirp
DA43 EE DAAB            inc     dirno
DA46 A9 AC              lda     #dirb>>8
DA48 BE 00AC            ldx     dirb,y          get first char filename
DA4B 18         8       clc
DA4C 60         9       rts

                ;---------------------------------------------------------------
                ;               read next directory sector
```

```
                        ;---------------------------------------------------------------------
DA4D AE F0AC    reddir  ldx     dirb+$f0
DA50 AC F1AC            ldy     dirb+$f1        link tsa
DA53 AD D9AB            lda     dirq            first dir
DA56 D0 11              bne     ndb
DA58 A2 00              ldx     #dir0tk
DA5A A0 03              ldy     #dir0sc
DA5C AD B5AB            lda     direct          directory @
DA5F F0 08              beq     ndb
DA61 A8                 tay
DA62 B1 E2              lda     [sysb],y        dir tsa from sis
DA64 AA                 tax
DA65 C8                 iny
DA66 B1 E2              lda     [sysb],y
DA68 A8                 tay
DA69 98         ndb     tya                     end directory
DA6A F0 DF              beq     8.
DA6C EE D9AB            inc     dirq            inc dir sector count
DA6F 8E D6AB            stx     dirtk           save dir tsa
DA72 8C D7AB            sty     dirsk
DA75 20 75DD            jsr     stdirb          dir mem
DA78 20 C3DD            jsr     readsect1       read
DA7B 09 02              ora     #$02
DA7D 60                 rts

                        ;---------------------------------------------------------------------
                        ;               write system sector
                        ;---------------------------------------------------------------------
DA7E A2 00      wrtsys  ldx     #systk          write system SIS sector
DA80 A0 01              ldy     #syssc          get track and sector
DA82 20 79DD            jsr     stsysb
DA85 20 8DDD            jsr     wrchsect1       write and check
DA88 09 01              ora     #$01
DA8A 60                 rts

DA8B AD D9AB    wrtsdi  lda     dirq            extern entry write sis/dir
DA8E F0 EE              beq     wrtsys

                        ;---------------------------------------------------------------------
                        ;               write current directory sector
                        ;---------------------------------------------------------------------
DA90 AE D6AB    wrtdir  ldx     dirtk           write directory sector
DA93 AC D7AB            ldy     dirsk
DA96 20 75DD            jsr     stdirb
DA99 20 8DDD            jsr     wrchsect1
DA9C 09 02              ora     #$02
DA9E 60                 rts

                        ;---------------------------------------------------------------------
                        ;               initialize fcb variables
                        ;---------------------------------------------------------------------
DA9F A0 04      invar   ldy     #flmo           setup info in fcb
DAA1 A2 01              ldx     #fmod           move mode file
DAA3 20 F0DA            jsr     movvar
DAA6 A0 14              ldy     #flln           move byte count
DAA8 A2 09              ldx     #chrc
DAAA 20 F0DA            jsr     movvar
DAAD 20 EDDA            jsr     movvar2
DAB0 A4 E0              ldy     varb
DAB2 AD B9AB            lda     modf            save mode open in varbuf
DAB5 99 10AB            sta     varo+rdwrf,y
DAB8 AD D9AB            lda     dirq            save dir blk and pos in dir
DABB 99 0EAB            sta     varo+dirc,y
DABE AD D8AB            lda     dirp
DAC1 99 0FAB            sta     varo+dirn,y
DAC4 A9 01              lda     #1              first tsl
DAC6 99 02AB            sta     varo+tslc,y
DAC9 A9 1E              lda     #$1e            init pos in first tsl
DACB 99 03AB            sta     varo+tsln,y
DACE A9 00              lda     #0              block count and pos in block
DAD0 A0 06              ldy     #inpc
DAD2 20 D7DA            jsr     movvx3
DAD5 A0 11              ldy     #f.rws          clear changed flags
DAD7 A2 03      movvx3  ldx     #3
DAD9 91 E0      movvx   sta     [varb],y        set X* A in fcb
DADB C8                 iny
DADC CA                 dex
DADD D0 FA              bne     movvx
DADF 60                 rts

                        ;---------------------------------------------------------------------
                        ;               various fcb/tsl subroutines
                        ;---------------------------------------------------------------------
DAE0 20 E3DA    movtsl2 jsr     movtsl
```

```
DAE3 B1 E0      movtsl  lda     [varb],y        move fcb,Y to tsl,X
DAE5 20 F9DA            jsr     exxy
DAE8 91 E4              sta     [tslb],y
DAEA 4C F7DA            jmp     movva1

