```
PAGE -   0
0000                            ;
0000                            ;       FILENAME: MONITOR
0000                            ;
0000                            ;-------------------------------------------------------------------
0000                            ;
0000                            ;       MACHINE CODE MONITOR FOR MOTOROLA 68000
0000                            ;
0000                            ;       Written by Rich Page       June 7, 1982
0000                            ;
0000                            ;-------------------------------------------------------------------
0000                            ;
0000                            ;       Register Allocation
0000                            ;
0000                            ;       A7      stack pointer
0000                            ;       A6      local pointer
0000                            ;       A5      global pointer
0000                            ;       A4
0000                            ;       A3-A0   scratch
0000                            ;
0000                            ;       D7-D0   scratch
0000                            ;
0000                            ;-------------------------------------------------------------------
0000                            ;
0000                            ;       Layout of low memory
0000                            ;
0000                            ;       From    To      Description
0000                            ;
0000                            ;       0000    00FF    Exception Vectors
0000                            ;       0100    0147    Memory Configuration Map
0000                            ;       0200    03FF    Free space for user asm lang globals
0000                            ;       0400    07FF    Macsbug globals
0000                            ;       0800    09FF    Mapped to FA0000 for DRIVERS.OBJ
0000                            ;       0A00    0BFF    Free
0000                            ;       0Cxx    0CFF    System Globals
0000                            ;       0D00    0Dxx    Syscom, miscinfo
0000                            ;       0E00    0EFF    Strbuf
0000                            ;       0F00    0FFF    Save Area for Zero Page
0000                            ;       1000    17FF    Data Area for DRIVERS.OBJ
0000                            ;       1800    2FFF    User Jump Table
0000                            ;       3000            Heap Bottom
0000                            ;
0000                            ;-------------------------------------------------------------------
0000                            ;
0000                            ;       Memory Configuration Map
0000                            ;
0000                            ;       $100    Ptr to Monitor Bottom (also base of the unit table)
0000                            ;       $104    Ptr to Monitor Top
0000                            ;       $108    unused for now
0000                            ;       $10C    Ptr to Bottom of Code Stack
0000                            ;       $110    Screen Base
0000                            ;       $114    Memory Top
0000                            ;       $118    Port to the Apple
0000                            ;       $11C    UART for Macsbug
0000                            ;       $120    Ptr to Macsbug Jump Table
PAGE -   1              FILE:MONITOR.TEXT


0000                            ;       $124    Ptr to GOTOXY routine
0000                            ;       $128    Ptr to SOROC driver
0000                            ;       $12C    Ptr to Default Soft Break Point Table
0000                            ;       $130    Ptr to UART driver
0000                            ;       $134    Ptr to built in parallel port
0000                            ;       $138    Start of the heap
0000                            ;       $13C    Start of the stack
0000                            ;       $140    Ptr to users last A6
0000                            ;       $144    Ptr to MEMORY: (ie. device #4: )
0000                            ;       $148    Ptr to Twiggy Driver
0000                            ;       $14C    RECORD MACHINE,FREE2,FREE3,FREE4:BYTE END
0000                            ;       $150    Ptr to debug card
0000                            ;       $154    Ptr to loader for intrinsic units
0000                            ;       $158    Ptr to four port card
0000                            ;       $15C    Ptr to external file system
0000                            ;
0000                            ;       Definition of $14C
0000                            ;
0000                            ;       MACHINE         0=LISA, 1=LITTLE
0000                            ;
0000                            ;-------------------------------------------------------------------
0000                            ;
PAGE -   2  MONITOR     FILE:MONITOR.TEXT


0000                                    .PROC   MONITOR,0
0000                            ;
0000                            ; misc equates
0000                            ;
0000  0000 0000                 NIL     EQU     0               ; matches Feb 1981 compiler (and beyond)
0000                            ;
0000                            ; iorslt word equates
0000                            ;
0000  0000 0000                 INOERR  EQU     0
0000  0000 0001                 IBADBLK EQU     1
0000  0000 0002                 IBADUNT EQU     2
0000  0000 0003                 IBADMOD EQU     3
0000  0000 0004                 ITIMOUT EQU     4
```

```
0000| 0000 0005            ILSTUNT EQU     5
0000| 0000 0006            ILSTFIL EQU     6
0000| 0000 0007            IBADTTL EQU     7
0000| 0000 0008            INOROOM EQU     8
0000| 0000 0009            INOUNIT EQU     9
0000| 0000 000A            INOFILE EQU     10
0000| 0000 000B            IDUPFIL EQU     11
0000| 0000 000C            INOTCLS EQU     12
0000| 0000 000D            INOTOPN EQU     13
0000| 0000 000E            IBADFMT EQU     14
0000| 0000 000F            ISTROVF EQU     15
0000|                      ;
0000|                      ; syscom definition
0000|                      ;
0000| 0000 0000            IORSLT  EQU     0
0000| 0000 0001            SYSUNT  EQU     IORSLT+1
0000| 0000 0002            HITIME  EQU     SYSUNT+1
0000| 0000 0004            LOTIME  EQU     HITIME+2
0000| 0000 0006            MSCNFO  EQU     LOTIME+2
0000| 0000 0008            CRTTYP  EQU     MSCNFO+2
0000| 0000 000A            CRTESC  EQU     CRTTYP+2
0000| 0000 0011            FILCNT  EQU     CRTESC+7
0000| 0000 0012            CLRSCRN EQU     FILCNT+1
0000| 0000 0013            CLRLINE EQU     CLRSCRN+1
0000| 0000 0016            CRTINFO EQU     CLRLINE+3
0000| 0000 001E            FLUSH   EQU     CRTINFO+8
0000| 0000 001F            EOF     EQU     FLUSH+1
0000| 0000 0020            SSTOP   EQU     EOF+1
0000| 0000 0021            BREAK   EQU     SSTOP+1
0000| 0000 0022            BADCHR  EQU     BREAK+1
0000| 0000 0023            CHRDEL  EQU     BADCHR+1
0000| 0000 0024            ALTMOD  EQU     CHRDEL+1
0000| 0000 0025            LINDEL  EQU     ALTMOD+1
0000| 0000 0026            ETX     EQU     LINDEL+1
0000| 0000 0027            PREFIX  EQU     ETX+1
0000| 0000 0028            ALPHLCK EQU     PREFIX+1
0000| 0000 002C            SYSCMSZ EQU     ALPHLCK+4
0000|                      ;
0000|                      ; filekind equates
0000|                      ;
0000| 0000 0000            UNTYPFL  EQU    0
PAGE -   3  MONITOR   FILE:MONITOR.TEXT


0000| 0000 0001            XDSKFILE EQU    1
0000| 0000 0002            CODEFILE EQU    2
0000| 0000 0003            TEXTFILE EQU    3
0000| 0000 0004            INFOFILE EQU    4
0000| 0000 0005            DATAFILE EQU    5
0000| 0000 0006            GRAFFILE EQU    6
0000| 0000 0007            FOTOFILE EQU    7
0000| 0000 0008            SECURDIR EQU    8
0000|                      ;
0000|                      ; directory equates
0000|                      ;
0000| 0000 004D            MAXDIR  EQU     77              ; max number of entries in 4 blocks
0000| 0000 0312            MAXDIRB EQU     786             ; max number of entries in 40 blocks
0000| 0000 0200            FBLKSIZ EQU     512
0000| 0000 0800            DIRSZ   EQU     4*FBLKSIZ       ; length of dir
0000| 0000 0002            DIRDISK EQU     2               ; block address of directory
0000|                      ;
0000|                      ; directory entry equates
0000|                      ;
0000| 0000 0000            FSTBLK  EQU     0               ; dir entry for DFIRSTBLK
0000| 0000 0002            LSTBLK  EQU     2               ; dir entry for DLSTBLK
0000| 0000 0004            FKIND   EQU     4               ; dir entry for file kind, status
0000|                      ; case fkind = securdir or untypedfile
0000| 0000 0006            DVID    EQU     6               ; dir entry for title field
0000| 0000 000E            DEOVBLK EQU     14              ; dir entry for end of volume field
0000| 0000 0010            DNUMFLS EQU     16              ; dir entry for number files
0000| 0000 0012            DLOADTM EQU     18
0000| 0000 0014            DLASTBT EQU     20              ; most recent date setting
0000|                      ; case fkind = normal files
0000| 0000 0006            DTID    EQU     6               ; dir entry for title field
0000| 0000 0016            LSTBYTE EQU     22              ; dir entry for lastbyte
0000| 0000 0018            DACCESS EQU     24              ; dir entry for date
0000| 0000 001A            DELENG  EQU     26              ; length in bytes of dir entry
0000|                      ;
0000|                      ; close types
0000|                      ;
0000| 0000 0000            CNORMAL EQU     0
0000| 0000 0001            CLOCK   EQU     1
0000| 0000 0002            CPURGE  EQU     2
0000| 0000 0003            CCRUNCH EQU     3
0000|                      ;
0000|                      ; FIB -- file buffer equates
0000|                      ;
0000| 0000 0000            FWINDOW EQU     0
0000| 0000 0004            FEOLN   EQU     FWINDOW+4
0000| 0000 0005            FEOF    EQU     FEOLN+1
0000| 0000 0006            FSTATE  EQU     FEOF+1
0000| 0000 0008            FRECSZ  EQU     FSTATE+2
0000| 0000 000A            FISOPEN EQU     FRECSZ+2
0000| 0000 000B            FISBLKD EQU     FISOPEN+1
0000| 0000 000C            FUNIT   EQU     FISBLKD+1
0000| 0000 000E            FMACHIN EQU     FUNIT+2
```

```
0000| 0000 0016            FMAXBLK EQU     FMACHIN+8
0000| 0000 0018            FNXTBLK EQU     FMAXBLK+2
0000| 0000 001A            FREPCNT EQU     FNXTBLK+2
PAGE -   4  MONITOR   FILE:MONITOR.TEXT


0000| 0000 001C            FMODIFD EQU     FREPCNT+2
0000| 0000 001E            FHEADER EQU     FMODIFD+2
0000| 0000 0038            FSOFTBF EQU     FHEADER+DELENG
0000| 0000 003A            FMAXBYT EQU     FSOFTBF+2
0000| 0000 003C            FNXTBYT EQU     FMAXBYT+2
0000| 0000 003E            FBFCHNG EQU     FNXTBYT+2
0000| 0000 0040            FDEVVID EQU     FBFCHNG+2
0000| 0000 0050            FBUFFER EQU     FDEVVID+16
0000| 0000 0050            FIBSZ   EQU     FBUFFER
0000| 0000 0250            FIBSIZE EQU     FBUFFER+FBLKSIZ
0000|                      ;
0000|                      ; system file table
0000|                      ;
0000| 0000 00D8            FTBLSZ  EQU     9*24            ; size of system filenames table in bytes
0000|                      ;
0000|                      ; unit equates
0000|                      ;
0000| 0000 0001            CONSOLE EQU     1
0000| 0000 0002            SYSTERM EQU     2
0000|                      ;
0000|                      ; UTBL entry equates
0000|                      ;
0000| 0000 0014            MAXU    EQU     20
0000| 0000 0000            UVID    EQU     0
0000| 0000 0008            FREEBYT EQU     UVID+8          ; was logged before dev/vol logic was written
0000| 0000 0009            UISBLKD EQU     FREEBYT+1
0000| 0000 000A            UEOVBLK EQU     UISBLKD+1
0000| 0000 000C            UTBLSIZ EQU     UEOVBLK+2
0000| 0000 00FC            UTABLSZ EQU     <MAXU+1>*UTBLSIZ
0000|                      ;
0000|                      ; prefix equates
0000|                      ;
0000| 0000 0000            P_RLF   .EQU    0
0000| 0000 0001            P_NDFS  .EQU    1
0000| 0000 0002            P_ERSEL .EQU    2
0000| 0000 0003            P_ERSES .EQU    3
0000| 0000 0004            P_HOME  .EQU    4
0000| 0000 0005            P_BCKSP .EQU    5
0000| 0000 0006            P_CLRSC .EQU    6
0000| 0000 0007            P_CLRLN .EQU    7
0000|                      ;
0000|                      ; unittable equates
0000|                      ;
0000| 0000 0001            INBIT   EQU     1
0000| 0000 0002            OUTBIT  EQU     2
0000| 0000 0004            CLRBIT  EQU     4
0000| 0000 0008            BSYBIT  EQU     8
0000| 0000 000F            ALLBIT  EQU     INBIT+OUTBIT+CLRBIT+BSYBIT
0000|                      ;
0000|                      ; devices
0000|                      ;
0000|                      ; 0     working device
0000|                      ; 1     Twiggy 1
0000|                      ; 2     Twiggy 2
0000|                      ; 3     Built in parallel port
PAGE -   5  MONITOR   FILE:MONITOR.TEXT


0000|                      ; 4     Four port card port 0
0000|                      ; 5     Four port card port 1
0000|                      ; 6     Four port card port 2
0000|                      ; 7     Four port card port 3 (Marksman only)
0000|                      ;
0000|                      ; device table equates
0000|                      ;
0000|                      ; NOTE: the device table must be a power of 2 bytes per entry
0000|                      ;
0000| 0000 0007            MAXDEV  EQU     7
0000| 0000 0000            DEVNAME EQU     0
0000| 0000 0010            DEVBASE EQU     DEVNAME+16
0000| 0000 0014            DEVMTBL EQU     DEVBASE+4
0000| 0000 0018            DEVJTBL EQU     DEVMTBL+4
0000| 0000 001C            DEVGLBS EQU     DEVJTBL+4
0000| 0000 0020            DNTRYSZ EQU     DEVGLBS+4
0000| 0000 0005            DEVNSHF EQU     5               ; shift count = DNTRYSZ
0000| 0000 0100            DTBLSIZ EQU     DNTRYSZ*<MAXDEV+1>
0000|                      ;
0000|                      ; LOGN table of logged in units
0000|                      ;
0000| 0000 0004            LOGENSZ EQU     4
0000| 0000 0020            LOGNSZ  EQU     LOGENSZ*<MAXDEV+1>
0000|                      ;
0000|                      ; UT17 table of volume names for devices 1 .. MAXDEV
0000|                      ;
0000| 0000 0000            UT17VID EQU     0
0000| 0000 0008            UT17VSZ EQU     UT17VID+8
0000| 0000 00A8            UT17TSZ EQU     UT17VSZ*<MAXU+1>
0000| 0000 0498            UT17SIZ EQU     UT17TSZ*<MAXDEV>
0000|                      ;
0000|                      ; mount table equates
```

```
0000|                           ;
0000|  0000 0054               MTBLSIZ EQU     4*<MAXU+1>
0000|  0000 024C               MTBLSZ5 EQU     MTBLSIZ*7
0000|                           ;
0000|                           ; boot device set by CPU rom at boot
0000|                           ;
0000|  0000 01B3               BOOTDVC EQU     $1B3
0000|                           ;
0000|                           ; PIA REGISTERS are offsets from $3FF41 (or its equivalent)
0000|                           ;
0000|  0000 0000               INDATA  EQU     $0
0000|  0000 0004               INCSR   EQU     $4
0000|  0000 0002               OUTDATA EQU     $2
0000|  0000 0006               OUTCSR  EQU     $6
0000|                           ;
0000|                           ; PIA REGISTERS for corvus/pippin disk interface
0000|                           ;
0000|  0000 0000               ORB     EQU     0                       ; output regs
0000|  0000 0008               ORA     EQU     8
0000|  0000 0008               IRA     EQU     ORA                     ; input regs
0000|  0000 0000               IRB     EQU     ORB
0000|  0000 0010               DDRB    EQU     $10                     ; data direction regs
0000|  0000 0018               DDRA    EQU     $18
PAGE -   6  MONITOR   FILE:MONITOR.TEXT


0000|  0000 0058               ACR     EQU     $58                     ; aux control
0000|  0000 0060               PCR     EQU     $60                     ; peripheral control
0000|  0000 0068               IFR     EQU     $68                     ; int flags
0000|  0000 0070               IER     EQU     $70                     ; int enable
0000|  0000 0078               NHS     EQU     $78                     ; reg A no hand shake
0000|  0000 0004               CMDSIZE EQU     4                       ; number of bytes in cmd string
0000|  0000 0200               DSKBLK  EQU     512                     ; number of bytes in a block
0000|  0000 0005               PCMNDSZ EQU     5                       ; bytes-1 in pippin cmd string
0000|  0000 00FF               BLKSIZE EQU     255                     ; words-1 in pippin block
0000|  0000 0009               HDRSIZE EQU     9                       ; words-1 in pippin header
0000|                           ;
0000|                           ; OFFSET EQUS for dskread and dskwrt routines
0000|                           ;
0000|  FFFF FFFC               IOCMD   EQU     -4
0000|  FFFF FFFD               IODRV   EQU     -3
0000|  FFFF FFFE               BLOCKL  EQU     -2
0000|  FFFF FFFF               BLOCKH  EQU     -1
0000|                           ;
0000|                           ; OFFSET EQUS for pippin read and write routines
0000|                           ;
0000|  FFFF FFFA               PCMND   EQU     -6
0000|  FFFF FFFB               BLKH    EQU     -5
0000|  FFFF FFFC               BLKM    EQU     -4
0000|  FFFF FFFD               BLKL    EQU     -3
0000|  FFFF FFFE               RETRY   EQU     -2
0000|  FFFF FFFF               THRESH  EQU     -1
0000|                           ;
0000|                           ; globals for the MONITOR
0000|                           ;
0000|  0000 0800               INBUFSZ EQU     $800
0000|                           ;INBUFF  EQU     $1000
0000|  0000 1800               UJMPTBL EQU     $1800
0000|  0000 3000               UJMPTOP EQU     $3000
0000|                           ;
0000|                           ; MACROS FOR ACCESS TO SYSTEM GLOBALS
0000|                           ;
0000|                           ; see MONGLBLS.TEXT complete declarations
0000|                           ;
0000|                           ; SYSCOM: ↑SYSCOMREC;                   (*POINTER TO SYSTEM COMM AREA*)
0000|                           ; EMPTYHEAP: ↑INTEGER;                  (*POINTER TO EMPTY HEAP TOP*)
0000|                           ; HEAPTOP: ↑INTEGER;                    (*POINTER TO CURRENT HEAP TOP*)
0000|                           ; ZFIRST: ↑INTEGER;                     (*POINTER TO LOADER/UNLOADER*)
0000|                           ; INPUTFIB: FIBP;                       (*FILES*)
0000|                           ; OUTPUTFIB: FIBP;
0000|                           ; SYSTERM: FIBP;
0000|                           ; SWAPFIB: FIBP;
0000|                           ; CODEFIB: FIBP;
0000|                           ; SYVID: PVID;                          (*SYSUNIT VOLID*)
0000|                           ; FREE4A: ↑INTEGER;                     (*USED BY MS*)
0000|                           ; DKVID: PVID;                          (*DEFAULT VOLID*)
0000|                           ; FREE4B: ↑INTEGER;                     (*USED BY MS*)
0000|                           ; THEDATE: DATEREC;                     (*TODAY...SET IN FILER OR SIGN ON*)
0000|                           ; FILLER: STRING[FILLRLN];              (*NULLS FOR CARRIAGE DELAY*)
0000|                           ; UTABLE: ↑UNITABL;                     (*POINTER TO UNIT TABLE FOR VOLUMES*)
0000|                           ; FILETBLP: ↑FILETABLE;                 (*POINTER TO TABLE SYSTEM FILENAMES*)
PAGE -   7  MONITOR   FILE:MONITOR.TEXT


0000|                           ; STACKPTR: ↑INTEGER;                   (*POINTER TO STACK BASE*)
0000|                           ; INBUFFP: ↑INTEGER;                    (*POINTER TO USER CODE BUFFER*)
0000|                           ; STRBUF: ↑STRING[255];                 (*POINTER TO ASM LANG STRING BUFFER*)
0000|                           ; DBUGFLAG: BOOLEAN;                    (*XEQUTE UNDER DEBUG FLAG*)
0000|                           ; RUNUSER: BOOLEAN;                     (*USER/SYM DEBUGGER STATUS FLAG*)
0000|                           ; USERFIB: FIBP;                        (*POINTER TO USER OBJ FILE BUFFER*)
0000|                           ; ERRP: ↑INTEGER;                       (*POINTER SYM DEBUGGER ERROR HANDLER*)
0000|                           ; UJPT: ↑INTEGER;                       (*POINTER USERS JUMP TABLE*)
0000|                           ; UREG: ↑INTEGER;                       (*POINTER REG SAVE AREA FOR USER*)
0000|                           ; DREG: ↑INTEGER;                       (*POINTER REG SAVE AREA FOR DEBUGGER*)
0000|                           ;
0000|                                   .MACRO  PUTSYSC                 ; put into pointer to system comm area
```

```
0000|                           MOVE.L  %1,-4(A5)
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  GETSYSC                 ; get from pointer to system comm area
0000|                           MOVE.L  -4(A5),%1
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  PUTEMTH                 ; put into empty heap pointer
0000|                           MOVE.L  %1,-8(A5)
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  GETEMTH                 ; get from empty heap pointer
0000|                           MOVE.L  -8(A5),%1
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  ADDRNP                  ; @pointer to heap top
0000|                           LEA     -12(A5),%1
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  PUTNP                   ; put into pointer to heap top
0000|                           MOVE.L  %1,-12(A5)
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  GETNP                   ; get from pointer to heap top
0000|                           MOVE.L  -12(A5),%1
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  PUTZFST                 ; put into load/unload pointer
0000|                           MOVE.L  %1,-16(A5)
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  GETZFST                 ; get from load/unload pointer
0000|                           MOVE.L  -16(A5),%1
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  PUTIFIB                 ; put into input fib pointer
0000|                           MOVE.L  %1,-20(A5)
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  GETIFIB                 ; get from input fib pointer
0000|                           MOVE.L  -20(A5),%1
0000|                           .ENDM
0000|                   ;
PAGE -   8  MONITOR   FILE:MONITOR.TEXT


0000|                           .MACRO  PUTOFIB                 ; put into output fib pointer
0000|                           MOVE.L  %1,-24(A5)
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  GETOFIB                 ; get from output fib pointer
0000|                           MOVE.L  -24(A5),%1
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  PUTTFIB                 ; put into systerm fib pointer
0000|                           MOVE.L  %1,-28(A5)
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  GETTFIB                 ; get from systerm fib pointer
0000|                           MOVE.L  -28(A5),%1
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  PUTSFIB                 ; put into swap fib pointer
0000|                           MOVE.L  %1,-32(A5)
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  GETSFIB                 ; get from swap fib pointer
0000|                           MOVE.L  -32(A5),%1
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  PUTCFIB                 ; put into code fib pointer
0000|                           MOVE.L  %1,-36(A5)
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  GETCFIB                 ; get from code fib pointer
0000|                           MOVE.L  -36(A5),%1
0000|                           .ENDM
0000|                   ;
0000|                   ;       -40 USED BY MS.OBJ FOR CRT INFO
0000|                   ;
0000|                           .MACRO  GETSYV                  ; get pointer to sysunit volid
0000|                           MOVE.L  -44(A5),%1
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  PUTSYV                  ; put pointer to sysunit volid
0000|                           MOVE.L  %1,-44(A5)
0000|                           .ENDM
0000|                   ;
0000|                   ;       -48 USED BY MS.OBJ FOR UPPER/LOWER TOGGLE, 3 BYTES FREE
0000|                   ;
0000|                           .MACRO  GETDKV                  ; get pointer to default volid
0000|                           MOVE.L  -52(A5),%1
0000|                           .ENDM
0000|                   ;
0000|                           .MACRO  PUTDKV                  ; put pointer to default volid
0000|                           MOVE.L  %1,-52(A5)
```

```
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  ADDRTHD                 ; pointer to THEDATE word
0000|                               LEA     -54(A5),%1
0000|                               .ENDM
PAGE -   9  MONITOR   FILE:MONITOR.TEXT


0000|                       ;
0000|                               .MACRO  ADDRFLR                 ; pointer to FILLER string
0000|                               LEA     -66(A5),%1
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  PUTUTBL                 ; put into unit table pointer
0000|                               MOVE.L  %1,-70(A5)
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  GETUTBL                 ; get from unit table pointer
0000|                               MOVE.L  -70(A5),%1
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  PUTSTBL                 ; put into sys fnames table pointer
0000|                               MOVE.L  %1,-74(A5)
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  GETSTBL                 ; get from sys fnames table pointer
0000|                               MOVE.L  -74(A5),%1
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  PUTSPTR                 ; put into stack pointer base
0000|                               MOVE.L  %1,-78(A5)
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  GETSPTR                 ; get from stack pointer base
0000|                               MOVE.L  -78(A5),%1
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  PUTIBF                  ; put into inbuff pointer
0000|                               MOVE.L  %1,-82(A5)
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  GETIBF                  ; get from inbuff pointer
0000|                               MOVE.L  -82(A5),%1
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  PUTSTRB                 ; put into string buffer pointer
0000|                               MOVE.L  %1,-86(A5)
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  GETSTRB                 ; get from string buffer pointer
0000|                               MOVE.L  -86(A5),%1
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  DBUGSET                 ; set debug flag
0000|                               MOVE.B  #1,-87(A5)
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  DBUGCLR                 ; clear debug flag
0000|                               MOVE.B  #0,-87(A5)
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  DBUGTST                 ; test debug flag
0000|                               TST.B   -87(A5)
PAGE -  10  MONITOR   FILE:MONITOR.TEXT


0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  RUNUSET                 ; set runuser flag
0000|                               MOVE.B  #1,-88(A5)
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  RUNUCLR                 ; clear runuser flag
0000|                               MOVE.B  #0,-88(A5)
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  RUNUTST                 ; test runuser flag
0000|                               TST.B   -88(A5)
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  PUTUFIB                 ; put into pointer to user obj fib
0000|                               MOVE.L  %1,-92(A5)
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  GETUFIB                 ; get from pointer to user obj fib
0000|                               MOVE.L  -92(A5),%1
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  PUTERRP                 ; put into pointer to error proc
0000|                               MOVE.L  %1,-96(A5)
0000|                               .ENDM
0000|                       ;
0000|                               .MACRO  GETERRP                 ; get from pointer to error proc
0000|                               MOVE.L  -96(A5),%1
0000|                               .ENDM
0000|                       ;
```

```
0000                            .MACRO  PUTUJPT                 ; put into pointer to user JT
0000                            MOVE.L  %1,-100(A5)
0000                            .ENDM
0000                    ;
0000                            .MACRO  GETUJPT                 ; get from pointer to user JT
0000                            MOVE.L  -100(A5),%1
0000                            .ENDM
0000                    ;
0000                            .MACRO  PUTUREG                 ; put into pointer to user regs
0000                            MOVE.L  %1,-104(A5)
0000                            .ENDM
0000                    ;
0000                            .MACRO  GETUREG                 ; get from pointer to user regs
0000                            MOVE.L  -104(A5),%1
0000                            .ENDM
0000                    ;
0000                            .MACRO  PUTDREG                 ; put into pointer to sym debug regs
0000                            MOVE.L  %1,-108(A5)
0000                            .ENDM
0000                    ;
0000                            .MACRO  GETDREG                 ; get from pointer to sym debug regs
0000                            MOVE.L  -108(A5),%1
0000                            .ENDM
0000                    ;
0000                            .MACRO  PUTUFST                 ; put into user load/unload ptr
PAGE -  11  MONITOR   FILE:MONITOR.TEXT


0000                            MOVE.L  %1,-112(A5)
0000                            .ENDM
0000                    ;
0000                            .MACRO  GETUFST                 ; get from user load/unload ptr
0000                            MOVE.L  -112(A5),%1
0000                            .ENDM
0000                    ;
0000                            .MACRO  PUTACCA                 ; put into access record ptr
0000                            MOVE.L  %1,-116(A5)
0000                            .ENDM
0000                    ;
0000                            .MACRO  GETACCA                 ; get from access record ptr
0000                            MOVE.L  -116(A5),%1
0000                            .ENDM
0000                    ;
0000                            .MACRO  PUTDTBL                 ; put into device table ptr
0000                            MOVE.L  %1,-120(A5)
0000                            .ENDM
0000                    ;
0000                            .MACRO  GETDTBL                 ; get from device table ptr
0000                            MOVE.L  -120(A5),%1
0000                            .ENDM
0000                    ;
0000                            .MACRO  PUTEBST                 ; put into ebstop
0000                            MOVE.B  %1,-121(A5)
0000                            .ENDM
0000                    ;
0000                            .MACRO  GETEBST                 ; get from ebstop
0000                            MOVE.B  -121(A5),%1
0000                            .ENDM
0000                    ;
0000                            .MACRO  PUTNXCH                 ; put into next exec char
0000                            MOVE.B  %1,-122(A5)
0000                            .ENDM
0000                    ;
0000                            .MACRO  GETNXCH                 ; get from next exec char
0000                            MOVE.B  -122(A5),%1
0000                            .ENDM
0000                    ;
0000                            .MACRO  PUTXFIB                 ; put into execfile fib
0000                            MOVE.L  %1,-126(A5)
0000                            .ENDM
0000                    ;
0000                            .MACRO  GETXFIB                 ; get from execfile fib
0000                            MOVE.L  -126(A5),%1
0000                            .ENDM
0000                    ;
0000                            .MACRO  PUTDTOW                 ; put into disk timeout word
0000                            MOVE.W  %1,-128(A5)
0000                            .ENDM
0000                    ;
0000                            .MACRO  DECDTOW                 ; decr the disk timeout word
0000                            SUB.W   #1,-128(A5)
0000                            .ENDM
0000                    ;
PAGE -  12  MONITOR   FILE:MONITOR.TEXT


0000                            .MACRO  ADDRSAT                 ; ptr reg save area top (A3-A4/D4-D7)
0000                            LEA     -132(A5),%1
0000                            .ENDM
0000                    ;
0000                            .MACRO  ADDRSAB                 ; ptr reg save area bottom (A3-A4/D4-D7)
0000                            LEA     -156(A5),%1
0000                            .ENDM
0000                    ;
0000                            .MACRO  PUTLOGN                 ; put into LOGN ptr
0000                            MOVE.L  %1,-160(A5)
```

```
0000|                                   .ENDM
0000|                           ;
0000|                                   .MACRO  GETLOGN                 ; get from LOGN ptr
0000|                                   MOVE.L  -160(A5),%1
0000|                                   .ENDM
0000|                           ;
0000|                                   .MACRO  PUTUT17                 ; put into UT17 ptr
0000|                                   MOVE.L  %1,-164(A5)
0000|                                   .ENDM
0000|                           ;
0000|                                   .MACRO  GETUT17                 ; get from UT17 ptr
0000|                                   MOVE.L  -164(A5),%1
0000|                                   .ENDM
0000|                           ;
0000|                                   .MACRO  PUTPRNT                 ; put ptr to printer driver
0000|                                   MOVE.L  %1,-168(A5)
0000|                                   .ENDM
0000|                           ;
0000|                                   .MACRO  GETPRNT                 ; get ptr to printer driver
0000|                                   MOVE.L  -168(A5),%1
0000|                                   .ENDM
0000|                           ;
0000|                                   .MACRO  PUTPRBA                 ; put ptr to printer driver
0000|                                   MOVE.L  %1,-172(A5)
0000|                                   .ENDM
0000|                           ;
0000|                                   .MACRO  GETPRBA                 ; get ptr to printer driver
0000|                                   MOVE.L  -172(A5),%1
0000|                                   .ENDM
0000|                           ;
0000|                           ;   Macro TrapTo is used to call the hardware interface routines.
0000|                           ;   Parameter %1 is the magic number associated with the routine begin called.