DAED 20 F0DA    movvar2 jsr     movvar
DAF0 B1 E4      movvar  lda     [tslb],y        move tsl,Y to fcb,X
DAF2 20 F9DA            jsr     exxy
DAF5 91 E0              sta     [varb],y
DAF7 E8         movva1  inx
DAF8 C8                 iny

DAF9 48         exxy    pha                     exchange X,Y
DAFA 8A                 txa
DAFB 48                 pha
DAFC 98                 tya
DAFD AA                 tax
DAFE 68                 pla
DAFF A8                 tay
DB00 68                 pla
DB01 60                 rts

DB02 A0 00      cltsl   ldy     #0              clear tsl sector
DB04 98                 tya
DB05 AA                 tax
DB06 91 E4      mtsla   sta     [tslb],y        store X* in tsl open w w+
DB08 C8                 iny
DB09 CA                 dex
DB0A D0 FA              bne     mtsla
DB0C 60                 rts

DB0D 98         stotsl  tya                     store tsa in tsl
DB0E 48                 pha
DB0F A0 03              ldy     #tsln           get pos
DB11 B1 E0              lda     [varb],y
DB13 A8                 tay
DB14 68                 pla
DB15 C8         stotsl1 iny                     store a,x tsa
DB16 91 E4              sta     [tslb],y        first save sector
DB18 88                 dey
DB19 8A                 txa
DB1A 91 E4              sta     [tslb],y        and track
DB1C 60                 rts

                ;---------------------------------------------------------------
                ;               next data tsa
                ;---------------------------------------------------------------
DB1D A6 E0      nexts1  ldx     varb            point to next tsa
DB1F FE 07AB            inc     varo+blkc,x     incr sector count
DB22 D0 03              bne     ntsa
DB24 FE 08AB            inc     varo+blkc+1,x
DB27 FE 03AB    ntsa    inc     varo+tsln,x     incr pos in tsl
DB2A FE 03AB            inc     varo+tsln,x
DB2D 60                 rts

DB2E 20 27D9    nchwts1 jsr     chwrbl          first save current data sector
DB31 B0 63              bcs     tsg9
DB33 20 1DDB    nexts1  jsr     nexts1          get next tsa tsl (Z=1 => end)
DB36 D0 12              bne     ntsg1           test overflow => get next
DB38 18                 clc
DB39 A0 01              ldy     #1              forw. link
DB3B B1 E4              lda     [tslb],y
DB3D F0 10              beq     bacts1          no => end
DB3F 20 A6DB            jsr     nrdtsl          read next
DB42 B0 52              bcs     tsg9
DB44 A0 03              ldy     #tsln
DB46 A9 04              lda     #4
DB48 91 E0              sta     [varb],y
DB4A 20 89DB    ntsg1   jsr     tslget
DB4D D0 47              bne     tsg9            not at end

                ;---------------------------------------------------------------
                ;               previous data tsa
                ;---------------------------------------------------------------
DB4F A6 E0      bacts1  ldx     varb            point to previous tsa
DB51 BD 07AB            lda     varo+blkc,x     decr sector count
DB54 D0 03              bne     ntsb
DB56 DE 08AB            dec     varo+blkc+1,x
DB59 DE 07AB    ntsb    dec     varo+blkc,x     decr pos in tsl
DB5C DE 03AB            dec     varo+tsln,x
DB5F DE 03AB            dec     varo+tsln,x
DB62 A9 00      bat8    lda     #0              set Z=1
DB64 60                 rts
```

```
DB65 20 27D9    bchwtsl jsr     chwrbl          first save current data sector
DB68 B0 2C              bcs     tsq9
DB6A 20 97DB    bactsl  jsr     chbeg           test begin
DB6D F0 F3              beq     bat8
DB6F 20 4FDB            jsr     bactsl          get tsa backwards
DB72 20 97DB            jsr     chbeg
DB75 F0 EB              beq     bat8
DB77 BD 03AB            lda     varo+tsln,x
DB7A C9 04              cmp     #4              test begin other tsl sectors
DB7C B0 0B              bcs     tslget          no => get tsa
DB7E 20 BADB            jsr     brdtsl          read bw tsl (check write)
DB81 B0 13              bcs     tsq9            err
DB83 A0 03              ldy     #tsln           set pos to end tsl
DB85 A9 FE              lda     #$fe
DB87 91 E0              sta     [varb],y

                ;---------------------------------------------------------------------------
                ;               data tsa from tsl
                ;---------------------------------------------------------------------------
DB89 A0 03      tslget  ldy     #tsln           get tsa from tsl
DB8B B1 E0              lda     [varb],y
DB8D A8                 tay
DB8E B1 E4              lda     [tslb],y
DB90 AA                 tax
DB91 C8                 iny
DB92 B1 E4              lda     [tslb],y
DB94 A8                 tay
DB95 18         tsq8    clc
DB96 60         tsq9    rts

DB97 A0 02      chbeg   ldy     #tslc           check 1st data tsa
DB99 B1 E0              lda     [varb],y
DB9B C9 01              cmp     #1
DB9D D0 F6              bne     tsq8            not first tsl
DB9F C8                 iny
DBA0 B1 E0              lda     [varb],y
DBA2 C9 1E              cmp     #$1e
DBA4 18                 clc
DBA5 60                 rts

                ;---------------------------------------------------------------------------
                ;               next tsl sector
                ;---------------------------------------------------------------------------
DBA6 20 ECDB    nrdtsl  jsr     chwrts          changed tsl => write
DBA9 B0 53              bcs     rts9
DBAB A0 00              ldy     #0              get next tsl from link
DBAD B1 E4              lda     [tslb],y
DBAF 48                 pha
DBB0 C8                 iny
DBB1 B1 E4              lda     [tslb],y
DBB3 A6 E0              ldx     varb
DBB5 FE 02AB            inc     varo+tslc,x
DBB8 90 20              bcc     rtsb            read

                ;---------------------------------------------------------------------------
                ;               previous tsl sector
                ;---------------------------------------------------------------------------
DBBA 20 ECDB    brdtsl  jsr     chwrts          read tsl backwards
DBBD B0 3F              bcs     rts9
DBBF A0 02              ldy     #2              get backward link
DBC1 B1 E4              lda     [tslb],y
DBC3 48                 pha
DBC4 C8                 iny
DBC5 B1 E4              lda     [tslb],y
DBC7 A6 E0              ldx     varb
DBC9 DE 02AB            dec     varo+tslc,x     decr count tsl
DBCC 90 0C              bcc     rtsb

                ;---------------------------------------------------------------------------
                ;               read tsl sector
                ;---------------------------------------------------------------------------
DBCE AE DBAB    redtsl1 ldx     dirp            read 1st tsl sector
DBD1 BD 0EAC            lda     dirb+$e,x
DBD4 48                 pha
DBD5 BD 0FAC            lda     dirb+$f,x
DBD8 A6 E0              ldx     varb
DBDA A8         rtsb    tay
DBDB 9D 05AB            sta     varo+tslsk,x    save tsa tsl
DBDE 68                 pla
DBDF 9D 04AB            sta     varo+tsltk,x
DBE2 AA                 tax
DBE3 20 7DDD    rdtsl   jsr     sttslb          read tsl sector
DBE6 20 C3DD            jsr     readsect1
DBE9 09 04              ora     #$04
```

```
DBEB 60                 rts

                        ;---------------------------------------------------------------------
                        ;               write current tsl sector
                        ;---------------------------------------------------------------------
DBEC A6 E0      chwrts  ldx     varb            check write tsl
DBEE 1E 12AB            asl     varo+tslf,x
DBF1 90 0B              bcc     rts9            not changed
DBF3 20 FFDB    wrttsl  jsr     tslts
DBF6 20 7DDD            jsr     sttslb
DBF9 20 A8DD            jsr     writsect1
DBFC 09 04              ora     #$04
DBFE 60         rts9    rts

DBFF A6 E0      tslts   ldx     varb            get tsa tsl sector
DC01 BC 05AB            ldy     varo+tslsk,x
DC04 BD 04AB            lda     varo+tsltk,x
DC07 AA                 tax
DC08 60                 rts

                        ;*********************************************************************
                        ;               disk bitmap allocation subroutines
                        ;*********************************************************************