0000|                           ;
0000|                                   .MACRO  TRAPTO                  ; trap to hardware interface
0000|                                   MOVE.L  D7,-(SP)                ; save register
0000|                                   MOVE.W  #%1,D7                  ; routine number
0000|                                   TRAP    #5                      ; trap
0000|                                   MOVE.L  (SP)+,D7                ; restore register
0000|                                   .ENDM
0000|                           ;
0000|                           ;       $C00 thru $C07 used by IU Trap Handler
0000|                           ;
0000|                           ; start of monitor
0000|                           ;
0000|                                   RORG    0
PAGE -  13  MONITOR   FILE:MONITOR.TEXT


0000|                           ;
0000|                                   .INCLUDE MONBEGN.TEXT
0000|                           ;
0000|                           ;       FILENAME MONBEGN
0000|                           ;
0000|
0000| 6000 ****                 MONBASE BRA     INITIAL
0004|                           ;
0004|                           ; intial values for syscom
0004|                           ;
0004| 00                        TOPSYSC .BYTE   0               ; IORSLT
0005| 05                                .BYTE   5               ; SYSUNT
0006| 0000                              .WORD   0               ; HITIME
0008| 0000                              .WORD   0               ; LOTIME
000A| 000D                      OFSMISC .WORD   $000D           ; MSCNFO
000C| 0003                              .WORD   $0003           ; CRTTYP
000E| 1E1B 5459                        .WORD   $1E1B,$5459     ; CRTESC
0012| 0B0C 0008                        .WORD   $0B0C,$0008
0016| 2A00 004C                        .WORD   $2A00,$004C     ; CLRLINE & CLRSCRN
001A| 0018 0050                        .WORD   $0018,$0050     ; CRTINFO
001E| 0A0B 0C08                        .WORD   $0A0B,$0C08
0022| 0603                              .WORD   $0603           ; EOF & FLUSH
0024| 1300                              .WORD   $1300           ; BREAK & SSTOP
0026| 3F08                              .WORD   $3F08           ; CHRDEL & BADCHR
0028| 1B7F                              .WORD   $1B7F           ; LINDEL & ALTMOD
002A| 0300                              .WORD   $0300           ; PREFIX & ETX
002C| 0800                              .WORD   $0800           ; BKSPACE
002E| 0000                              .WORD   0
0030| 0000                      BOTSYSC .WORD   0
0032|                           ;
0032|                           ; initial values for unit table
0032|                           ;
0032| 00                        TOPUTBL .BYTE   0               ; unit #0
0033| 20 20 20 20 20 20 20              .ASCII  '       '
003A| 00 00                             .BYTE   0,0
003C| 0000                              .WORD   0
003E| 07                                .BYTE   7               ; unit #1
003F| 43 4F 4E 53 4F 4C 45             .ASCII  'CONSOLE'
0046| 01 00                             .BYTE   1,0
0048| 0000                              .WORD   0
004A| 07                                .BYTE   7               ; unit #2
004B| 53 59 53 54 45 52 4D             .ASCII  'SYSTERM'
0052| 01 00                             .BYTE   1,0
0054| 0000                              .WORD   0
0056| 00                                .BYTE   0               ; unit #3
0057| 20 20 20 20 20 20 20              .ASCII  '       '
005E| 00 00                             .BYTE   0,0
0060| 0000                              .WORD   0
```

```
0062| 00                              .BYTE   0               ; unit #4
0063| 20 20 20 20 20 20 20           .ASCII  '       '
006A| 00 01                           .BYTE   0,1
006C| 0000                           .WORD   0
006E| 00                              .BYTE   0               ; unit #5
006F| 20 20 20 20 20 20 20           .ASCII  '       '
0076| 00 01                           .BYTE   0,1
PAGE -  14  MONITOR   FILE: MONBEGN.TEXT


0078| 0000                           .WORD   0
007A| 07                              .BYTE   7               ; unit #6
007B| 50 52 49 4E 54 45 52           .ASCII  'PRINTER'
0082| 00 00                           .BYTE   0,0
0084| 0000                           .WORD   0
0086| 00                              .BYTE   0               ; unit #7
0087| 20 20 20 20 20 20 20           .ASCII  '       '
008E| 00 00                           .BYTE   0,0
0090| 0000                           .WORD   0
0092| 00                              .BYTE   0               ; unit #8
0093| 20 20 20 20 20 20 20           .ASCII  '       '
009A| 00 00                           .BYTE   0,0
009C| 0000                           .WORD   0
009E| 00                              .BYTE   0               ; unit #9
009F| 20 20 20 20 20 20 20           .ASCII  '       '
00A6| 00 01                           .BYTE   0,1
00A8| 0000                           .WORD   0
00AA| 00                              .BYTE   0               ; unit #10
00AB| 20 20 20 20 20 20 20           .ASCII  '       '
00B2| 00 01                           .BYTE   0,1
00B4| 0000                           .WORD   0
00B6| 00                              .BYTE   0               ; unit #11
00B7| 20 20 20 20 20 20 20           .ASCII  '       '
00BE| 00 01                           .BYTE   0,1
00C0| 0000                           .WORD   0
00C2| 00                              .BYTE   0               ; unit #12
00C3| 20 20 20 20 20 20 20           .ASCII  '       '
00CA| 00 01                           .BYTE   0,1
00CC| 0000                           .WORD   0
00CE| 00                              .BYTE   0               ; unit #13
00CF| 20 20 20 20 20 20 20           .ASCII  '       '
00D6| 00 01                           .BYTE   0,1
00D8| 0000                           .WORD   0
00DA| 00                              .BYTE   0               ; unit #14
00DB| 20 20 20 20 20 20 20           .ASCII  '       '
00E2| 00 01                           .BYTE   0,1
00E4| 0000                           .WORD   0
00E6| 00                              .BYTE   0               ; unit #15
00E7| 20 20 20 20 20 20 20           .ASCII  '       '
00EE| 00 01                           .BYTE   0,1
00F0| 0000                           .WORD   0
00F2| 00                              .BYTE   0               ; unit #16
00F3| 20 20 20 20 20 20 20           .ASCII  '       '
00FA| 00 01                           .BYTE   0,1
00FC| 0000                           .WORD   0
00FE| 00                              .BYTE   0               ; unit #17
00FF| 20 20 20 20 20 20 20           .ASCII  '       '
0106| 00 01                           .BYTE   0,1
0108| 0000                           .WORD   0
010A| 00                              .BYTE   0               ; unit #18
010B| 20 20 20 20 20 20 20           .ASCII  '       '
0112| 00 01                           .BYTE   0,1
0114| 0000                           .WORD   0
0116| 00                              .BYTE   0               ; unit #19
0117| 20 20 20 20 20 20 20           .ASCII  '       '
PAGE -  15  MONITOR   FILE: MONBEGN.TEXT


011E| 00 01                           .BYTE   0,1
0120| 0000                           .WORD   0
0122| 00                              .BYTE   0               ; unit #20
0123| 20 20 20 20 20 20 20           .ASCII  '       '
012A| 00 01                           .BYTE   0,1
012C| 0000                           .WORD   0
012E| 0000                  BOTUTBL  .WORD   0
0130|                         ;
0130|                         ; initial values for system filenames
0130|                         ;
0130| 0E                    TOPSYSF  .BYTE   14
0131| 2A 41 53 53 45 4D 42           .ASCII  '*ASSEMBLER.OBJ         '
0138| 4C 45 52 2E 4F 42 4A
013F| 20 20 20 20 20 20 20
0146| 20 20
0148| 0D                              .BYTE   13
0149| 2A 43 4F 4D 50 49 4C           .ASCII  '*COMPILER.OBJ          '
0150| 45 52 2E 4F 42 4A 20
0157| 20 20 20 20 20 20 20
015E| 20 20
0160| 0B                              .BYTE   11
0161| 2A 45 44 49 54 4F 52           .ASCII  '*EDITOR.OBJ            '
0168| 2E 4F 42 4A 20 20 20
016F| 20 20 20 20 20 20 20
0176| 20 20
0178| 0A                              .BYTE   10
0179| 2A 46 49 4C 45 52 2E           .ASCII  '*FILER.OBJ             '
0180| 4F 42 4A 20 20 20 20
```

```
0187| 20 20 20 20 20 20 20
018E| 20 20
0190| 0B                              .BYTE   11
0191| 2A 4C 49 4E 4B 45 52            .ASCII  '*LINKER.OBJ          '
0198| 2E 4F 42 4A 20 20 20
019F| 20 20 20 20 20 20 20
01A6| 20 20
01A8| 0D                              .BYTE   13
01A9| 2A 44 45 42 55 47 47            .ASCII  '*DEBUGGER.OBJ        '
01B0| 45 52 2E 4F 42 4A 20
01B7| 20 20 20 20 20 20 20
01BE| 20 20
01C0| 0F                              .BYTE   15
01C1| 2A 55 43 53 44 45 44            .ASCII  '*UCSDEDITOR.OBJ      '
01C8| 49 54 4F 52 2E 4F 42
01CF| 4A 20 20 20 20 20 20
01D6| 20 20
01D8| 09                              .BYTE   9
01D9| 2A 43 4F 44 45 2E 4F            .ASCII  '*CODE.OBJ            '
01E0| 42 4A 20 20 20 20 20
01E7| 20 20 20 20 20 20 20
01EE| 20 20
01F0| 0B                              .BYTE   11
01F1| 2A 53 59 53 4D 47 52            .ASCII  '*SYSMGR.OBJ          '
01F8| 2E 4F 42 4A 20 20 20
01FF| 20 20 20 20 20 20 20
0206| 20 20
PAGE -  16  MONITOR   FILE: MONBEGN.TEXT


0208| 0000                  BOTSYSF .WORD   0
020A|                       ;
020A|                       ;       initialzation code starts here
020A|                       ;
0002* 0208
020A| 46FC 2000             INITIAL MOVE.W  #$2000,SR               ; turn on interrupts
020E| 2E78 013C                     MOVE.L  $13C,A7                 ; set up default SP
0212| 207C 0000 0100               MOVE.L  #$00000100,A0
0218| 43FA ****                     LEA     UNITBL,A1
021C| 20C9                          MOVE.L  A1,(A0)+                ; start of monitor
021E| 43FA ****                     LEA     THEEND,A1
0222| 20C9                          MOVE.L  A1,(A0)+                ; end of monitor
0224| 6100 ****                     BSR     SETUPA5
0228|                               PUTSYSC #$00000D00              ; set up ptr to syscom
0228| 2B7C 0000 0D00 FFFC   #       MOVE.L   #$00000D00,-4(A5)
0230|                       ;
0230|                       ;       initialize syscom area
0230|                       ;
0230|                               GETSYSC A1
0230| 226D FFFC             #       MOVE.L   -4(A5),A1
0234| 41FA FDCE                     LEA     TOPSYSC,A0
0238| 303C 002C                     MOVE.W  #BOTSYSC-TOPSYSC,D0
023C| 12D8                  INITSYS MOVE.B  (A0)+,(A1)+             ; copy default into syscom area
023E| 5340                          SUB.W   #1,D0
0240| 66FA                          BNE.S   INITSYS
0242|                       ;
0242|                       ;       allocate globals on the heap
0242|                       ;
0242| 2078 0138                     MOVE.L  $138,A0                 ; get bottom of heap
0246|                               PUTIBF  A0
0246| 2B48 FFAE             #       MOVE.L   A0,-82(A5)
024A| D0FC 0800                     ADD.W   #INBUFSZ,A0             ; allocate INBUFF
024E|                               PUTIFIB A0
024E| 2B48 FFEC             #       MOVE.L   A0,-20(A5)
0252| D0FC 0250                     ADD.W   #FIBSIZE,A0             ; allocate INPUTFIB
0256|                               PUTOFIB A0
0256| 2B48 FFE8             #       MOVE.L   A0,-24(A5)
025A| D0FC 0250                     ADD.W   #FIBSIZE,A0             ; allocate OUTPUTFIB
025E|                               PUTTFIB A0
025E| 2B48 FFE4             #       MOVE.L   A0,-28(A5)
0262| D0FC 0250                     ADD.W   #FIBSIZE,A0             ; allocate SYSTERM FIB
0266|                               PUTSFIB A0
0266| 2B48 FFE0             #       MOVE.L   A0,-32(A5)
026A| D0FC 0050                     ADD.W   #FIBSZ,A0               ; allocate SWAPFIB
026E|                               PUTCFIB A0
026E| 2B48 FFDC             #       MOVE.L   A0,-36(A5)
0272| D0FC 0050                     ADD.W   #FIBSZ,A0               ; allocate CODEFIB
0276|                               PUTSTBL A0
0276| 2B48 FFB6             #       MOVE.L   A0,-74(A5)
027A| D0FC 00D8                     ADD.W   #FTBLSZ,A0              ; allocate sys fnames table
027E|                               PUTDREG A0
027E| 2B48 FF94             #       MOVE.L   A0,-108(A5)
0282| D0FC 0048                     ADD.W   #72,A0                  ; allocate sym debuggers reg save area
0286|                               PUTUREG A0
0286| 2B48 FF98             #       MOVE.L   A0,-104(A5)
PAGE -  17  MONITOR   FILE: MONBEGN.TEXT


028A| D0FC 0048                     ADD.W   #72,A0                  ; allocate users reg save area
028E|                               PUTACCA A0
028E| 2B48 FF8C             #       MOVE.L   A0,-116(A5)
0292| 5048                          ADD.W   #8,A0                   ; allocate access record
0294|                               PUTDTBL A0
0294| 2B48 FF88             #       MOVE.L   A0,-120(A5)
0298| D0FC 034C                     ADD.W   #DTBLSIZ+MTBLSZ5,A0     ; allocate device and mount tables
029C|                               PUTLOGN A0
```

```
029C| 2B48 FF60            #       MOVE.L   A0,-160(A5)
02A0| D0FC 0020                    ADD.W    #LOGNSZ,A0              ; allocate LOGN bitmap
02A4|                             PUTUT17  A0
02A4| 2B48 FF5C            #       MOVE.L   A0,-164(A5)
02A8| D0FC 0498                    ADD.W    #UT17SIZ,A0             ; allocate UT17 volume name table
02AC|                             PUTSYV   A0
02AC| 2B48 FFD4            #       MOVE.L   A0,-44(A5)
02B0| D0FC 0018                    ADD.W    #24,A0                  ; allocate sysunit volume id
02B4|                             PUTDKV   A0
02B4| 2B48 FFCC            #       MOVE.L   A0,-52(A5)
02B8| D0FC 0018                    ADD.W    #24,A0                  ; allocate prefix volume id
02BC|                             PUTXFIB  A0
02BC| 2B48 FF82            #       MOVE.L   A0,-126(A5)
02C0| D0FC 0250                    ADD.W    #FIBSIZE,A0             ; allocate exec file fib
02C4|                             PUTUTBL  A0
02C4| 2B48 FFBA            #       MOVE.L   A0,-70(A5)
02C8| D0FC 00FC                    ADD.W    #UTABLSZ,A0             ; allocate unit table
02CC| 2438 0144                    MOVE.L   $144,D2
02D0| 67**                        BEQ.S    @1                      ; if sizeof(memory) > 0 then
02D2| 21C8 0144                    MOVE.L   A0,$144
02D6| 2202                        MOVE.L   D2,D1                   ; allocate MEMORY:
02D8| C3FC 0200                    MULS     #FBLKSIZ,D1
02DC| D1C1                        ADD.L    D1,A0
02DE|                      @1     PUTNP    A0                      ; save new heap top in NP
02D0* 0C
02DE| 2B48 FFF4            #       MOVE.L   A0,-12(A5)
02E2|                             PUTEMTH  A0                      ; save NP as empty heap
02E2| 2B48 FFF8            #       MOVE.L   A0,-8(A5)
02E6|                             PUTSPTR  A7                      ; save SP as default
02E6| 2B4F FFB2            #       MOVE.L   A7,-78(A5)
02EA| 3F02                        MOVE.W   D2,-(A7)                ; save memory size on top of stack
02EC|                      ;
02EC|                      ;       initialize syvid and dkvid
02EC|                      ;
02EC|                             GETSYV   A0
02EC| 206D FFD4            #       MOVE.L   -44(A5),A0
02F0| 4210                        MOVE.B   #0,(A0)                 ; intialize sysunit volid to ''
02F2|                             GETDKV   A0
02F2| 206D FFCC            #       MOVE.L   -52(A5),A0
02F6| 4210                        MOVE.B   #0,(A0)                 ; intialize default volid to ''
02F8|                      ;
02F8|                      ;       initialize those with fixed allocation
02F8|                      ;
02F8|                             PUTSTRB  #$00000E00              ; save ptr to STRBUF
02F8| 2B7C 0000 0E00 FFAA  #       MOVE.L   #$00000E00,-86(A5)
0300|                             ADDRTHD  A0
0300| 41ED FFCA            #       LEA      -54(A5),A0
PAGE -  18  MONITOR   FILE: MONBEGN.TEXT


0304| 30BC A4E5                    MOVE.W   #$A4E5,(A0)             ; initialize the DATE to May 14, 1982
0308|                             ADDRFLR  A0
0308| 41ED FFBE            #       LEA      -66(A5),A0
030C| 4210                        MOVE.B   #0,(A0)                 ; initialize FILLER
030E|                             PUTEBST  #0                      ; initialize ebstop to nul
030E| 422D FF87            #       MOVE.B   #0,-121(A5)
0312|                             PUTNXCH  #0                      ; initialize next exec char to nul
0312| 422D FF86            #       MOVE.B   #0,-122(A5)
0316|                      ;
0316|                      ;       set up pointer to printer driver
0316|                      ;
0316| 41FA ****                    LEA      PRTDRVR,A0
031A|                             PUTPRNT  A0
031A| 2B48 FF58            #       MOVE.L   A0,-168(A5)
031E|                      ;
031E|                      ;       set up trap $E to point to trap handler
031E|                      ;
031E| 41FA ****                    LEA      HANDLER,A0
0322| 21C8 00B8                    MOVE.L   A0,$B8                  ; initialize TRAP #$E
0326|                      ;
0326|                      ;       set up trap $D to point to trap handler
0326|                      ;
0326| 41FA ****                    LEA      TRP2MAX,A0
032A| 21C8 00B4                    MOVE.L   A0,$B4                  ; initialize TRAP #$D
032E|                      ;
032E|                      ;       set up trap $C to point to trap handler
032E|                      ;
032E| 41FA ****                    LEA      SOFTBPT,A0
0332| 21C8 00B0                    MOVE.L   A0,$B0                  ; initialize TRAP #$C
0336|                      ;
0336|                      ;       set up CHK exception to point to trap handler
0336|                      ;
0336| 41FA ****                    LEA      CHKERR,A0
033A| 21C8 0018                    MOVE.L   A0,$18                  ; initialize CHK exception
033E|                      ;
033E|                      ;       save zero page at $F00
033E|                      ;
033E| 91C8                        MOVE.L   #$00000000,A0
0340| 227C 0000 0F00              MOVE.L   #$00000F00,A1
0346| 303C 003F                    MOVE.W   #63,D0                  ; zero page length is 64 longs
034A| 22D8               MOVE0P   MOVE.L   (A0)+,(A1)+
034C| 51C8 FFFC                    DBF      D0,MOVE0P
0350|                      ;
0350|                      ;       initialize unit table
0350|                      ;
0350|                             GETUTBL  A1
```

```
0350| 226D FFBA              #         MOVE.L   -70(A5),A1
0354| 41FA FCDC                       LEA      TOPUTBL,A0
0358| 303C 00FC                       MOVE.W   #BOTUTBL-TOPUTBL,D0
035C|                        INITUTBL
035C| 12D8                           MOVE.B   (A0)+,(A1)+
035E| 5340                           SUB.W    #1,D0
0360| 66FA                           BNE.S    INITUTBL
0362|                        ;
0362|                        ;        initialize system filenames table
PAGE -  19  MONITOR   FILE: MONBEGN.TEXT


0362|                        ;
0362|                                GETSTBL A1
0362| 226D FFB6              #        MOVE.L   -74(A5),A1
0366| 41FA FDC8                       LEA      TOPSYSF,A0
036A| 303C 00D8                       MOVE.W   #BOTSYSF-TOPSYSF,D0
036E|                        INITSYSF
036E| 12D8                           MOVE.B   (A0)+,(A1)+
0370| 5340                           SUB.W    #1,D0
0372| 66FA                           BNE.S    INITSYSF
0374|                        ;
0374|                        ;        zero #4: (ie. MEMORY:)
0374|                        ;
0374|                        ;        note the memory size in blocks has already been pushed on top of stack
0374|                        ;
0374| 6100 ****                      BSR      ZEROMEM
0378|                        ;
0378|                        ;        initialize the device table
0378|                        ;
0378| 6100 ****                      BSR      INITDEV
037C|                        ;
037C|                        ;        test for disks, setup device table and setup the mount table
037C|                        ;
037C| 4AB8 0148                      TST.L    $148
0380| 67**                           BEQ.S    NOTWGDR
0382|                        ;
0382| 1F3C 0001                      MOVE.B   #1,-(A7)
0386| 3F3C 0002                      MOVE.W   #2,-(A7)
038A| 6100 ****                      BSR      RDMTABL
038E|                        ;
038E| 1F3C 0001                      MOVE.B   #1,-(A7)
0392| 3F3C 0001                      MOVE.W   #1,-(A7)
0396| 6100 ****                      BSR      RDMTABL
0380* 18
039A|                        NOTWGDR
039A| 1F3C 0001                      MOVE.B   #1,-(A7)
039E| 3F3C 0003                      MOVE.W   #3,-(A7)
03A2| 6100 ****                      BSR      RDMTABL
03A6|                        ;
03A6| 1F3C 0001                      MOVE.B   #1,-(A7)
03AA| 3F3C 0007                      MOVE.W   #7,-(A7)
03AE| 6100 ****                      BSR      RDMTABL
03B2|                        ;
03B2| 1F3C 0001                      MOVE.B   #1,-(A7)
03B6| 3F3C 0006                      MOVE.W   #6,-(A7)
03BA| 6100 ****                      BSR      RDMTABL
03BE|                        ;
03BE| 1F3C 0001                      MOVE.B   #1,-(A7)
03C2| 3F3C 0005                      MOVE.W   #5,-(A7)
03C6| 6100 ****                      BSR      RDMTABL
03CA|                        ;
03CA| 1F3C 0001                      MOVE.B   #1,-(A7)
03CE| 3F3C 0004                      MOVE.W   #4,-(A7)
03D2| 6100 ****                      BSR      RDMTABL
03D6|                        ;
03D6|                        ;        setup 60hz handler for automatic unmount of profile
PAGE -  20  MONITOR   FILE: MONBEGN.TEXT


03D6|                        ;
03D6| 4E71                           NOP
03D8|                        ;
03D8|                        ;        get root volume directory, then init syvid, dkvid, thedate
03D8|                        ;
03D8| 6100 ****                      BSR      GETRDIR
03DC|                        ;
03DC|                        ;        initialize files
03DC|                        ;
03DC| 6100 ****                      BSR      INITFLS
03E0|                        ;
03E0|                        ;        update SYSCOM area with MON.MISCINFO
03E0|                        ;
03E0| 6100 ****                      BSR      INITMI
03E4|                        ;
03E4|                        ;        try to execute MON.STARTUP
03E4|                        ;
03E4| 6100 ****                      BSR      XSTRTUP
03E8|                        ;
03E8|                        ;        print welcome message and startup monitor at the command line
03E8|                        ;
03E8| 6000 ****                      BRA      WELCOME
03EC|                        ;
03EC| 08                     FIL1STR .BYTE    8
03ED| 43 4F 4E 53 4F 4C 45        .ASCII   'CONSOLE: '
03F4| 3A 20
```

```
03F6|                          ;
03F6| 08                       FIL2STR .BYTE   8
03F7| 53 59 53 54 45 52 4D             .ASCII  'SYSTERM: '
03FE| 3A 20
0400|                          ;
0400|                          ;       INITIALIZE FILES
0400|                          ;
03DE* 0022
0400|                          INITFLS
0400|                          ;
0400|                          ;       initialize INPUTFIB
0400|                          ;
0400|                                  GETIFIB A1                      ; @file
0400| 226D FFEC                #       MOVE.L   -20(A5),A1
0404| 2F09                             MOVE.L  A1,-(A7)
0406| 4869 0050                        PEA     FBUFFER(A1)
040A| 4267                             MOVE.W  #0,-(A7)
040C| 6100 ****                       BSR     FINIT                   ; initialize fib
0410|                                  GETIFIB -(A7)                   ; @file
0410| 2F2D FFEC                #       MOVE.L   -20(A5),-(A7)
0414| 487A FFD6                        PEA     FIL1STR                 ; @title
0418| 3F3C 0001                        MOVE.W  #1,-(A7)                ; openold:=true
041C| 42A7                             CLR.L   -(A7)                   ; junk
041E| 6100 ****                       BSR     FOPEN
0422|                          ;
0422|                          ;       initialize OUTPUTFIB
0422|                          ;
0422|                                  GETOFIB A1                      ; @file
0422| 226D FFE8                #       MOVE.L   -24(A5),A1
PAGE -  21  MONITOR   FILE:  MONBEGN.TEXT


0426| 2F09                             MOVE.L  A1,-(A7)
0428| 4869 0050                        PEA     FBUFFER(A1)
042C| 4267                             MOVE.W  #0,-(A7)
042E| 6100 ****                       BSR     FINIT                   ; initialize fib
0432|                                  GETOFIB -(A7)                   ; @file
0432| 2F2D FFE8                #       MOVE.L   -24(A5),-(A7)
0436| 487A FFB4                        PEA     FIL1STR                 ; @title
043A| 3F3C 0001                        MOVE.W  #1,-(A7)                ; openold:=true
043E| 42A7                             CLR.L   -(A7)                   ; junk
0440| 6100 ****                       BSR     FOPEN
0444|                          ;
0444|                          ;       initialize SYSTERM FIB
0444|                          ;
0444|                                  GETTFIB A1                      ; @file
0444| 226D FFE4                #       MOVE.L   -28(A5),A1
0448| 2F09                             MOVE.L  A1,-(A7)
044A| 4869 0050                        PEA     FBUFFER(A1)
044E| 4267                             MOVE.W  #0,-(A7)
0450| 6100 ****                       BSR     FINIT                   ; initialize fib
0454|                                  GETTFIB -(A7)                   ; @file
0454| 2F2D FFE4                #       MOVE.L   -28(A5),-(A7)
0458| 487A FF9C                        PEA     FIL2STR                 ; @title
045C| 3F3C 0001                        MOVE.W  #1,-(A7)                ; openold:=true
0460| 42A7                             CLR.L   -(A7)                   ; junk
0462| 6100 ****                       BSR     FOPEN
0466|                          ;
0466| 4E75                             RTS
0468|                          ;
0468| 0C                       SMSCNFO .BYTE   12
0469| 4D 4F 4E 2E 4D 49 53             .ASCII  'MON.MISCINFO '
0470| 43 49 4E 46 4F 20
0476|                          ;
0476|                          ;       INITMI -- initialize SYSCOM with MON.MISCINFO
0476|                          ;
0476| 0000 0050                MISCFIB EQU     FIBSZ
0476| 0000 0250                MISCBUF EQU     MISCFIB+FBLKSIZ
0476|                          ;
03E2* 0094
0476| 4E56 FDB0                INITMI  LINK    A6,#-MISCBUF
047A| 486E FFB0                        PEA     -MISCFIB(A6)            ; @file
047E| 42A7                             CLR.L   -(A7)
0480| 3F3C FFFF                        MOVE.W  #-1,-(A7)               ; untyped file
0484| 6100 ****                       BSR     FINIT                   ; initialize fib
0488| 486E FFB0                        PEA     -MISCFIB(A6)            ; @file
048C| 487A FFDA                        PEA     SMSCNFO                 ; @title
0490| 3F3C 0001                        MOVE.W  #1,-(A7)                ; openold:=true
0494| 42A7                             CLR.L   -(A7)                   ; junk
0496| 6100 ****                       BSR     FOPEN                   ; open the file
049A|                                  GETSYSC A0
049A| 206D FFFC                #       MOVE.L   -4(A5),A0
049E| 4A10                             TST.B   (A0)                    ; exit if ioresult is nonzero
04A0| 66**                            BNE.S   INITMIX
04A2| 42A7                             CLR.L   -(SP)                   ; Room for function result
04A4| 486E FFB0                        PEA     -MISCFIB(A6)            ; File address
04A8| 486E FDB0                        PEA     -MISCBUF(A6)            ; Buffer address
PAGE -  22  MONITOR   FILE:  MONBEGN.TEXT


04AC| 3F3C 0001                        MOVE.W  #1,-(SP)                ; # Blocks to read
04B0| 4267                             MOVE.W  #0,-(SP)                ; Block to read
04B2| 3F3C 0001                        MOVE.W  #1,-(SP)                ; Read:=true
04B6| 6100 ****                       BSR     BLKIO
04BA| 4A5F                             TST.W   (A7)+                   ; test #blocks read, exit if zero
04BC| 6700 ****                       BEQ     INITMIX
```

```
04C0| 41EE FDB0                        LEA     -MISCBUF(A6),A0         ; compute @source
04C4| D0FC 003A                        ADD.W   #58,A0
04C8|                                  GETSYSC A1                      ; compute @destination
04C8| 226D FFFC                 #      MOVE.L   -4(A5),A1
04CC| 5C49                             ADD.W   #OFSMISC-TOPSYSC,A1
04CE| 7013                             MOVE.L  #19,D0
04D0| 32D8                     INITMIL MOVE.W  (A0)+,(A1)+             ; copy MISCINFO from file to SYSCOM
04D2| 5340                             SUB.W   #1,D0
04D4| 66FA                             BNE.S   INITMIL
04BE* 0018
04A0* 34
04D6| 4E5E                     INITMIX UNLK    A6
04D8| 4E75                             RTS
04DA|                           ;
04DA|                           ;      PROCEDURE ZEROMEM(BLKS:INTEGER)
04DA|                           ;
04DA|                           ;      stack
04DA|                           ;      8       number of blocks
04DA|                           ;      4       return address
04DA|                           ;      0       old A6
04DA|                           ;
0376* 0164
04DA| 4E56 F800                  ZEROMEM LINK    A6,#-2048               ; allocate 2k for directory
04DE| 342E 0008                         MOVE.W  8(A6),D2
04E2| 67**                             BEQ.S   @3                      ; if no memory then exit
04E4|                           ;
04E4|                           ;      zero the directory
04E4|                           ;
04E4| 204F                             MOVE.L  A7,A0
04E6| 303C 01FF                         MOVE.W  #511,D0                 ; fill 512 longs with zero
04EA| 4298                     @1      MOVE.L  #0,(A0)+
04EC| 51C8 FFFC                         DBF     D0,@1
04F0|                           ;
04F0|                           ;      fill in direntry[0]
04F0|                           ;
04F0| 204F                             MOVE.L  A7,A0                   ; fill in only nonzero fields
04F2| 317C 0006 0002                    MOVE.W  #6,LSTBLK(A0)           ; first..last is 0..6
04F8| 43E8 0006                         LEA     DVID(A0),A1
04FC| 22FC 064D 454D                    MOVE.L  #$064D454D,(A1)+        ; set DVID to MEMORY
0502| 22FC 4F52 5900                    MOVE.L  #$4F525900,(A1)+
0508| 3142 000E                         MOVE.W  D2,DEOVBLK(A0)          ; set number of blocks (no files)
050C| 317C A4E5 0014                    MOVE.W  #$A4E5,DLASTBT(A0)      ; set DATE to May 14, 1982
0512|                           ;
0512|                           ;      copy the directory to #4:
0512|                           ;
0512| 204F                             MOVE.L  A7,A0
0514| 2278 0144                         MOVE.L  $144,A1
0518| D2FC 0400                         ADD.W   #2*FBLKSIZ,A1
051C| 303C 01FF                         MOVE.W  #511,D0                 ; copy 512 longs
PAGE -  23  MONITOR   FILE: MONBEGN.TEXT


0520| 22D8                     @2      MOVE.L  (A0)+,(A1)+
0522| 51C8 FFFC                         DBF     D0,@2
04E2* 42
0526| 4E5E                     @3      UNLK    A6
0528| 205F                             MOVE.L  (A7)+,A0
052A| 544F                             ADD.W   #2,A7
052C| 4ED0                             JMP     (A0)
052E|                           ;
052E|                           ;      initialize the device table
052E|                           ;
037A* 01B4
052E| 4E56 FFE0                  INITDEV LINK    A6,#-DNTRYSZ
0532| 204F                             MOVE.L  A7,A0
0534| 4280                             CLR.L   D0
0536| 20C0                             MOVE.L  D0,(A0)+        ; clear device name
0538| 20C0                             MOVE.L  D0,(A0)+
053A| 20C0                             MOVE.L  D0,(A0)+
053C| 20C0                             MOVE.L  D0,(A0)+
053E| 70FF                             MOVE.L  #-1,D0
0540| 20C0                             MOVE.L  D0,(A0)+        ; base of hardware
0542| 20C0                             MOVE.L  D0,(A0)+        ; mount table
0544| 20C0                             MOVE.L  D0,(A0)+        ; driver jump table
0546| 20C0                             MOVE.L  D0,(A0)+        ; global data area
0548|                                  GETDTBL A1
0548| 226D FF88                 #      MOVE.L   -120(A5),A1
054C| 4240                             MOVE.W  #0,D0
054E| 204F                     DEVLOOP MOVE.L  A7,A0
0550| 45FA ****                        LEA     DEVJMPT,A2
0554| 4EF2 0000                        JMP     0(A2,D0)
0552* 0006
0558| 6000 ****                DEVJMPT BRA     DEVNUM0
055C| 6000 ****                        BRA     DEVNUM1
0560| 6000 ****                        BRA     DEVNUM2
0564| 6000 ****                        BRA     DEVNUM3
0568| 6000 ****                        BRA     DEVNUM4
056C| 6000 ****                        BRA     DEVNUM5
0570| 6000 ****                        BRA     DEVNUM6
0574| 6000 ****                        BRA     DEVNUM7
0578|                           DEVNUM0 GETDTBL A3
055A* 001E
0578| 266D FF88                 #      MOVE.L   -120(A5),A3
057C| D6FC 0100                        ADD.W   #DTBLSIZ,A3
0580| 214B 0014                        MOVE.L  A3,DEVMTBL(A0)
0584| 6000 ****                        BRA     DOCOPY
```

```
0588|                          DEVNUM1 GETDTBL A3
055E* 002A
0588|  266D FF88              #       MOVE.L   -120(A5),A3
058C|  D6FC 0100                      ADD.W   #DTBLSIZ,A3
0590|  214B 0014                      MOVE.L  A3,DEVMTBL(A0)
0594|  6000 ****                      BRA     DOCOPY
0562* 0036
0598|  D6FC 0054              DEVNUM2 ADD.W   #MTBLSIZ,A3
059C|  214B 0014                      MOVE.L  A3,DEVMTBL(A0)
05A0|  60**                          BRA.S   DOCOPY
0566* 003C
PAGE -  24  MONITOR   FILE: MONBEGN.TEXT


05A2|  2178 0134 0010        DEVNUM3 MOVE.L  $134,DEVBASE(A0)
05A8|  D6FC 0054                      ADD.W   #MTBLSIZ,A3
05AC|  214B 0014                      MOVE.L  A3,DEVMTBL(A0)
05B0|  60**                          BRA.S   DOCOPY
056A* 0048
05B2|  2178 0158 0010        DEVNUM4 MOVE.L  $158,DEVBASE(A0)
05B8|  67**                          BEQ.S   @1
05BA|  06A8 0000 2001 0010           ADD.L   #$2001,DEVBASE(A0)
05B8* 08
05C2|  D6FC 0054              @1      ADD.W   #MTBLSIZ,A3
05C6|  214B 0014                      MOVE.L  A3,DEVMTBL(A0)
05CA|  60**                          BRA.S   DOCOPY
056E* 005E
05CC|  2178 0158 0010        DEVNUM5 MOVE.L  $158,DEVBASE(A0)
05D2|  67**                          BEQ.S   @1
05D4|  06A8 0000 2801 0010           ADD.L   #$2801,DEVBASE(A0)
05D2* 08
05DC|  D6FC 0054              @1      ADD.W   #MTBLSIZ,A3
05E0|  214B 0014                      MOVE.L  A3,DEVMTBL(A0)
05E4|  60**                          BRA.S   DOCOPY
0572* 0074
05E6|  2178 0158 0010        DEVNUM6 MOVE.L  $158,DEVBASE(A0)
05EC|  67**                          BEQ.S   @1
05EE|  06A8 0000 3001 0010           ADD.L   #$3001,DEVBASE(A0)
05EC* 08
05F6|  D6FC 0054              @1      ADD.W   #MTBLSIZ,A3
05FA|  214B 0014                      MOVE.L  A3,DEVMTBL(A0)
05FE|  60**                          BRA.S   DOCOPY
0576* 008A
0600|  2178 0158 0010        DEVNUM7 MOVE.L  $158,DEVBASE(A0)
0606|  67**                          BEQ.S   @1
0608|  06A8 0000 3801 0010           ADD.L   #$3801,DEVBASE(A0)
0606* 08
0610|  D6FC 0054              @1      ADD.W   #MTBLSIZ,A3
0614|  214B 0014                      MOVE.L  A3,DEVMTBL(A0)
05FE* 18
05E4* 32
05CA* 4C
05B0* 66
05A0* 76
0596* 0082
0586* 0092
0618|  7207                  DOCOPY  MOVE.L  #DNTRYSZ/4-1,D1
061A|  22D8                  CPYLOOP MOVE.L  (A0)+,(A1)+
061C|  51C9 FFFC                      DBF     D1,CPYLOOP
0620|  5840                          ADD.W   #4,D0
0622|  0C40 0020                      CMP.W   #4*<MAXDEV+1>,D0
0626|  6600 FF26                      BNE     DEVLOOP
062A|                          ;
062A|                          ; clear LOGN bitmap
062A|                          ;
062A|                                  GETLOGN A0
062A|  206D FF60              #       MOVE.L   -160(A5),A0
062E|  303C 0007                      MOVE.W  #MAXDEV,D0
0632|  4298                  @1      CLR.L   (A0)+
PAGE -  25  MONITOR   FILE: MONBEGN.TEXT


0634|  51C8 FFFC                      DBF     D0,@1
0638|                          ;
0638|                          ; clear UT17 volume name table
0638|                          ;
0638|                                  GETUT17 A0
0638|  206D FF5C              #       MOVE.L   -164(A5),A0
063C|  303C 0092                      MOVE.W  #UT17SIZ/8-1,D0
0640|  4298                  @2      CLR.L   (A0)+
0642|  4298                          CLR.L   (A0)+
0644|  51C8 FFFA                      DBF     D0,@2
0648|  4E5E                          UNLK    A6
064A|  4E75                          RTS
064C|                          ;
064C|  0008                  W1LPL   .WORD   8
064E|  57 65 6C 63 6F 6D 65   W1PL    .ASCII  'Welcome '
0655|  20
0656|                          ;
0656|  0010                  W2LPL   .WORD   16
0658|  20 74 6F 20 4C 69 73   W2PL    .ASCII  ' to Lisa Monitor'
065F|  61 20 4D 6F 6E 69 74
0666|  6F 72
0668|                          ;
0668|                          ;       XSTRTUP -- execute MON.STARTUP
0668|                          ;
```

```
0668|                          XSTRTUP DBUGCLR                 ; just execute system.startup
0668| 422D FFA9               #       MOVE.B   #0,-87(A5)
066C| 41FA ****                       LEA      STARTUP,A0
0670|                                 GETSTRB A1
0670| 226D FFAA               #       MOVE.L   -86(A5),A1
0674| 22D8                            MOVE.L   (A0)+,(A1)+    ; 16 bytes is 4 longs
0676| 22D8                            MOVE.L   (A0)+,(A1)+
0678| 22D8                            MOVE.L   (A0)+,(A1)+
067A| 22D8                            MOVE.L   (A0)+,(A1)+
067C| 4241                            CLR.W    D1
067E| 6100 ****                       BSR      TRYEXEC        ; MON.STARTUP doesn't have EXEC/
0682| 4A40                            TST.W    D0
0684| 67**                            BEQ.S    @2
0686| 4A9F                            TST.L    (A7)+          ; delete return address
0688| 6100 ****                       BSR      GETREGS
068C| 6100 ****                       BSR      INITPRG
0690| 6100 ****                       BSR      HOMCRSR
0694| 6100 ****                       BSR      CLRSCR
0698| 6000 ****                       BRA      CMDLOOP
0684* 16
069C| 4E75                    @2      RTS
069E|                          ;
066E* 0030
069E| 0B                      STARTUP .BYTE    11
069F| 4D 4F 4E 2E 53 54 41           .ASCII   'MON.STARTUP'
06A6| 52 54 55 50
06AA|                          ;
06AA|                          ;       WELCOME -- prints welcome message
06AA|                          ;
06AA| 6100 ****               WELCOME BSR      GETREGS
06AE| 6100 ****                       BSR      HOMCRSR
PAGE -  26  MONITOR   FILE: MONBEGN.TEXT


06B2| 6100 ****                       BSR      CLRSCR
06B6| 3F3C 0008                       MOVE.W   #8,-(A7)
06BA| 6100 ****               WELCOM2 BSR      WCRLF          ; print CRLF's to middle of screen
06BE| 5357                            SUB.W    #1,(A7)
06C0| 66F8                            BNE.S    WELCOM2
06C2| 4A5F                            TST.W    (A7)+          ; delete CRLF counter
06C4| 43FA FF88                       LEA      W1PL,A1
06C8| 343A FF82                       MOVE.W   W1LPL,D2
06CC| 6100 ****                       BSR      PRINT          ; print 'Welcome '
06D0|                                 GETSYV   A1
06D0| 226D FFD4               #       MOVE.L   -44(A5),A1
06D4| 1419                            MOVE.B   (A1)+,D2
06D6| 4882                            EXT.W    D2
06D8| 6100 ****                       BSR      PRINT          ; print sysunit volid
06DC| 43FA FF7A                       LEA      W2PL,A1
06E0| 343A FF74                       MOVE.W   W2LPL,D2
06E4| 6100 ****                       BSR      PRINT          ; print ' to Lisa Monitor'
06E8| 6100 ****                       BSR      WCRLF
06EC| 6000 ****                       BRA      CMDLOOP
06F0|
06F0|                          ; ----- START OF THE RESIDENT PART OF THE MONITOR -----
06F0|
06F0|                          ;
06F0|                          ;       UNIT TABLE -- VECTORS TO DRIVERS
06F0|                          ;
06F0| 0000                     UNITBL  .WORD    0              ; Unit  0: illegal
06F2| 0000 0000                       .WORD    0,0
06F6| 000F                            .WORD    ALLBIT         ; Unit  1: CONSOLE:
06F8| 0000 0000                       .WORD    0,0     ; DRVR
06FC| 000F                            .WORD    ALLBIT         ; Unit  2: SYSTERM: (non-echoing keyboard)
06FE| 0000 0000                       .WORD    0,0     ; DRVR
0702| 0000                            .WORD    0              ; Unit  3: not used
0704| 0000 0000                       .WORD    0,0
0708| 000F                            .WORD    ALLBIT         ; Unit  4: disk drive 0
070A| 0000 0000                       .WORD    0,0     ; DRVR
070E| 000F                            .WORD    ALLBIT         ; Unit  5: disk drive 1
0710| 0000 0000                       .WORD    0,0     ; DRVR
0714| 0006                            .WORD    OUTBIT+CLRBIT  ; Unit  6: PRINTER:
0716| 0000 0000                       .WORD    0,0     ; DRVR
071A| 0005                            .WORD    INBIT+CLRBIT   ; Unit  7: REMIN:
071C| 0000 0000                       .WORD    0,0     ; DRVR
0720| 0006                            .WORD    OUTBIT+CLRBIT  ; Unit  8: REMOUT:
0722| 0000 0000                       .WORD    0,0     ; DRVR
0726| 000F                            .WORD    ALLBIT         ; Unit  9: disk drive 3
0728| 0000 0000                       .WORD    0,0     ; DRVR
072C| 000F                            .WORD    ALLBIT         ; Unit 10: disk drive 4
072E| 0000 0000                       .WORD    0,0     ; DRVR
0732| 000F                            .WORD    ALLBIT         ; Unit 11: disk drive 5
0734| 0000 0000                       .WORD    0,0     ; DRVR
0738| 000F                            .WORD    ALLBIT         ; Unit 12: disk drive 6
073A| 0000 0000                       .WORD    0,0     ; DRVR
073E| 000F                            .WORD    ALLBIT         ; Unit 13: disk drive 7
0740| 0000 0000                       .WORD    0,0     ; DRVR
0744| 000F                            .WORD    ALLBIT         ; Unit 14: disk drive 8
0746| 0000 0000                       .WORD    0,0     ; DRVR
PAGE -  27  MONITOR   FILE: MONBEGN.TEXT


074A| 000F                            .WORD    ALLBIT         ; Unit 15: disk drive 9
074C| 0000 0000                       .WORD    0,0     ; DRVR
0750| 000F                            .WORD    ALLBIT         ; Unit 16: disk drive 10
0752| 0000 0000                       .WORD    0,0     ; DRVR
```

```
0756| 000F                              .WORD   ALLBIT          ; Unit 17: disk drive 11
0758| 0000 0000                         .WORD   0,0     ; DRVR
075C| 000F                              .WORD   ALLBIT          ; Unit 18: disk drive 12
075E| 0000 0000                         .WORD   0,0     ; DRVR
0762| 000F                              .WORD   ALLBIT          ; Unit 19: disk drive 13
0764| 0000 0000                         .WORD   0,0     ; DRVR
0768| 000F                              .WORD   ALLBIT          ; Unit 20: disk drive 14
076A| 0000 0000                         .WORD   0,0     ; DRVR
076E|                           ;
076E|                           ;
076E|                           ;
076E| 6000 ****               JMPTBL  BRA     ZZLOADIT        ; offset = 0
0772| 6000 ****                       BRA     ZZUNLOAD        ; offset = 4
0776| 6000 ****                       BRA     FWRTCHAR        ; offset = 8
077A| 6000 ****                       BRA     FWRITELN        ; offset = 12
077E| 6000 ****                       BRA     FREADCHR        ; offset = 16
0782| 6000 ****                       BRA     FREADLN         ; offset = 20
0786| 6000 ****                       BRA     FINIT           ; offset = 24
078A| 6000 ****                       BRA     FOPEN           ; offset = 28
078E| 6000 ****                       BRA     BLKIO           ; offset = 32
0792| 6000 ****                       BRA     MNEW            ; offset = 36
0796| 6000 ****                       BRA     MMRK            ; offset = 40
079A| 6000 ****                       BRA     MRLS            ; offset = 44
079E| 6000 ****                       BRA     MEMA            ; offset = 48
07A2| 6000 ****                       BRA     UCLR            ; offset = 52
07A6| 6000 ****                       BRA     UREAD           ; offset = 56
07AA| 6000 ****                       BRA     UWRITE          ; offset = 60
07AE| 6000 ****                       BRA     UBUSY           ; offset = 64
07B2| 6000 ****                       BRA     ZZIORES         ; offset = 68
07B6| 6000 ****                       BRA     REMOVE1         ; offset = 72
07BA| 6000 ****                       BRA     FCLOSE          ; offset = 76
07BE| 6000 ****                       BRA     MHALT           ; offset = 80
07C2| 6000 ****                       BRA     MIOERR          ; offset = 84
07C6| 6000 ****                       BRA     MGOTOXY         ; offset = 88
07CA| 6000 ****                       BRA     RCERR           ; offset = 92
07CE| 6000 ****                       BRA     SCERR           ; offset = 96
07D2| 6000 ****                       BRA     LAUNCH          ; offset = 100
07D6| 6000 ****                       BRA     FGET            ; offset = 104
07DA| 6000 ****                       BRA     FPUT            ; offset = 108
07DE| 6000 ****                       BRA     FSEEK           ; offset = 112
07E2| 6000 ****                       BRA     SCANTTL         ; offset = 116
07E6| 6000 ****                       BRA     FREEPROC        ; offset = 120
07EA| 6000 ****                       BRA     VOLSRCH         ; offset = 124
07EE| 6000 ****                       BRA     DIRSRCH         ; offset = 128
07F2| 6000 ****                       BRA     FRESET          ; offset = 132
07F6| 6000 ****                       BRA     INSNTRY         ; offset = 136
07FA| 6000 ****                       BRA     DELNTRY         ; offset = 140
07FE| 6000 ****                       BRA     ADDR2SO         ; offset = 144
0802| 6000 ****                       BRA     PMADDR2S        ; offset = 148
0806| 6000 ****                       BRA     ADDRDRVR        ; offset = 152
080A| 6000 ****                       BRA     HDSKINIT        ; offset = 156
PAGE -  28  MONITOR   FILE: MONBEGN.TEXT


080E| 6000 ****                       BRA     DSKREAD         ; offset = 160
0812| 6000 ****                       BRA     DSKWRT          ; offset = 164
0816| 6000 ****                       BRA     SETHDSK         ; offset = 168
081A| 6000 ****                       BRA     HDSKCSZ         ; offset = 172
081E| 6000 ****                       BRA     RDMTABL         ; offset = 176
0822| 6000 ****                       BRA     REVISION        ; offset = 180
0826|                           ;
0826|                           ;       Validate index, check for stack overflow, and jump to routine
0826|                           ;
0826| 21CE 0140               HANDLER MOVE.L  A6,$140
082A| 41FA ****                       LEA     GETINDX,A0      ; update PC with entry point
082E| 2F48 0002                       MOVE.L  A0,2(A7)
0832| 4E73                            RTE                     ; get back to callers stack
0834|                           ;
0834|                           ;       DON'T MOVE HANDLER !