                        ;---------------------------------------------------------------------
                        ;               search contiguous free space
                        ;---------------------------------------------------------------------
DC09 98         frblks  tya                     map position
DC0A 48                 pha
DC0B AD 12C0            lda     dosmode         free sector search?
DC0E 29 02              and     #$02
DC10 F0 52              beq     frbl9
DC12 88                 dey
DC13 C8         10      iny
DC14 F0 4E              beq     frbl9           end of bitmap => not found
DC16 B1 E2              lda     [sysb],y        all occupied
DC18 F0 F9              beq     10.
DC1A A9 00              lda     #0
DC1C 38                 sec
DC1D 48                 pha
DC1E 68         11      pla
DC1F 2A                 rola                    shift bit
DC20 B0 F1              bcs     10.             next 8 sectors
DC22 48                 pha
DC23 31 E2              and     [sysb],y        bitmap with free map
DC25 F0 F7              beq     11.             not free
DC27 8C C0AB            sty     blss            save position
DC2A A6 E9              ldx     rwpoin+1        sectors to search
DC2C 90 21              bcc     25.             branch
                        ;
DC2E 84 E8      30      sty     rwpoin          check if existing sector
DC30 98                 tya
DC31 29 03              and     #$03
DC33 09 24              ora     #$24
DC35 A8                 tay                     format map
DC36 68                 pla                     bit
DC37 48                 pha
DC38 31 E2              and     [sysb],y        check formatted map
DC3A 08                 php                     save test
DC3B A4 E8              ldy     rwpoin          restore Y
DC3D 28                 plp
DC3E D0 DE              bne     11.b            legal sector => gap not found
DC40 68                 pla                     remove bit pattern
                        ;
DC41 C8         20      iny                     next 8 sectors
DC42 F0 20              beq     frbl9           end of bitmap
DC44 A9 00              lda     #0              bit 1
DC46 38                 sec
DC47 2A         21      rola                    shift pattern
DC48 B0 F7              bcs     20.             end => next byte free map
DC4A 48                 pha
DC4B 31 E2              and     [sysb],y        bit test
DC4D F0 DF              beq     30.b            not free => check
DC4F 84 E8      25      sty     rwpoin          save Y
DC51 20 12DD            jsr     smap            allocation count
DC54 18                 clc
DC55 A8                 tay
DC56 E8         2       inx
DC57 F0 08              beq     8.              found
DC59 88                 dey
DC5A 10 FA              bpl     2.              until count
DC5C A4 E8              ldy     rwpoin
DC5E 68                 pla                     bit pattern
DC5F D0 E6              bne     21.b            branch to next bit
```

```
                        ;
DC61 68                 8       pla                     remove bit
DC62 68                         pla                     remove old sector start
DC63 60                         rts
DC64 68                 frb19   pla                     not found reset BLSS
DC65 8D C0AB            blks3   sta     blss            save last
DC68 60                 9       rts

                        ;---------------------------------------------------------------
                        ;               search and allocate free sector
                        ;---------------------------------------------------------------
DC69 20 02DD            blksrc  jsr     nexsec          sector not yet used
DC6C 90 FA                      bcc     9.b
DC6E AC C0AB                    ldy     blss            free data sector start from BLSS
DC71 20 94DC                    jsr     secsrc
DC74 90 EF                      bcc     blks3
DC76 20 1ADD                    jsr     defdba          data sector allocation
DC79 20 94DC                    jsr     secsrc
DC7C 90 E7                      bcc     blks3

DC7E A0 2B              tslsrc  ldy     #s.tbam         tsl search pattern
DC80 B1 E2                      lda     [sysb],y
DC82 48                         pha
DC83 88                         dey                     tsl starting track
DC84 20 23DD                    jsr     defdb1
DC87 68                         pla
DC88 31 E2                      and     [sysb],y
DC8A D0 15                      bne     bls5            found
DC8C C8                         iny
DC8D 20 94DC                    jsr     secsrc
DC90 90 D6                      bcc     9.b

DC92 A0 60              dirsrc  ldy     #$60            directory search
DC94 C0 60              secsrc  cpy     #$60            search and lock free sector
DC96 B0 05                      bcs     2.
DC98 A0 5F                      ldy     #$5f
DC9A C8                 10      iny
DC9B F0 61                      beq     diful           end of bit map
DC9D B1 E2              2       lda     [sysb],y
DC9F F0 F9                      beq     10.             all occupied
DCA1 A2 FF              bls5    ldx     #$ff
DCA3 E8                 1       inx                     mask position
DCA4 4A                         lsra
DCA5 90 FC                      bcc     1.b
DCA7 98                         tya
DCA8 48                         pha                     save map position
DCA9 8A                         txa
DCAA 48                         pha                     save bit pos
DCAB A9 FF                      lda     #$ff
DCAD 18                         clc
DCAE 2A                 1       rola                    reset mask
DCAF CA                         dex
DCB0 10 FC                      bpl     1.b
DCB2 31 E2                      and     [sysb],y        lock map bit
DCB4 91 E2                      sta     [sysb],y
DCB6 98                         tya
DCB7 E9 60                      sbc     #$60
DCB9 85 E8                      sta     rwpoin          map track
DCBB A0 28                      ldy     #s.acnt
DCBD B1 E2                      lda     [sysb],y        allocation count
DCBF E9 01                      sbc     #1
DCC1 8D C2AB                    sta     alcnt           sectors left
DCC4 A9 03                      lda     #3              default track divide-1
DCC6 90 09                      bcc     2.f
DCC8 C8                         iny
DCC9 B1 E2                      lda     [sysb],y        track shift 2, 1 or 0
DCCB 88                         dey                     at alloc count
DCCC 0A                         asla                    4, 2 or 0
DCCD F0 02                      beq     2.
DCCF E9 00                      sbc     #0              3 or 1
DCD1 48                 2       pha
DCD2 25 E8                      and     rwpoin          mask map sector byte
DCD4 0A                         asla                    sector times 8
DCD5 0A                         asla
DCD6 0A                         asla
DCD7 AA                         tax
DCD8 68                         pla
DCD9 F0 05                      beq     2.f
DCDB 46 E8              1       lsr     rwpoin          shift track
DCDD 4A                         lsra
DCDE D0 FB                      bne     1.b
DCE0 8A                 2       txa
DCE1 A6 E8                      ldx     rwpoin          physical track
DCE3 85 E8                      sta     rwpoin
```