0834|                           ;
0834| 0000 00B8               MAXINDX EQU     HANDLER-JMPTBL
0834|                           ;
082C* 0008
0834| 301F                    GETINDX MOVE.W  (A7)+,D0        ; get index word
0836| 6B**                            BMI.S   INDXER0         ; if (index<0) or (index>maxindx) then
0838| 0C40 00B8                       CMPI.W  #MAXINDX,D0     ;   goto indxer0
083C| 6E**                            BGT.S   INDXER0
083E| 4A40                            TST.W   D0              ; skip overflow check for ZZLOADIT
0840| 67**                            BEQ.S   @1
0842| 6100 ****                       BSR     SETUPA5         ; get pointer to system globals
0846| 61**                            BSR.S   OVFCHK
0840* 06
0848| 41FA FF24               @1      LEA     JMPTBL,A0       ; get address of jump table
084C| 4EF0 0000                       JMP     0(A0,D0)        ; go to it
0850|                           ;
0850|                           ;       MHALT -- programmed halt
0850|                           ;
07C0* 0090
0850| 3F3C 0002               MHALT   MOVE.W  #2,-(A7)        ; push error# 2
0854| 60**                            BRA.S   INDXERR
0856|                           ;
0856|                           ;       RCERR -- range check error from paslib routine
0856|                           ;
0856|                           RCERR   RUNUTST
07CC* 008A
0856| 4A2D FFA8                #      TST.B    -88(A5)
```

```
085A| 67**                          BEQ.S   @1
085C| 40E7                          MOVE    SR,-(A7)
085E| 3F3C 0006                     MOVE.W  #6,-(A7)
0862| 6000 ****                     BRA     SOFTBO
085A* 0A
0866| 3F3C 0003              @1      MOVE.W  #3,-(A7)        ; push error# 3
086A| 60**                          BRA.S   INDXERR
086C|                          ;
086C|                          ;       SCERR -- string index error
086C|                          ;
086C|                          SCERR   RUNUTST
07D0* 009C
086C| 4A2D FFA8               #       TST.B    -88(A5)
PAGE -  29  MONITOR   FILE: MONBEGN.TEXT


0870| 67**                          BEQ.S   @1
0872| 40E7                          MOVE    SR,-(A7)
0874| 3F3C 0006                     MOVE.W  #6,-(A7)
0878| 6000 ****                     BRA     SOFTBO
0870* 0A
087C| 3F3C 0004              @1      MOVE.W  #4,-(A7)        ; push error# 4
0880| 60**                          BRA.S   INDXERR
0882|                          ;
0882|                          ;       OVFCHK -- stack overflow check routine
0882|                          ;
0846* 3A
0882| 48E7 00C0              OVFCHK  MOVEM.L A0/A1,-(A7)     ; get some room
0886| 204F                          MOVE.L  A7,A0
0888| 91FC 0000 0100                SUB.L   #256,A0         ; make sure at least 256 free bytes
088E|                                 GETNP   A1
088E| 226D FFF4               #       MOVE.L   -12(A5),A1
0892| B1C9                          CMP.L   A1,A0           ; stack overflow ?
0894| 6E**                          BGT.S   OVFCHKX
0896| 3F3C 0001                     MOVE.W  #1,-(A7)        ; yes, error
089A| 6000 ****                     BRA     INDXERR
0894* 08
089E| 4CDF 0300              OVFCHKX MOVEM.L (A7)+,A0/A1     ; no, just return
08A2| 4E75                          RTS
08A4|                          ;
08A4| 000C                    HLPL    .WORD   12
08A6| 46 61 74 61 6C 20 45   HPL     .ASCII  'Fatal Error '
08AD| 72 72 6F 72 20
08B2|                          ;
08B2|                          ;       INDXERR - Fatal trap handler errors come here.
08B2|                          ;
08B2|                          ;       error
08B2|                          ;
08B2|                          ;       #0 = trap index out of range
08B2|                          ;       #1 = stack overflow
08B2|                          ;       #2 = programmed halt
08B2|                          ;       #3 = value range error
08B2|                          ;       #4 = invalid string index
08B2|                          ;       #5 = can't read root volume
08B2|                          ;
08B2|                          ;       stack
08B2|                          ;               4       error#
08B2|                          ;               0       old A6
08B2|                          ;               -2      errornumber
08B2|                          ;
083C* 74
0836* 7A
08B2| 4267                    INDXER0 MOVE.W  #0,-(A7)        ; push error# 0
089C* 0018
0880* 32
086A* 48
0854* 5E
08B4| 4E56 FFFE              INDXERR LINK    A6,#-2          ; allocate: errornumber
08B8| 43FA FFEC                      LEA     HPL,A1          ; address
08BC| 343A FFE6                      MOVE.W  HLPL,D2         ; and length
08C0| 6100 ****                      BSR     PRINT
PAGE -  30  MONITOR   FILE: MONBEGN.TEXT


08C4| 43EE FFFE                      LEA     -2(A6),A1       ; address
08C8| 7402                          MOVE.L  #2,D2           ; and length
08CA| 32BC 0030                     MOVE.W  #$30,(A1)       ; initialize error number to '0'
08CE| 302E 0004                     MOVE.W  4(A6),D0
08D2| 48C0                          EXT.L   D0
08D4| 6000 ****                     BRA     PRNTERR
08D8|                          ;
08D8|                          ;       CHKERR -- range check error from CHK instruction
08D8|                          ;
08D8| 2F0D                    CHKERR  MOVE.L  A5,-(A7)
08DA| 6100 ****                      BSR     SETUPA5         ; get pointer to system globals
08DE|                                 RUNUTST
08DE| 4A2D FFA8               #       TST.B    -88(A5)
08E2| 2A5F                          MOVE.L  (A7)+,A5
08E4| 67**                          BEQ.S   @1
08E6| 3F3C 0006                     MOVE.W  #6,-(A7)
08EA| 6000 ****                     BRA     SOFTBO
08E4* 08
08EE| 3F3C 0003              @1      MOVE.W  #3,-(A7)        ; push error# 3
08F2| 60C0                          BRA.S   INDXERR
08F4|                          ;
08F4|                          ;       MIOERR -- io check routine
```

```
08F4|                           ;
07C4* 0130
08F4| 6000 ****                MIOERR  BRA     IOCHK
08F8|                           ;
08F8|                           ;       MGOTOXY -- gotoxy
08F8|                           ;
07C8* 0130
08F8| 307C 0D00                MGOTOXY MOVE.W  #$D00,A0
08FC| 321F                              MOVE.W  (A7)+,D1
08FE| 301F                              MOVE.W  (A7)+,D0
0900| 3428 0018                        MOVE.W  $18(A0),D2
0904| 5342                              SUB.W   #1,D2
0906| B042                              CMP.W   D2,D0
0908| 6F**                              BLE.S   @1
090A| 3002                              MOVE.W  D2,D0
0908* 02
090C| 4A40                    @1        TST.W   D0
090E| 6A**                              BPL.S   @2
0910| 4240                              CLR.W   D0
090E* 02
0912| 3428 0016                @2        MOVE.W  $16(A0),D2
0916| 5342                              SUB.W   #1,D2
0918| B242                              CMP.W   D2,D1
091A| 6F**                              BLE.S   @3
091C| 3202                              MOVE.W  D2,D1
091A* 02
091E| 4A41                    @3        TST.W   D1
0920| 6A**                              BPL.S   @4
0922| 4241                              CLR.W   D1
0920* 02
0924| 243C 001B 3D00          @4        MOVE.L  #$1B3D00,D2
092A| 1401                              MOVE.B  D1,D2
092C| 0602 0020                        ADD.B   #$20,D2
PAGE -  31  MONITOR   FILE: MONBEGN.TEXT


0930| E18A                              LSL.L   #8,D2
0932| 1400                              MOVE.B  D0,D2
0934| 0602 0020                        ADD.B   #$20,D2
0938| 2F02                              MOVE.L  D2,-(A7)
093A| 204F                              MOVE.L  A7,A0
093C| 487A ****                        PEA     @5
0940| 3F3C 0002                        MOVE.W  #2,-(A7)
0944| 2F08                              MOVE.L  A0,-(A7)
0946| 3F3C 0004                        MOVE.W  #4,-(A7)
094A| 4267                              CLR.W   -(A7)
094C| 4267                              CLR.W   -(A7)
094E| 6000 ****                        BRA     UWRITE
093E* 0014
0952| 4A9F                    @5        TST.L   (A7)+
0954| 4E75                              RTS
0956|                           ;
0956|                           ;       FUNCTION GETMTBL(DEV:INTEGER):LONGINT
0956|                           ;
0956|                           ;       stack:
0956|                           ;       10      function result
0956|                           ;       8       device number
0956|                           ;       4       return address
0956|                           ;       0       Old A6
0956|                           ;
0956| 4E56 0000                GETMTBL LINK    A6,#0
095A|                                    GETDTBL A0
095A| 206D FF88                #         MOVE.L   -120(A5),A0
095E| 302E 0008                          MOVE.W  8(A6),D0
0962| EB48                              LSL.W   #DEVNSHF,D0
0964| 2D70 0014 000A                    MOVE.L  DEVMTBL(A0,D0.W),10(A6)
096A| 4E5E                              UNLK    A6
096C| 205F                              MOVE.L  (A7)+,A0
096E| 544F                              ADD.W   #2,A7
0970| 4ED0                              JMP     (A0)
0972|                           ;
0972|                           ;       FUNCTION GETBASE(DEV:INTEGER):LONGINT
0972|                           ;
0972|                           ;       stack:
0972|                           ;       10      function result
0972|                           ;       8       device number
0972|                           ;       4       return address
0972|                           ;       0       Old A6
0972|                           ;
0972| 4E56 0000                GETBASE LINK    A6,#0
0976|                                    GETDTBL A0
0976| 206D FF88                #         MOVE.L   -120(A5),A0
097A| 302E 0008                          MOVE.W  8(A6),D0
097E| EB48                              LSL.W   #DEVNSHF,D0
0980| 2D70 0010 000A                    MOVE.L  DEVBASE(A0,D0.W),10(A6)
0986| 4E5E                              UNLK    A6
0988| 205F                              MOVE.L  (A7)+,A0
098A| 544F                              ADD.W   #2,A7
098C| 4ED0                              JMP     (A0)
098E|                           ;
098E|                           ;
PAGE -  32  MONITOR   FILE: MONBEGN.TEXT


098E|                           ;
098E| 0839 0002 00FC F801      NEWTMP1 BTST    #2,$FCF801              ; test for 60hz interrupt
```

```
0996| 66**                          BNE.S   NOT60HZ
0998|                          DECRTOW DECDTOW                         ; decrement the time out word
0998| 536D FF80                #       SUB.W    #1,-128(A5)
099C| 66**                          BNE.S   NOT60HZ
099E| 487A ****                     PEA     QUITDSK
09A2| 2F5F 0002                     MOVE.L  (A7)+,2(A7)             ; if time out then fudge return address
099C* 08
0996* 0E
09A6| 2F12                     NOT60HZ MOVE.L  (A2),-(A7)
09A8| 4E75                          RTS                             ; goto old level handler
09AA|                          ;
09AA|                          ;
09AA|                          ;
09A0* 000A
09AA| 2C6A 0004                QUITDSK MOVE.L  4(A2),A6                ; restore the stack frame
09AE| 2E4A                          MOVE.L  A2,A7                   ; and cut the stack back
09B0| 40C0                          MOVE    SR,D0
09B2| 46FC 2700                     MOVE.W  #$2700,SR
09B6| 21DF 0064                     MOVE.L  (A7)+,$64               ; restore old level 1 handler
09BA| 4A9F                          TST.L   (A7)+                   ; discard saved A6
09BC| 46C0                          MOVE    D0,SR
09BE| 536E FFFE                     SUB.W   #1,-2(A6)               ; decrement counter
09C2| 67**                          BEQ.S   @1                      ; quit if zero
09C4| 302E 0008                     MOVE.W  8(A6),D0
09C8| EB48                          LSL.W   #DEVNSHF,D0
09CA|                               GETDTBL A0
09CA| 206D FF88                #       MOVE.L   -120(A5),A0
09CE| 43FA ****                     LEA     PJMPTBL,A1              ; time out a Profile with headers
09D2| B3F0 0018                     CMP.L   DEVJTBL(A0,D0.W),A1
09D6| 6700 ****                     BEQ     RDMTLP
09DA| 43FA ****                     LEA     NJMPTBL,A1              ; time out a Profile without headers
09DE| B3F0 0018                     CMP.L   DEVJTBL(A0,D0.W),A1
09E2| 6700 ****                     BEQ     RDMTLP
09C2* 22
09E6| 0C6E 0003 0008           @1      CMP.W   #3,8(A6)                ; Built in port ?
09EC| 66**                          BNE.S   @2
09EE| 4280                          MOVE.L  #0,D0
09F0|                               TRAPTO  108                     ; yes, set port free
09F0| 2F07                #       MOVE.L   D7,-(SP)
09F2| 3E3C 006C            #       MOVE.W   #108,D7
09F6| 4E45                #       TRAP     #5
09F8| 2E1F                #       MOVE.L   (SP)+,D7
09EC* 0C
09FA| 6000 ****              @2      BRA     REINITJ
09FE|                          ;
09FE|                          ;       PROCEDURE RDMTABL(SETWORK:BOOLEAN; DEV:INTEGER)
09FE|                          ;
09FE|                          ;       test for hard disk, setup device table and setup the mount table
09FE|                          ;
09FE|                          ;       stack:
09FE|                          ;       10      set working device
09FE|                          ;       8       device number
09FE|                          ;       4       return address
PAGE -  33  MONITOR   FILE: MONBEGN.TEXT


09FE|                          ;       0       Old A6
09FE|                          ;       -2      counter
09FE|                          ;       -514    buffer
09FE|                          ;
0820* 01DE
09FE| 4E56 FDFE                RDMTABL LINK    A6,#-514
0A02| 204F                          MOVE.L  A7,A0
0A04| 303C 0100                     MOVE.W  #DSKBLK/2,D0
0A08| 4258                     @1      MOVE.W  #0,(A0)+                ; initialize the mount table
0A0A| 5340                          SUB.W   #1,D0
0A0C| 66FA                          BNE.S   @1
0A0E| 0C6E 0001 0008               CMP.W   #1,8(A6)
0A14| 6700 ****                     BEQ     SETTWGY
0A18| 0C6E 0002 0008               CMP.W   #2,8(A6)
0A1E| 6700 ****                     BEQ     SETTWGY
0A22| 42A7                          CLR.L   -(A7)
0A24| 3F2E 0008                     MOVE.W  8(A6),-(A7)             ; push device number
0A28| 6100 FF48                     BSR     GETBASE
0A2C| 201F                          MOVE.L  (A7)+,D0                ; see if ptr to the port is zero
0A2E| 6700 ****                     BEQ     MVMTABL
0A32| 47FA FF5A                     LEA     NEWTMP1,A3              ; and load A3 with @temp handler
0A36| 2F0E                          MOVE.L  A6,-(A7)                ; save A6
0A38| 2F38 0008                     MOVE.L  $8,-(A7)                ; save exception vector 3
0A3C| 244F                          MOVE.L  A7,A2                   ; save A7 for trap to NOCARD
0A3E| 41FA ****                     LEA     NOCARD,A0               ; set exception vector
0A42| 21C8 0008                     MOVE.L  A0,$8
0A46| 0C6E 0007 0008               CMP.W   #7,8(A6)
0A4C| 67**                          BEQ.S   @2
0A4E| 42A7                          CLR.L   -(A7)                   ; allocate space for disk size
0A50| 3F2E 0008                     MOVE.W  8(A6),-(A7)             ; push device number
0A54| 6100 ****                     BSR     CDSKINIT                ; for pippin and corvus
0A58| 4A9F                          TST.L   (A7)+                   ; discard disk size
0A5A| 60**                          BRA.S   @3
0A4C* 0E
0A5C| 6100 ****              @2      BSR     MDSKRES                 ; for marksman
0A5A* 04
0A60| 21DF 0008              @3      MOVE.L  (A7)+,$8                ; restore exception vector 3
0A64| 2C5F                          MOVE.L  (A7)+,A6                ; restore A6
0A66|                          ; WHAT DO WE DO IF INTERRUPTS ARE SHUT OFF AND WE CAN'T ABORT THE DISK READ ???
0A66| 3F2E 0008                     MOVE.W  3(A6),-(A7)             ; push device number
```

```
0A6A| 6100 ****              BSR     SETHDSK                 ; set up ptr to driver jump table
0A6E|                      RDMTINIT
0A6E| 3D7C 0004 FFFE        MOVE.W  #4,-2(A6)               ; set counter
09E4* 0090
09D8* 009C
0A74|                      RDMTLP
0A74| 220F                 MOVE.L  A7,D1                   ; save address of buffer
0A76| 302E 0008            MOVE.W  8(A6),D0
0A7A| 6100 ****            BSR     GETJTBL                 ; get devjtbl ptr
0A7E|                      PUTDTOW #60                     ; setup disk time out word 1.0 second
0A7E| 3B7C 003C FF80      #         MOVE.W   #60,-128(A5)
0A84| 43FA ****            LEA     MJMPTBL,A1
0A88| B1C9                 CMP.L   A1,A0                   ; if marksman then
0A8A| 66**                 BNE.S   @1
0A8C|                      PUTDTOW #120                    ;   setup disk time out word 2.0 second
PAGE -  34  MONITOR   FILE: MONBEGN.TEXT


0A8C| 3B7C 0078 FF80      #         MOVE.W   #120,-128(A5)
0A8A* 06
0A92| 2F0E                 @1      MOVE.L  A6,-(A7)        ; save A6
0A94| 2F38 0064            MOVE.L  $64,-(A7)               ; save old level 1 handler
0A98| 244F                 MOVE.L  A7,A2                   ; save A7 for trap to QUITDSK
0A9A| 40C0                 MOVE    SR,D0
0A9C| 46FC 2700            MOVE.W  #$2700,SR
0AA0| 21CB 0064            MOVE.L  A3,$64                  ; setup temp handler for level 1
0AA4| 46C0                 MOVE    D0,SR
0AA6| 4267                 CLR.W   -(A7)                   ; allocate RC
0AA8| 2F0F                 MOVE.L  A7,-(A7)                ; push @RC
0AAA| 3F3C 0001            MOVE.W  #1,-(A7)                ; push drv
0AAE| 2F3C 0000 0001       MOVE.L  #1,-(A7)                ; push block count
0AB4| 302E 0008            MOVE.W  8(A6),D0                ; get device number
0AB8| E148                 LSL.W   #8,D0
0ABA| 4840                 SWAP    D0
0ABC| 303C 0007            MOVE.W  #7,D0                   ; fill in block number
0AC0| 2F00                 MOVE.L  D0,-(A7)                ; push dev# and block#
0AC2| 2F01                 MOVE.L  D1,-(A7)                ; push @buffer
0AC4| 6100 ****            BSR     DSKREAD                 ; read the mount table
0AC8| 321F                 MOVE.W  (A7)+,D1                ; save RC
0ACA| 40C0                 MOVE    SR,D0
0ACC| 46FC 2700            MOVE.W  #$2700,SR
0AD0| 21DF 0064            MOVE.L  (A7)+,$64               ; restore old level 1 handler
0AD4| 2C5F                 MOVE.L  (A7)+,A6                ; restore A6
0AD6| 46C0                 MOVE    D0,SR
0AD8| 4A01                 TST.B   D1                      ; test RC
0ADA| 6B00 ****            BMI     REINIT
0ADE| 302E 0008            MOVE.W  8(A6),D0
0AE2| EB48                 LSL.W   #DEVNSHF,D0
0AE4|                      GETDTBL A1
0AE4| 226D FF88            #       MOVE.L   -120(A5),A1
0AE8| 43F1 0000            LEA     DEVNAME(A1,D0.W),A1     ; address of device n
0AEC| 41EF 0038            LEA     $40-8(A7),A0            ; address of name
0AF0| 0210 0007            AND.B   #7,(A0)
0AF4| 66**                 BNE.S   CHKBLK7                 ; make length(devicename) 1..7
0AF6| 10BC 0001            MOVE.B  #1,(A0)
0AF4* 04
0AFA| 42A7                 CHKBLK7 CLR.L   -(A7)
0AFC| 3F2E 0008            MOVE.W  8(A6),-(A7)             ; push device number
0B00| 6100 ****            BSR     HDSKINIT
0B04| 261F                 MOVE.L  (A7)+,D3                ; save disk size
0B06| 4267                 CLR.W   -(A7)
0B08| 3F2E 0008            MOVE.W  8(A6),-(A7)
0B0C| 6100 ****            BSR     HDSKCSZ
0B10| 301F                 MOVE.W  (A7)+,D0
0B12| E0AB                 LSR.L   D0,D3                   ; divide disk size by cluster size
0B14| 204F                 MOVE.L  A7,A0
0B16| 303C 0100            MOVE.W  #$100,D0                ; index to start of voltable
0B1A| D0C0                 ADD.W   D0,A0
0B1C| 3218                 LOOPVOL MOVE.W  (A0)+,D1        ; get start word
0B1E| 3401                 MOVE.W  D1,D2
0B20| 0241 E000            AND.W   #$E000,D1               ; exit if drv = 0
0B24| 67**                 BEQ.S   BREINIT
0B26| 0C41 E000            CMP.W   #$E000,D1               ; exit if end of table
PAGE -  35  MONITOR   FILE: MONBEGN.TEXT


0B2A| 67**                 BEQ.S   CHKMTBL
0B2C| 0C41 4000            CMP.W   #$4000,D1
0B30| 62**                 BHI.S   BREINIT                 ; exit if drv > 4
0B32| 0242 1FFF            AND.W   #$1FFF,D2
0B36| 67**                 BEQ.S   BREINIT
0B38| B443                 CMP.W   D3,D2
0B3A| 62**                 BHI.S   BREINIT
0B3C| 3218                 MOVE.W  (A0)+,D1                ; get stop word
0B3E| 3801                 MOVE.W  D1,D4
0B40| 0241 E000            AND.W   #$E000,D1               ; ok if top3 = 0
0B44| 67**                 BEQ.S   @1
0B46| 0C41 8000            CMP.W   #$8000,D1               ; ok if top3 = 4
0B4A| 66**                 BNE.S   BREINIT
0B44* 06
0B4C| 0244 1FFF            @1      AND.W   #$1FFF,D4
0B50| 67**                 BEQ.S   BREINIT
0B52| B843                 CMP.W   D3,D4
0B54| 62**                 BHI.S   BREINIT
0B56| B444                 CMP.W   D4,D2                   ; compare start and stop
0B58| 62**                 BHI.S   BREINIT
```

```
0B5A|  5840                          ADD.W   #4,D0
0B5C|  0C40 0200                     CMP.W   #$200,D0
0B60|  66BA                          BNE.S   LOOPVOL
0B58*  08
0B54*  0C
0B50*  10
0B4A*  16
0B3A*  26
0B36*  2A
0B30*  30
0B24*  3C
0B62|  60**                BREINIT   BRA.S   BREINI2
0B2A*  38
0B64|  4A2E 000A            CHKMTBL   TST.B   10(A6)                ; if setwork then
0B68|  67**                          BEQ.S   @1
0B6A|  0C40 0100                     CMP.W   #$100,D0              ;   exit if volume table is empty
0B6E|  67F2                          BEQ.S   BREINIT
0B68*  06
0B70|  41EF 0040           @1        LEA     $40(A7),A0
0B74|  4242                          CLR.W   D2                    ; unit:=0
0B76|  283C E000 0000                MOVE.L  #$E0000000,D4         ; load mask
0B7C|  6100 ****           LOOPTBL   BSR     UNITISB               ; repeat
0B80|  66**                          BNE.S   @1                    ;   if not unitisblocked(unit) then
0B82|  4A98                          TST.L   (A0)+                 ;     exit if mtbl(unit) <> 0000 0000
0B84|  66**                          BNE.S   BREINI2
0B86|  60**                          BRA.S   LOOPEND               ;   else
0B80*  06
0B88|  2018                @1        MOVE.L  (A0)+,D0              ;     if mtbl(unit) <> 0000 0000 then
0B8A|  67**                          BEQ.S   LOOPEND
0B8C|  0C42 0004                     CMP.W   #4,D2                 ;       if unit=4 then
0B90|  66**                          BNE.S   @2
0B92|  0C80 FFFF 0000                CMP.L   #$FFFF0000,D0         ;         exit if mtbl(unit) <> FFFF 0000
0B98|  66**                          BNE.S   BREINI2
0B9A|  60**                          BRA.S   LOOPEND               ;       else
0B90*  0A
PAGE -  36  MONITOR   FILE: MONBEGN.TEXT


0B9C|  43EF 0100           @2        LEA     $100(A7),A1           ;         begin
0BA0|  2219                @3        MOVE.L  (A1)+,D1              ;           get next voltable entry
0BA2|  B081                          CMP.L   D1,D0                 ;           if voltbl<>mtbl(unit) then
0BA4|  67**                          BEQ.S   LOOPEND
0BA6|  C284                          AND.L   D4,D1                 ;             exit if end of voltable
0BA8|  B284                          CMP.L   D4,D1
0BAA|  66F4                          BNE.S   @3
0B98*  12
0B84*  26
0B62*  48
0BAC|  6000 ****           BREINI2   BRA     REINIT                ;         end
0BA4*  0A
0B9A*  14
0B8A*  24
0B86*  28
0BB0|  5242                LOOPEND   ADD.W   #1,D2                 ;   unit:=unit+1;
0BB2|  0C42 0015                     CMP.W   #21,D2                ; until unit=21;
0BB6|  66C4                          BNE     LOOPTBL
0BB8|  302E 0008           VALIDB7   MOVE.W  8(A6),D0
0BBC|  EB48                          LSL.W   #DEVNSHF,D0
0BBE|                                GETDTBL A1
0BBE|  226D FF88           #         MOVE.L   -120(A5),A1
0BC2|  43F1 0000                     LEA     DEVNAME(A1,D0.W),A1   ; address of device n
0BC6|  41EF 0038                     LEA     $40-8(A7),A0          ; address of name
0BCA|  2298                CPYDEVN   MOVE.L  (A0)+,(A1)            ; copy device name into device table
0BCC|  2358 0004                     MOVE.L  (A0)+,4(A1)
0BD0|  4A2E 000A           CPYDEVE   TST.B   10(A6)                ; if setwork then
0BD4|  67**                          BEQ.S   @2                    ;   begin
0BD6|                                GETDTBL A0                    ;     address of device 0
0BD6|  206D FF88           #         MOVE.L   -120(A5),A0
0BDA|  7007                          MOVE.L  #DNTRYSZ/4-1,D0
0BDC|  20D9                @1        MOVE.L  (A1)+,(A0)+           ;     copy entry from dev# n to dev# 0
0BDE|  51C8 FFFC                     DBF     D0,@1
0BD4*  0C
0BE2|  6000 ****           @2        BRA     MVMTABL               ;   end
0A20*  01C6
0A16*  01D0
0BE6|  302E 0008           SETTWGY   MOVE.W  8(A6),D0
0BEA|  3200                          MOVE.W  D0,D1
0BEC|  0641 0030                     ADD.W   #$30,D1               ; make device number into ascii
0BF0|  EB48                          LSL.W   #DEVNSHF,D0
0BF2|                                GETDTBL A0
0BF2|  206D FF88           #         MOVE.L   -120(A5),A0
0BF6|  43F0 0000                     LEA     0(A0,D0.W),A1
0BFA|  22BC 0644 5249                MOVE.L  #$06445249,(A1)       ; set device name to 6 DRIVEx
0C00|  337C 5645 0004                MOVE.W  #$5645,4(A1)
0C06|  1341 0006                     MOVE.B  D1,6(A1)
0C0A|  342E 0008                     MOVE.W  8(A6),D2
0C0E|  4882                          EXT.W   D2
0C10|  E98A                          LSL.L   #4,D2                 ; device number times 16
0C12|  2238 0148                     MOVE.L  $148,D1
0C16|  9282                          SUB.L   D2,D1
0C18|  2341 0018                     MOVE.L  D1,DEVJTBL(A1)        ; set up jump table ptr
0C1C|  2F09                          MOVE.L  A1,-(A7)              ; save address of device n
0C1E|  42A7                          CLR.L   -(A7)
PAGE -  37  MONITOR   FILE: MONBEGN.TEXT
```

```
0C20| 3F2E 0008                    MOVE.W  8(A6),-(A7)
0C24| 6100 ****                    BSR     HDSKINIT
0C28| 201F                         MOVE.L  (A7)+,D0
0C2A| 0080 2000 0000               OR.L    #$20000000,D0
0C30| 41EE FDFE                    LEA     -514(A6),A0
0C34| D0FC 0040                    ADD.W   #$40,A0
0C38| 2140 0014                    MOVE.L  D0,20(A0)               ; start=0, stop=1582
0C3C| 225F                         MOVE.L  (A7)+,A1                ; restore address of device n
0C3E| 4A2E 000A                    TST.B   10(A6)                  ; if setwork then
0C42| 67**                         BEQ.S   @1
0C44| 302E 0008                    MOVE.W  8(A6),D0
0C48| 5340                         SUB.W   #1,D0
0C4A| B038 01B3                    CMP.B   BOOTDVC,D0              ;    if device-1 = bootdvc then
0C4E| 6780                         BEQ.S   CPYDEVE                 ;      go copy device n to device 0
0C42* 0C
0C50| 60**                   @1    BRA.S   MVMTABL
0C52| 43FA ****             TRYNJMP LEA     NJMPTBL,A1              ; set up for a Profile without headers
0C56| 2189 0018                    MOVE.L  A1,DEVJTBL(A0,D0.W)
0C5A| 6000 FE12                    BRA     RDMTINIT
0BAE* 00B0
0ADC* 0182
0C5E| 302E 0008             REINIT  MOVE.W  8(A6),D0
0C62| E848                         LSL.W   #DEVNSHF,D0
0C64|                              GETDTBL A0
0C64| 206D FF88             #       MOVE.L   -120(A5),A0
0C68| 43FA ****                    LEA     PJMPTBL,A1              ; did we fail a Profile with headers
0C6C| B3F0 0018                    CMP.L   DEVJTBL(A0,D0.W),A1
0C70| 67E0                         BEQ.S   TRYNJMP
0C72| 302E 0008             REINITJ MOVE.W  8(A6),D0
0C76| E848                         LSL.W   #DEVNSHF,D0
0C78|                              GETDTBL A0
0C78| 206D FF88             #       MOVE.L   -120(A5),A0
0C7C| 72FF                         MOVE.L  #-1,D1
0C7E| 2181 0018                    MOVE.L  D1,DEVJTBL(A0,D0.W)     ; unmount, init jump table ptr
0C82| 204F                         MOVE.L  A7,A0
0C84| 303C 0100                    MOVE.W  #DSKBLK/2,D0
0C88| 4258                   @2    MOVE.W  #0,(A0)+                ; re-init the mount table
0C8A| 5340                         SUB.W   #1,D0
0C8C| 66FA                         BNE.S   @2
0C50* 3C
0BE4* 00AA
0A30* 025E
0C8E| 42A7                   MVMTABL CLR.L   -(A7)
0C90| 3F2E 0008                    MOVE.W  8(A6),-(A7)             ; push device number
0C94| 6100 FCC0                    BSR     GETMTBL
0C98| 2257                         MOVE.L  (A7),A1                 ; copy @mtbl into A1
0C9A| 41EE FDFE                    LEA     -514(A6),A0
0C9E| D0FC 0040                    ADD.W   #$40,A0
0CA2| 303C 002A                    MOVE.W  #MTBLSIZ/2,D0
0CA6| 32D8                   @1    MOVE.W  (A0)+,(A1)+             ; copy mount table into heap
0CA8| 5340                         SUB.W   #1,D0
0CAA| 66FA                         BNE.S   @1
0CAC| 225F                         MOVE.L  (A7)+,A1                ; pop @mtbl into A1
0CAE| 4AB8 0144                    TST.L   $144
0CB2| 67**                         BEQ.S   @2
PAGE -  38  MONITOR   FILE: MONBEGN.TEXT


0CB4| 237C FFFF 0000 0010          MOVE.L  #$FFFF0000,16(A1)       ; make MEMORY: be #4:
0CBC| 60**                         BRA.S   @3
0CB2* 0A
0CBE| 42A9 0010             @2      CLR.L   16(A1)
0CBC* 04
0CC2| 4E5E                   @3      UNLK    A6
0CC4| 2E9F                         MOVE.L  (A7)+,(A7)              ; delete 4 bytes
0CC6| 4E75                         RTS
0CC8|                        ;
0CC8|                        ;       Trap to here if no PIA
0CC8|                        ;
0A40* 0288
0CC8| 2E4A                   NOCARD  MOVE.L  A2,A7                   ; cut back stack
0CCA| 21DF 0008                    MOVE.L  (A7)+,$8                ; restore exception vector 3
0CCE| 2C5F                         MOVE.L  (A7)+,A6                ; restore A6
0CD0| 60BC                         BRA.S   MVMTABL
0CD2|                        ;
0CD2|                        ;       GETRDIR -- GET ROOT VOLUME AND INITIALIZE SYVID AND DKVID
0CD2|                        ;
0CD2|                        ;       -26     days:array [0..12] of integer