```
DCE5 68                 pla                     bit position
DCE6 05 E8              ora     rwpoin          add offset
DCE8 48                 pha                     map sector
DCE9 B1 E2              lda     [sysb],y        alloc count
DCEB 85 E8              sta     rwpoin
DCED 68                 pla
DCEE 0A         1       asla                    shift map sector
DCEF 46 E8              lsr     rwpoin
DCF1 D0 FB              bne     1.b             until too far
DCF3 4A                 lsra                    shift one back
DCF4 A8                 tay
DCF5 C8                 iny                     physical sector
DCF6 8E C3AB            stx     altrk           allocated ts
DCF9 8C C4AB            sty     alsec
DCFC 68                 pla                     map position
DCFD 60                 rts
DCFE A9 02      diful   lda     #e2
DD00 38         4       sec                     disk full
DD01 60                 rts

DD02 CE C2AB    nexsec  dec     alcnt           sector already allocated
DD05 30 F9              bmi     4.b
DD07 EE C4AB            inc     alsec
DD0A AE C3AB            ldx     altrk
DD0D AC C4AB            ldy     alsec
DD10 18                 clc
DD11 60                 rts

                ;---------------------------------------------------------------
                ;               allocation conversion
                ;---------------------------------------------------------------
DD12 A0 28      smap    ldy     #s.acnt         system allocation count
DD14 B1 E2              lda     (sysb),y
DD16 38                 sec
DD17 E9 01              sbc     #1              subtract one in advance
DD19 60                 rts                     C = 0 old version

DD1A A2 02      defdba  ldx     #2              default data track 2
DD1C 20 12DD            jsr     smap
DD1F 90 05              bcc     trtomap
DD21 A0 2C              ldy     #s.dbas
DD23 B1 E2      defdb1  lda     [sysb],y        for tsl search
DD25 AA                 tax
DD26 20 12DD    trtomap jsr     smap            convert track X to mapposition Y
DD29 A9 02              lda     #2
DD2B 90 03              bcc     1.
DD2D C8                 iny
DD2E B1 E2              lda     [sysb],y
DD30 A8         1       tay
DD31 8A                 txa
DD32 0A         1       asla
DD33 88                 dey
DD34 10 FC              bpl     1.
DD36 6A                 rora
DD37 69 60              adc     #$60
DD39 A8                 tay
DD3A 60         9       rts

                ;---------------------------------------------------------------
                ;               free sector
                ;---------------------------------------------------------------
DD3B B1 E4      tslfre  lda     [tslb],y        get tsa from tsl and unlock
DD3D AA                 tax
DD3E C8                 iny
DD3F B1 E4              lda     [tslb],y
DD41 F0 F7              beq     9.b             escape from delete next tsl
DD43 A8                 tay
DD44 88         blkfre  dey
DD45 84 E8              sty     rwpoin          physical sector -1
DD47 20 12DD            jsr     smap
DD4A F0 0B              beq     2.f             1 to 1
DD4C 90 09              bcc     2.f             version 1
DD4E 24 E8              bit     rwpoin
DD50 D0 22              bne     9.f             not first sector
DD52 46 E8      1       lsr     rwpoin          shift until map sector
DD54 4A                 lsra
DD55 D0 FB              bne     1.b
DD57 20 26DD    2       jsr     trtomap
DD5A 38                 sec
DD5B A5 E8              lda     rwpoin
DD5D 88                 dey
DD5E C8         1       iny                     adjust map position
DD5F E9 08              sbc     #8              until in range
DD61 B0 FB              bcs     1.b
```