0CD2|                        ;       -30     heaptop
0CD2|                        ;
0CD2| 4E56 FFE2             GETRDIR LINK    A6,#-30
0CD6|                              GETNP   -30(A6)                 ; mark(heahtop)
0CD6| 2D6D FFF4 FFE2        #       MOVE.L   -12(A5),-30(A6)
0CDC|                              GETSYSC A0
0CDC| 206D FFFC             #       MOVE.L   -4(A5),A0
0CE0| 1428 0001                    MOVE.B  SYSUNT(A0),D2
0CE4| 4882                         EXT.W   D2
0CE6| 6100 ****                    BSR     FTCHDIR                 ; get directory
0CEA| 4A06                         TST.B   D6
0CEC| 67**                         BEQ.S   @1
0CEE| 3F3C 0005                    MOVE.W  #5,-(A7)                ; push error# 5
0CF2| 6000 FBC0                    BRA.S   INDXERR
0CEC* 08
0CF6| 204A                   @1      MOVE.L  A2,A0
0CF8|                              ADDRTHD A1                      ; initialize thedate
0CF8| 43ED FFCA             #       LEA      -54(A5),A1
```

```
0CFC| 32A8 0014                    MOVE.W  DLASTBT(A0),(A1)
0D00| 43E8 0006                    LEA     DVID(A0),A1
0D04|                              GETSYV  A0                      ; initialize syvid
0D04| 206D FFD4              #     MOVE.L   -44(A5),A0
0D08| 20D1                        MOVE.L  (A1),(A0)+
0D0A| 20E9 0004                    MOVE.L  4(A1),(A0)+
0D0E|                              GETDKV  A0                      ; initialize dkvid
0D0E| 206D FFCC              #     MOVE.L   -52(A5),A0
0D12| 20D1                        MOVE.L  (A1),(A0)+
0D14| 20E9 0004                    MOVE.L  4(A1),(A0)+
0D18|                         ;
0D18|                              TRAPTO  80
0D18| 2F07                  #     MOVE.L   D7,-(SP)
0D1A| 3E3C 0050              #     MOVE.W   #80,D7
0D1E| 4E45                  #     TRAP     #5
0D20| 2E1F                  #     MOVE.L   (SP)+,D7
0D22| 4A44                        TST.W   D4
PAGE -  39  MONITOR   FILE: MONBEGN.TEXT


0D24| 66**                        BNE.S   @2
0D26| 4A43                        TST.W   D3
0D28| 66**                        BNE.S   @2
0D2A| 4A42                        TST.W   D2
0D2C| 66**                        BNE.S   @2
0D2E| 0C41 0001                  CMP.W   #1,D1
0D32| 66**                        BNE.S   @2
0D34| 0C40 07BC                  CMP.W   #1980,D0
0D38| 67**                        BEQ.S   @5
0D32* 06
0D2C* 0C
0D28* 10
0D24* 14
0D3A| 41EE FFE6            @2    LEA     -26(A6),A0
0D3E| 2248                        MOVE.L  A0,A1
0D40| 4259                        CLR.W   (A1)+
0D42| 32FC 001F                  MOVE.W  #31,(A1)+
0D46| 32FC 001C                  MOVE.W  #28,(A1)+
0D4A| 32FC 001F                  MOVE.W  #31,(A1)+
0D4E| 32FC 001E                  MOVE.W  #30,(A1)+
0D52| 32FC 001F                  MOVE.W  #31,(A1)+
0D56| 32FC 001E                  MOVE.W  #30,(A1)+
0D5A| 32FC 001F                  MOVE.W  #31,(A1)+
0D5E| 32FC 001F                  MOVE.W  #31,(A1)+
0D62| 32FC 001E                  MOVE.W  #30,(A1)+
0D66| 32FC 001F                  MOVE.W  #31,(A1)+
0D6A| 32FC 001E                  MOVE.W  #30,(A1)+
0D6E| 32FC 001F                  MOVE.W  #31,(A1)+
0D72| 0440 076C                  SUB.W   #1900,D0
0D76| 3400                        MOVE.W  D0,D2
0D78| 0242 0003                  AND.W   #3,D2           ; if year mod 4 = 0 then days[2]:=29
0D7C| 66**                        BNE.S   @3
0D7E| 3D7C 001D FFEA              MOVE.W  #29,-22(A6)
0D7C* 06
0D84| 4242                  @3    MOVE.W  #0,D2           ; m:=0
0D86|                                                     ; repeat
0D86| 5442                  @4    ADD.W   #2,D2           ;   m:=m+1
0D88| 0C42 001A                  CMP.W   #26,D2          ;   if m=13 then exit
0D8C| 67**                        BEQ.S   @5
0D8E| 9270 2000                  SUB.W   0(A0,D2.W),D1   ;   d:=d-days[m]
0D92| 6EF2                        BGT.S   @4              ; until d<=0
0D94| D270 2000                  ADD.W   0(A0,D2.W),D1   ; d:=d+days[m]
0D98| E24A                        LSR.W   #1,D2           ; month is 1..12
0D9A| E949                        LSL.W   #4,D1           ; day is 1..31
0D9C| D441                        ADD.W   D1,D2
0D9E| E148                        LSL.W   #8,D0           ; year is 0..99
0DA0| E348                        LSL.W   #1,D0
0DA2| D440                        ADD.W   D0,D2
0DA4|                              ADDRTHD A1
0DA4| 43ED FFCA              #     LEA      -54(A5),A1
0DA8| 3282                        MOVE.W  D2,(A1)
0DAA|                         @5    PUTNP   -30(A6)         ; release(heaptop)
0D8C* 1C
0D38* 70
0DAA| 2B6E FFE2 FFF4         #     MOVE.L   -30(A6),-12(A5)
PAGE -  40  MONITOR   FILE: MONBEGN.TEXT


0DB0| 4E5E                        UNLK    A6
0DB2| 4E75                        RTS
0DB4|
0DB4|
0DB4|
0DB4|                             .INCLUDE HARDDISK.TEXT
0DB4|
0DB4|                             FILENAME: HARDDISK
0DB4|
0DB4|
0DB4|
0DB4|                             PROCEDURE SETHDSK(DEV:INTEGER)
0DB4|
0DB4|                             stack
0DB4|                             8       Device number
0DB4|                             4       Return address
0DB4|                             0       Old A6
0DB4|
```

```
0DB4|                          ;       Gets called during boot to setup the DEVJTBL for each device
0DB4|                          ;
0A6C* 0348
0DB4| 4E56 0000                SETHDSK LINK    A6,#0
0DB8| 0C6E 0007 0008                   CMP.W   #7,8(A6)
0DBE| 67**                            BEQ.S   TRYMRK
0DC0| 42A7                            CLR.L   -(A7)
0DC2| 3F2E 0008                       MOVE.W  8(A6),-(A7)             ; push device number
0DC6| 6100 FBAA                       BSR     GETBASE
0DCA| 205F                            MOVE.L  (A7)+,A0
0DCC| 1010                    TSTSTAT MOVE.B  IRB(A0),D0              ; get the status port
0DCE| 0240 0003                       AND.W   #3,D0
0DD2| D040                            ADD.W   D0,D0
0DD4| 4EFB 00**                       JMP     STATJMP(D0)
0DD6* 02
0DD8| 60F2                    STATJMP BRA.S   TSTSTAT                 ; case 8: corvus off, pippin on the way
0DDA| 60F0                            BRA.S   TSTSTAT                 ; case 9: corvus on the way
0DDC| 60**                            BRA.S   TRYPIP                  ; case A: pippin off, pippin ready
0DDE| 60**                            BRA.S   TRYCRV                  ; case B: cable disc, corvus ready
0DBE* 20
0DE0| 43FA ****                TRYMRK  LEA     MJMPTBL,A1
0DE4| 60**                            BRA.S   SETDISK
0DDC* 08
0DE6| 43FA ****                TRYPIP  LEA     PJMPTBL,A1
0DEA| 60**                            BRA.S   SETDISK
0DDE* 0C
0DEC| 43FA ****                TRYCRV  LEA     CJMPTBL,A1
0DEA* 04
0DE4* 0A
0DF0| 302E 0008                SETDISK MOVE.W  8(A6),D0
0DF4| EB48                            LSL.W   #DEVNSHF,D0
0DF6|                                 GETDTBL A0
0DF6| 206D FF88                #        MOVE.L   -120(A5),A0
0DFA| 2189 0018                       MOVE.L  A1,DEVJTBL(A0,D0.W)
0DFE| 6000 ****                       BRA     INITXIT
0E02|                          ;
0E02|                          ;       GETJTBL moves the address of jump table into A0 for device number in D0
PAGE - .41  MONITOR   FILE: HARDDISK.TEXT


0E02|                          ;
0E02|                          ;       GETJTBL returns to call+4 if address of jump table is negative
0E02|                          ;
0E02|                          GETJTBL GETDTBL A0
0E02| 206D FF88                #        MOVE.L   -120(A5),A0
0E06| EB48                            LSL.W   #DEVNSHF,D0
0E08| 2030 0018                       MOVE.L  DEVJTBL(A0,D0.W),D0
0E0C| 6B**                            BMI.S   @1
0E0E| 2040                            MOVE.L  D0,A0
0E10| 4E75                            RTS
0E0C* 04
0E12| 205F                    @1      MOVE.L  (A7)+,A0
0E14| 5848                            ADD.W   #4,A0
0E16| 4ED0                            JMP     (A0)
0E18|                          ;
0E18|                          ;       HARD DISK ENTRY POINTS
0E18|                          ;
0C26* 01F2
0E18|                          HDSKINIT
0E18| 302F 0004                       MOVE.W  4(A7),D0
0E1C| 61E4                            BSR     GETJTBL
0E1E| 4ED0                            JMP     (A0)
0E20| 4E71                            NOP
0E22|                          ; error return from GETJTBL
0E22| 42AF 0006                       CLR.L   6(A7)
0E26| 60**                            BRA.S   CSZEXIT
0E28|                          DSKREAD
0E28| 4240                            CLR.W   D0
0E2A| 102F 0008                       MOVE.B  8(A7),D0
0E2E| 61D2                            BSR     GETJTBL
0E30| 4EE8 0004                       JMP     4(A0)
0E34|                          ; error return from GETJTBL
0E34| 206F 0012                DISKERR MOVE.L  18(A7),A0       ; get @RC
0E38| 30BC 00FF                       MOVE.W  #$FF,(A0)       ; indicate error
0E3C| 6000 ****                       BRA     DISKIOX
0E40|                          DSKWRT
0E40| 4240                            CLR.W   D0
0E42| 102F 0008                       MOVE.B  8(A7),D0
0E46| 61BA                            BSR     GETJTBL
0E48| 4EE8 0008                       JMP     8(A0)
0E4C|                          ; error return from GETJTBL
0E4C| 60E6                            BRA.S   DISKERR
0E4E|                          HDSKCSZ
0E4E| 302F 0004                       MOVE.W  4(A7),D0
0E52| 61AE                            BSR     GETJTBL
0E54| 4EE8 000C                       JMP     12(A0)
0E58|                          ; error return from GETJTBL
0E58| 426F 0006                       CLR.W   6(A7)
0E5C| 60**                            BRA.S   CSZEXIT
0E5E|                          ;
0E5E|                          ;       CORVUS JUMP TABLE
0E5E|                          ;
0DEE* 0070
0E5E| 6000 ****                CJMPTBL BRA     CDSKINIT
0E62| 6000 ****                       BRA     CDSKREAD
PAGE -  42  MONITOR   FILE: HARDDISK.TEXT
```

```
0E66| 6000 ****                    BRA     CDSKWRT
0E6A| 6000 ****                    BRA     CDSKCSZ
0E6E|                         ;
0E6E|                         ;       FUNCTION CDSKINIT(DEV:INTEGER):LONGINT
0E6E|                         ;
0E60* 000E
0E6E|                         CDSKINIT
0E6E| 4E56 0000                        LINK    A6,#0
0E72| 61**                            BSR.S   PRTINIT
0E74| 2D7C 0000 52E4 000A              MOVE.L  #21220,10(A6)           ; return disk size
0E00* 007C
0E7C| 4E5E                      INITXIT UNLK    A6
0E5C* 20
0E26* 56
0E7E| 205F                      CSZEXIT MOVE.L  (A7)+,A0
0E80| 544F                            ADD.W   #2,A7
0E82| 4ED0                            JMP     (A0)
0E64* 0020
0E84|                         CDSKREAD
0E84| 41FA ****                        LEA     CDSKRD,A0
0E88| 6000 ****                        BRA     DISKIO                  ; go do Corvus disk read
0E68* 0024
0E8C|                         CDSKWRT
0E8C| 41FA ****                        LEA     CDSKWR,A0
0E90| 6000 ****                        BRA     DISKIO                  ; go do Corvus disk write
0E6C* 0028
0E94|                         CDSKCSZ
0E94| 3F7C 0003 0006                  MOVE.W  #3,6(A7)                ; return cluster size (ie shift count)
0E9A| 60E2                            BRA     CSZEXIT
0E9C|                         ;
0E9C|                         ;       PROCEDURE PRTINIT
0E9C|                         ;
0E72* 28
0E9C| 42A7                      PRTINIT CLR.L   -(A7)
0E9E| 3F2E 0008                        MOVE.W  8(A6),-(A7)             ; push device number
0EA2| 6100 FACE                        BSR     GETBASE
0EA6| 205F                            MOVE.L  (A7)+,A0                ; get via base address
0EA8| 117C 000A 0060                  MOVE.B  #$0A,PCR(A0)            ; set ctrl CA2 pulse mode strobe
0EAE| 4228 0018                        MOVE.B  #$00,DDRA(A0)           ; set port A bits to input
0EB2| 10BC 0018                        MOVE.B  #$18,ORB(A0)            ; en=true, dir=in, cmd=false
0EB6| 117C 007C 0010                  MOVE.B  #$7C,DDRB(A0)           ; set port B bits 0,1,7=in, 2,3,4,5,6=out
0EBC| 4E75                            RTS
0EBE|                         ;
0EBE|                         ;       PROCEDURE DISKIO  (VAR RC:INTEGER;
0EBE|                         ;                          DRIVE:INTEGER;
0EBE|                         ;                          BLKCOUNT:LONGINT;
0EBE|                         ;                          BLKNUMBER:LONGINT;
0EBE|                         ;                          VAR BUFFER);
0EBE|                         ;
0EBE|                         ;       Stack:
0EBE|                         ;
0EBE|                         ;               22      @RC
0EBE|                         ;               20      Drive Number
0EBE|                         ;               16      Block Count
0EBE|                         ;               12      Block Number
PAGE -  43  MONITOR   FILE: HARDDISK.TEXT


0EBE|                         ;               8       @Buffer
0EBE|                         ;               4       Return Address
0EBE|                         ;               0       Old A6
0EBE|                         ;              -4       Count
0EBE|                         ;              -8       Blk
0EBE|                         ;             -12       Addr
0EBE|                         ;             -16       TempCount
0EBE|                         ;             -20       Address of driver
0EBE|                         ;             -22       Device number
0EBE|                         ;             -26       Hardware base for device 3
0EBE|                         ;             -30       Hardware base for device n
0EBE|                         ;
0EBE|                         ;       Note: incoming block number is split into dev# and blk#
0EBE|                         ;
0E92* 002C
0E8A* 0034
0EBE| 4E56 FFE2              DISKIO  LINK    A6,#-30
0EC2| 48E7 7000                        MOVEM.L D1-D3,-(A7)
0EC6| 2D48 FFEC                        MOVE.L  A0,-20(A6)              ; save address of driver
0ECA| 4240                            CLR.W   D0
0ECC| 102E 000C                        MOVE.B  12(A6),D0               ; form device number
0ED0| 3D40 FFEA                        MOVE.W  D0,-22(A6)
0ED4| 42A7                            CLR.L   -(A7)
0ED6| 3F00                            MOVE.W  D0,-(A7)                ; push device number n
0ED8| 6100 FA98                        BSR     GETBASE                 ; pushes hardware base
0EDC| 2D5F FFE2                        MOVE.L  (A7)+,-30(A6)
0EE0| 42A7                            CLR.L   -(A7)
0EE2| 3F3C 0003                        MOVE.W  #3,-(A7)                ; push device number 3
0EE6| 6100 FA8A                        BSR     GETBASE                 ; pushes hardware base
0EEA| 201F                            MOVE.L  (A7)+,D0
0EEC| 2D40 FFE6                        MOVE.L  D0,-26(A6)
0EF0| B0AE FFE2                        CMP.L   -30(A6),D0              ; Built in port ?
0EF4| 66**                            BNE.S   @0
0EF6| 7001                            MOVE.L  #1,D0
0EF8|                                 TRAPTO  108                     ; yes, set port busy
0EF8| 2F07                    #       MOVE.L   D7,-(SP)
```

```
0EFA| 3E3C 006C            #       MOVE.W   #108,D7
0EFE| 4E45                 #       TRAP     #5
0F00| 2E1F                 #       MOVE.L   (SP)+,D7
0EF4* 0C
0F02| 422E 000C             @0     CLR.B    12(A6)                ; make block number valid
0F06| 2D6E 0010 FFFC               MOVE.L   16(A6),-4(A6)         ; count:=blockcount
0F0C| 2D6E 000C FFF8               MOVE.L   12(A6),-8(A6)         ; blk:=blocknumber
0F12| 2D6E 0008 FFF4               MOVE.L   8(A6),-12(A6)         ; addr:=@buffer
0F18| 4AAE FFFC             @1     TST.L    -4(A6)                ; while count <> 0 do
0F1C| 67**                         BEQ.S    @2                    ; begin
0F1E| 2D6E FFFC FFF0               MOVE.L   -4(A6),-16(A6)        ;   tempcount:=count
0F24| 2F2E FFE2                    MOVE.L   -30(A6),-(A7)         ;   push hardware base for device n
0F28| 2F2E 0016                    MOVE.L   22(A6),-(A7)          ;   push @RC
0F2C| 3F2E 0014                    MOVE.W   20(A6),-(A7)          ;   push drive number
0F30| 486E FFFC                    PEA      -4(A6)                ;   push @count
0F34| 2F2E FFF8                    MOVE.L   -8(A6),-(A7)          ;   push blk
0F38| 2F2E FFF4                    MOVE.L   -12(A6),-(A7)         ;   push addr
0F3C| 206E FFEC                    MOVE.L   -20(A6),A0            ;   get address of driver
0F40| 4E90                         JSR      (A0)                  ;   call the driver
PAGE -  44  MONITOR   FILE: HARDDISK.TEXT


0F42| 206E 0016                    MOVE.L   22(A6),A0             ;   get @RC
0F46| 4A28 0001                    TST.B    1(A0)                 ;   if RC > 127 then exit
0F4A| 6B**                         BMI.S    @2
0F4C| 202E FFF0                    MOVE.L   -16(A6),D0
0F50| 90AE FFFC                    SUB.L    -4(A6),D0             ;   x:=tempcount-count
0F54| D1AE FFF8                    ADD.L    D0,-8(A6)             ;   blk:=blk+x
0F58| E188                         LSL.L    #8,D0
0F5A| D080                         ADD.L    D0,D0
0F5C| D1AE FFF4                    ADD.L    D0,-12(A6)            ;   addr:=addr+512*x
0F60| 60B6                         BRA.S    @1                    ; end
0F4A* 16
0F1C* 44
0F62| 202E FFE6             @2     MOVE.L   -26(A6),D0
0F66| B0AE FFE2                    CMP.L    -30(A6),D0            ; Built in port ?
0F6A| 66**                         BNE.S    @3
0F6C| 4280                         MOVE.L   #0,D0
0F6E|                              TRAP0    108                   ; yes, set port free
0F6E| 2F07                 #       MOVE.L    D7,-(SP)
0F70| 3E3C 006C            #       MOVE.W   #108,D7
0F74| 4E45                 #       TRAP     #5
0F76| 2E1F                 #       MOVE.L   (SP)+,D7
0F6A* 0C
0F78| 4CDF 000E             @3     MOVEM.L  (A7)+,D1-D3
0F7C| 4E5E                         UNLK     A6
0E3E* 0140
0F7E| 205F                 DISKIOX MOVE.L   (A7)+,A0
0F80| DEFC 0012                    ADD.W    #18,A7
0F84| 4ED0                         JMP      (A0)
0F86|                      ;
0F86|                      ;       PROCEDURE CDSKRD  (BASEADDR:LONGINT;
0F86|                      ;                           VAR RC:INTEGER;
0F86|                      ;                           DRIVE:INTEGER;
0F86|                      ;                           VAR COUNT:LONGINT;
0F86|                      ;                           BLKNUMBER:LONGINT;
0F86|                      ;                           VAR BUFFER);
0F86|                      ;
0F86|                      ;       Stack:
0F86|                      ;
0F86|                      ;               26      base address
0F86|                      ;               22      @RC             ptr to word 0..255
0F86|                      ;               20..21  drive           word
0F86|                      ;               16      @Count          ptr to long
0F86|                      ;               12..15  Block Number    long
0F86|                      ;               8       @Buffer         ptr
0F86|                      ;               4       Return Address
0F86|                      ;               0       Old A6
0F86|                      ;
0E86* 0100
0F86|                      CDSKRD
0F86| 4E56 FFFC                    LINK     A6,#-4
0F8A| 206E 0010                    MOVE.L   16(A6),A0             ; get @count
0F8E| 5390                         SUB.L    #1,(A0)               ; decrement count
0F90| 206E 001A                    MOVE.L   26(A6),A0             ; get base address
0F94| 1D7C 0032 FFFC               MOVE.B   #$32,IOCMD(A6)        ; set command to read
0F9A| 1D6E 0015 FFFD               MOVE.B   21(A6),IODRV(A6)      ; set drive number
PAGE -  45  MONITOR   FILE: HARDDISK.TEXT


0FA0| 1D6E 000F FFFE               MOVE.B   15(A6),BLOCKL(A6)     ; set block lsb
0FA6| 1D6E 000E FFFF               MOVE.B   14(A6),BLOCKH(A6)     ; set block msb
0FAC| 117C 00FF 0018               MOVE.B   #$FF,DDRA(A0)         ; port A bits to output
0FB2| 0890 0003                    BCLR     #3,ORB(A0)            ; bidir drvr to output
0FB6| 43EE FFFC                    LEA      IOCMD(A6),A1          ; @command string
0FBA| 7003                         MOVE.L   #CMDSIZE-1,D0         ; length-1 of command string
0FBC| 0810 0001             DSKRD1 BTST     #1,IRB(A0)            ; ready ?
0FC0| 67FA                         BEQ.S    DSKRD1
0FC2| 1159 0008                    MOVE.B   (A1)+,ORA(A0)         ; move each byte
0FC6| 51C8 FFF4                    DBF      D0,DSKRD1
0FCA| 700D                         MOVE.L   #13,D0                ; delay approx 30us
0FCC| 51C8 FFFE             DSKRD2 DBF      D0,DSKRD2
0FD0| 0810 0000             DSKRD3 BTST     #0,IRB(A0)            ; bus direction in ?
0FD4| 66FA                         BNE.S    DSKRD3
0FD6| 4228 0018                    MOVE.B   #0,DDRA(A0)           ; port A bits to input
0FDA| 08D0 0003                    BSET     #3,ORB(A0)            ; bidir drvr to input
```

```
0FDE| 4280                          CLR.L   D0
0FE0| 0810 0001              DSKRD4  BTST    #1,IRB(A0)              ; ready ?
0FE4| 67FA                          BEQ.S   DSKRD4
0FE6| 1028 0008                     MOVE.B  IRA(A0),D0              ; get status byte into D0
0FEA| 226E 0016                     MOVE.L  22(A6),A1
0FEE| 3280                          MOVE.W  D0,(A1)                 ; return status as RC
0FF0| 226E 0008                     MOVE.L  8(A6),A1                ; address of block buffer
0FF4| 703F                          MOVE.L  #DSKBLK/8-1,D0          ; length-1 of block
0FF6| 0810 0001              DSKRD5  BTST    #1,IRB(A0)              ; ready ?
0FFA| 67FA                          BEQ.S   DSKRD5
0FFC| 41E8 0008                     LEA     IRA(A0),A0              ; @input reg
1000| 12D0                    DSKRD6  MOVE.B  (A0),(A1)+              ; move each byte
1002| 12D0                            MOVE.B  (A0),(A1)+
1004| 12D0                            MOVE.B  (A0),(A1)+
1006| 12D0                            MOVE.B  (A0),(A1)+
1008| 12D0                            MOVE.B  (A0),(A1)+
100A| 12D0                            MOVE.B  (A0),(A1)+
100C| 12D0                            MOVE.B  (A0),(A1)+
100E| 12D0                            MOVE.B  (A0),(A1)+
1010| 51C8 FFEE                       DBF     D0,DSKRD6
1014| 6000 ****                       BRA     FINISH
1018|                          ;
1018|                          ;      PROCEDURE CDSKWR  (BASEADDR:LONGINT;
1018|                          ;                         VAR RC:INTEGER;
1018|                          ;                         DRIVE:INTEGER;
1018|                          ;                         VAR COUNT:LONGINT;
1018|                          ;                         BLKNUMBER:LONGINT;
1018|                          ;                         VAR BUFFER);
1018|                          ;
1018|                          ;      Stack:
1018|                          ;
1018|                          ;              26      base address
1018|                          ;              22      @RC             ptr to word 0..255
1018|                          ;              20..21  drive           word
1018|                          ;              16      @Count          ptr to long
1018|                          ;              12..15  Block Number    long
1018|                          ;              8       @Buffer
1018|                          ;              4       Return Address
1018|                          ;              0       Old A6
PAGE -  46  MONITOR   FILE: HARDDISK.TEXT


1018|                          ;
0E8E* 018A
1018| 4E56 FFFC                CDSKWR  LINK    A6,#-4
101C| 206E 0010                       MOVE.L  16(A6),A0               ; get @count
1020| 5390                            SUB.L   #1,(A0)                 ; decrement count
1022| 206E 001A                       MOVE.L  26(A6),A0               ; get base address
1026| 1D7C 0033 FFFC                  MOVE.B  #$33,IOCMD(A6)          ; set command to write
102C| 1D6E 0015 FFFD                  MOVE.B  21(A6),IODRV(A6)        ; set drive number
1032| 1D6E 000F FFFE                  MOVE.B  15(A6),BLOCKL(A6)       ; set block lsb
1038| 1D6E 000E FFFF                  MOVE.B  14(A6),BLOCKH(A6)       ; set block msb
103E| 117C 00FF 0018                  MOVE.B  #$FF,DDRA(A0)           ; port A bits to output
1044| 0890 0003                       BCLR    #3,ORB(A0)              ; bidir drvr to output
1048| 43EE FFFC                       LEA     IOCMD(A6),A1            ; @command string
104C| 7003                            MOVE.L  #CMDSIZE-1,D0           ; length-1 of command string
104E| 0810 0001              DSKWRT2 BTST    #1,IRB(A0)              ; ready ?
1052| 67FA                          BEQ.S   DSKWRT2
1054| 1159 0008                     MOVE.B  (A1)+,ORA(A0)           ; move each byte
1058| 51C8 FFF4                       DBF     D0,DSKWRT2
105C| 0810 0001              DSKWRT3 BTST    #1,IRB(A0)              ; ready ?
1060| 67FA                          BEQ.S   DSKWRT3
1062| 43E8 0008                     LEA     ORA(A0),A1              ; @output reg
1066| 206E 0008                     MOVE.L  8(A6),A0                ; address of block buffer
106A| 703F                          MOVE.L  #DSKBLK/8-1,D0          ; length-1 of block
106C| 1298                    DSKWRT4 MOVE.B  (A0)+,(A1)              ; move each byte
106E| 1298                            MOVE.B  (A0)+,(A1)
1070| 1298                            MOVE.B  (A0)+,(A1)
1072| 1298                            MOVE.B  (A0)+,(A1)
1074| 1298                            MOVE.B  (A0)+,(A1)
1076| 1298                            MOVE.B  (A0)+,(A1)
1078| 1298                            MOVE.B  (A0)+,(A1)
107A| 1298                            MOVE.B  (A0)+,(A1)
107C| 51C8 FFEE                       DBF     D0,DSKWRT4
1080| 206E 001A                     MOVE.L  26(A6),A0               ; get base address
1084| 700D                          MOVE.L  #13,D0                  ; delay approx 30us
1086| 51C8 FFFE              DSKWRT5 DBF     D0,DSKWRT5
108A| 0810 0000              DSKWRT6 BTST    #0,IRB(A0)              ; bus direction in ?
108E| 66FA                          BNE.S   DSKWRT6
1090| 4228 0018                     MOVE.B  #0,DDRA(A0)             ; port A bits to input
1094| 08D0 0003                     BSET    #3,ORB(A0)              ; bidir drvr to input
1098| 4280                          CLR.L   D0
109A| 0810 0001              DSKWRT7 BTST    #1,IRB(A0)              ; ready ?
109E| 67FA                          BEQ.S   DSKWRT7
10A0| 1028 0008                     MOVE.B  IRA(A0),D0              ; get status byte into D0
10A4| 226E 0016                     MOVE.L  22(A6),A1
10A8| 3280                          MOVE.W  D0,(A1)                 ; return status as RC
10AA| 6000 ****                     BRA     FINISH
10AE|                          ;
10AE|                          ;      PIPPIN JUMP TABLE (headers)
10AE|                          ;
10AE| 6000 ****                PJMPTBL BRA     PDSKINIT
10B2| 6000 ****                       BRA     PDSKREAD
10B6| 6000 ****                       BRA     PDSKWRT
10BA| 6000 ****                       BRA     PDSKCSZ
10BE|                          ;
```

```
10BE|                           ;       PIPPIN JUMP TABLE (no headers)
PAGE -  47  MONITOR  FILE: HARDDISK.TEXT


10BE|                           ;
10BE| 6000 ****                NJMPTBL BRA     PDSKINIT
10C2| 6000 ****                        BRA     NDSKREAD
10C6| 6000 ****                        BRA     NDSKWRT
10CA| 6000 ****                        BRA     PDSKCSZ
10CE|                           ;
10CE|                           ;       FUNCTION PDSKINIT(DEV:INTEGER):LONGINT
10CE|                           ;
10C0* 000E
10B0* 001E
10CE|                           PDSKINIT
10CE| 4E56 0000                        LINK    A6,#0
10D2| 6100 FDC8                        BSR     PRTINIT
10D6| 2D7C 0000 2560 000A              MOVE.L  #9568,10(A6)            ; return disk size
10DE| 6000 FD9C                        BRA     INITXIT
10B4* 002E
10E2|                           PDSKREAD
10E2| 41FA ****                        LEA     PDSKRD,A0
10E6| 6000 FDD6                        BRA     DISKIO                  ; go do Pippin disk read
10B8* 0032
10EA|                           PDSKWRT
10EA| 41FA ****                        LEA     PDSKWR,A0
10EE| 6000 FDCE                        BRA     DISKIO                  ; go do Pippin disk write
10C4* 002E
10F2|                           NDSKREAD
10F2| 41FA ****                        LEA     NDSKRD,A0
10F6| 6000 FDC6                        BRA     DISKIO                  ; go do Pippin disk read
10C8* 0032
10FA|                           NDSKWRT
10FA| 41FA ****                        LEA     NDSKWR,A0
10FE| 6000 FDBE                        BRA     DISKIO                  ; go do Pippin disk write
10CC* 0036
10BC* 0046
1102|                           PDSKCSZ
1102| 3F7C 0003 0006                  MOVE.W  #3,6(A7)                ; return cluster size (ie shift count)
1108| 6000 FD74                        BRA     CSZEXIT
110C|                           ;
110C|                           ;
110C|                           ;
110C| 00 05 0A 0F 04 09 0E      INTRLV  .BYTE   0,5,10,15,4,9,14,3,8,13,2,7,12,1,6,11   ; 9:1 INTERLEAVE
1113| 03 08 0D 02 07 0C 01
111A| 06 0B
111C|                           ;
111C|                           ;
111C|                           ;
111C| 72F0                      REMAP   MOVEQ   #-$10,D1                ; = $F0
111E| C200                              AND.B   D0,D1                   ; mask high 4 bits
1120| 0240 000F                        AND.W   #$0F,D0                 ; mask low  4 bits
1124| D23B 00E6                        ADD.B   INTRLV(D0),D1           ; add in remapped low 4 bits
1128| 4E75                            RTS
112A|                           ;
112A|                           ;
112A|                           ;
112A| 10BC 0008                 FINDD2  MOVE.B  #$08,ORB(A0)            ; en=true, dir=in, cmd=true
112E| 4228 0018                        MOVE.B  #$00,DDRA(A0)           ; set port A bits to input
PAGE -  48  MONITOR  FILE: HARDDISK.TEXT


1132| 0810 0001                 WFB1    BTST    #1,ORB(A0)              ; wait for busy
1136| 66FA                              BNE.S   WFB1
1138| 1228 0008                        MOVE.B  IRA(A0),D1              ; get port A in D1
113C| 4200                              CLR.B   D0
113E| B202                              CMP.B   D2,D1                   ; did pippin return state requested ?
1140| 66**                            BNE.S   SNDR1
1142| 103C 0055                        MOVE.B  #$55,D0
1140* 04
1146| 4210                      SNDR1   MOVE.B  #$00,ORB(A0)            ; en=true, dir=out, cmd=true
1148| 117C 00FF 0018                   MOVE.B  #$FF,DDRA(A0)           ; set port A bits to output
114E| 1140 0008                        MOVE.B  D0,ORA(A0)              ; send reply 00 or 55
1152| 10BC 0010                        MOVE.B  #$10,ORB(A0)            ; en=true, dir=out, cmd=false
1156| 0810 0001                 WFNB1   BTST    #1,ORB(A0)              ; wait for not busy
115A| 67FA                              BEQ.S   WFNB1
115C| 4228 0018                        MOVE.B  #$00,DDRA(A0)           ; set port A bits to input
1160| 10BC 0018                        MOVE.B  #$18,ORB(A0)            ; en=true, dir=in, cmd=false
1164| 4E75                            RTS
1166|                           ;
1166|                           ;
1166|                           ;
1166| 7401                      STAT01  MOVE.L  #1,D2                   ; try to find state 01
1168| 61C0                              BSR     FINDD2
116A| 4A00                              TST.B   D0                      ; if state 01 was found then
116C| 6600 ****                        BNE     COPY6                   ; go send command bytes else
1170| 61B8                              BSR     FINDD2                  ; try to find state 01 again
1172| 4A00                              TST.B   D0                      ; if state 01 was not found then
1174| 6700 ****                        BEQ     DSKERR                  ; return disk error
116E* 000A
1178| 10BC 0010                 COPY6   MOVE.B  #$10,ORB(A0)            ; en=true, dir=out, cmd=false
117C| 117C 00FF 0018                   MOVE.B  #$FF,DDRA(A0)           ; set port A bits to output
1182| 43EE FFFA                        LEA     -6(A6),A1
1186| 1151 0078                 COPY6LP MOVE.B  (A1),NHS(A0)
118A| 1159 0008                        MOVE.B  (A1)+,ORA(A0)
118E| 1151 0078                        MOVE.B  (A1),NHS(A0)
```

```
1192| 1159 0008                    MOVE.B  (A1)+,ORA(A0)
1196| 1151 0078                    MOVE.B  (A1),NHS(A0)
119A| 1159 0008                    MOVE.B  (A1)+,ORA(A0)
119E| 1151 0078                    MOVE.B  (A1),NHS(A0)
11A2| 1159 0008                    MOVE.B  (A1)+,ORA(A0)
11A6| 1151 0078                    MOVE.B  (A1),NHS(A0)
11AA| 1159 0008                    MOVE.B  (A1)+,ORA(A0)
11AE| 1151 0078                    MOVE.B  (A1),NHS(A0)
11B2| 1159 0008                    MOVE.B  (A1)+,ORA(A0)
11B6| 4E75                        RTS
11B8|                       ;
11B8|                       ;
11B8|                       ;
11B8| 1D68 0008 FFFC        GETSTAT MOVE.B  IRA(A0),-4(A6)
11BE| 1D68 0008 FFFD                MOVE.B  IRA(A0),-3(A6)
11C4| 1D68 0008 FFFE                MOVE.B  IRA(A0),-2(A6)
11CA| 1D68 0008 FFFF                MOVE.B  IRA(A0),-1(A6)
11D0| 4E75                        RTS
11D2|                       ;
11D2|                       ;
11D2|                       ;
PAGE -  49  MONITOR   FILE: HARDDISK.TEXT


11D2| 206E 0016             STRTRD  MOVE.L  22(A6),A0               ; set RC to zero
11D6| 4250                        CLR.W   (A0)
11D8| 206E 001A                    MOVE.L  26(A6),A0               ; get base address
11DC| 422E FFFA                    MOVE.B  #0,PCMND(A6)            ; set command to read
11E0| 422E FFFB                    MOVE.B  #0,BLKH(A6)             ; set block number
11E4| 1D6E 000E FFFC                MOVE.B  14(A6),BLKM(A6)
11EA| 102E 000F                    MOVE.B  15(A6),D0               ; lsb of block number
11EE| 1200                        MOVE.B  D0,D1
11F0| 4A6E FFF8                    TST.W   -8(A6)
11F4| 67**                        BEQ.S   @1
11F6| 6100 FF24                    BSR     REMAP
11F4* 04
11FA| 1D41 FFFD             @1      MOVE.B  D1,BLKL(A6)             ; replace block number
11FE| 1D7C 000A FFFE                MOVE.B  #10,RETRY(A6)           ; set retry count
1204| 1D7C 0004 FFFF                MOVE.B  #4,THRESH(A6)           ; set threshhold
120A| 6100 FF5A                    BSR     STAT01                  ; get 01 byte and send read command
120E| 7402                        MOVE.L  #2,D2                   ; get 02 byte
1210| 6100 FF18                    BSR     FINDD2
1214| 4A00                        TST.B   D0                      ; disk error if not in read state
1216| 6700 ****                    BEQ     DSKERR
121A| 4E75                        RTS
121C|                       ;
121C|                       ;       PROCEDURE PDSKRD  (BASEADDR:LONGINT;
121C|                       ;                          VAR RC:INTEGER;
121C|                       ;                          DRIVE:INTEGER;
121C|                       ;                          VAR COUNT:LONGINT;
121C|                       ;                          BLKNUMBER:LONGINT;
121C|                       ;                          VAR BUFFER);
121C|                       ;
121C|                       ;       Stack:
121C|                       ;
121C|                       ;               26      base address
121C|                       ;               22      @RC             ptr to word 0..255
121C|                       ;               20..21  drive           word
121C|                       ;               16      @Count          ptr to long
121C|                       ;               12..15  Block Number    long
121C|                       ;               8       @Buffer
121C|                       ;               4       Return Address
121C|                       ;               0       Old A6
121C|                       ;               -6      Command Buffer
121C|                       ;               -8      Header flag
121C|                       ;               -28     Header Buffer
121C|                       ;
10E4* 0138
121C| 303C 0001             PDSKRD  MOVE.W  #1,D0                   ; headers := true
1220| 60**                        BRA.S   LDSKRD
10F4* 012E
1222| 4240                 NDSKRD  MOVE.W  #0,D0                   ; headers := false
1220* 02
1224| 4E56 FFE4             LDSKRD  LINK    A6,#-28
1228| 3D40 FFF8                    MOVE.W  D0,-8(A6)
122C| 206E 0010                    MOVE.L  16(A6),A0               ; get @count
1230| 5390                        SUB.L   #1,(A0)                 ; decrement count
1232| 619E                        BSR     STRTRD                  ; try read first time
1234| 1D68 0008 FFFC                MOVE.B  IRA(A0),-4(A6)          ; get pippin status
PAGE -  50  MONITOR   FILE: HARDDISK.TEXT


123A| 1D68 0008 FFFD                MOVE.B  IRA(A0),-3(A6)
1240| 1D68 0008 FFFE                MOVE.B  IRA(A0),-2(A6)
1246| 1D68 0008 FFFF                MOVE.B  IRA(A0),-1(A6)
124C| 4A6E FFFE                    TST.W   -2(A6)
1250| 6A**                        BPL.S   RDNRES
1252| 6100 FF7E                    BSR     STRTRD                  ; try read second time
1256| 1D68 0008 FFFC                MOVE.B  IRA(A0),-4(A6)          ; get pippin status
125C| 1D68 0008 FFFD                MOVE.B  IRA(A0),-3(A6)
1262| 1D68 0008 FFFE                MOVE.B  IRA(A0),-2(A6)
1268| 1D68 0008 FFFF                MOVE.B  IRA(A0),-1(A6)
1250* 1C
126E| 7403                 RDNRES  MOVE.L  #3,D2
1270| 4AAE FFFC                    TST.L   -4(A6)
1274| 6600 ****                    BNE     DSKERR
```

```
1278| 4201                          CLR.B   D1              ;INIT CSUM
127A| 41E8 0008                    LEA     IRA(A0),A0
127E| 4A6E FFF8                    TST.W   -8(A6)
1282| 67**                         BEQ.S   RSKPHDR
1284| 43EE FFE4                    LEA     -28(A6),A1
1288| 343C 0009                    MOVE.W  #HDRSIZE,D2
128C|                       READHDR
128C| 1010                          MOVE.B  (A0),D0         ;GET BYTE FROM DISK
128E| B101                          EOR.B   D0,D1           ;INCLUDE IN RUNNING CHECKSUM
1290| 12C0                          MOVE.B  D0,(A1)+        ;AND STORE IT IN BUFFER
1292| 1010                          MOVE.B  (A0),D0         ;GET BYTE FROM DISK
1294| B101                          EOR.B   D0,D1           ;INCLUDE IN RUNNING CHECKSUM
1296| 12C0                          MOVE.B  D0,(A1)+        ;AND STORE IT IN BUFFER
1298| 51CA FFF2                     DBF     D2,READHDR      ;REPEAT UNTIL DONE
1282* 18
129C|                       RSKPHDR
129C| 226E 0008                    MOVE.L  8(A6),A1                ; get address of BUFFER
12A0| 343C 00FF                    MOVE.W  #BLKSIZE,D2
12A4|                       READLP                          ; read the data bytes
12A4| 1010                          MOVE.B  (A0),D0         ;GET BYTE FROM DISK
12A6| B101                          EOR.B   D0,D1           ;INCLUDE IN RUNNING CHECKSUM
12A8| 12C0                          MOVE.B  D0,(A1)+        ;AND STORE IT IN BUFFER
12AA| 1010                          MOVE.B  (A0),D0         ;GET BYTE FROM DISK
12AC| B101                          EOR.B   D0,D1           ;INCLUDE IN RUNNING CHECKSUM
12AE| 12C0                          MOVE.B  D0,(A1)+        ;AND STORE IT IN BUFFER
12B0| 51CA FFF2                     DBF     D2,READLP       ;REPEAT UNTIL DONE
12B4| 4A01                          TST.B   D1
12B6| 67**                         BEQ.S   FINISH
12B8| 4A6E FFF8                    TST.W   -8(A6)
12BC| 67**                         BEQ.S   FINISH
12BE| 4A2E FFEA                    TST.B   -22(A6)
12C2| 6A**                         BPL.S   FINISH
12C4| 7404                          MOVE.L  #4,D2
12C6|                       ;
1276* 0050
1218* 00AE
1176* 0150
12C6| 206E 0016             DSKERR  MOVE.L  22(A6),A0               ; set RC
12CA| 4402                          NEG.B   D2                      ; error is 128..255
12CC| 3082                          MOVE.W  D2,(A0)
12CE|                       ;
12C2* 0A
PAGE -  51  MONITOR   FILE: HARDDISK.TEXT


12BC* 10
12B6* 16
10AC* 0222
1016* 02B8
12CE| 4E5E                 FINISH  UNLK    A6
12D0| 205F                          MOVE.L  (A7)+,A0
12D2| DEFC 0016                    ADD.W   #22,A7
12D6| 4ED0                          JMP     (A0)
12D8|                       ;
12D8|                       ;
12D8|                       ;
12D8| 206E 0016             STRTWRT MOVE.L  22(A6),A0               ; set RC to zero
12DC| 4250                          CLR.W   (A0)
12DE| 206E 001A                    MOVE.L  26(A6),A0               ; get base address
12E2| 1D7C 0002 FFFA                MOVE.B  #2,PCMND(A6)            ; set command to write/verify
12E8| 422E FFFB                    MOVE.B  #0,BLKH(A6)             ; set block number
12EC| 1D6E 000E FFFC                MOVE.B  14(A6),BLKM(A6)
12F2| 102E 000F                    MOVE.B  15(A6),D0               ; lsb of block number
12F6| 1200                          MOVE.B  D0,D1
12F8| 4A6E FFF8                    TST.W   -8(A6)
12FC| 67**                         BEQ.S   @1
12FE| 6100 FE1C                    BSR     REMAP
12FC* 04
1302| 1D41 FFFD             @1      MOVE.B  D1,BLKL(A6)             ; replace block number
1306| 1D7C 000A FFFE                MOVE.B  #10,RETRY(A6)           ; set retry count
130C| 1D7C 0004 FFFF                MOVE.B  #4,THRESH(A6)           ; set threshhold
1312| 6100 FE52                    BSR     STAT01                  ; get 01 byte and send write command
1316| 7404                          MOVE.L  #4,D2
1318| 6100 FE10                    BSR     FINDD2                  ; get 04 byte
131C| 4A00                          TST.B   D0
131E| 67A6                          BEQ     DSKERR                  ; disk error if not in write/verify state
1320| 10BC 0010                    MOVE.B  #$10,ORB(A0)            ; en=true, dir=out, cmd=false
1324| 117C 00FF 0018                MOVE.B  #$FF,DDRA(A0)           ; set port A bits to output
132A| 4A6E FFF8                    TST.W   -8(A6)
132E| 67**                         BEQ.S   WSKPHDR
1330| 43EE FFE4                    LEA     -28(A6),A1
1334| 303C 0009                    MOVE.W  #HDRSIZE,D0
1338| 1151 0078             HDRLOOP MOVE.B  (A1),NHS(A0)            ; send out header bytes
133C| 1159 0008                    MOVE.B  (A1)+,ORA(A0)
1340| 1151 0078                    MOVE.B  (A1),NHS(A0)
1344| 1159 0008                    MOVE.B  (A1)+,ORA(A0)
1348| 51C8 FFEE                    DBF     D0,HDRLOOP
132E* 1C
134C| 226E 0008             WSKPHDR MOVE.L  8(A6),A1                ; get address of BUFFER
1350| 303C 00FF                    MOVE.W  #BLKSIZE,D0
1354| 1151 0078             WRTLOOP MOVE.B  (A1),NHS(A0)            ; send out data bytes
1358| 1159 0008                    MOVE.B  (A1)+,ORA(A0)
135C| 1151 0078                    MOVE.B  (A1),NHS(A0)
1360| 1159 0008                    MOVE.B  (A1)+,ORA(A0)
1364| 51C8 FFEE                    DBF     D0,WRTLOOP
1368| 7406                          MOVE.L  #6,D2
```

```
136A| 6100 FDBE                    BSR     FINDD2
136E| 4A00                         TST.B   D0
1370| 6700 FF54                    BEQ     DSKERR
PAGE -  52  MONITOR   FILE: HARDDISK.TEXT


1374| 4E75                         RTS
1376|                         ;
1376|                         ;       PROCEDURE PDSKWR  (BASEADDR:LONGINT;
1376|                         ;                          VAR RC:INTEGER;
1376|                         ;                          DRIVE:INTEGER;
1376|                         ;                          VAR COUNT:LONGINT;
1376|                         ;                          BLKNUMBER:LONGINT;
1376|                         ;                          VAR BUFFER);
1376|                         ;
1376|                         ;       Stack:
1376|                         ;
1376|                         ;               26      base address
1376|                         ;               22      @RC             ptr to word 0..255
1376|                         ;               20..21  drive           word
1376|                         ;               16      @Count          ptr to long
1376|                         ;               12..15  Block Number    long
1376|                         ;               8       @Buffer
1376|                         ;               4       Return Address
1376|                         ;               0       Old A6
1376|                         ;               -6      Command Buffer
1376|                         ;               -8      Header flag
1376|                         ;               -28     Header Buffer
1376|                         ;
10EC* 028A
1376| 303C 0001               PDSKWR  MOVE.W  #1,D0                   ; headers := true
137A| 60**                            BRA.S   LDSKWR
10FC* 0280
137C| 4240                   NDSKWR  MOVE.W  #0,D0                   ; headers := false
137A* 02
137E| 4E56 FFE4               LDSKWR  LINK    A6,#-28
1382| 3D40 FFF8                       MOVE.W  D0,-8(A6)
1386| 41EE FFE4                       LEA     -28(A6),A0              ; @header buffer
138A| 4281                           CLR.L   D1
138C| 20C1                           MOVE.L  D1,(A0)+
138E| 30C1                           MOVE.W  D1,(A0)+
1390| 30FC 8000                       MOVE.W  #$8000,(A0)+            ; sign bit on in byte 6 indicates csum
1394| 20C1                           MOVE.L  D1,(A0)+
1396| 20C1                           MOVE.L  D1,(A0)+
1398| 20C1                           MOVE.L  D1,(A0)+
139A| 123C 0080                     MOVE.B  #$80,D1                 ; csum for header = $80
139E| 202E 0008                     MOVE.L  8(A6),D0
13A2| 2040                           MOVE.L  D0,A0
13A4| E258                           ROR.W   #1,D0
13A6| 64**                          BCC.S   @2
13A8| 343C 01FF                     MOVE.W  #511,D2
13AC| 1018                   @1      MOVE.B  (A0)+,D0                ; CSUM on byte boundary
13AE| B101                           EOR.B   D0,D1
13B0| 51CA FFFA                     DBF     D2,@1
13B4| 60**                          BRA.S   @4
13A6* 0E
13B6| 343C 007F             @2      MOVE.W  #127,D2
13BA| 2018                   @3      MOVE.L  (A0)+,D0                ; CSUM on even word boundary
13BC| B181                           EOR.L   D0,D1                   ; long csum
13BE| 51CA FFFA                     DBF     D2,@3
13C2| 3001                           MOVE.W  D1,D0
PAGE -  53  MONITOR   FILE: HARDDISK.TEXT


13C4| 4841                           SWAP    D1
13C6| B141                           EOR.W   D0,D1                   ; word csum
13B4* 12
13C8| 3F01                   @4      MOVE.W  D1,-(A7)
13CA| B317                           EOR.B   D1,(A7)
13CC| 1D5F FFEF                     MOVE.B  (A7)+,-17(A6)           ; final csum in byte 11
13D0| 206E 0010                     MOVE.L  16(A6),A0               ; get @count
13D4| 5390                           SUB.L   #1,(A0)                 ; decrement count
13D6| 6100 FF00                     BSR     STRTWRT                 ; try write first time
13DA| 6100 FDDC                     BSR     GETSTAT                 ; write complete get pippin status
13DE| 4A6E FFFE                     TST.W   -2(A6)
13E2| 6A**                          BPL.S   WRTNRES
13E4| 6100 FEF2                     BSR     STRTWRT                 ; try write second time
13E8| 6100 FDCE                     BSR     GETSTAT                 ; write complete get pippin status
13E2* 08
13EC| 7407                   WRTNRES MOVE.L  #7,D2
13EE| 4AAE FFFC                     TST.L   -4(A6)
13F2| 6600 FED2                     BNE     DSKERR
13F6| 6000 FED6                     BRA     FINISH
13FA|                           ;
13FA|                           ; equates for Marksman interface
13FA|                           ;
13FA|                           ; reset for breadboard is thru 139 where as 4-port card uses PB6 on PIA2
13FA| 00FC 7001                 MRKRES  EQU     $FC7001         ; address to read to reset marksman
13FA| 0000 0000                 REG0    EQU     $00             ; offset to read/write register 0
13FA| 0000 0008                 REG1    EQU     $08
13FA| 0000 0010                 REG2    EQU     $10
13FA| 0000 0018                 REG3    EQU     $18
13FA| 0000 0020                 REG4    EQU     $20
13FA| 0000 0028                 REG5    EQU     $28
13FA| 0000 0030                 REG6    EQU     $30
13FA| 0000 0038                 REG7    EQU     $38
```

```
13FA| 0000 0040              REG8     EQU      $40
13FA| 0000 0048              REG9     EQU      $48
13FA| 0000 0050              REGA     EQU      $50
13FA| 0000 0058              REGB     EQU      $58
13FA| 0000 0060              REGC     EQU      $60
13FA| 0000 0068              REGD     EQU      $68
13FA| 0000 0070              REGE     EQU      $70
13FA| 0000 0078              REGF     EQU      $78
13FA| 0000 0000              XPOVRUN  EQU      0
13FA| 0000 0001              CDOVRUN  EQU      1
13FA| 0000 0002              XPCPT    EQU      2
13FA| 0000 0003              XPBUSY   EQU      3
13FA| 0000 0004              CDCPT    EQU      4
13FA| 0000 0005              IMTBUSY  EQU      5
13FA| 0000 0007              DTRQST_  EQU      7
13FA| 0000 00AB              IDFIELD  EQU      $AB
13FA|                        ;
13FA|                        ;        MARKSMAN JUMP TABLE
13FA|                        ;
13FA| 6000 ****              MJMPTBL  BRA      MDSKINIT
13FE| 6000 ****                       BRA      MDSKREAD
1402| 6000 ****                       BRA      MDSKWRT
1406| 6000 ****                       BRA      MDSKCSZ
PAGE -  54  MONITOR   FILE: HARDDISK.TEXT


140A|                        ;
140A|                        ;
140A|                        ;
140A| 207C 00FC 7001         MDSKRES  MOVE.L   #MRKRES,A0
1410| 0890 0006                       BCLR     #6,(A0)
1414| 4E71                            NOP
1416| 08D0 0006                       BSET     #6,(A0)
141A| 203C 0007 A120                  MOVE.L   #500000,D0              ; get count for 2 seconds
1420| 5380                   @1       SUB.L    #1,D0
1422| 66FC                            BNE.S    @1                     ; wait for marksman to reset
1424| 4E75                            RTS
1426|                        ;
1426|                        ;        FUNCTION MDSKINIT(DEV:INTEGER):LONGINT
1426|                        ;
13FC* 002A
1426|                        MDSKINIT
1426| 4E56 0000                       LINK     A6,#0
142A| 4E71                            NOP
142C| 2D7C 0002 0C10 000A    @1       MOVE.L   #134160,10(A6)          ; return disk size
1434| 6000 FA46                       BRA      INITXIT
1400* 0038
1438|                        MDSKREAD
1438| 41FA ****                       LEA      MDSKRD,A0
143C| 6000 FA80                       BRA      DISKIO                 ; go do Marksman disk read
1404* 003C
1440|                        MDSKWRT
1440| 41FA ****                       LEA      MDSKWR,A0
1444| 6000 FA78                       BRA      DISKIO                 ; go do Marksman disk write
1408* 0040
1448|                        MDSKCSZ
1448| 3F7C 0006 0006                  MOVE.W   #6,6(A7)                ; return cluster size (ie shift count)
144E| 6000 FA2E                       BRA      CSZEXIT
1452|                        ;
1452|                        ;        PROCEDURE MDSKRD  (BASEADDR:LONGINT;
1452|                        ;                           VAR RC:INTEGER;
1452|                        ;                           DRIVE:INTEGER;
1452|                        ;                           VAR COUNT:LONGINT;
1452|                        ;                           BLKNUMBER:LONGINT;
1452|                        ;                           VAR BUFFER);
1452|                        ;
1452|                        ;        Stack:
1452|                        ;
1452|                        ;                 26       base address
1452|                        ;                 22       @RC             ptr to word 0..255
1452|                        ;                 20..21   drive           word
1452|                        ;                 16       @Count          ptr to long
1452|                        ;                 12..15   Block Number    long
1452|                        ;                 8        @Buffer
1452|                        ;                 4        Return Address
1452|                        ;                 0        Old A6
1452|                        ;                 -1       reg9
1452|                        ;                 ...
1452|                        ;                 -9       reg1
1452|                        ;                 -10      reg0
1452|                        ;
143A* 0018
PAGE -  55  MONITOR   FILE: HARDDISK.TEXT


1452| 4E56 FFF6              MDSKRD   LINK     A6,#-10
1456| 202E 000C                       MOVE.L   12(A6),D0               ; get Block Number
145A| 81FC 0028                       DIVS     #40,D0                 ; D0 msw is Sector, D0 lsw is quotient
145E| 4281                            CLR.L    D1
1460| 3200                            MOVE.W   D0,D1
1462| 83FC 0006                       DIVS     #6,D1                  ; D1 msw is Head,   D1 lsw is Cylinder
1466| 3401                            MOVE.W   D1,D2
1468| E05A                            ROR.W    #8,D2
146A| 43EE FFF6                       LEA      -10(A6),A1
146E| 12FC 0058                       MOVE.B   #$58,(A1)+              ; reg0 -- read command
1472| 4219                            MOVE.B   #0,(A1)+                ; reg1 -- drive number
```

```
1474| 12C2                          MOVE.B  D2,(A1)+                ; reg2 -- cylinder MSB
1476| 12C1                          MOVE.B  D1,(A1)+                ; reg3 -- cylinder LSB
1478| 4841                          SWAP    D1
147A| 12C1                          MOVE.B  D1,(A1)+                ; reg4 -- head number
147C| 4840                          SWAP    D0
147E| 12C0                          MOVE.B  D0,(A1)+                ; reg5 -- sector number
1480| 4219                          MOVE.B  #0,(A1)+                ; reg6 -- unused
1482| 12FC 0005                     MOVE.B  #5,(A1)+                ; reg7 -- number retries
1486| 7428                          MOVE.L  #40,D2
1488| 9400                          SUB.B   D0,D2                   ; left:=40-sector#
148A| 206E 0010                     MOVE.L  16(A6),A0
148E| B490                          CMP.L   (A0),D2                 ; if left > count then
1490| 6D**                          BLT.S   @0
1492| 2410                          MOVE.L  (A0),D2                 ;   left:=count
1490* 02
1494| 2602                    @0    MOVE.L  D2,D3                   ; copy left to counter
1496| 12C2                          MOVE.B  D2,(A1)+                ; reg8 -- number of sectors to read
1498| 4219                          MOVE.B  #0,(A1)+                ; reg9 -- unused
149A| 206E 001A              @13   MOVE.L  26(A6),A0               ; get base address
149E| 203C 0001 B207               MOVE.L  #111111,D0              ; get count for 1 second
14A4| 5380                    @1    SUB.L   #1,D0
14A6| 66**                          BNE.S   @12                     ; if timeout then reset marksman
14A8| 6100 FF60                     BSR     MDSKRES
14AC| 60EC                          BRA.S   @13
14A6* 06
14AE| 0810 0005              @12   BTST    #IMTBUSY,(A0)           ; wait for IMTBUSY to go false
14B2| 66F0                          BNE.S   @1
14B4| 303C 0048                     MOVE.W  #REG9,D0                ; offset to last register
14B8| 11A1 0000              @2    MOVE.B  -(A1),0(A0,D0)          ; send next byte to disk
14BC| 5140                          SUB.W   #REG1,D0                ; subtract offset between registers
14BE| 6AF8                          BPL.S   @2                      ; done ?
14C0|
14C0| 226E 0008                     MOVE.L  8(A6),A1                ; get address of buffer
14C4| 303C 01FF              @3    MOVE.W  #511,D0
14C8| 0C42 0001                     CMP.W   #1,D2
14CC| 67**                          BEQ.S   @11
14CE| 303C 000F                     MOVE.W  #15,D0
14D2|
14CC* 04
14D2| 1210                    @11   move.b  (A0),D1           ; 1st read
14D4| B210                          cmp.b   (A0),D1           ; same the next time?
14D6| 66FA                          bne.s   @11               ;  nope, keep looking
14D8| 0801 0005                     btst    #IMTbusy,D1       ; busy still ok?
PAGE -  56  MONITOR   FILE: HARDDISK.TEXT


14DC| 67**                          beq.s   @5                ;  nope, assume error
14DE| 0801 0007                     btst    #DTrqst_,D1       ; well, how about data ready?
14E2| 66EE                          bne.s   @11               ;  sorry, look again
14E4|
14E4| 0C42 0001              @9    CMP.W   #1,D2
14E8| 67**                          BEQ.S   @4
14EA| 0348 0078                     MOVEP.L REGF(A0),D1             ; copy the data from fifo to buffer
14EE| 22C1                          MOVE.L  D1,(A1)+
14F0| 0348 0078                     MOVEP.L REGF(A0),D1
14F4| 22C1                          MOVE.L  D1,(A1)+
14F6| 0348 0078                     MOVEP.L REGF(A0),D1
14FA| 22C1                          MOVE.L  D1,(A1)+
14FC| 0348 0078                     MOVEP.L REGF(A0),D1
1500| 22C1                          MOVE.L  D1,(A1)+
1502| 0348 0078                     MOVEP.L REGF(A0),D1
1506| 22C1                          MOVE.L  D1,(A1)+
1508| 0348 0078                     MOVEP.L REGF(A0),D1
150C| 22C1                          MOVE.L  D1,(A1)+
150E| 0348 0078                     MOVEP.L REGF(A0),D1
1512| 22C1                          MOVE.L  D1,(A1)+
1514| 0348 0078                     MOVEP.L REGF(A0),D1
1518| 22C1                          MOVE.L  D1,(A1)+
151A| 51C8 FFC8                     DBF     D0,@9
151E| 60**                          BRA.S   @10
14E8* 36
1520| 12E8 0078              @4    MOVE.B  REGF(A0),(A1)+
1524| 51C8 FFFA                     DBF     D0,@4
151E* 08
1528| 5343                    @10   SUB.W   #1,D3                   ; decr the counter
152A| 6698                          BNE.S   @3
152C| 226E 0010                     MOVE.L  16(A6),A1
1530| 9591                          SUB.L   D2,(A1)                 ; count:=count-left
1532|
14DC* 54
1532| 1010                    @5    move.b  (A0),D0           ; get valid status
1534| B010                          cmp.b   (A0),D0
1536| 66FA                          bne.s   @5
1538| 0800 0005                     btst    #IMTbusy,D0       ; wait for valid CDcpt
153C| 66F4                          bne.s   @5
153E| 0800 0004                     btst    #CDcpt,D0         ; then wait for command completion
1542| 67EE                          beq.s   @5
1544|
1544| 4240                    @7    CLR.W   D0
1546| 226E 0016                     MOVE.L  22(A6),A1               ; get the address of status
154A| 1028 0008                     MOVE.B  REG1(A0),D0             ; get the status byte
154E| 0C00 0005                     CMP.B   #5,D0                   ; if status is bad then
1552| 6F**                          BLE.S   @8
1554| 4400                          NEG.B   D0                      ;    negate the byte
1552* 02
1556| 3280                    @8    MOVE.W  D0,(A1)                 ; copy status to user
```

```
1558| 6000 FD74                      BRA     FINISH
155C|                          ;
155C|                          ;       PROCEDURE MDSKWR  (BASEADDR:LONGINT;
155C|                          ;                          VAR RC:INTEGER;
155C|                          ;                          DRIVE:INTEGER;
PAGE -  57  MONITOR   FILE: HARDDISK.TEXT


155C|                          ;                          VAR COUNT:LONGINT;
155C|                          ;                          BLKNUMBER:LONGINT;
155C|                          ;                          VAR BUFFER);
155C|                          ;
155C|                          ;       Stack:
155C|                          ;
155C|                          ;               26      base address
155C|                          ;               22      @RC             ptr to word 0..255
155C|                          ;               20..21  drive           word
155C|                          ;               16      @Count          ptr to long
155C|                          ;               12..15  Block Number    long
155C|                          ;               8       @Buffer
155C|                          ;               4       Return Address
155C|                          ;               0       Old A6
155C|                          ;
1442* 011A
155C| 4E56 FFF6                MDSKWR  LINK    A6,#-10
1560| 202E 000C                        MOVE.L  12(A6),D0               ; get Block Number
1564| 81FC 0028                        DIVS    #40,D0                  ; D0 msw is Sector, D0 lsw is quotient
1568| 4281                            CLR.L   D1
156A| 3200                            MOVE.W  D0,D1
156C| 83FC 0006                        DIVS    #6,D1                   ; D1 msw is Head,   D1 lsw is Cylinder
1570| 3401                            MOVE.W  D1,D2
1572| E05A                            ROR.W   #8,D2
1574| 43EE FFF6                       LEA     -10(A6),A1
1578| 12FC 0060                       MOVE.B  #$60,(A1)+              ; reg0 -- write command
157C| 4219                            MOVE.B  #0,(A1)+                ; reg1 -- drive number
157E| 12C2                            MOVE.B  D2,(A1)+                ; reg2 -- cylinder MSB
1580| 12C1                            MOVE.B  D1,(A1)+                ; reg3 -- cylinder LSB
1582| 4841                            SWAP    D1
1584| 12C1                            MOVE.B  D1,(A1)+                ; reg4 -- head number
1586| 4840                            SWAP    D0
1588| 12C0                            MOVE.B  D0,(A1)+                ; reg5 -- sector number
158A| 12FC 00AB                       MOVE.B  #IDFIELD,(A1)+          ; reg6 -- id field
158E| 4219                            MOVE.B  #0,(A1)+                ; reg7 -- unused
1590| 7428                            MOVE.L  #40,D2
1592| 9400                            SUB.B   D0,D2                   ; left:=40-sector#
1594| 206E 0010                       MOVE.L  16(A6),A0
1598| B490                            CMP.L   (A0),D2                 ; if left > count then
159A| 6D**                           BLT.S   @0
159C| 2410                            MOVE.L  (A0),D2                 ;   left:=count
159A* 02
159E| 2602                    @0      MOVE.L  D2,D3                   ; copy left to counter
15A0| 12C2                            MOVE.B  D2,(A1)+                ; reg8 -- number of sectors to write
15A2| 4219                            MOVE.B  #0,(A1)+                ; reg9 -- unused
15A4| 206E 001A                @13     MOVE.L  26(A6),A0               ; get base address
15A8| 203C 0001 B207                    MOVE.L  #111111,D0              ; get count for 1 second
15AE| 5380                    @1      SUB.L   #1,D0
15B0| 66**                            BNE.S   @12                     ; if timeout then reset marksman
15B2| 6100 FE56                        BSR     MDSKRES
15B6| 60EC                            BRA.S   @13
15B0* 06
15B8| 0810 0005                @12     BTST    #IMTBUSY,(A0)           ; wait for IMTBUSY to go false
15BC| 66F0                            BNE.S   @1
15BE| 303C 0048                        MOVE.W  #REG9,D0                ; offset to last register
PAGE -  58  MONITOR   FILE: HARDDISK.TEXT


15C2| 11A1 0000                @2      MOVE.B  -(A1),0(A0,D0)          ; send next byte to disk
15C6| 5140                            SUB.W   #REG1,D0                ; subtract offset between registers
15C8| 6AF8                            BPL.S   @2                      ; done ?
15CA| 226E 0008                        MOVE.L  8(A6),A1                ; get address of buffer
15CE| 303C 01FF                @3      MOVE.W  #511,D0
15D2| 0C42 0001                        CMP.W   #1,D2
15D6| 67**                            BEQ.S   @11
15D8| 303C 000F                        MOVE.W  #15,D0
15DC|
15D6* 04
15DC| 1210                    @11     move.b  (A0),D1         ; fetch status
15DE| B210                            cmp.b   (A0),D1
15E0| 66FA                            bne.s   @11             ; and wait for settling
15E2| 0801 0005                        btst    #IMTbusy,D1     ;  unbusy?
15E6| 67**                            beq.s   @5              ; yes, assume error
15E8| 0801 0007                        btst    #DTrqst_,D1     ; else, look at data ready
15EC| 66EE                            bne.s   @11             ;  sorry, still not available
15EE|
15EE| 0C42 0001                @9      CMP.W   #1,D2
15F2| 67**                            BEQ.S   @4
15F4| 2219                            MOVE.L  (A1)+,D1                ; copy the data from fifo to buffer
15F6| 03C8 0078                        MOVEP.L D1,REGF(A0)
15FA| 2219                            MOVE.L  (A1)+,D1
15FC| 03C8 0078                        MOVEP.L D1,REGF(A0)
1600| 2219                            MOVE.L  (A1)+,D1
1602| 03C8 0078                        MOVEP.L D1,REGF(A0)
1606| 2219                            MOVE.L  (A1)+,D1
1608| 03C8 0078                        MOVEP.L D1,REGF(A0)
160C| 2219                            MOVE.L  (A1)+,D1
160E| 03C8 0078                        MOVEP.L D1,REGF(A0)
```

```
1612| 2219                          MOVE.L  (A1)+,D1
1614| 03C8 0078                     MOVEP.L D1,REGF(A0)
1618| 2219                          MOVE.L  (A1)+,D1
161A| 03C8 0078                     MOVEP.L D1,REGF(A0)
161E| 2219                          MOVE.L  (A1)+,D1
1620| 03C8 0078                     MOVEP.L D1,REGF(A0)
1624| 51C8 FFC8                     DBF     D0,@9
1628| 60**                          BRA.S   @10
15F2* 36
162A| 1159 0078              @4      MOVE.B  (A1)+,REGF(A0)
162E| 51C8 FFFA                     DBF     D0,@4
1628* 08
1632| 5343                    @10     SUB.W   #1,D3                   ; decr the counter
1634| 6698                          BNE.S   @3
1636| 226E 0010                     MOVE.L  16(A6),A1
163A| 9591                          SUB.L   D2,(A1)                 ; count:=count-left
163C|
15E6* 54
163C| 1010                    @5      move.b  (A0),D0         ; get valid status
163E| B010                          cmp.b   (A0),D0
1640| 66FA                          bne.s   @5
1642| 0800 0005                     btst    #IMTbusy,D0     ; make sure of valid CDcpt
1646| 66F4                          bne.s   @5
1648| 0800 0004                     btst    #CDcpt,D0       ; and wait for command complete
164C| 67EE                          beq.s   @5
PAGE -  59  MONITOR   FILE: HARDDISK.TEXT


164E|
164E| 4240                    @7      CLR.W   D0
1650| 226E 0016                     MOVE.L  22(A6),A1               ; get the address of status
1654| 1028 0008                     MOVE.B  REG1(A0),D0             ; get the status byte
1658| 0C00 0005                     CMP.B   #5,D0                   ; if status is bad then
165C| 6F**                          BLE.S   @8
165E| 4400                          NEG.B   D0                      ;    negate the byte
165C* 02
1660| 3280                    @8      MOVE.W  D0,(A1)                 ; copy status to user
1662| 6000 FC6A                     BRA     FINISH
1666|                           ;
1666|                           ;
1666|                           .INCLUDE MONIO.TEXT
1666|                           ;
1666|                           ;       FILENAME MONIO
1666|                           ;
1666|
1666|                           ;       REGISTERS
1666|                           ;
1666|                           ;       A7      stack pointer
1666|                           ;       A6-A2   unused
1666|                           ;       A1      ptr to pia
1666|                           ;       A0      scratch
1666|                           ;       D7      ptr system globals
1666|                           ;       D6-D0   input paramters and scratch
1666|
1666|                           ;
1666|                           ;       EXECDRVR -- does the unit read from an exec file
1666|                           ;
1666|                           EXECDRVR
1666| 3800                          MOVE.W  D0,D4           ; save the unit number
1668| 2041                          MOVE.L  D1,A0           ; save the address
166A| 4A42                          TST.W   D2
166C| 67**                          BEQ.S   @2
166E| 6100 ****              @1      BSR     FROMEXEC
1672| 10C0                          MOVE.B  D0,(A0)+
1674| 5342                          SUB.W   #1,D2
1676| 66F6                          BNE.S   @1
166C* 0A
1678| 4E75                    @2      RTS
167A|                           ;
167A|                           ;
167A|                           ;
167A| 4840                    UNITDEV SWAP    D0
167C| 3800                          MOVE.W  D0,D4
167E| EA48                          LSR.W   #5,D0
1680| 0240 000F                     AND.W   #$F,D0
1684| 3F00                          MOVE.W  D0,-(A7)
1686| 42A7                          CLR.L   -(A7)
1688| 3F00                          MOVE.W  D0,-(A7)
168A| 6100 F2CA                     BSR     GETMTBL
168E| 205F                          MOVE.L  (A7)+,A0
1690| 4280                          CLR.L   D0
1692| 301F                          MOVE.W  (A7)+,D0                ; device number
PAGE -  60  MONITOR   FILE: MONIO.TEXT


1694| 0244 001F                     AND.W   #$1F,D4
1698| E54C                          LSL.W   #2,D4                   ; unit number*4
169A| 4E75                          RTS
169C|                           ;
169C|                           ;
169C|                           ;
169C| 48E7 8880               CHECKCD MOVEM.L D0/D4/A0,-(A7)
16A0| 61D8                          BSR.S   UNITDEV
16A2| 3030 4000                     MOVE.W  0(A0,D4),D0
16A6| 8070 4002                     OR.W    2(A0,D4),D0             ; set cc=NE if dsk or mem is mounted
```

```
16AA| 4CDF 0111                    MOVEM.L (A7)+,D0/D4/A0
16AE| 4E75                         RTS
16B0|                        ;
16B0|                        ; Input:
16B0|                        ;       D0      equal to unitnumber*256         byte
16B0|                        ;       D3      block number positive           word
16B0|                        ;
16B0|                        ; Output:
16B0|                        ;       D0      MSB reflects write protect      word
16B0|                        ;       D3      if -1 then memory               word
16B0|                        ;       D4      drive number                    word
16B0|                        ;       D5      volumestart+blocknumber         long
16B0|                        ;       D6      lastblock of this volume        long
16B0|                        ;
16B0|                        ; Note: if starting block for this volume is $FFFF then it is memory
16B0|                        ;
16B0| 61C8                   GETINFO BSR     UNITDEV                 ; D0=device number, D4=unit number*4
16B2| C940                         EXG     D4,D0                   ; D4=device number, D0=unit number*4
16B4| 3C04                         MOVE.W  D4,D6                   ; save device number for call to HDSKCSZ
16B6| E09C                         ROR.L   #8,D4                   ; move dev# to high byte of long
16B8| 3A30 0000                    MOVE.W  0(A0,D0),D5             ; get strt cluster
16BC| 0C45 FFFF                    CMP.W   #$FFFF,D5
16C0| 67**                         BEQ.S   NOTDISK
16C2| 2F04                         MOVE.L  D4,-(A7)                ; push rotated dev# to bias D5 and D6
16C4| 0285 0000 1FFF               AND.L   #$1FFF,D5               ; make strt cluster a long
16CA| 48E7 8080                    MOVEM.L D0/A0,-(A7)
16CE| 4267                         CLR.W   -(A7)
16D0| 3F06                         MOVE.W  D6,-(A7)                ; push device number
16D2| 6100 F77A                    BSR     HDSKCSZ                 ; get cluster size as shift count
16D6| 381F                         MOVE.W  (A7)+,D4
16D8| 4CDF 0101                    MOVEM.L (A7)+,D0/A0
16DC| E9AD                         LSL.L   D4,D5                   ; strtblock:=strtcluster*clustersize
16DE| DA97                         ADD.L   (A7),D5                 ; bias strtblock by dev#
16E0| 0283 0000 7FFF               AND.L   #$7FFF,D3               ; make blocknumber a long
16E6| DA83                         ADD.L   D3,D5                   ; volumestart+blocknumber
16E8| 3C30 0002                    MOVE.W  2(A0,D0),D6             ; get last cluster
16EC| 0286 0000 1FFF               AND.L   #$1FFF,D6               ; make last cluster a long
16F2| E9AE                         LSL.L   D4,D6                   ; lastblock:=lastcluster*clustersize
16F4| DC9F                         ADD.L   (A7)+,D6                ; bias strtblock by dev#, pop dev#
16F6| 3830 0000                    MOVE.W  0(A0,D0),D4             ; compute drv
16FA| E75C                         ROL.W   #3,D4                   ; move top 3 bits to bottom
16FC| 0244 0007                    AND.W   #7,D4                   ; and mask off the junk bits 4-15
1700| 3030 0002                    MOVE.W  2(A0,D0),D0             ; MSB reflects write protect
1704| 4E75                         RTS
16C0* 44
PAGE -  61  MONITOR   FILE: MONIO.TEXT


1706| 3003                   NOTDISK MOVE.W  D3,D0                   ; save strt block for mem r/w
1708| C1FC 0200                      MULS    #FBLKSIZ,D0
170C| D0B8 0144                    ADD.L   $144,D0                 ; compute it in bytes for them
1710| 3605                         MOVE.W  D5,D3                   ; set D3 negative
1712| 4E75                         RTS
1714|                        ;
1714|                        ;       IO ERROR
1714|                        ;
1714| 303C 0001              CDERROR MOVE.W  #IBADBLK,D0
1718| 4E75                         RTS
171A|                        ;
171A|                        ;       WRITE PROTECT ERROR
171A|                        ;
171A| 303C 0003              WRTPRT  MOVE.W  #IBADMOD,D0
171E| 4E75                         RTS
1720|                        ;
1720|                        ;
1720|                        ;
1720| 4241                   CDBUSY  MOVE.W  #0,D1
1722| 4240                         MOVE.W  #INOERR,D0      ; ioresult:=inoerror
1724| 4E75                         RTS
1726|                        ;
1726|                        ;
1726|                        ;
1726| 4240                   CDCLEAR MOVE.W  #INOERR,D0      ; ioresult:=inoerror
1728| 4E75                         RTS
172A|                        ;
172A|                        ; Input parameters:
172A|                        ;
172A|                        ;       D0 -- unit number*256   byte
172A|                        ;       D1 -- buffer address    long
172A|                        ;       D2 -- number bytes      word
172A|                        ;       D3 -- block number      word
172A|                        ;
172A| 4A42                   CDREAD  TST.W   D2              ; is length neg ?