```
DD63 69 08              adc     #8              adjust and set carry
DD65 AA                 tax
DD66 A9 00              lda     #0
DD68 2A         1       rola                    mask pattern
DD69 CA                 dex
DD6A 10 FC              bpl     1.b
DD6C C0 60              cpy     #$60
DD6E 90 04              bcc     9.              something wrong
DD70 11 E2              ora     [sysb],y
DD72 91 E2              sta     [sysb],y        unlock
DD74 60         9       rts

                ;*********************************************************************
                ;               disk read / write
                ;*********************************************************************
DD75 A9 AC      stdirb  lda     #dirb>>8        directory memory
DD77 D0 0A              bne     1.f
DD79 A5 E3      stsysb  lda     sysb+1          system memory
DD7B D0 06              bne     1.f
DD7D A5 E5      sttslb  lda     tslb+1          tsl memory
DD7F D0 02              bne     1.f
DD81 A5 E7      stinpb  lda     inpb+1          data memory
DD83 85 E9      1       sta     rwpoin+1
DD85 A9 00              lda     #0
DD87 85 E8              sta     rwpoin
DD89 AD B4AB            lda     drive
DD8C 60                 rts

                ;---------------------------------------------------------------------
                ;               write/check sector
                ;---------------------------------------------------------------------
                ;A drive, X track, Y sector
DD8D 20 D7DD    wrchsect1 jsr   ready           write check sector
DD90 B0 3B              bcs     rs11
DD92 A9 03              lda     #3
DD94 8D B8AB            sta     psec
DD97 20 0BCC    wc1     jsr     wshw            write
DD9A B0 14              bcs     ws9             error write
DD9C 20 11CC            jsr     cshw            check
DD9F 90 2C              bcc     rs11            ok!
DDA1 CE B8AB            dec     psec            try 3 times
DDA4 D0 F1              bne     wc1
DDA6 F0 0C              beq     ws91            write error mess

                ;---------------------------------------------------------------------
                ;               write sector
                ;---------------------------------------------------------------------
DDA8 20 D7DD    writsect1 jsr   ready           write sector routine
DDAB B0 20              bcs     rs11            not ready
DDAD 20 0BCC            jsr     wshw
DDB0 A9 D0      ws9     lda     #$d0            if C write protected
DDB2 70 19              bvs     rs11
DDB4 A9 80      ws91    lda     #$80            if C write error
DDB6 D0 15              bne     rs11

                ;---------------------------------------------------------------------
                ;               check sector
                ;---------------------------------------------------------------------
DDB8 20 D7DD    checsect1 jsr   ready           check sector
DDBB B0 10              bcs     rs11
DDBD 20 11CC            jsr     cshw
DDC0 4C CBDD            jmp     rs9

                ;---------------------------------------------------------------------
                ;               read sector
                ;---------------------------------------------------------------------
DDC3 20 D7DD    readsect1 jsr   ready           read sector routine
DDC6 B0 05              bcs     rs11            not ready
DDC8 20 0ECC            jsr     rshw
DDCB A9 90      rs9     lda     #$90
DDCD 8C B8AB    rs11    sty     psec            error sector address
DDD0 AE B6AB            ldx     xtrk
DDD3 AC B7AB            ldy     ysec
DDD6 60                 rts

                ;---------------------------------------------------------------------
                ;               check dtsa, convert sa and select drive
                ;---------------------------------------------------------------------
DDD7 8D B4AB    ready   sta     drive
DDDA 8E B6AB            stx     xtrk
DDDD 8C B7AB            sty     ysec
DDE0 CD 00CC            cmp     drives          max drives
DDE3 B0 50              bcs     rdy14
DDE5 A0 20              ldy     #s.mode
```