172C| 6BE6                         BMI     CDERROR
172E| 4A43                         TST.W   D3              ; is block neg ?
1730| 6BE2                         BMI     CDERROR
1732| 6100 FF7C                    BSR     GETINFO         ; set D5 to strt and D6 to last
1736| 4A43                         TST.W   D3
1738| 6B00 ****                    BMI     MEMREAD         ; memory ?
173C|                        ;
173C|                        ; register useage:
173C|                        ;
173C|                        ;       D0 -- used by dskread   A0 -- used by dskread
173C|                        ;       D1 -- @buffer           A1 -- used by dskread
173C|                        ;       D2 -- length left
173C|                        ;       D3 -- length of fractional read
```

```
173C|                          ;       D4 -- drv number
173C|                          ;       D5 -- curr block
173C|                          ;       D6 -- last block
173C|                          ;
173C| 3602                             MOVE.W  D2,D3           ; save low 9 bits
173E| E04A                             LSR.W   #8,D2
1740| E24A                             LSR.W   #1,D2           ; number of blocks to read
PAGE -  62  MONITOR   FILE: MONIO.TEXT


1742| 0282 0000 007F                   AND.L   #$7F,D2
1748| 2005                             MOVE.L  D5,D0
174A| D082                             ADD.L   D2,D0
174C| 5380                             SUB.L   #1,D0           ; forming start+blockcount-1
174E| BC80                             CMP.L   D0,D6           ; reading beyond the end of vol ?
1750| 63C2                             BLS     CDERROR
1752| 4A82                             TST.L   D2              ; any full blocks to read ?
1754| 67**                             BEQ.S   @1
1756| 4267                             CLR.W   -(A7)           ; allocate RC
1758| 2F0F                             MOVE.L  A7,-(A7)        ; push @RC
175A| 3F04                             MOVE.W  D4,-(A7)        ; push drv
175C| 2F02                             MOVE.L  D2,-(A7)        ; push block count
175E| 2F05                             MOVE.L  D5,-(A7)        ; push block number
1760| 2F01                             MOVE.L  D1,-(A7)        ; push @buffer
1762| 6100 F6C4                       BSR     DSKREAD
1766| 301F                             MOVE.W  (A7)+,D0
1768| 4A00                             TST.B   D0
176A| 6BA8                             BMI     CDERROR         ; read error ?
1754* 16
176C| 0243 01FF              @1        AND.W   #$1FF,D3
1770| 67**                             BEQ.S   CDRWXIT
1772| DA82                             ADD.L   D2,D5           ; compute next starting block
1774| BC45                             CMP.W   D5,D6
1776| 639C                             BLS     CDERROR         ; reading beyond the end of vol ?
1778| E18A                             LSL.L   #8,D2
177A| D482                             ADD.L   D2,D2
177C| D282                             ADD.L   D2,D1           ; @buffer:=@buffer+512*blockcount
177E| 9EFC 0200              @2        SUB.W   #DSKBLK,A7
1782| 204F                             MOVE.L  A7,A0
1784| 2F08                             MOVE.L  A0,-(A7)        ; save A0 (address of fractional buff)
1786| 4267                             CLR.W   -(A7)           ; allocate RC
1788| 2F0F                             MOVE.L  A7,-(A7)        ; push @RC
178A| 3F04                             MOVE.W  D4,-(A7)        ; push drv
178C| 2F3C 0000 0001                  MOVE.L  #1,-(A7)        ; push block count
1792| 2F05                             MOVE.L  D5,-(A7)        ; push block number
1794| 2F08                             MOVE.L  A0,-(A7)        ; push address of fractional buff
1796| 6100 F690                       BSR     DSKREAD
179A| 301F                             MOVE.W  (A7)+,D0
179C| 205F                             MOVE.L  (A7)+,A0        ; restore A0 (address of fractional buff)
179E| 2241                             MOVE.L  D1,A1
17A0| 12D8              @3        MOVE.B  (A0)+,(A1)+
17A2| 5343                             SUB.W   #1,D3
17A4| 66FA                             BNE.S   @3
17A6| DEFC 0200                       ADD.W   #DSKBLK,A7
17AA| 4A00                             TST.B   D0
17AC| 6B00 FF66                       BMI     CDERROR         ; read error ?
1770* 3E
17B0| 4240              CDRWXIT   MOVE.W  #INOERR,D0      ; ioresult:=inoerror
17B2| 4E75                       RTS
17B4|                          ;
17B4|                          ; register useage:
17B4|                          ;
17B4|                          ;       D0 -- start and counter A0 -- from
17B4|                          ;       D1 -- @buffer           A1 -- to
17B4|                          ;       D2 -- length left
PAGE -  63  MONITOR   FILE: MONIO.TEXT


17B4|                          ;       D3 -- starting block
17B4|                          ;
173A* 007A
17B4| 2040              MEMREAD   MOVE.L  D0,A0           ; from = ($144) + blk*512
17B6| 2241                       MOVE.L  D1,A1           ; to = @buffer
17B8|                          ;
17B8|                          ;       shared by mem r/w to do the move
17B8|                          ;
17B8| 2009              MOVEMEM   MOVE.L  A1,D0           ; first check dest for word boundary
17BA| E248                       LSR.W   #1,D0           ; we assume source is !
17BC| 64**                       BCC.S   MOVFAST
17BE| 3002              MOVSLOW   MOVE.W  D2,D0
17C0| 67EE                       BEQ     CDRWXIT
17C2| 5340                       SUB.W   #1,D0
17C4| 12D8              @1        MOVE.B  (A0)+,(A1)+     ; byte boundary, move bytes
17C6| 51C8 FFFC                  DBF     D0,@1
17CA| 60E4                       BRA     CDRWXIT
17BC* 0E
17CC| 3002              MOVFAST   MOVE.W  D2,D0           ; word boundary, move quads !
17CE| E648                       LSR.W   #3,D0
17D0| 67**                       BEQ.S   @2              ; any quads ?
17D2| 5340                       SUB.W   #1,D0
17D4| 22D8              @1        MOVE.L  (A0)+,(A1)+
17D6| 22D8                       MOVE.L  (A0)+,(A1)+
17D8| 51C8 FFFA                  DBF     D0,@1
17D0* 0A
17DC| 0242 0007              @2        AND.W   #7,D2
17E0| 60DC                       BRA     MOVSLOW
```

```
17E2|                          ;
17E2|                          ; register useage:
17E2|                          ;
17E2|                          ;       D0 -- start and counter A0 -- from
17E2|                          ;       D1 -- @buffer           A1 -- to
17E2|                          ;       D2 -- length left
17E2|                          ;       D3 -- starting block
17E2|                          ;
17E2|                          MEMWRITE
17E2| 2240                             MOVE.L  D0,A1           ; to = ($144) + blk*512
17E4| 2041                             MOVE.L  D1,A0           ; from = @buffer
17E6| 60D0                             BRA     MOVEMEM
17E8|                          ;
17E8|                          ;
17E8|                          ;
17E8| 4A42                     CDWRITE TST.W   D2              ; is length neg ?
17EA| 6B00 FF28                        BMI     CDERROR
17EE| 4A43                             TST.W   D3              ; is block neg ?
17F0| 6B00 FF22                        BMI     CDERROR
17F4| 6100 FEBA                        BSR     GETINFO
17F8| 4A43                             TST.W   D3
17FA| 6BE6                             BMI     MEMWRITE        ; memory ?
17FC| 4A40                             TST.W   D0
17FE| 6B00 FF1A                        BMI     WRTPRT
1802|                          ;
1802|                          ; register useage:
1802|                          ;
PAGE -  64  MONITOR   FILE: MONIO.TEXT


1802|                          ;       D0 -- used by dskwrt    A0 -- used by dskwrt
1802|                          ;       D1 -- @buffer           A1 -- used by dskwrt
1802|                          ;       D2 -- length left
1802|                          ;       D3 -- length of write
1802|                          ;       D4 -- drv number
1802|                          ;       D5 -- curr block
1802|                          ;       D6 -- last block
1802|                          ;
1802| 0642 01FF                        ADD.W   #511,D2         ; round up to nearest block multiple
1806| E04A                             LSR.W   #8,D2
1808| E24A                             LSR.W   #1,D2           ; number of blocks to read
180A| 0282 0000 007F                   AND.L   #$7F,D2
1810| 2005                             MOVE.L  D5,D0
1812| D082                             ADD.L   D2,D0
1814| 5380                             SUB.L   #1,D0           ; forming start+blockcount-1
1816| BC80                             CMP.L   D0,D6           ; reading beyond the end of vol ?
1818| 6300 FEFA                        BLS     CDERROR
181C| 4267                             CLR.W   -(A7)           ; allocate RC
181E| 2F0F                             MOVE.L  A7,-(A7)        ; push @RC
1820| 3F04                             MOVE.W  D4,-(A7)        ; push drv
1822| 2F02                             MOVE.L  D2,-(A7)        ; push block count
1824| 2F05                             MOVE.L  D5,-(A7)        ; push block number
1826| 2F01                             MOVE.L  D1,-(A7)        ; push @buffer
1828| 6100 F616                        BSR     DSKWRT
182C| 301F                             MOVE.W  (A7)+,D0
182E| 4A00                             TST.B   D0
1830| 6B00 FEE2                        BMI     CDERROR         ; read error ?
1834| 6000 FF7A                        BRA     CDRWXIT
1838|                          ;
1838|                          ;       DRIVER TRANSFER TABLE
1838|                          ;
1838| 6000 ****                DRVRTBL BRA     RDDATA
183C| 6000 ****                        BRA     WRDATA
1840| 6000 ****                        BRA     UNITCLR
1844| 0000 0000                        .WORD   0,0
1848| 6000 ****                        BRA     UNITBSY
184C|                          ;
184C|                          ;       ROUTINE TO PUT A BYTE TO PIA
184C|                          ;
184C| 4A29 0006                PUT     TST.B   OUTCSR(A1)
1850| 6AFA                             BPL.S   PUT             ; wait till ready
1852| 4A29 0002                        TST.B   OUTDATA(A1)     ; reset ready flag
1856| 1340 0002                        MOVE.B  D0,OUTDATA(A1)  ; send lo byte of D0
185A| 4E75                             RTS
185C|                          ;
185C|                          ;       ROUTINE TO GET A BYTE FROM PIA
185C|                          ;
185C| 4A29 0004                GET     TST.B   INCSR(A1)
1860| 6AFA                             BPL.S   GET             ; wait for data avail
1862| 1011                             MOVE.B  INDATA(A1),D0   ; get data and reset flag
1864| 4E75                             RTS
1866|                          ;
1866|                          ;       FORCED IORESULT OF LOST UNIT FOR UNIT I/O TO APPLE
1866|                          ;
1866| 303C 0005                NOAPPLE MOVE.W  #ILSTUNT,D0
PAGE -  65  MONITOR   FILE: MONIO.TEXT


186A| 4E75                             RTS
186C|                          NOAPPL2
186C| E048                             LSR.W   #8,D0           ; get unit number into lower byte
186E| 0C40 0002                        CMP.W   #2,D0           ; unit 1 or 2 ?
1872| 62F2                             BHI.S   NOAPPLE
1874| 4240                             MOVE.W  #INOERR,D0      ; ioresult:=0 if unitwrite to unit 1 or 2
1876| 4E75                             RTS
1878|                          ;
```

```
1878|                          ;       GET IORESULT FROM APPLE INTO D0 AND RETURN
1878|                          ;
1878| 61E2                     GETRSLT BSR     GET             ; get high byte
187A| E148                             LSL.W   #8,D0
187C| 61DE                             BSR     GET             ; get low byte
187E| 4E75                             RTS
1880|                          ;
1880|                          ;       SEND IO COMMAND AND UNIT NUMBER
1880|                          ;
1880| 61CA                     SENDCMD BSR     PUT             ; send IO command byte
1882| E048                             LSR.W   #8,D0           ; get unit number into lower byte
1884| 61C6                             BSR     PUT             ; send unit number
1886| 4E75                             RTS
1888|                          ;
1888|                          ;       SEND HEADER: address count blocknumber
1888|                          ;
1888| 2001                     SENDHDR MOVE.L  D1,D0           ; send 4 byte address
188A| E198                             ROL.L   #8,D0
188C| 61BE                             BSR     PUT             ; high byte first
188E| E198                             ROL.L   #8,D0
1890| 61BA                             BSR     PUT
1892| E198                             ROL.L   #8,D0
1894| 61B6                             BSR     PUT
1896| E198                             ROL.L   #8,D0           ; low byte last
1898| 61B2                             BSR     PUT
189A| 3002                             MOVE.W  D2,D0           ; send 2 byte count
189C| E158                             ROL.W   #8,D0
189E| 61AC                             BSR     PUT             ; high byte
18A0| E158                             ROL.W   #8,D0
18A2| 61A8                             BSR     PUT             ; low byte
18A4| 3003                             MOVE.W  D3,D0           ; send 2 byte blocknumber
18A6| E158                             ROL.W   #8,D0
18A8| 61A2                             BSR     PUT             ; high byte
18AA| E158                             ROL.W   #8,D0
18AC| 619E                             BSR     PUT             ; low byte
18AE| 2041                             MOVE.L  D1,A0
18B0| 4E75                             RTS
18B2|                          ;
18B2|                          ;       UNIT BUSY
18B2|                          ;
184A* 0068
18B2| 5840                     UNITBSY ADD.W   #4,D0           ; setup IO command
18B4| 6100 FDE6                         BSR     CHECKCD
18B8| 6600 FE66                         BNE     CDBUSY
18BC| 4AB8 0118                         TST.L   $118
18C0| 67A4                             BEQ     NOAPPLE
18C2| 61BC                             BSR     SENDCMD         ; send IO command and unit number
PAGE -  66  MONITOR   FILE: MONIO.TEXT


18C4| 6196                             BSR     GET             ; get high byte
18C6| E148                             LSL.W   #8,D0
18C8| 6192                             BSR     GET             ; get low byte
18CA| 3200                             MOVE.W  D0,D1
18CC| 60AA                             BRA     GETRSLT         ; get IORESULT and return
18CE|                          ;
18CE|                          ;       UNIT CLEAR
18CE|                          ;
1842* 008C
18CE| 5240                     UNITCLR ADD.W   #1,D0           ; setup IO command
18D0| 6100 FDCA                         BSR     CHECKCD
18D4| 6600 FE50                         BNE     CDCLEAR
18D8| 4AB8 0118                         TST.L   $118
18DC| 6788                             BEQ     NOAPPLE
18DE| 61A0                             BSR     SENDCMD         ; send IO command and unit number
18E0| 6096                             BRA     GETRSLT         ; get IORESULT and return
18E2|                          ;
18E2|                          ;       READ DATA STREAM
18E2|                          ;
183A* 00A8
18E2| 5440                     RDDATA  ADD.W   #2,D0           ; setup IO command
18E4| 6100 FDB6                         BSR     CHECKCD
18E8| 6600 FE40                         BNE     CDREAD
18EC| 4AB8 0118                         TST.L   $118
18F0| 6700 FF74                         BEQ     NOAPPLE
18F4| 618A                             BSR     SENDCMD         ; send IO command and unit number
18F6| 6190                             BSR     SENDHDR         ; send header and setup A0
18F8| 60**                             BRA.S   @2
18FA| 4A29 0004               @1        TST.B   INCSR(A1)
18FE| 6AFA                             BPL.S   @1              ; wait for data avail
1900| 10D1                             MOVE.B  INDATA(A1),(A0)+; get data and reset flag
18F8* 08
1902| 5342                     @2      SUB.W   #1,D2           ; dec count
1904| 64F4                             BCC.S   @1
1906| 6000 FF70                         BRA     GETRSLT         ; get IORESULT and return
190A|                          ;
190A|                          ;       WRITE DATA STREAM
190A|                          ;
183E* 00CC
190A| 5640                     WRDATA  ADD.W   #3,D0           ; setup IO command
190C| 6100 FD8E                         BSR     CHECKCD
1910| 6600 FED6                         BNE     CDWRITE
1914| 4AB8 0118                         TST.L   $118
1918| 6700 FF52                         BEQ     NOAPPL2
191C| 6100 FF62                         BSR     SENDCMD         ; send IO command and unit number
1920| 6100 FF66                         BSR     SENDHDR         ; send header and setup A0
```

```
1924| 60**                          BRA.S   @2
1926| 4A29 0006              @1     TST.B   OUTCSR(A1)
192A| 6AFA                          BPL.S   @1              ; wait till ready
192C| 4A29 0002                    TST.B   OUTDATA(A1)     ; reset ready flag
1930| 1358 0002                    MOVE.B  (A0)+,OUTDATA(A1)
1924* 0E
1934| 5342                  @2     SUB.W   #1,D2           ; dec count
1936| 64EE                          BCC.S   @1
1938| 6000 FF3E                     BRA     GETRSLT         ; get IORESULT and return
PAGE -  67  MONITOR   FILE: MONIO.TEXT


193C|                         ;
193C|                         ;       DEFAULT DRIVER FOR ALL UNIT IO
193C|                         ;
193C| 327C 0118              DRVR   MOVE.W  #$118,A1        ; set up base register for PIA
1940| 2251                          MOVE.L  (A1),A1
1942| 3F00                          MOVE.W  D0,-(A7)
1944| 4840                          SWAP    D0              ; move machine/dev/unit number to upper word
1946| 301F                          MOVE.W  (A7)+,D0
1948| 0240 001F                     AND.W   #$1F,D0
194C| E148                          LSL.W   #8,D0           ; move unit number to upper byte
194E| 0246 000E                     AND.W   #$E,D6
1952| DC46                          ADD.W   D6,D6           ; map 1,2,4,8 to 0,4,8,16
1954| 41FA FEE2                     LEA     DRVRTBL,A0
1958| 4EF0 6000                     JMP     0(A0,D6)        ; goto to appropiate drvr
195C|                         ;
195C|                         ;       GET UNIT NUMBER AND VALIDATE
195C|                         ;
195C| 225F                   GETUNIT MOVE.L  (A7)+,A1        ; save return address
195E| 3A1F                          MOVE.W  (A7)+,D5        ; get unit number in D5
1960| 67**                          BEQ.S   @3              ; bad if zero
1962| 3005                          MOVE.W  D5,D0           ; save unit number for driver
1964| 0245 001F                     AND.W   #$1F,D5
1968| 0C45 0014                     CMP.W   #MAXU,D5
196C| 62**                          BHI.S   @3              ; bad if greater than MAXU
196E| CBFC 0006                     MULS    #6,D5           ; times UNITBL entry size
1972| 41FA ED7C                     LEA     UNITBL,A0
1976| CC70 5000                     AND.W   0(A0,D5),D6     ; test IO direction
197A| 67**                          BEQ.S   @2
197C| 0C40 0002                     CMP.W   #2,D0           ; unit 1 or 2 ?
1980| 62**                          BHI.S   @6
1982| 0C46 0001                     CMP.W   #INBIT,D6       ; read ?
1986| 66**                          BNE.S   @6
1988|                               GETEBST D4              ; test ebstop, if non zero then inexec
1988| 182D FF87              #      MOVE.B   -121(A5),D4
198C| 67**                          BEQ.S   @6
198E| 41FA FCD6                     LEA     EXECDRVR,A0
1992| 60**                          BRA.S   @7
198C* 06
1986* 0C
1980* 12
1994| 2A30 5002              @6     MOVE.L  2(A0,D5),D5     ; get address of driver
1998| 66**                          BNE.S   @1
199A| 41FA FFA0                     LEA     DRVR,A0
1992* 0A
199E| 4ED1                   @7     JMP     (A1)
1998* 06
19A0| 2045                   @1     MOVE.L  D5,A0           ; move address of driver into A0
19A2| 4ED1                          JMP     (A1)            ; and return
197A* 28
19A4| 7003                   @2     MOVE.L  #IBADMOD,D0     ; bad I/O direction
19A6| 60**                          BRA.S   @4
196C* 3A
1960* 46
19A8| 7002                   @3     MOVE.L  #IBADUNT,D0     ; bad unit number
19AA|                          @4     GETSYSC A0              ; get pointer to syscom in A0
19A6* 02
PAGE -  68  MONITOR   FILE: MONIO.TEXT


19AA| 206D FFFC              #      MOVE.L   -4(A5),A0
19AE| 1080                          MOVE.B  D0,IORSLT(A0)
19B0| 4A59                          TST.W   (A1)+           ; skip JSR (A0) since iorslt is nonzero
19B2| 4ED1                          JMP     (A1)
19B4|                         ;
19B4|                         ;       UNIT CLEAR
19B4|                         ;
19B4| 7C04                   UCLR   MOVE.L  #CLRBIT,D6      ; save UREQ in D6
19B6| 61A4                          BSR     GETUNIT         ; get unit number and validate
19B8| 4E90                          JSR     (A0)            ; go to driver
19BA| 60**                          BRA.S   IOEXIT
19BC|                         ;
19BC|                         ;       UNIT WRITE
19BC|                         ;
19BC| 7C02                   UWRITE MOVE.L  #OUTBIT,D6      ; save UREQ in D6
19BE| 60**                          BRA.S   UIO
19C0|                         ;
19C0|                         ;       UNIT READ
19C0|                         ;
19C0| 7C01                   UREAD  MOVE.L  #INBIT,D6       ; save UREQ in D6
19BE* 02
19C2| 361F                    UIO    MOVE.W  (A7)+,D3        ; get async param
19C4| 4843                          SWAP    D3              ; into the high word
19C6| 361F                          MOVE.W  (A7)+,D3        ; get blocknumber
19C8| 341F                          MOVE.W  (A7)+,D2        ; get length
```

```
19CA| 221F                          MOVE.L  (A7)+,D1        ; get address
19CC| 618E                          BSR     GETUNIT         ; get unit number and validate
19CE| 4E90                          JSR     (A0)            ; go to driver
19D0|                       IOEXIT  GETSYSC A0              ; get pointer to syscom in A0
19BA* 14
19D0| 206D FFFC             #       MOVE.L   -4(A5),A0
19D4| 1080                          MOVE.B  D0,IORSLT(A0)
19D6| 4E75                          RTS
19D8|                       ;
19D8|                       ;       UNIT BUSY
19D8|                       ;
19D8| 7C08                  UBUSY   MOVE.L  #BSYBIT,D6      ; save UREQ in D6
19DA| 6180                          BSR     GETUNIT         ; get unit number and validate
19DC| 4E90                          JSR     (A0)            ; go to driver
19DE| 241F                          MOVE.L  (A7)+,D2
19E0| 0241 0001                     ANDI.W  #1,D1
19E4| 3F01                          MOVE.W  D1,-(A7)
19E6| 2F02                          MOVE.L  D2,-(A7)
19E8| 60E6                          BRA.S   IOEXIT
19EA|                       ;
19EA|                       ;       FUNCTION ADDRDRVR:PTR
19EA|                       ;
19EA|                       ;       returns the address of the default driver
19EA|                       ;
19EA|                       ;       stack:
19EA|                       ;
19EA|                       ;       4       function result
19EA|                       ;       0       return address
19EA|                       ;
PAGE -  69  MONITOR   FILE: MONIO.TEXT


19EA|                       ADDRDRVR
19EA| 41FA FF50                      LEA     DRVR,A0
19EE| 2F48 0004                      MOVE.L  A0,4(A7)
19F2| 4E75                          RTS
19F4|                       ;
19F4|                       ;       IORESULT
19F4|                       ;
19F4| 225F                  ZZIORES MOVE.L  (A7)+,A1
19F6|                               GETSYSC A0              ; get ioresult
19F6| 206D FFFC             #       MOVE.L   -4(A5),A0
19FA| 1010                          MOVE.B  (A0),D0
19FC| 4880                          EXT.W   D0
19FE| 3F00                          MOVE.W  D0,-(A7)        ; and return it
1A00| 4ED1                          JMP     (A1)
1A02|                       ;
1A02|                       ;       ALLOCATE DYNAMIC MEMORY
1A02|                       ;
1A02| 321F                  MNEW    MOVE.W  (A7)+,D1        ; get number of words into D1
1A04|                               GETNP   A0              ; get current heap top in A0
1A04| 206D FFF4             #       MOVE.L   -12(A5),A0
1A08| 225F                          MOVE.L  (A7)+,A1        ; get address of ptr param in A1
1A0A| 2288                          MOVE.L  A0,(A1)         ; set pointer param to new mem space
1A0C| D0C1                          ADD.W   D1,A0           ; point A0 above dyn mem area
1A0E| D0C1                          ADD.W   D1,A0           ; byte wise
1A10|                               PUTNP   A0              ; save new heap top
1A10| 2B48 FFF4             #       MOVE.L   A0,-12(A5)
1A14| 4E75                          RTS
1A16|                       ;
1A16|                       ;       MARK HEAP
1A16|                       ;
1A16| 225F                  MMRK    MOVE.L  (A7)+,A1        ; get address of ptr param in A1
1A18|                               GETNP   (A1)            ; save top of heap in pointer param
1A18| 22AD FFF4             #       MOVE.L   -12(A5),(A1)
1A1C| 4E75                          RTS
1A1E|                       ;
1A1E|                       ;       RELEASE HEAP
1A1E|                       ;
1A1E| 205F                  MRLS    MOVE.L  (A7)+,A0        ; get address of ptr param in A0
1A20|                               PUTNP   (A0)            ; cut back heap
1A20| 2B50 FFF4             #       MOVE.L   (A0),-12(A5)
1A24| 4E75                          RTS
1A26|                       ;
1A26|                       ;       MEMAVAIL -- number of free words in data area
1A26|                       ;
1A26| 245F                  MEMA    MOVE.L  (A7)+,A2
1A28| 204F                          MOVE.L  A7,A0
1A2A|                               GETNP   A1
1A2A| 226D FFF4             #       MOVE.L   -12(A5),A1
1A2E| 91C9                          SUB.L   A1,A0           ; get number of bytes left
1A30| 2008                          MOVE.L  A0,D0
1A32| 0280 FFFF FFFE               AND.L   #$FFFFFFFE,D0   ; make into number of words
1A38| E298                          ROR.L   #1,D0
1A3A| 2F00                          MOVE.L  D0,-(A7)        ; return number of words
1A3C| 4ED2                          JMP     (A2)
1A3E|
PAGE -  70  MONITOR   FILE:MONITOR.TEXT


1A3E|
1A3E|                       ;
1A3E|                               .INCLUDE PRTDRVR.TEXT
1A3E|                       ;
1A3E|                       ;       FILENAME: PRTDRVR
1A3E|                       ;
```

```
1A3E|
1A3E|                          ;
1A3E|                          ;       D0 -- unit number       word
1A3E|                          ;       D1 -- buffer address    long
1A3E|                          ;       D2 -- number bytes      word
1A3E|                          ;       D3 -- async parm,block# long
1A3E|                          ;       A0 -- address of base   long
1A3E|                          ;
1A3E| 0246 000E                PRTDRVR ANDI    #$E,D6                  ; force entry type for indexed jump
1A42| 4EFB 60**                        JMP     ACTNTBL(D6)             ;  and do it
1A46|                          ;
1A44* 02
1A46|                          ACTNTBL
1A46| 60**                             BRA.S   PRTREAD
1A48| 60**                             BRA.S   PRTWRT
1A4A| 60**                             BRA.S   PRTCLR
1A4C| 4E71                             NOP                             ; no entry here
1A4E|                          ;       BRA.S   PRTBUSY
1A4E| 4241                     PRTBUSY CLR     D1                      ; assume unitbusy is false
1A46* 08
1A50| 6000 ****                PRTREAD BRA     PRTEXIT
1A4A* 08
1A54| 48E7 FC00                PRTCLR  MOVEM.L D0-D5,-(A7)
1A58| 61**                             BSR.S   PRTSETUP
1A5A| 183C 0011                       MOVE.B  #$11,D4
1A5E| 6100 ****                       BSR     PRTSEND
1A62| 4CDF 003F                       MOVEM.L (A7)+,D0-D5
1A66| 6000 ****                       BRA     PRTEXIT
1A6A|                          ;
1A6A|                          ;
1A6A|                          ;
1A58* 10
1A6A|                          PRTSETUP
1A6A|                                  GETPRBA A0                      ; get base of pia
1A6A| 206D FF54                #       MOVE.L  -172(A5),A0
1A6E| 117C 000E 0060                   MOVE.B  #$0E,PCR(A0)            ; set ctrl CA2 high
1A74| 117C 00FF 0018                   MOVE.B  #$FF,DDRA(A0)           ; port A bits to output
1A7A| 0890 0003                       BCLR    #3,ORB(A0)              ; bidir drvr to output
1A7E| 0890 0002                       BCLR    #2,ORB(A0)              ; enable the driver
1A82| 4E75                           RTS
1A84|                          ;
1A84|                          ;
1A84|                          ;
1A48* 3A
1A84| 48E7 FC00                PRTWRT  MOVEM.L D0-D5,-(A7)
1A88| 4843                             SWAP    D3                      ; get async parm
1A8A| 61DE                             BSR.S   PRTSETUP
1A8C| 2241                             MOVE.L  D1,A1
1A8E| 5342                             SUB.W   #$1,D2                  ; adjust for the DBF
PAGE -  71  MONITOR   FILE: PRTDRVR.TEXT


1A90| 0C03 000C                       CMP.B   #$0C,D3
1A94| 67**                            BEQ.S   @10
1A96| 1819                     @1      MOVE.B  (A1)+,D4                ; get character into D4
1A98| 0803 0002                       BTST    #2,D3                   ; dle?
1A9C| 66**                            BNE.S   @3
1A9E| 0C04 0010                       CMP.B   #$10,D4                 ; is it a dle character
1AA2| 66**                            BNE.S   @3
1AA4| 5342                           SUB.W   #1,D2                   ; adjust character count
1AA6| 4244                           CLR.W   D4
1AA8| 1819                           MOVE.B  (A1)+,D4                ; get # of spaces to send
1AAA| 0404 0020                       SUB.B   #$20,D4
1AAE| 3A04                           MOVE.W  D4,D5
1AB0| 183C 0020                       MOVE.B  #$20,D4                 ; set character to space
1AB4| 61**                     @2      BSR.S   PRTSEND                 ; send the suckers
1AB6| 51CD FFFC                       DBF     D5,@2
1ABA| 60**                            BRA.S   @4
1AA2* 18
1A9C* 1E
1ABC| 61**                     @3      BSR.S   PRTSEND
1ABE| 0803 0003                       BTST    #3,D3                   ; send LF
1AC2| 66**                            BNE.S   @4
1AC4| 0C04 000D                       CMP.B   #$0D,D4                 ; was last character a CR
1AC8| 66**                            BNE.S   @4
1ACA| 183C 000A                       MOVE.B  #$0A,D4                 ; set character to LF
1ACE| 61**                            BSR.S   PRTSEND
1AC8* 06
1AC2* 0C
1ABA* 14
1AD0| 51CA FFC4                @4      DBF     D2,@1                   ; are we done yet?
1AD4| 60**                            BRA.S   @11
1A94* 40
1AD6| 1819                     @10     MOVE.B  (A1)+,D4
1AD8| 61**                            BSR.S   PRTSEND                 ; no interpretation to do
1ADA| 51CA FFFA                       DBF     D2,@10                  ; more to do?