```
DDE7 B1 E2            lda   [sysb],y    mode
DDE9 0A               asla
DDEA 08               php                     C=1 no lookup N=1 side offset
DDEB 29 40            and   #$40              V=1 80 tracks diskette
DDED 69 40            adc   #$40
DDEF C8               iny
DDF0 8A               txa
DDF1 F0 09            beq   5.f               track 0
DDF3 D1 E2            cmp   [sysb],y          max track
DDF5 B0 3D            bcs   rdy14a
DDF7 50 03            bvc   5.f
DDF9 09 80            ora   #$80              80 tracks floppy drive
DDFB AA               tax
DDFC AD B7AB    5     lda   ysec
DDFF 38               sec
DE00 E9 01            sbc   #1
DE02 F0 24            beq   4.f               sector 1
DE04 C8               iny
DE05 D1 E2            cmp   [sysb],y          max sector
DE07 B0 2B            bcs   rdy14a
DE09 C8               iny
DE0A 28               plp
DE0B 08               php
DE0C B0 14            bcs   6.                no lookup
DE0E 10 06            bpl   3.                side lookup
DE10 D1 E2            cmp   [sysb],y          side offset
DE12 90 02            bcc   3.
DE14 F1 E2            sbc   [sysb],y          sector/side
DE16 A8         3     tay
DE17 B1 E2            lda   [sysb],y          lookup logic -> physic
DE19 A0 23            ldy   #s.msec
DE1B 90 03            bcc   1.                no side offset
DE1D 18               clc
DE1E 71 E2            adc   [sysb],y
DE20 E9 00      1     sbc   #0
DE22 D1 E2      6     cmp   [sysb],y
DE24 90 02            bcc   4.
DE26 09 80            ora   #$80              select side 1
DE28 28         4     plp                     adjust stack
DE29 A8               tay
DE2A C8               iny
DE2B AD B4AB          lda   drive
DE2E 20 08CC          jsr   rdyhw
DE31 A9 C0            lda   #$c0
DE33 60               rts
DE34 28         rdy14a plp
DE35 A9 B0      rdy14 lda   #$b0              error dts address
DE37 38               sec
DE38 60               rts

                ;*********************************************************************
                ;               print error message
                ;*********************************************************************
DE39 90 4F      ermes1 bcc    erm9
DE3B 48               pha
DE3C A8               tay
DE3D 10 05            bpl   1.                soft
DE3F 20 62C0          jsr   seter             redirect to stdio
DE42 29 F0            and   #$f0
DE44 8D C0AB    1     sta   blss
DE47 A0 FF            ldy   #$ff
DE49 D0 06            bne   erm1
DE4B C8         erm2  iny                     next message
DE4C B9 2EDF          lda   ermtb,y
DE4F 10 FA            bpl   erm2
DE51 C8         erm1  iny
DE52 B9 2EDF          lda   ermtb,y
DE55 F0 05            beq   erm10             end
DE57 CD C0AB          cmp   blss
DE5A D0 EF            bne   erm2
DE5C C8         erm10 iny                     print
DE5D B9 2EDF          lda   ermtb,y
DE60 30 14            bmi   erm11             end
DE62 C9 0A            cmp   #10
DE64 90 05            bcc   erm12
DE66 20 23C0          jsr   out
DE69 90 F1            bcc   erm10

DE6B 8C C0AB    erm12 sty   blss              subr print
DE6E 20 A7DE          jsr   ermp
DE71 AC C0AB          ldy   blss
DE74 D0 E6            bne   erm10

DE76 A8         erm11 tay                     end message
```

```
DE77 C8                 iny
DE78 F0 0B              beq     80.f            newline
DE7A C8                 iny
DE7B F0 0E              beq     10.f            error
DE7D C8                 iny
DE7E F0 17              beq     11.f            protected
DE80 68                 pla                     only number
DE81 48                 pha
DE82 20 38C0            jsr     hexout
DE85 20 2FC0    80      jsr     crlf
DE88 38                 sec
DE89 68                 pla
DE8A 60         erm9    rts

DE8B 20 3BC0    10      jsr     print
DE8E 206572726F         fcc     ' error',0
DE93 7200
DE95 90 EE              bcc     80.b

DE97 20 3BC0    11      jsr     print
DE9A 2070726F74         fcc     ' protected',0
DE9F 6563746564
DEA4 00
DEA5 90 DE              bcc     80.b

DEA7 A8         ermp    tay                     print subroutine
DEA8 88                 dey
DEA9 F0 26              beq     erm31           filespec
DEAB 88                 dey
DEAC F0 1E              beq     erm30           filename
DEAE 20 D4DE            jsr     erspa           drive
DEB1 AD B4AB            lda     drive
DEB4 20 35C0            jsr     hnout
DEB7 88                 dey
DEB8 D0 D0              bne     erm9
DEBA 20 F4DE            jsr     ercol           read/write error
DEBD AD B6AB            lda     xtrk
DEC0 20 38C0            jsr     hexout
DEC3 20 F4DE            jsr     ercol
DEC6 AD B8AB            lda     psec
DEC9 4C 38C0            jmp     hexout

DECC 20 FBDE    erm30   jsr     prnamd          print filespec
DECF 90 03              bcc     erspa
DED1 20 0BDF    erm31   jsr     prfild          print filename
DED4 A9 20      erspa   lda     #' '
DED6 D0 1E              bne     1.f

                ;---------------------------------------------------------------------------
                ;               print drive/directory
                ;---------------------------------------------------------------------------
DED8 AD EEAB    prdriv  lda     fdrive
DEDB 48                 pha
DEDC 29 03              and     #3
DEDE 20 35C0            jsr     hnout
DEE1 20 F4DE            jsr     ercol
DEE4 68                 pla
DEE5 29 1C              and     #$1c
DEE7 F0 1F              beq     9.
DEE9 4A                 lsra
DEEA 4A                 lsra
DEEB 09 40              ora     #'@'
DEED 20 23C0            jsr     out
DEF0 A9 2F              lda     #'/'
DEF2 D0 02              bne     1.
DEF4 A9 3A      ercol   lda     #':'
DEF6 4C 23C0    1       jmp     out

                ;---------------------------------------------------------------------------
                ;               print filespec
                ;---------------------------------------------------------------------------
DEF9 90 03      prname1 bcc     1.f             no filespec
DEFB 20 D8DE    prnamd  jsr     prdriv
DEFE A0 00      1       ldy     #0              print name from filenamebuffer
DF00 B9 E0AB    prn1    lda     namb,y
DF03 20 22DF            jsr     prno
DF06 D0 F8              bne     prn1
DF08 60         9       rts

                ;---------------------------------------------------------------------------
                ;               print filename
                ;---------------------------------------------------------------------------
DF09 90 03      prfile1 bcc     1.f
DF0B 20 D8DE    prfild  jsr     prdriv
```

```
DF0E A0 00      1       ldy     #0              print name from directoy
DF10 8A                 txa                     save X
DF11 48                 pha
DF12 AE D8AB            ldx     dirp            assume correct position
DF15 CA                 dex
DF16 E8         prn2    inx
DF17 BD 00AC            lda     dirb,x
DF1A 20 22DF            jsr     prno
DF1D D0 F7              bne     prn2
DF1F 68                 pla
DF20 AA                 tax
DF21 60                 rts
DF22 29 7F      prno    and     #$7f
DF24 F0 06              beq     prno9
DF26 20 23C0            jsr     out
DF29 C8                 iny
DF2A C0 0E              cpy     #14             max length
DF2C 18         prno9   clc
DF2D 60                 rts

                ;error messages
DF2E            ermtb
DF2E 014D656D2E         fcc     e1,'Mem. full',$ff
DF33 2066756C6C
DF38 FF
DF39 024469736B         fcc     e2,'Disk full',$ff
DF3E 2066756C6C
DF43 FF
DF44 03424F46FF         fcc     e3,'BOF',$ff
DF49 04454F46FF         fcc     e4,'EOF',$ff
DF4E 1046696C65         fcc     e10,'Filename',$fe
DF53 6E616D65FE
DF58 1144657669         fcc     e11,'Device',$fe
DF5D 6365FE
DF60 1246696C65         fcc     e12,'Filename missing',$ff
DF65 6E616D6520
DF6A 6D69737369
DF6F 6E67FF
DF72 13026E6F74         fcc     e13,2,'not found',$ff
DF77 20666F756E
DF7C 64FF
DF7E 154F707469         fcc     e15,'Option',$fe
DF83 6F6EFE
DF86 1644617461         fcc     e16,'Data',$fe
DF8B FE
                ;       fcc     e20,'No more entries',$ff
DF8C 21016F7065         fcc     e21,1,'open file',$ff
DF91 6E2066696C
DF96 65FF
                ;       fcc     e25,1,'delete',$fd
                ;       fcc     e26,1,'write',$fd
                ;       fcc     e27,1,'read',$fd
DF98 2801706572         fcc     e28,1,'permission denied',$ff
DF9D 6D69737369
DFA2 6F6E206465
DFA7 6E696564FF
DFAC 2901696E63         fcc     e29,1,'incorrect format',$ff
DFB1 6F72726563
DFB6 7420666F72
DFBB 6D6174FF
                ;       fcc     e30,'File not open',$ff
                ;       fcc     e31,'Fd',$fe
DFBF 8057726974         fcc     $80,'Write',3,$fe
DFC4 6503FE
DFC7 9052656164         fcc     $90,'Read',3,$fe
DFCC 03FE
DFCE B044723A74         fcc     $b0,'Dr:tr:sc',3,$fe
DFD3 723A736303
DFD8 FE
DFD9 C044726976         fcc     $c0,'Drive',4,' not ready',$ff
DFDE 6504206E6F
DFE3 7420726561
DFE8 6479FF
DFEB D04469736B         fcc     $d0,'Disk',4,$fd
DFF0 04FD
DFF2 0045727206F        fcc     0,'Error ',$fc
DFF7 7220FC

Errors detected: 0
```