1AD4* 08
1ADE| 4CDF 003F                @11     MOVEM.L (A7)+,D0-D5
1A68* 007A
1A52* 0090
1AE2| 4240                     PRTEXIT CLR     D0                      ; set IORSLT value
1AE4| 4E75                           RTS
1AE6|                          ;
1AE6|                          ;
1AE6|                          ;
1AD8* 0C
```

```
1ACE* 16
1ABC* 28
1AB4* 30
1A60* 0086
1AE6| 0810 0000              PRTSEND BTST    #0,IRB(A0)
1AEA| 66FA                           BNE.S   PRTSEND                 ; wait for PE/
1AEC| 117C 007F 0068                 MOVE.B  #$7F,IFR(A0)            ; clear CA1 flag
1AF2| 1144 0008                      MOVE.B  D4,ORA(A0)              ; output the data
1AF6| 117C 000C 0060                 MOVE.B  #$0C,PCR(A0)            ; set ctrl CA2 low
1AFC| 117C 000E 0060                 MOVE.B  #$0E,PCR(A0)            ; set ctrl CA2 high
1B02| 0828 0001 0068         @1      BTST    #1,IFR(A0)              ; CA1 set yet
PAGE -  72  MONITOR   FILE: PRTDRVR.TEXT


1B08| 67F8                           BEQ.S   @1
1B0A| 4E75                           RTS
1B0C|                        ;
1B0C|
1B0C|                        ;
1B0C|                        .INCLUDE FILPRC1.TEXT
1B0C|                        ;
1B0C|                        ;       FILENAME FILPRC1
1B0C|                        ;
1B0C|
1B0C|                        ;
1B0C|                        ;       FINIT(F:FIB; W:WP; RECWORDS:INTEGER);
1B0C|                        ;
1B0C| 205F                 FINIT   MOVE.L  (A7)+,A0                ; return address
1B0E| 301F                         MOVE.W  (A7)+,D0                ; recwords
1B10| 225F                         MOVE.L  (A7)+,A1                ; ptr to window
1B12| 245F                         MOVE.L  (A7)+,A2                ; file ptr
1B14| 426A 0006                   MOVE.W  #0,FSTATE(A2)           ; fstate:=fjandw
1B18| 422A 000A                   MOVE.B  #0,FISOPEN(A2)          ; fisopen:=false
1B1C| 157C 0001 0005             MOVE.B  #1,FEOF(A2)             ; feof:=true
1B22| 157C 0001 0004             MOVE.B  #1,FEOLN(A2)            ; feoln:=true
1B28| 2489                       MOVE.L  A1,FWINDOW(A2)          ; fwindow:=ptr to window
1B2A| 4A40                       TST.W   D0
1B2C| 67**                       BEQ.S   @1
1B2E| 0C40 FFFE                 CMP.W   #-2,D0                  ; if recwords = 0 or -2 then
1B32| 66**                       BNE.S   @2
1B2C* 06
1B34| 4229 0001           @1      MOVE.B  #0,1(A1)                ; fwindow^[1]:=null
1B38| 357C 0001 0008             MOVE.W  #1,FRECSZ(A2)           ; frecsize:=1
1B3E| 4A40                       TST.W   D0                      ; if recwords = 0 then
1B40| 66**                       BNE.S   FINITX
1B42| 357C 0001 0006             MOVE.W  #1,FSTATE(A2)           ; fstate:=fneedchar
1B48| 60**                       BRA.S   FINITX
1B32* 16
1B4A| 4A40                 @2      TST.W   D0                      ; if recwords < 0 then
1B4C| 6A**                       BPL.S   @3
1B4E| 4292                       MOVE.L  #NIL,FWINDOW(A2)        ; fwindow:=nil
1B50| 426A 0008                   MOVE.W  #0,FRECSZ(A2)           ; frecsize:=0
1B54| 60**                       BRA.S   FINITX
1B4C* 08
1B56| D040                 @3      ADD.W   D0,D0
1B58| 3540 0008                   MOVE.W  D0,FRECSZ(A2)           ; frecsize:=2*recwords
1B54* 06
1B48* 12
1B40* 1A
1B5C| 4ED0                 FINITX  JMP     (A0)
1B5E|                        ;
1B5E|                        ;       SCANTITLE(VAR FTITLE:STRING;
1B5E|                        ;                 VAR FVID:VID;
1B5E|                        ;                 VAR FTID:TID;
1B5E|                        ;                 VAR FBLOCKS:INTEGER;
1B5E|                        ;                 VAR FKIND:FILEKIND):BOOLEAN
1B5E|                        ;
1B5E|                        ;       stack
1B5E|                        ;
PAGE -  73  MONITOR   FILE: FILPRC1.TEXT


1B5E|                        ;               28      FUNC RESULT
1B5E|                        ;               24      @FTITLE
1B5E|                        ;               20      @FVID
1B5E|                        ;               16      @FTID
1B5E|                        ;               12      @FBLOCKS
1B5E|                        ;               8       @FKIND
1B5E|                        ;               4       RETURN ADDRESS
1B5E|                        ;               0       Old A6
1B5E|                        ;
1B5E| 4E56 FF00             SCANTTL LINK    A6,#-256                ; allocate space for string[255]
1B62| 6100 ED1E                   BSR     OVFCHK
1B66| 206E 0014                   MOVE.L  20(A6),A0
1B6A| 4210                       MOVE.B  #0,(A0)                 ; fvid:=''
1B6C| 206E 0010                   MOVE.L  16(A6),A0
1B70| 4210                       MOVE.B  #0,(A0)                 ; ftid:=''
1B72| 206E 000C                   MOVE.L  12(A6),A0
1B76| 4250                       MOVE.W  #0,(A0)                 ; fblocks:=0
1B78| 206E 0008                   MOVE.L  8(A6),A0
1B7C| 4228 0001                   MOVE.B  #UNTYPFL,1(A0)          ; fkind:=untypedfile
1B80| 42AE 001C                   MOVE.L  #0,28(A6)               ; scantitle:=false
1B84| 206E 0018                   MOVE.L  24(A6),A0               ; get address of string
1B88| 43EE FF00                   LEA     -256(A6),A1             ; get address of string copy
1B8C| 1018                       MOVE.B  (A0)+,D0                ; get string length
1B8E| 6700 ****                   BEQ     SCANTTLX                ; exit if zero
```

```
1B92| 2D49 0018                    MOVE.L  A1,24(A6)               ; save pointer to the copy
1B96| 12C0                         MOVE.B  D0,(A1)+                ; copy the length
1B98| 12D8                SCNSTRC  MOVE.B  (A0)+,(A1)+             ; copy the string
1B9A| 5300                         SUB.B   #1,D0
1B9C| 66FA                         BNE.S   SCNSTRC
1B9E| 266E 0014                    MOVE.L  20(A6),A3               ; @fvid
1BA2| 206E 0018                    MOVE.L  24(A6),A0               ; @ftitle
1BA6| 7201                         MOVE.L  #1,D1                   ; i:=1
1BA8| B210                @1       CMP.B   (A0),D1                 ; while i<=len(ftitle) do
1BAA| 6E**                         BGT.S   FAIRTTL
1BAC| 1030 1000                    MOVE.B  0(A0,D1.W),D0           ; ch:=ftitle[i]
1BB0| 0C00 0020                    CMP.B   #$20,D0                 ; if ch<=' ' then
1BB4| 6E**                         BGT.S   @3
1BB6| 6100 ****                    BSR     DEL1CH                  ; delete(ftitle,1,i)
1BBA| 60EC                         BRA.S   @1
1BB4* 06
1BBC| 6100 ****            @3       BSR     UPSHFT                  ; upshift(ch)
1BC0| 1180 1000                    MOVE.B  D0,0(A0,D1.W)           ; ftitle[i]:=ch
1BC4| 5241                         ADD.W   #1,D1                   ; i:=i+1
1BC6| 60E0                         BRA.S   @1
1BAA* 1C
1BC8| 4A10                FAIRTTL  TST.B   (A0)                    ; if len(ftitle)=0 then exit
1BCA| 6700 ****                    BEQ     SCANTTLX
1BCE| 0C28 002A 0001                CMP.B   #42,1(A0)               ; if ftitle[1] = '*' then
1BD4| 66**                         BNE.S   NOTSTAR
1BD6|                              GETSYV  A1                      ;   fvid:=syvid
1BD6| 226D FFD4             #       MOVE.L  -44(A5),A1
1BDA| 244B                         MOVE.L  A3,A2
1BDC| 24D9                         MOVE.L  (A1)+,(A2)+
1BDE| 24D9                         MOVE.L  (A1)+,(A2)+
1BE0| 24D9                         MOVE.L  (A1)+,(A2)+
PAGE -  74  MONITOR   FILE: FILPRC1.TEXT


1BE2| 24D9                         MOVE.L  (A1)+,(A2)+
1BE4| 24D9                         MOVE.L  (A1)+,(A2)+
1BE6| 24D9                         MOVE.L  (A1)+,(A2)+
1BE8| 7201                         MOVE.L  #1,D1
1BEA| 6100 ****                    BSR     DEL1CH                  ;   delete(ftitle,1,1)
1BD4* 18
1BEE| 4281                NOTSTAR  MOVE.L  #0,D1
1BF0| 60**                         BRA.S   @7
1BF2| 5201                @0       ADD.B   #1,D1
1BF4| 0C30 003A 1000                CMP.B   #58,0(A0,D1.W)          ; i:=pos(':',ftitle)
1BFA| 67**                         BEQ.S   @1
1BF0* 0A
1BFC| B210                @7       CMP.B   (A0),D1
1BFE| 66F2                         BNE.S   @0
1C00| 4281                         MOVE.L  #0,D1
1C02| 60**                         BRA.S   @2
1BFA* 08
1C04| 0C01 0001                @1       CMP.B   #1,D1                   ; if i<=1 then
1C08| 66**                         BNE.S   @5
1C02* 06
1C0A| 244B                @2       MOVE.L  A3,A2                   ;   if len(fvid)=0 then
1C0C| 4A12                         TST.B   (A2)
1C0E| 66**                         BNE.S   @3
1C10|                              GETDKV  A1                      ;     fvid:=dkvid
1C10| 226D FFCC             #       MOVE.L  -52(A5),A1
1C14| 24D9                         MOVE.L  (A1)+,(A2)+
1C16| 24D9                         MOVE.L  (A1)+,(A2)+
1C18| 24D9                         MOVE.L  (A1)+,(A2)+
1C1A| 24D9                         MOVE.L  (A1)+,(A2)+
1C1C| 24D9                         MOVE.L  (A1)+,(A2)+
1C1E| 24D9                         MOVE.L  (A1)+,(A2)+
1C0E* 10
1C20| 0C01 0001                @3       CMP.B   #1,D1                   ;   if i=1 then
1C24| 66**                         BNE.S   @4
1C26| 6100 ****                    BSR     DEL1CH                  ;     delete(ftitle,1,1)
1C24* 04
1C2A| 60**                @4       BRA.S   TSTFVID
1C08* 22
1C2C| 5301                @5       SUB.B   #1,D1
1C2E| 0C01 0017                    CMP.B   #23,D1                  ; if i-1 <= vidleng then
1C32| 6E**                         BGT.S   TSTFVID
1C34| 3F01                         MOVE.W  D1,-(A7)
1C36| 2F08                         MOVE.L  A0,-(A7)
1C38| 3F3C 0001                    MOVE.W  #1,-(A7)
1C3C| 3F01                         MOVE.W  D1,-(A7)
1C3E| 2F0B                         MOVE.L  A3,-(A7)
1C40| 6100 ****                    BSR     TTLCOPY                 ;   fvid:=copy(ftitle,1,i-1)
1C44| 361F                         MOVE.W  (A7)+,D3
1C46| 5243                         ADD.W   #1,D3
1C48| 7201                @6       MOVE.L  #1,D1
1C4A| 6100 ****                    BSR     DEL1CH                  ;   delete(ftitle,1,i)
1C4E| 5303                         SUB.B   #1,D3
1C50| 66F6                         BNE.S   @6
1C32* 1E
1C2A* 26
PAGE -  75  MONITOR   FILE: FILPRC1.TEXT


1C52| 4A13                TSTFVID  TST.B   (A3)                    ; if len(fvid)=0 then exit
1C54| 6700 ****                    BEQ     SCANTTLX
1C58| 4281                         MOVE.L  #0,D1
1C5A| 60**                         BRA.S   @14
```

```
1C5C| 5241                  @0      ADD.W   #1,D1
1C5E| 0C30 005B 1000                CMP.B   #91,0(A0,D1.W)          ; i:=pos('[',ftitle)
1C64| 67**                          BEQ.S   @1
1C5A* 0A
1C66| B210                  @14     CMP.B   (A0),D1
1C68| 66F2                          BNE.S   @0
1C6A| 4281                          MOVE.L  #0,D1
1C6C| 60**                          BRA.S   @2                      ; if i>0 then
1C64* 08
1C6E| 5301                  @1      SUB.B   #1,D1                   ;   i:=i-1
1C70| 60**                          BRA.S   @3                      ; else
1C6C* 04
1C72| 1210                  @2      MOVE.B  (A0),D1                 ;   i:=length(ftitle)
1C74| 4282                          MOVE.L  #0,D2                   ;   ok:=false
1C70* 04
1C76| 0C01 000F              @3      CMP.B   #15,D1                  ; if i <= tidleng then
1C7A| 6E00 ****                     BGT     DOSUFIX
1C7E| 4A01                          TST.B   D1                      ; if i>0 then
1C80| 67**                          BEQ.S   @5
1C82| 2448                          MOVE.L  A0,A2                   ;   get address of src
1C84| 226E 0010                     MOVE.L  16(A6),A1               ;   get address of dst
1C88| 4A1A                          TST.B   (A2)+                   ;   skip src string length
1C8A| 1001                          MOVE.B  D1,D0                   ;   get string length
1C8C| 12C0                          MOVE.B  D0,(A1)+                ;   copy string length
1C8E| 12DA                  @4      MOVE.B  (A2)+,(A1)+             ;   ftid:=copy(ftitle,1,i)
1C90| 5300                          SUB.B   #1,D0
1C92| 66FA                          BNE.S   @4
1C94| 1410                          MOVE.B  (A0),D2                 ;   delete(ftitle,1,i)
1C96| 9401                          SUB.B   D1,D2
1C98| D0C1                          ADD.W   D1,A0
1C9A| 1082                          MOVE.B  D2,(A0)
1C80* 1A
1C9C| 4A10                  @5      TST.B   (A0)                    ; if len(ftitle)=0 then
1C9E| 66**                          BNE.S   @13
1CA0| 7401                          MOVE.L  #1,D2                   ;   ok:=true
1CA2| 6000 ****                     BRA     DOSUFIX                 ; else
1C9E* 06
1CA6| 4282                  @13     MOVE.L  #0,D2                   ;   ok:=false
1CA8| 4283                          MOVE.L  #0,D3
1CAA| 60**                          BRA.S   @15
1CAC| 5203                  @6      ADD.B   #1,D3
1CAE| 0C30 005D 3000                CMP.B   #93,0(A0,D3.W)          ; rbrack:=pos(']',ftitle)
1CB4| 67**                          BEQ.S   @7
1CAA* 0A
1CB6| B610                  @15     CMP.B   (A0),D3
1CB8| 66F2                          BNE.S   @6
1CBA| 4283                          MOVE.L  #0,D3
1CB4* 06
1CBC| 0C03 0002              @7      CMP.B   #2,D3                   ; if rbrack=2 then
1CC0| 66**                          BNE.S   @8
1CC2| 7401                          MOVE.L  #1,D2                   ;   ok:=true
PAGE -  76  MONITOR   FILE: FILPRC1.TEXT


1CC4| 60**                          BRA.S   DOSUFIX                 ; else
1CC0* 04
1CC6| 6F**                  @8      BLE.S   DOSUFIX                 ;   if rbrack>2 then
1CC8| 7401                          MOVE.L  #1,D2                   ;     ok:=true
1CCA| 7202                          MOVE.L  #2,D1                   ;     i:=2
1CCC| 4240                  @9      CLR.W   D0                      ;     repeat
1CCE| 1030 1000                     MOVE.B  0(A0,D1.W),D0           ;       ch:=ftitle[i]
1CD2| 0C00 0030                     CMP.B   #'0',D0                 ;       if ch in DIGITS then
1CD6| 6D**                          BLT.S   @10
1CD8| 0C00 0039                     CMP.B   #'9',D0
1CDC| 6E**                          BGT.S   @10
1CDE| 226E 000C                     MOVE.L  12(A6),A1
1CE2| 3811                          MOVE.W  (A1),D4
1CE4| C9FC 000A                     MULS    #10,D4                  ;         fblocks:=fblocks*10
1CE8| D840                          ADD.W   D0,D4                   ;         fblocks:=fblocks+ord(ch)
1CEA| 0444 0030                     SUB.W   #'0',D4                 ;         fblocks:=fblocks-ord('0')
1CEE| 3284                          MOVE.W  D4,(A1)
1CF0| 60**                          BRA.S   @11                     ;       else
1CDC* 14
1CD6* 1A
1CF2| 4282                  @10     MOVE.L  #0,D2                   ;         ok:=false
1CF0* 02
1CF4| 5201                  @11     ADD.B   #1,D1                   ;       i:=i+1
1CF6| B601                          CMP.B   D1,D3                   ;     until i=rbrack or not ok
1CF8| 67**                          BEQ.S   @12
1CFA| 4A42                          TST.W   D2
1CFC| 66CE                          BNE.S   @9
1CF8* 04
1CFE| 0C01 0003              @12     CMP.B   #3,D1                   ;     if i=3 then
1D02| 66**                          BNE.S   DOSUFIX
1D04| 0C03 0003                     CMP.B   #3,D3                   ;     if rbrack=3 then
1D08| 66**                          BNE.S   DOSUFIX
1D0A| 5301                          SUB.B   #1,D1                   ;     if ftitle[i-1]='*' then
1D0C| 0C30 002A 1000                CMP.B   #42,0(A0,D1.W)
1D12| 66**                          BNE.S   DOSUFIX
1D14| 226E 000C                     MOVE.L  12(A6),A1               ;       fblocks:=-1
1D18| 32BC FFFF                     MOVE.W  #-1,(A1)
1D1C| 7401                          MOVE.L  #1,D2                   ;       ok:=true
1D12* 0A
1D08* 14
1D02* 1A
1CC6* 56
```

```
1CC4* 58
1CA4* 007A
1C7C* 00A2
1D1E| 2D42 001C             DOSUFIX MOVE.L  D2,28(A6)               ; scantitle:=ok
1D22| 6700 ****                     BEQ     SCANTTLX                ; if not ok then exit
1D26| 206E 0010                     MOVE.L  16(A6),A0
1D2A| 1010                         MOVE.B  (A0),D0
1D2C| 0C00 0005                     CMP.B   #5,D0                   ; if len(ftid) > 5 then
1D30| 6D00 ****                     BLT     SCANTTLX
1D34| 2F08                  DOSFX   MOVE.L  A0,-(A7)                ;   push source string address
1D36| 5940                         SUB.W   #4,D0                   ;   compute length-4
1D38| 3F00                         MOVE.W  D0,-(A7)                ;   push starting index
1D3A| 3F3C 0005                     MOVE.W  #5,-(A7)                ;   push size to copy
PAGE -  77  MONITOR   FILE: FILPRC1.TEXT


1D3E| 2F2E 0018                     MOVE.L  24(A6),-(A7)            ;   push address of result
1D42| 6100 ****                     BSR     TTLCOPY
1D46| 246E 0008                     MOVE.L  8(A6),A2
1D4A| 206E 0018                     MOVE.L  24(A6),A0
1D4E| 43FA ****                     LEA     SFXTEXT,A1
1D52| 6100 ****                     BSR     CMPSFX
1D56| 67**                         BEQ.S   @1
1D58| 157C 0003 0001                MOVE.B  #TEXTFILE,1(A2)
1D5E| 60**                         BRA.S   SCANTTLX
1D56* 08
1D60| 206E 0018             @1      MOVE.L  24(A6),A0
1D64| 43FA ****                     LEA     SFXCODE,A1
1D68| 6100 ****                     BSR     CMPSFX
1D6C| 67**                         BEQ.S   @2
1D6E| 157C 0002 0001                MOVE.B  #CODEFILE,1(A2)
1D74| 60**                         BRA.S   SCANTTLX
1D6C* 08
1D76| 206E 0018             @2      MOVE.L  24(A6),A0
1D7A| 43FA ****                     LEA     SFXBACK,A1
1D7E| 6100 ****                     BSR     CMPSFX
1D82| 67**                         BEQ.S   @3
1D84| 157C 0003 0001                MOVE.B  #TEXTFILE,1(A2)
1D8A| 60**                         BRA.S   SCANTTLX
1D82* 08
1D8C| 206E 0018             @3      MOVE.L  24(A6),A0
1D90| 43FA ****                     LEA     SFXINFO,A1
1D94| 6100 ****                     BSR     CMPSFX
1D98| 67**                         BEQ.S   @4
1D9A| 157C 0004 0001                MOVE.B  #INFOFILE,1(A2)
1DA0| 60**                         BRA.S   SCANTTLX
1D98* 08
1DA2| 206E 0018             @4      MOVE.L  24(A6),A0
1DA6| 43FA ****                     LEA     SFXGRAF,A1
1DAA| 6100 ****                     BSR     CMPSFX
1DAE| 67**                         BEQ.S   @5
1DB0| 157C 0006 0001                MOVE.B  #GRAFFILE,1(A2)
1DB6| 60**                         BRA.S   SCANTTLX
1DAE* 08
1DB8| 206E 0018             @5      MOVE.L  24(A6),A0
1DBC| 43FA ****                     LEA     SFXFOTO,A1
1DC0| 6100 ****                     BSR     CMPSFX
1DC4| 67**                         BEQ.S   SCANTTLX
1DC6| 157C 0007 0001                MOVE.B  #FOTOFILE,1(A2)
1DC4* 06
1DB6* 14
1DA0* 2A
1D8A* 40
1D74* 56
1D5E* 6C
1D32* 009A
1D24* 00A8
1C56* 0176
1BCC* 0200
1B90* 023C
1DCC|                       SCANTTLX
PAGE -  78  MONITOR   FILE: FILPRC1.TEXT


1DCC| 4E5E                         UNLK    A6
1DCE| 205F                         MOVE.L  (A7)+,A0        ; pop return address
1DD0| DEFC 0016                   ADD.W   #22,A7          ; delete parameters
1DD4| 4ED0                         JMP     (A0)
1DD6|                       ;
1D50* 0086
1DD6| 05                    SFXTEXT .BYTE   5
1DD7| 2E 54 45 58 54                .ASCII  '.TEXT'
1DDC|                       ;
1D66* 0076
1DDC| 05                    SFXCODE .BYTE   5
1DDD| 2E 43 4F 44 45                .ASCII  '.CODE'
1DE2|                       ;
1D7C* 0066
1DE2| 05                    SFXBACK .BYTE   5
1DE3| 2E 42 41 43 4B                .ASCII  '.BACK'
1DE8|                       ;
1D92* 0056
1DE8| 05                    SFXINFO .BYTE   5
1DE9| 2E 49 4E 46 4F                .ASCII  '.INFO'
1DEE|                       ;
1DA8* 0046
```

```
1DEE| 05                     SFXGRAF .BYTE   5
1DEF| 2E 47 52 41 46                .ASCII  '.GRAF'
1DF4|                        ;
1DBE* 0036
1DF4| 05                     SFXFOTO .BYTE   5
1DF5| 2E 46 4F 54 4F                .ASCII  '.FOTO'
1DFA|                        ;
1DFA|                        ;       DEL1CH -- deletes one character from FTITLE
1DFA|                        ;
1DFA|                        ;       D1      is the character position to delete from
1DFA|                        ;       A0      is a ptr to FTITLE
1DFA|                        ;
1DFA|                        ;       A1-A2   used as scratch
1DFA|                        ;       D2      used as scratch
1DFA|                        ;
1C4C* 01AE
1C28* 01D2
1BEC* 020E
1BB8* 0242
1DFA| 5310                   DEL1CH  SUB.B   #1,(A0)                 ; decrement length
1DFC| 3401                           MOVE.W  D1,D2
1DFE| 9410                           SUB.B   (A0),D2                 ; IF I<=LEN(FTITLE) THEN
1E00| 6E**                           BGT.S   @4
1E02| 4402                           NEG.B   D2
1E04| 45F0 1000                     LEA     0(A0,D1.W),A2           ; dst ptr
1E08| 43F0 1001                     LEA     1(A0,D1.W),A1           ; src ptr
1E0C| 14D9                   @2      MOVE.B  (A1)+,(A2)+             ; copy each byte
1E0E| 5342                           SUB.W   #1,D2
1E10| 64FA                           BCC.S   @2
1E00* 10
1E12| 4E75                   @4      RTS
1E14|                        ;
1E14|                        ;       CMPSFX -- compares 5 chars at A0 to 5 chars at A1
PAGE -  79  MONITOR   FILE: FILPRC1.TEXT


1E14|                        ;                 equal or not equal is set into D0
1E14|                        ;
1DC2* 0052
1DAC* 0068
1D96* 007E
1D80* 0094
1D6A* 00AA
1D54* 00C0
1E14| 4240                   CMPSFX  MOVE.W  #0,D0                   ; assume false
1E16| B388                           CMPM.L  (A0)+,(A1)+             ; first 4 equal ?
1E18| 66**                           BNE.S   CMPSFXX
1E1A| B348                           CMPM.W  (A0)+,(A1)+             ; next 2 equal ?
1E1C| 66**                           BNE.S   CMPSFXX
1E1E| 7001                           MOVE.L  #1,D0                   ; result is true
1E1C* 02
1E18* 06
1E20| 4A40                   CMPSFXX TST.W   D0
1E22| 4E75                           RTS
1E24|                        ;
1E24|                        ;       TTLCOPY -- DST:=COPY(SRC,I,J)
1E24|                        ;
1E24|                        ; Parameters: ST.L - Source string address
1E24|                        ;             ST.W - Starting index
1E24|                        ;             ST.W - Size to copy
1E24|                        ;             ST.L - Address of result
1E24|                        ;
1D44* 00E0
1C42* 01E2
1E24| 48E7 E0C0               TTLCOPY MOVEM.L D0-D2/A0-A1,-(SP)
1E28| 206F 0018                       MOVE.L  24(SP),A0       ; Address for result
1E2C| 302F 001C                       MOVE.W  28(SP),D0       ; Size to copy
1E30| 322F 001E                       MOVE.W  30(SP),D1       ; Index
1E34| 5341                           SUBQ.W  #1,D1
1E36| 226F 0020                       MOVE.L  32(SP),A1       ; Address of source string
1E3A| 4242                           CLR.W   D2              ; to get length of source
1E3C| 1419                           MOVE.B  (A1)+,D2        ;
1E3E| 9441                           SUB.W   D1,D2
1E40| 9440                           SUB.W   D0,D2
1E42| 6D**                           BLT.S   Y.ERROR         ; Error if too little source
1E44| D2C1                           ADDA.W  D1,A1           ; Point to first byte to copy
1E46| 10C0                           MOVE.B  D0,(A0)+        ; Store result length
1E48| 60**                           BRA.S   Y.TEST
1E4A| 10D9                   Y.LOOP: MOVE.B  (A1)+,(A0)+
1E48* 02
1E4C| 5340                   Y.TEST: SUBQ.W  #1,D0
1E4E| 6AFA                           BPL.S   Y.LOOP
1E50| 2F6F 0014 0020         Y.LEAVE:MOVE.L  20(SP),32(SP)   ; Set up return address
1E56| 4CDF 0307                       MOVEM.L (SP)+,D0-D2/A0-A1
1E5A| DEFC 000C                       ADD.W   #12,SP          ; Delete parameters
1E5E| 4E75                           RTS
1E42* 1C
1E60| 4210                   Y.ERROR:CLR.B   (A0)             ; Set result to null for now
1E62| 60EC                           BRA.S   Y.LEAVE         ; And return
1E64|                        ;
1E64|                        FREEPROC
PAGE -  80  MONITOR   FILE: FILPRC1.TEXT


1E64| 4E4F                           TRAP    #$F             ; Not to be used (was FETCHDR = $120)
1E66|                        ;
```

```
1E66|                          ;       UNITISBLKD(LUNIT:INTEGER)
1E66|                          ;
1E66|                          ;       parameters
1E66|                          ;
1E66|                          ;               D2      lunit
1E66|                          ;
1E66| 3F02                    UNITISB MOVE.W  D2,-(A7)
1E68| 0257 001F                       AND.W   #$1F,(A7)               ; get just unit#
1E6C| 0C57 0004                       CMP.W   #4,(A7)
1E70| 67**                           BEQ.S   @1
1E72| 0C57 0005                       CMP.W   #5,(A7)
1E76| 67**                           BEQ.S   @1
1E78| 0C57 0008                       CMP.W   #8,(A7)
1E7C| 62**                           BHI.S   @1
1E7E| 44FC 0004                       MOVE    #4,CCR                  ; cc=Z
1E82| 60**                           BRA.S   @2
1E7C* 06
1E76* 0C
1E70* 12
1E84| 44FC 0000                 @1    MOVE    #0,CCR                  ; cc=NZ
1E82* 04
1E88| 544F                       @2    ADD.W   #2,A7
1E8A| 4E75                             RTS
1E8C|                          ;
1E8C|                          ;       UT17IDX
1E8C|                          ;
1E8C|                          ;       Input:  D2 = dev#*32+unit#
1E8C|                          ;
1E8C|                          ;       Output: D2 = (dev#-1)*21*8+unit#*8
1E8C|                          ;
1E8C| 0442 0020                 UT17IDX SUB.W   #$20,D2                 ; subtract 1 from dev#
1E90| 3F02                             MOVE.W  D2,-(A7)
1E92| 0257 001F                       AND.W   #$1F,(A7)               ; form unit#
1E96| EA4A                           LSR.W   #DEVNSHF,D2
1E98| C5FC 0015                       MULS    #MAXU+1,D2              ; form dev#*21
1E9C| D45F                           ADD.W   (A7)+,D2
1E9E| E74A                           LSL.W   #3,D2                   ; multiply by size of UT17vid
1EA0| 4E75                             RTS
1EA2|                          ;
1EA2|                          ;       FTCHDIR(LUNIT:INTEGER; VAR DIRBUF:DIRECTORY)
1EA2|                          ;
1EA2|                          ;       parameters
1EA2|                          ;
1EA2|                          ;               D2      lunit
1EA2|                          ;               A2      address of directory
1EA2|                          ;
1EA2|                          ;       -2      length
1EA2|                          ;       -6      block
1EA2|                          ;       -8      first:boolean
1EA2|                          ;       -12     nextaddr
1EA2|                          ;
1EA2|                          ;       Note: iorslt for the I/O is returned in D6
1EA2|                          ;
PAGE -  81  MONITOR   FILE: FILPRC1.TEXT


1EA2| 4E56 FFF4                 FTCHDIR LINK    A6,#-12
1EA6| 48E7 FCD8                         MOVEM.L D0-D5/A0/A1/A3/A4,-(A7) ; save registers
1EAA| 61BA                             BSR     UNITISB
1EAC| 66**                           BNE.S   @0
1EAE| 303C 0002                       MOVE.W  #IBADUNT,D0
1EB2| 60**                           BRA.S   @3
1EB4|                          @0     GETNP   A3
1EAC* 06
1EB4| 266D FFF4                 #     MOVE.L   -12(A5),A3
1EB8| 244B                           MOVE.L  A3,A2
1EBA| D7FC 0000 0800                 ADD.L   #DIRSZ,A3               ; allocate four blocks for the directory
1EC0|                                  PUTNP   A3
1EC0| 2B4B FFF4                 #     MOVE.L   A3,-12(A5)
1EC4| 3D7C 0800 FFFE                 MOVE.W  #DIRSZ,-2(A6)           ; length
1ECA| 2D7C 0000 0002 FFFA           MOVE.L  #DIRDISK,-6(A6)         ; block
1ED2| 3D7C 0001 FFF8                 MOVE.W  #1,-8(A6)               ; first:=true
1ED8| 2D4A FFF4                       MOVE.L  A2,-12(A6)              ; nextaddr:=address of directory buffer
1EDC| 6100 E9A4                 @17   BSR     OVFCHK
1EE0| 3F02                           MOVE.W  D2,-(A7)                ; save dev#/unit#
1EE2| 2F0A                           MOVE.L  A2,-(A7)                ; save address of directory buffer
1EE4|                          ;
1EE4|                          ;       read the directory
1EE4|                          ;
1EE4| 222E FFF4                       MOVE.L  -12(A6),D1              ; address of directory buffer
1EE8| 3F02                           MOVE.W  D2,-(A7)                ; dev#/unit#
1EEA| 342E FFFE                       MOVE.W  -2(A6),D2               ; length
1EEE| 262E FFFA                       MOVE.L  -6(A6),D3               ; block
1EF2| 7C01                           MOVE.L  #INBIT,D6               ; save UREQ in D6
1EF4| 6100 FA66                       BSR     GETUNIT                 ; get unit number and validate
1EF8| 4E90                           JSR     (A0)                    ; go to driver
1EFA|                          ;
1EFA|                          ;       test iorslt
1EFA|                          ;
1EFA| 6100 FAD4                       BSR     IOEXIT                  ; record iorslt
1EFE| 245F                           MOVE.L  (A7)+,A2                ; restore address of directory buffer
1F00| 341F                           MOVE.W  (A7)+,D2                ; restore dev#/unit#
1EB2* 4E
1F02| 3C00                       @3    MOVE.W  D0,D6                   ; save iosrlst in temp
1F04| 6700 ****                       BEQ     @1
1F08|                          ;
```

```
1F08|                          ;       if error then log out this unit & clear uvid (in utbl or UT17)
1F08|                          ;
1F08| 3802                            MOVE.W  D2,D4
1F0A| E64C                            LSR.W   #3,D4
1F0C| 0244 003C                       AND.W   #$3C,D4
1F10|                                 GETLOGN A1
1F10| 226D FF60                #      MOVE.L   -160(A5),A1
1F14| 2A31 4000                       MOVE.L  0(A1,D4.W),D5
1F18| 0585                            BCLR    D2,D5                   ; log it out
1F1A| 2385 4000                       MOVE.L  D5,0(A1,D4.W)
1F1E| 0C42 001F                       CMP.W   #$1F,D2                 ; if dev# < 1 then
1F22| 62**                           BHI.S   @12                     ;   begin
1F24|                                 GETUTBL A0
1F24| 206D FFBA                #      MOVE.L   -70(A5),A0
1F28| C5FC 000C                       MULS    #UTBLSIZ,D2
PAGE -  82  MONITOR   FILE: FILPRC1.TEXT


1F2C| 41F0 2000                       LEA     UVID(A0,D2),A0          ;     uvid(unit#):='' and exit
1F30| 4298                     @11    MOVE.L  #0,(A0)+
1F32| 4298                            MOVE.L  #0,(A0)+                ;   end
1F34| 6000 ****                       BRA     FTCHDRX                 ; else
1F38|                          @12    GETUT17 A0                      ;   begin
1F22* 14
1F38| 206D FF5C                #      MOVE.L   -164(A5),A0
1F3C| 6100 FF4E                       BSR     UT17IDX
1F40| 41F0 2000                       LEA     UT17VID(A0,D2),A0       ;     uvid(dev#,unit#):='' and exit
1F44| 60EA                            BRA.S   @11                     ;   end
1F46|                          ;
1F46| 4A6E FFF8                @16    TST.W   -8(A6)
1F4A| 67**                           BEQ.S   @2
1F4C| 302A 0010                       MOVE.W  DNUMFLS(A2),D0
1F50| 5440                            ADD.W   #2,D0
1F52| C1FC 001A                       MULS    #DELENG,D0
1F56| 0440 0800                       SUB.W   #DIRSZ,D0
1F5A| 6B00 ****                       BMI     @2
1F5E| 0640 01FF                       ADD.W   #511,D0
1F62| E048                            LSR.W   #8,D0
1F64| E248                            LSR.W   #1,D0
1F66| 0C40 0024                       CMP.W   #36,D0
1F6A| 6D**                           BLT.S   @18
1F6C| 303C 0024                       MOVE.W  #36,D0
1F6A* 04
1F70| C1FC 0200                @18    MULS    #512,D0
1F74| 3D40 FFFE                       MOVE.W  D0,-2(A6)               ; length
1F78|                                 GETNP   A3
1F78| 266D FFF4                #      MOVE.L   -12(A5),A3
1F7C| D7C0                            ADD.L   D0,A3                   ; grow directory on the heap
1F7E|                                 PUTNP   A3
1F7E| 2B4B FFF4                #      MOVE.L   A3,-12(A5)
1F82| 2D7C 0000 0006 FFFA             MOVE.L  #DIRDISK+4,-6(A6)       ; block
1F8A| 426E FFF8                       MOVE.W  #0,-8(A6)               ; first:=false
1F8E| 06AE 0000 0800 FFF4             ADD.L   #DIRSZ,-12(A6)          ; nextaddr:=nextaddr+2048
1F96| 6000 FF44                       BRA     @17
1F9A|                          ;
1F9A|                          ;       test for male/female directory
1F9A|                          ;
1F06* 0094
1F9A| 0C6A 0006 0002           @1     CMP.W   #6,2(A2)                ; female directory ?
1FA0| 67**                           BEQ.S   @2
1FA2| 0C6A 002A 0002                 CMP.W   #42,2(A2)
1FA8| 679C                            BEQ.S   @16
1FAA| 103C 0009                       MOVE.B  #INOUNIT,D0             ; no
1FAE| 6100 FA20                       BSR     IOEXIT
1FB2| 6000 FF4E                       BRA.S   @3
1FB6|                          ;
1FB6|                          ;       compare the uvid (in utbl or UT17) with dvid (in the directory)
1FB6|                          ;
1FA0* 14
1F5C* 005A
1F4A* 6A
1FB6| 7201                     @2     MOVE.L  #1,D1                   ; yes, ok:=true
1FB8| 0C42 001F                       CMP.W   #$1F,D2                 ; if dev# < 1 then
PAGE -  83  MONITOR   FILE: FILPRC1.TEXT


1FBC| 62**                           BHI.S   @14
1FBE|                                 GETUTBL A0
1FBE| 206D FFBA                #      MOVE.L   -70(A5),A0
1FC2| 3602                            MOVE.W  D2,D3                   ;     copy lunit to compute @uvid
1FC4| C7FC 000C                       MULS    #UTBLSIZ,D3
1FC8| 41F0 3000                       LEA     UVID(A0,D3),A0          ;     get @uvid(unit#)
1FCC| 60**                           BRA.S   @15
1FBC* 10
1FCE| 3F02                     @14    MOVE.W  D2,-(A7)                ; else
1FD0|                                 GETUT17 A0
1FD0| 206D FF5C                #      MOVE.L   -164(A5),A0
1FD4| 6100 FEB6                       BSR     UT17IDX
1FD8| 41F0 2000                       LEA     UT17VID(A0,D2),A0       ;     get @uvid(dev#,unit#)
1FDC| 341F                            MOVE.W  (A7)+,D2
1FCC* 10
1FDE| 2648                     @15    MOVE.L  A0,A3
1FE0| 261B                            MOVE.L  (A3)+,D3
1FE2| 43EA 0006                       LEA     DVID(A2),A1
1FE6| B691                            CMP.L   (A1),D3                 ; if uvid<>dvid then
1FE8| 66**                           BNE.S   @5                      ;   go validate directory
```

```
1FEA| 261B                          MOVE.L  (A3)+,D3
1FEC| B6A9 0004                     CMP.L   4(A1),D3
1FF0| 66**                          BNE.S   @5
1FF2|                       ;
1FF2|                       ;       make sure that the device is logged in
1FF2|                       ;
1FF2| 3802                          MOVE.W  D2,D4
1FF4| E64C                          LSR.W   #3,D4
1FF6| 0244 003C                     AND.W   #$3C,D4
1FFA|                               GETLOGN A1
1FFA| 226D FF60               #       MOVE.L   -160(A5),A1
1FFE| 2A31 4000                     MOVE.L  0(A1,D4.W),D5
2002| 0505                          BTST    D2,D5                   ; exit if device is logged in
2004| 6600 ****                     BNE     FTCHDRX
2008|                       ;
2008|                       ;       validate directory (A0 = @uvid, A2 = @dirbuf, D1 = ok, D2 = lunit)
2008|                       ;
1FF0* 16
1FE8* 1E
2008| 7001                  @5      MOVE.L  #1,D0                   ; linx:=1
200A| 47EA 001A                     LEA     DELENG(A2),A3           ; address of first dir entry
200E| B06A 0010              @6      CMP.W   DNUMFLS(A2),D0          ; while linx <= dnumfiles do
2012| 6E**                          BGT.S   @8
2014| 162B 0006                     MOVE.B  DTID(A3),D3             ; if (length(dtid) >= 0) or
2018| 6B**                          BMI.S   @7
201A| 0C03 000F                     CMP.B   #15,D3                  ;    (length(dtid) > tidleng) or
201E| 6E**                          BGT.S   @7
2020| 3613                          MOVE.W  FSTBLK(A3),D3
2022| B66B 0002                     CMP.W   LSTBLK(A3),D3           ;    (dlastblk < dfirstblk) or
2026| 6E**                          BGT.S   @7
2028| 362B 0016                     MOVE.W  LSTBYTE(A3),D3          ;    (dlastbyte <= 0) or
202C| 6B**                          BMI.S   @7
202E| 0C43 0200                     CMP.W   #FBLKSIZ,D3             ;    (dlastbyte > fblksize) or
2032| 6E**                          BGT.S   @7
2034| 0C6B C800 0018                CMP.W   #$C800,DACCESS(A3)
PAGE -  84  MONITOR   FILE: FILPRC1.TEXT


203A| 64**                          BCC.S   @7                      ;    (daccess.year >= 100) then
203C| 5240                          ADD.W   #1,D0
203E| D6FC 001A                     ADD.W   #DELENG,A3
2042| 60CA                          BRA.S   @6
203A* 08
2032* 10
202C* 16
2026* 1C
201E* 24
2018* 2A
2044| 4281                  @7      MOVE.L  #0,D1                   ;    begin
2046| 48E7 E0A0                     MOVEM.L D0-D2/A0/A2,-(A7)       ;      ok:=false; delentry(linx,dirbuf)
204A| 3F00                          MOVE.W  D0,-(A7)
204C| 2F0A                          MOVE.L  A2,-(A7)
204E| 6100 ****                     BSR     DELNTRY
2052| 4CDF 0507                     MOVEM.L (A7)+,D0-D2/A0/A2
2056| 60B6                          BRA.S   @6                      ;    end else linx:=linx+1
2058|                       ;
2058|                       ;       if dirbuf has beeen changed then write it back
2058|                       ;
2012* 44
2058| 3802                  @8      MOVE.W  D2,D4
205A| E64C                          LSR.W   #3,D4
205C| 0244 003C                     AND.W   #$3C,D4
2060|                               GETLOGN A1
2060| 226D FF60               #       MOVE.L   -160(A5),A1
2064| 2A31 4000                     MOVE.L  0(A1,D4.W),D5
2068| 05C5                          BSET    D2,D5                   ; log it in
206A| 2385 4000                     MOVE.L  D5,0(A1,D4.W)
206E| 4A41                  @10     TST.W   D1                      ; if not ok then
2070| 66**                          BNE.S   @9
2072| 48E7 60A0                     MOVEM.L D1/D2/A0/A2,-(A7)
2076| 3F02                          MOVE.W  D2,-(A7)                ;   unit
2078| 2F0A                          MOVE.L  A2,-(A7)                ;   dirbuf
207A| 6100 ****                     BSR     WRTDIR                  ;   write directory
207E| 6100 F950                     BSR     IOEXIT                  ;   and record iorslt
2082| 4CDF 0506                     MOVEM.L (A7)+,D1/D2/A0/A2
2086| 4A40                          TST.W   D0                      ; test iorslt
2088| 6600 FE78                     BNE.S   @3
2070* 1A
208C| 4246                  @9      MOVE.W  #INOERR,D6
208E| 0C42 001F                     CMP.W   #$1F,D2                 ; if dev# < 1 then
2092| 62**                          BHI.S   @13                     ;   begin
2094| 316A 000E 000A                MOVE.W  DEOVBLK(A2),UEOVBLK(A0) ;     compute eov
2092* 06
209A| 43EA 0006              @13     LEA     DVID(A2),A1             ;   end
209E| 2091                          MOVE.L  (A1),(A0)               ; uvid(unit):=dvid
20A0| 2169 0004 0004                MOVE.L  4(A1),4(A0)
2006* 00A0
1F36* 0170
20A6| 4CDF 1B3F              FTCHDRX MOVEM.L (A7)+,D0-D5/A0/A1/A3/A4 ; restore registers
20AA| 4E5E                          UNLK    A6
20AC| 4E75                          RTS
20AE|                       ;
20AE|                       ;       VOLSRCH(VAR FVID:VID; LOOKHARD:BOOLEAN; VAR FDIR:DIRP):INTEGER;
PAGE -  85  MONITOR   FILE: FILPRC1.TEXT
```

```
20AE|                          ;
20AE|                          ;       stack
20AE|                          ;
20AE|                          ;               18      func result
20AE|                          ;               14      @FVID
20AE|                          ;               12      LOOKHARD
20AE|                          ;               8       @FDIR
20AE|                          ;               4       Return address
20AE|                          ;               0       Old A6
20AE|                          ;              -2       Device
20AE|                          ;             -26       tvid
20AE|                          ;             -42       tvid2
20AE|                          ;             -44       slashflag
20AE|                          ;             -48       heaptop
20AE|                          ;
20AE|                          ;       registers:
20AE|                          ;
20AE|                          ;       D0 -- ok                A0 -- @fvid & @tvid
20AE|                          ;       D1 -- physunit          A1 -- @utable
20AE|                          ;       D2 -- lunit             A2 -- fdir
20AE|                          ;       D3 -- device & index    A3 -- @fdir & temp
20AE|                          ;       D4 -- temp              A4 -- temp
20AE|                          ;       D5 -- copy of A3        A5 -- global ptr
20AE|                          ;       D6 -- temp              A6 -- local ptr
20AE|                          ;       D7 --                   A7 -- SP
20AE|                          ;
20AE| 4E56 FFD0               VOLSRCH LINK    A6,#-48
20B2|                                  GETNP   -48(A6)
20B2| 2D6D FFF4 FFD0         #         MOVE.L  -12(A5),-48(A6)
20B8| 4243                             CLR.W   D3                      ; device:=0
20BA| 426E FFD4                       CLR.W   -44(A6)                 ; slashflag:=false
20BE| 266E 000E                       MOVE.L  14(A6),A3               ; @fvid
20C2| 49EE FFE6                       LEA     -26(A6),A4              ; @tvid
20C6| 204C                           MOVE.L  A4,A0                   ; save @tvid
20C8| 383C 0005                       MOVE.W  #24/4-1,D4
20CC| 28DB                   @1      MOVE.L  (A3)+,(A4)+             ; copy fvid to tvid
20CE| 51CC FFFC                       DBF     D4,@1
20D2| 4244                           CLR.W   D4
20D4| 1810                           MOVE.B  (A0),D4                 ; get length of tvid
20D6| 6700 ****                       BEQ     BRVOLSX
20DA| 4246                           CLR.W   D6
20DC| 5246                   LSLASH  ADD.W   #1,D6
20DE| 0C30 002F 6000                 CMP.B   #'/',0(A0,D6.W)         ; is nextchar a slash
20E4| 67**                           BEQ.S   CSLASH
20E6| B846                           CMP.W   D6,D4
20E8| 66F2                           BNE.S   LSLASH
20EA| 6000 ****                       BRA     DEVNUMB
20E4* 08
20EE| 3D7C 0001 FFD4         CSLASH  MOVE.W  #1,-44(A6)              ; slashflag:=true
20F4| 0C28 0026 0001                 CMP.B   #'&',1(A0)              ; if firstchar = '&' then
20FA| 66**                           BNE.S   @2                      ;   while true do
20FC| 383C 0001                       MOVE.W  #1,D4                   ;     begin
2100| 5244                   @1      ADD.W   #1,D4                   ;       index:=index+1
2102| BC44                           CMP.W   D4,D6                   ;       exit if nextchar is slash
2104| 67**                           BEQ.S   CHKDEV
PAGE -  86  MONITOR   FILE: FILPRC1.TEXT


2106| 0C30 0030 4000                 CMP.B   #'0',0(A0,D4.W)         ;       if nextchar in DIGITS then
210C| 6D**                           BLT.S   BRVOLSX
210E| 0C30 0039 4000                 CMP.B   #'9',0(A0,D4.W)
2114| 6E**                           BGT.S   BRVOLSX
2116| C7FC 000A                       MULS    #10,D3                  ;         device:=device*10
211A| D630 4000                       ADD.B   0(A0,D4.W),D3           ;         device:=device+ord(nextchar)
211E| 0403 0030                       SUB.B   #'0',D3                 ;         device:=device-ord('0')
2122| 60DC                           BRA.S   @1                      ;     end
20FA* 28
2124| 4243                   @2      CLR.W   D3                      ; device:=0*DNTRYSZ
2126| 3806                           MOVE.W  D6,D4
2128| 5344                           SUB.W   #1,D4                   ; go delete slash if length(devname) = 0
212A| 67**                           BEQ.S   DELDEV
212C| 0C44 0007                       CMP.W   #7,D4                   ; exit if length(devname) > 7
2130| 62**                           BHI.S   BRVOLSX
2132| 49EE FFD6                       LEA     -42(A6),A4
2136| 2890                           MOVE.L  (A0),(A4)               ; tvid2:=tvid
2138| 2968 0004 0004                 MOVE.L  4(A0),4(A4)
213E| 1884                           MOVE.B  D4,(A4)                 ; truncate tvid2 to just devname
2140|                          @3      GETDTBL A3
2140| 266D FF88               #       MOVE.L  -120(A5),A3
2144| D6C3                           ADD.W   D3,A3                   ; address of nth devname
2146| 49EE FFD6                       LEA     -42(A6),A4              ; address of tvid2
214A| 1814                           MOVE.B  (A4),D4
214C| 5204                           ADD.B   #1,D4                   ; number of bytes to compare
214E| B90B                   @4      CMPM.B  (A3)+,(A4)+
2150| 66**                           BNE.S   @5
2152| 5304                           SUB.B   #1,D4
2154| 66F8                           BNE.S   @4
2156| EA4B                           LSR.W   #DEVNSHF,D3             ; found match
2158| 60**                           BRA.S   DELDEV
2150* 08
215A| 0643 0020                 @5      ADD.W   #DNTRYSZ,D3
215E| 0C43 00E0                       CMP.W   #MAXDEV*DNTRYSZ,D3
2162| 63DC                           BLS.S   @3
2130* 32
2114* 4E
210C* 56
```

```
20D8* 008C
2164| 6000 ****          BRVOLSX BRA     VOLSRCX
2104* 62
2168| EB4B                CHKDEV  LSL.W   #DEVNSHF,D3
216A|                           GETDTBL A3                      ; address of zeroth devname
216A| 266D FF88           #       MOVE.L   -120(A5),A3
216E| 284B                        MOVE.L  A3,A4
2170| D8C3                        ADD.W   D3,A4                   ; address of nth devname
2172| EA4B                        LSR.W   #DEVNSHF,D3
2174| 1814                        MOVE.B  (A4),D4
2176| 5204                        ADD.B   #1,D4                   ; number of bytes to compare
2178| B90B                @1      CMPM.B  (A3)+,(A4)+
217A| 66**                        BNE.S   DELDEV
217C| 5304                        SUB.B   #1,D4
217E| 66F8                        BNE.S   @1
2180| 4243                        CLR.W   D3
217A* 06
2158* 28
PAGE -  87  MONITOR   FILE: FILPRC1.TEXT


212A* 56
2182| 4244                DELDEV  CLR.W   D4                      ; delete device name from tvid
2184| 1810                        MOVE.B  (A0),D4
2186| 9846                        SUB.W   D6,D4                   ; exit if length(fvid)=0
2188| 67DA                        BEQ.S   BRVOLSX
218A| 0C44 0007                  CMP.W   #7,D4                   ; exit if length(fvid)>7
218E| 62D4                        BHI.S   BRVOLSX
2190| 5246                        ADD.W   #1,D6                   ; amount to delete
2192| 2F08                        MOVE.L  A0,-(A7)
2194| 3F06                        MOVE.W  D6,-(A7)
2196| 3F04                        MOVE.W  D4,-(A7)
2198| 2F08                        MOVE.L  A0,-(A7)
219A| 6100 FC88                  BSR     TTLCOPY
20EC* 00B2
219E| 0C43 0007          DEVNUMB CMP.W   #MAXDEV,D3
21A2| 62C0                        BHI.S   BRVOLSX
21A4| EB4B                        LSL.W   #DEVNSHF,D3             ; shift dev# to higher bits
21A6| 67**                        BEQ.S   @0                      ; if dev# <> 0 then
21A8| 3D7C 0001 000C              MOVE.W  #1,12(A6)               ;   lookhard:=true
21A6* 06
21AE| 3D43 FFFE            @0      MOVE.W  D3,-2(A6)               ; reflect shifted dev# in -2(A6)
21B2|                           GETUTBL A1                      ; @utable
21B2| 226D FFBA           #       MOVE.L   -70(A5),A1
21B6| 246E 0008                  MOVE.L  8(A6),A2                ; get @fdir
21BA| 264A                        MOVE.L  A2,A3                   ; copy the address
21BC| 4293                        MOVE.L  #NIL,(A3)               ; fdir:=nil
21BE| 2A0B                        MOVE.L  A3,D5                   ; save A3 in D5
21C0| 4280                        MOVE.L  #0,D0                   ; ok:=false
21C2| 4281                        MOVE.L  #0,D1                   ; physunit:=false
21C4| 0C28 0023 0001              CMP.B   #35,1(A0)               ; if (tvid[1]='#') and
21CA| 66**                        BNE.S   @5
21CC| 0C10 0001                  CMP.B   #1,(A0)                 ;    (length(tvid)<>1) then
21D0| 67**                        BEQ.S   @5
21D2| 7001                        MOVE.L  #1,D0                   ;    ok:=true
21D4| 4242                        MOVE.W  #0,D2                   ;    lunit:=0
21D6| 7602                        MOVE.L  #2,D3                   ;    index:=2
21D8|                                                           ;    repeat
21D8| 0C30 0030 3000        @1      CMP.B   #'0',0(A0,D3.W)         ;      if tvid[index] in DIGITS then
21DE| 6D**                        BLT.S   @2
21E0| 0C30 0039 3000              CMP.B   #'9',0(A0,D3.W)
21E6| 6E**                        BGT.S   @2
21E8| C5FC 000A                  MULS    #10,D2                  ;        lunit:=lunit*10
21EC| D430 3000                  ADD.B   0(A0,D3.W),D2           ;        lunit:=lunit+ord(tvid[index])
21F0| 0402 0030                  SUB.B   #'0',D2                 ;        lunit:=lunit-ord('0')
21F4| 60**                        BRA.S   @3                      ;      else
21E6* 0E
21DE* 16
21F6| 4280                @2      MOVE.L  #0,D0                   ;        ok:=false
21F8| 60**                        BRA.S   @4                      ;        exit repeat loop (ie. not ok)
21F4* 04
21FA| 5243                @3      ADD.W   #1,D3                   ;      index:=index+1
21FC| B610                        CMP.B   (A0),D3                 ;    until (index>len(tvid)) or not ok
21FE| 6FD8                        BLE.S   @1
21F8* 06
PAGE -  88  MONITOR   FILE: FILPRC1.TEXT


2200| 2200                @4      MOVE.L  D0,D1                   ;    physunit:=ok and (lunit in 1..MAXU)
2202| 67**                        BEQ.S   @5
2204| 4A42                        TST.W   D2
2206| 67EE                        BEQ.S   @2
2208| 0C02 0014                  CMP.B   #MAXU,D2
220C| 6EE8                        BGT.S   @2
220E| 846E FFFE                  OR.W    -2(A6),D2               ;    lunit:=lunit ORed dev#*32
2212|                                                           ;    if physunit then
2212| 426E 000C                  MOVE.W  #0,12(A6)               ;      lookhard:=false
2216| 60**                        BRA.S   LNFVID0                 ;      goto lnfvid0
2202* 14
21D0* 46
21CA* 4C
2218|                   @5                                      ;    ok:=false
2218| 4A6E FFD4                  TST.W   -44(A6)                 ;    if not slashflag then
221C| 66**                        BNE.S   @12
221E| 4242                        CLR.W   D2                      ;    dev#:=0
2220| 60**                        BRA.S   @8
```

```
221C* 04
2222| 342E FFFE            @12     MOVE.W  -2(A6),D2
2220* 04
2226| 0642 0014            @8      ADD.W   #MAXU,D2              ; 2: lunit:=maxunit
222A|                      @6                                    ;    repeat
222A| 0C42 001F                    CMP.W   #$1F,D2
222E| 62**                         BHI.S   @10
2230| 3C02                        MOVE.W  D2,D6
2232| CDFC 000C                   MULS    #UTBLSIZ,D6
2236| 49F1 6000                   LEA     UVID(A1,D6),A4        ;      @uvid
223A| 60**                         BRA.S   @11
222E* 0C
223C| 3F02                  @10     MOVE.W  D2,-(A7)
223E|                              GETUT17 A4
223E| 286D FF5C             #       MOVE.L   -164(A5),A4
2242| 6100 FC48                    BSR     UT17IDX
2246| 49F4 2000                   LEA     UT17VID(A4,D2),A4
224A| 341F                        MOVE.W  (A7)+,D2
223A* 10
224C| 2648                  @11     MOVE.L  A0,A3                 ;      @tvid
224E| 1814                        MOVE.B  (A4),D4               ;      if tvid=utable1[lunit].uvid then
2250| 5204                        ADD.B   #1,D4
2252| B90B                  @9      CMPM.B  (A3)+,(A4)+
2254| 66**                         BNE.S   @7
2256| 5304                        SUB.B   #1,D4
2258| 66F8                        BNE.S   @9                    ;        begin
225A| 3D42 FFFE                   MOVE.W  D2,-2(A6)
225E| 026E 01E0 FFFE              AND.W   #$1E0,-2(A6)
2264| 7001                        MOVE.L  #1,D0                 ;          ok:=true; goto lnfvid0
2266| 60**                         BRA.S   LNFVID0               ;        end
2254* 12
2268| 5342                  @7      SUB.W   #1,D2                 ;      lunit:=lunit-1
226A| 3F02                        MOVE.W  D2,-(A7)
226C| 025F 001F                   AND.W   #$1F,(A7)+
2270| 66B8                        BNE.S   @6                    ;    until ok or lunit=0
2272| 4A6E FFD4                   TST.W   -44(A6)               ;    if not slashflag then
2276| 66**                         BNE.S   LNFVID0               ;      begin
PAGE -  89  MONITOR   FILE: FILPRC1.TEXT


2278| 0642 0020                   ADD.W   #$20,D2               ;        dev#:=dev#+1
227C| 0C42 0100                   CMP.W   #$20*<MAXDEV+1>,D2    ;        if dev#<>8 then goto 2
2280| 66A4                        BNE.S   @8                    ;      end
2282|                       ; Enter here with:
2282|                       ;       1) lunit = dev#*32+unit#  and -2(A6) = dev#*32
2282|                       ;    or 2) ok = false
2276* 0A
2266* 1A
2216* 6A
2282| 4A40                  LNFVID0 TST.W   D0                    ; if ok then
2284| 67**                        BEQ.S   NOTOKLH
2286| 6100 FBDE                   BSR     UNITISB               ; if unit is not blocked then
228A| 67**                        BEQ.S   NOTOKLH               ;   goto notoklh
228C|                              PUTNP   -48(A6)
228C| 2B6E FFD0 FFF4        #       MOVE.L   -48(A6),-12(A5)
2292| 6100 FC0E                   BSR     FTCHDIR               ; else ftchdir(lunit,fdir)
2296| 4A06                        TST.B   D6                    ; if iorslt=0 then
2298| 66**                        BNE.S   @2
229A| 2645                        MOVE.L  D5,A3
229C| 268A                        MOVE.L  A2,(A3)               ; update ptr to directory
229E| 7001                        MOVE.L  #1,D0                 ; ok:=true
22A0| 2648                        MOVE.L  A0,A3                 ; @tvid
22A2| 49EA 0006                   LEA     DVID(A2),A4           ; @dvid
22A6| 4A41                        TST.W   D1                    ; if physunit then
22A8| 67**                        BEQ.S   @1                    ;   begin
22AA| 26DC                        MOVE.L  (A4)+,(A3)+           ;   tvid:=fvid
22AC| 26DC                        MOVE.L  (A4)+,(A3)+           ;   end
22AE| 60**                        BRA.S   NOTOKLH               ; else
22A8* 06
22B0| 1814                  @1      MOVE.B  (A4),D4               ;   ok:=tvid=fdir1[0].dvid
22B2| B90B                  @9      CMPM.B  (A3)+,(A4)+
22B4| 66**                         BNE.S   @2
22B6| 5304                        SUB.B   #1,D4
22B8| 6AF8                        BPL.S   @9
22BA| 60**                        BRA.S   NOTOKLH               ;   ok:=true
22B4* 06
2298* 22
22BC| 4280                  @2      MOVE.L  #0,D0                 ;   ok:=false
22BA* 02
22AE* 0E
228A* 32
2284* 38
22BE| 4A40                  NOTOKLH TST.W   D0                    ; if not ok and lookhard then
22C0| 66**                        BNE.S   FIXFVID
22C2| 4A6E 000C                   TST.W   12(A6)
22C6| 67**                        BEQ.S   FIXFVID
22C8| 7414                  @1      MOVE.L  #MAXU,D2              ; 1: lunit:=maxunit
22CA| D46E FFFE                  @6      ADD.W   -2(A6),D2             ;    repeat
22CE| 6100 FB96                   BSR     UNITISB               ;      if unit is blocked then
22D2| 67**                        BEQ.S   @7
22D4| 0C42 001F                   CMP.W   #$1F,D2               ;        if (device=0) or
22D8| 63**                        BLS.S   @2
22DA| 4840                        SWAP    D0
22DC| 3002                        MOVE.W  D2,D0
22DE| 4840                        SWAP    D0
PAGE -  90  MONITOR   FILE: FILPRC1.TEXT
```

```
22E0| 6100 F3BA                        BSR     CHECKCD                 ;           (harddisk or memory) then
22E4| 67**                            BEQ.S   @7
22E6|                          @2      PUTNP   -48(A6)
22D8* 0C
22E6| 2B6E FFD0 FFF4           #       MOVE.L   -48(A6),-12(A5)
22EC| 6100 FBB4                        BSR     FTCHDIR                 ;           ftchdir(lunit,fdir)
22F0| 4A06                            TST.B   D6                      ;       if iorslt=0 then
22F2| 66**                            BNE.S   @7
22F4| 2645                            MOVE.L  D5,A3
22F6| 268A                            MOVE.L  A2,(A3)                 ;         update ptr to directory
22F8| 2648                            MOVE.L  A0,A3                   ;         @tvid
22FA| 49EA 0006                       LEA     DVID(A2),A4             ;         @dvid
22FE| 1814                            MOVE.B  (A4),D4                 ;         if tvid=dvid then
2300| 5204                            ADD.B   #1,D4
2302| B90B                    @9      CMPM.B  (A3)+,(A4)+
2304| 66**                            BNE.S   @7
2306| 5304                            SUB.B   #1,D4
2308| 66F8                            BNE.S   @9
230A| 3D42 0014                       MOVE.W  D2,20(A6)               ;           volsrch:=lunit, and exit
230E| 60**                            BRA.S   CDEVVOL                 ;         else
2304* 0A
22F2* 1C
22E4* 2A
22D2* 3C
2310| 4280                    @7      MOVE.L  #0,D0                   ;           ok:=false
2312| 946E FFFE                       SUB.W   -2(A6),D2
2316| 5342                            SUB.W   #1,D2                   ;     lunit:=lunit-1
2318| 66B0                            BNE.S   @6                      ;     until ok or lunit=0
231A| 4A6E FFD4                       TST.W   -44(A6)                 ;     if not slashflag then
231E| 66**                            BNE.S   FIXFVID                 ;       begin
2320| 066E 0020 FFFE                  ADD.W   #$20,-2(A6)             ;         dev#:=dev#+1
2326| 0C6E 0100 FFFE                  CMP.W   #$20*<MAXDEV+1>,-2(A6)  ;         if dev#<>8 then goto 1
232C| 669A                            BNE.S   @1                      ;       end
231E* 0E
22C6* 66
22C0* 6C
232E| 4A40                    FIXFVID TST.W   D0                      ; exit if not ok
2330| 67**                            BEQ.S   VOLSRCX
2332| 3D42 0014                       MOVE.W  D2,20(A6)               ; volsrch:=lunit
2336| 0C42 001F                       CMP.W   #$1F,D2                 ; if dev# = 0 then
233A| 62**                            BHI.S   CDEVVOL                 ;   begin
233C| 3C02                            MOVE.W  D2,D6
233E| CDFC 000C                       MULS    #UTBLSIZ,D6
2342| 49F1 6000                       LEA     UVID(A1,D6),A4          ;     @uvid
2346| 209C                            MOVE.L  (A4)+,(A0)              ;     tvid:=uvid
2348| 215C 0004                       MOVE.L  (A4)+,4(A0)             ;   end
233A* 10
230E* 3C
234C| 286E 000E               CDEVVOL MOVE.L  14(A6),A4               ; fvid:=''
2350| 4214                            CLR.B   (A4)
2352| 4A10                            TST.B   (A0)                    ; exit if length(tvid)=0
2354| 6700 ****                       BEQ     VOLSRCX
2358| 6100 FB0C                       BSR     UNITISB                 ; if unit is blocked then
235C| 66**                            BNE.S   @0
235E| 264C                    @5      MOVE.L  A4,A3
PAGE -  91  MONITOR   FILE: FILPRC1.TEXT


2360| 4A1C                            TST.B   (A4)+
2362| 60**                            BRA.S   @2
2364|                          @0      GETDTBL A3                      ;   begin
235C* 06
2364| 266D FF88                #       MOVE.L   -120(A5),A3
2368| 382E FFFE                       MOVE.W  -2(A6),D4
236C| EA4C                            LSR.W   #DEVNSHF,D4
236E| 67EE                            BEQ.S   @5
2370| EB4C                            LSL.W   #DEVNSHF,D4
2372| 47F3 4000                       LEA     DEVNAME(A3,D4.W),A3     ;     address of device name
2376| 181B                            MOVE.B  (A3)+,D4
2378| 6700 ****                       BEQ     VOLSRCX
237C| D91C                            ADD.B   D4,(A4)+
237E| 18DB                    @1      MOVE.B  (A3)+,(A4)+             ;     concat devname to fvid
2380| 5304                            SUB.B   #1,D4
2382| 66FA                            BNE.S   @1
2384| 266E 000E                       MOVE.L  14(A6),A3
2388| 5213                            ADD.B   #1,(A3)                 ;     concat slash to fvid
238A| 18FC 002F                       MOVE.B  #'/',(A4)+              ;   end
2362* 2A
238E| 1818                    @2      MOVE.B  (A0)+,D4
2390| D913                            ADD.B   D4,(A3)
2392| 18D8                    @3      MOVE.B  (A0)+,(A4)+             ; concat tvid to fvid
2394| 5304                            SUB.B   #1,D4
2396| 66FA                            BNE.S   @3
2398|
2398| 0C13 000F                       CMP.B   #15,(A3)
239C| 62**                            BHI.S   @4
239E| 60**                            BRA.S   VOLSRXT
239C* 02
23A0| 4E4F                    @4      TRAP    #$F
23A2|                          ;
23A2|                          ; error exit
23A2|                          ;
237A* 0028
2356* 004C
```

```
2330* 70
2166* 023C
23A2| 206E 000E            VOLSRCX MOVE.L  14(A6),A0               ; fvid:=''
23A6| 4210                        CLR.B   (A0)
23A8| 426E 0014                    MOVE.W  #0,20(A6)               ; volsrch:=0
239E* 0C
23AC| 4E5E                VOLSRXT UNLK    A6
23AE| 205F                        MOVE.L  (A7)+,A0                ; pop return address
23B0| DEFC 000C                    ADD.W   #12,A7                  ; delete parameters
23B4| 4ED0                        JMP     (A0)
23B6|                     ;
23B6|                     ;       DIRSRCH
23B6|                     ;
23B6|                     ;               18      func result
23B6|                     ;               14      @FTID
23B6|                     ;               12      FINDPERM
23B6|                     ;               8       @FDIR
23B6|                     ;               4       Return address
23B6|                     ;               0       Old A6
PAGE -  92  MONITOR   FILE: FILPRC1.TEXT


23B6|                     ;
23B6| 4E56 0000          DIRSRCH LINK    A6,#0
23BA| 226E 0008                  MOVE.L  8(A6),A1                ; get ptr to dirbuf
23BE| 426E 0014                  MOVE.W  #0,20(A6)               ; return 0 as default
23C2| 761A                      MOVE.L  #DELENG,D3              ; initialize offset
23C4| 206E 000E                  MOVE.L  14(A6),A0               ; point A0 at the filename
23C8| 3229 0010                  MOVE.W  DNUMFLS(A1),D1          ; get number of files
23CC| 67**                      BEQ.S   DIRSRCX                 ; number of files = 0 ?
23CE| D2FC 0020                  ADD.W   #DELENG+DTID,A1         ; point A1 at first entry
23D2| 48E7 0018          DIRSRLP MOVEM.L A3/A4,-(A7)             ; save A3 and A4
23D6| 2849                      MOVE.L  A1,A4                   ; A4 is used for title compare
23D8| 2648                      MOVE.L  A0,A3                   ; A3 is title to look for
23DA| 4242                      CLR.W   D2
23DC| 1410                      MOVE.B  (A0),D2                 ; length of name in bytes
23DE| B90B                @1    CMPM.B  (A3)+,(A4)+             ; check each byte for equal
23E0| 66**                      BNE.S   @2
23E2| 5342                      SUB.W   #1,D2                   ; compare length+1 bytes
23E4| 64F8                      BCC.S   @1
23E6| 5D49                      SUB.W   #DTID,A1                ; found it leave A1 at start of entry
23E8| 4CDF 1800                MOVEM.L (A7)+,A3/A4             ; restore A3 and A4
23EC| 60**                      BRA.S   @4
23E0* 0C
23EE| 4CDF 1800          @2    MOVEM.L (A7)+,A3/A4             ; restore A3 and A4
23F2| D2FC 001A          @3    ADD.W   #DELENG,A1              ; skip to next dir entry
23F6| 0643 001A                ADD.W   #DELENG,D3              ; also update offset
23FA| 5341                      SUB.W   #1,D1                   ; any files left ?
23FC| 66D4                      BNE.S   DIRSRLP
23FE| 60**                      BRA.S   DIRSRCX                 ; file not found
23EC* 12
2400| 302E 000C          @4    MOVE.W  12(A6),D0               ; if findperm = daccess.year<>100 then
2404| 3429 0018                MOVE.W  DACCESS(A1),D2
2408| 0242 FE00                AND.W   #$FE00,D2
240C| 0C42 C800                CMP.W   #$C800,D2
2410| 56C2                      SNE     D2
2412| 0242 0001                AND.W   #1,D2
2416| B440                      CMP.W   D0,D2
2418| 67**                      BEQ.S   @5
241A| 5C49                      ADD.W   #DTID,A1
241C| 60D4                      BRA.S   @3
2418* 04
241E| 3D43 0014          @5    MOVE.W  D3,20(A6)               ;   dirsearch:=offset
23FE* 22
23CC* 54
2422| 4E5E                DIRSRCX UNLK    A6
2424| 205F                        MOVE.L  (A7)+,A0                ; pop return address
2426| DEFC 000C                    ADD.W   #12,A7                  ; delete parameters
242A| 4ED0                        JMP     (A0)
242C|                     ;
242C|
242C|
242C|                     ;
242C|                             .INCLUDE FILPRC2.TEXT
242C|                     ;
242C|                     ;       FILENAME FILPRC2
242C|                     ;
PAGE -  93  MONITOR   FILE: FILPRC2.TEXT


242C|
242C|                     ;
242C|                     ;       FRESET(VAR F:FIB)
242C|                     ;
242C|                     ;       stack:
242C|                     ;
242C|                     ;               8       F
242C|                     ;               4       Return Address
242C|                     ;               0       Old A6
242C|                     ;
242C| 4E56 0000          FRESET  LINK    A6,#0
2430|                              GETSYSC A0
2430| 206D FFFC                #       MOVE.L   -4(A5),A0
2434| 4210                            MOVE.B  #0,(A0)                 ; iorslt:=inoerror
2436| 226E 0008                      MOVE.L  8(A6),A1
243A| 4A29 000A                      TST.B   FISOPEN(A1)             ; if fisopen then
```

```
243E| 67**                          BEQ.S   FRESETX
2440| 2F09                          MOVE.L  A1,-(A7)
2442| 6100 ****                    BSR     RESETER                 ;   reseter(F)
2446| 226E 0008                    MOVE.L  8(A6),A1
244A| 4A69 0008                    TST.W   FRECSZ(A1)              ;   if frecsz>0 then
244E| 6F**                          BLE.S   FRESETX
2450| 4A69 0006                    TST.W   FSTATE(A1)
2454| 66**                          BNE.S   @1
2456| 2F09                          MOVE.L  A1,-(A7)
2458| 6100 ****                    BSR     FGET
245C| 60**                          BRA.S   FRESETX
2454* 08
245E| 337C 0001 0006         @1      MOVE.W  #1,FSTATE(A1)
245C* 06
244E* 14
243E* 24
2464| 4E5E                    FRESETX UNLK    A6
2466| 205F                          MOVE.L  (A7)+,A0
2468| 584F                          ADD.W   #4,A7
246A| 4ED0                          JMP     (A0)
246C|                           ;
246C|                           ;       RESETER(VAR F:FIB)
246C|                           ;
246C|                           ;       stack:
246C|                           ;
246C|                           ;               8       F
246C|                           ;               4       Return Address
246C|                           ;               0       Old A6
246C|                           ;
2444* 0028
246C| 4E56 0000                 RESETER LINK    A6,#0
2470| 226E 0008                       MOVE.L  8(A6),A1
2474| 4269 001A                       MOVE.W  #0,FREPCNT(A1)          ; frepcnt:=0
2478| 4229 0005                       MOVE.B  #0,FEOF(A1)             ; feof:=false
247C| 4229 0004                       MOVE.B  #0,FEOLN(A1)            ; feoln:=false
2480| 4A29 000B                       TST.B   FISBLKD(A1)             ; if fisblkd then
2484| 6700 ****                       BEQ     RESETRX
2488| 3029 0016                       MOVE.W  FMAXBLK(A1),D0
248C| B069 0018                       CMP.W   FNXTBLK(A1),D0
PAGE -  94  MONITOR   FILE: FILPRC2.TEXT


2490| 5DC0                          SLT     D0                      ; bigger:=fnxtblk>fmaxblk
2492| 6C**                          BGE.S   @1                      ; if bigger then
2494| 3369 0018 0016                MOVE.W  FNXTBLK(A1),FMAXBLK(A1) ; fmaxblk:=fnxtblk
2492* 06
249A| 4A69 0038             @1      TST.W   FSOFTBF(A1)             ; if softbf then
249E| 6700 ****                    BEQ     @7
24A2| 4A40                          TST.W   D0                      ; if bigger then fmaxbyte:=fnxtbyte
24A4| 66**                          BNE.S   @2
24A6| 3229 0016                    MOVE.W  FMAXBLK(A1),D1          ; if fnxtblk=fmaxblk then
24AA| B269 0018                    CMP.W   FNXTBLK(A1),D1
24AE| 66**                          BNE.S   @3
24B0| 3429 003C                    MOVE.W  FNXTBYT(A1),D2          ;   if fnxtbyte>fmaxbyte then
24B4| B469 003A                    CMP.W   FMAXBYT(A1),D2
24B8| 6F**                          BLE.S   @3
24BA| 7001                          MOVE.L  #1,D0                   ;     bigger:=true
24A4* 16
24BC| 3369 003C 003A         @2      MOVE.W  FNXTBYT(A1),FMAXBYT(A1) ;     fmaxbyte:=fnxtbyte
24B8* 08
24AE* 12
24C2| 4A69 003E             @3      TST.W   FBFCHNG(A1)             ; if fbufchngd then
24C6| 67**                          BEQ.S   @6
24C8| 4269 003E                    MOVE.W  #0,FBFCHNG(A1)          ;   fbufchngd:=false
24CC| 337C 0001 001C                MOVE.W  #1,FMODIFD(A1)          ;   fmodified:=true
24D2| 4A40                          TST.W   D0                      ; if bigger then
24D4| 67**                          BEQ.S   @5
24D6| 41E9 0050                    LEA     FBUFFER(A1),A0          ;   fillchar(fbuffer[fnxtbyte],
24DA| D0E9 003C                    ADD.W   FNXTBYT(A1),A0          ;            fblksize-fnxtbyte,0)
24DE| 363C 0200                    MOVE.W  #FBLKSIZ,D3
24E2| 9669 003C                    SUB.W   FNXTBYT(A1),D3
24E6| 67**                          BEQ.S   @5
24E8| 4218                 @4      MOVE.B  #0,(A0)+
24EA| 5343                          SUB.W   #1,D3
24EC| 66FA                          BNE.S   @4
24E6* 06
24D4* 18
24EE| 3F00                 @5      MOVE.W  D0,-(A7)
24F0| 3029 000C                    MOVE.W  FUNIT(A1),D0            ; unitwrite(funit,fbuffer,fblksize,
24F4| 41E9 0050                    LEA     FBUFFER(A1),A0          ;            fheader.dfrstblk+fnxtblk-1
24F8| 2208                          MOVE.L  A0,D1
24FA| 343C 0200                    MOVE.W  #FBLKSIZ,D2
24FE| 41E9 001E                    LEA     FHEADER(A1),A0
2502| 3610                          MOVE.W  FSTBLK(A0),D3
2504| D669 0018                    ADD.W   FNXTBLK(A1),D3
2508| 5343                          SUB.W   #1,D3
250A| 6100 ****                    BSR     WRITEIT
250E| 301F                          MOVE.W  (A7)+,D0                ; if bigger and
2510| 67**                          BEQ.S   @6
2512| 226E 0008                    MOVE.L  8(A6),A1
2516| 41E9 001E                    LEA     FHEADER(A1),A0          ;   (fheader.dfkind=textfile) and
251A| 0C68 0003 0004                CMP.W   #TEXTFILE,FKIND(A0)
2520| 66**                          BNE.S   @6
2522| 3029 0018                    MOVE.W  FNXTBLK(A1),D0
2526| E258                          ROR.W   #1,D0                   ;   add(fnxtblk) then
2528| 64**                          BCC.S   @6
```

```
252A| 5269 0016                    ADD.W   #1,FMAXBLK(A1)          ;   fmaxblk:=fmaxblk+1
PAGE -  95  MONITOR   FILE: FILPRC2.TEXT


252E| 41E9 0050                    LEA     FBUFFER(A1),A0          ;   fillchar(fbuffer,fblksize,0)
2532| 363C 0200                    MOVE.W  #FBLKSIZ,D3
2536| 5269 0018                    ADD.W   #1,FNXTBLK(A1)
253A| 4240                        MOVE.W  #0,D0
253C| 60AA                        BRA.S   @4                      ;   go do unitwrite
2528* 14
2520* 1C
2510* 2C
24C6* 76
253E| 226E 0008            @6      MOVE.L  8(A6),A1
2542| 337C 0200 003C              MOVE.W  #FBLKSIZ,FNXTBYT(A1)    ; fnxtbyte:=fblksize
24A0* 00A8
2548| 4269 0018            @7      MOVE.W  #0,FNXTBLK(A1)          ; fnxtblk:=0
254C| 4A69 0038                    TST.W   FSOFTBF(A1)             ; if fsoftbf and
2550| 67**                        BEQ.S   RESETRX
2552| 41E9 001E                    LEA     FHEADER(A1),A0          ;   (fheader.dfkind=textfile) then
2556| 0C68 0003 0004              CMP.W   #TEXTFILE,FKIND(A0)
255C| 66**                        BNE.S   RESETRX
255E| 337C 0002 0018              MOVE.W  #2,FNXTBLK(A1)          ;   fnxtblk:=2
255C* 06
2550* 12
2486* 00DE
2564| 4E5E                RESETRX UNLK    A6
2566| 205F                        MOVE.L  (A7)+,A0
2568| 584F                        ADD.W   #4,A7
256A| 4ED0                        JMP     (A0)
256C|                       ;
256C|                       ;       INSNTRY(VAR FENTRY:DIRENTRY; FINX:DIRRANGE; FDIR:DIRP)
256C|                       ;
256C|                       ;       stack:
256C|                       ;
256C|                       ;               14      @FENTRY
256C|                       ;               12      FINX
256C|                       ;               8       FDIR
256C|                       ;               4       Return Address
256C|                       ;               0       Old A6
256C|                       ;
256C| 4E56 0000            INSNTRY LINK    A6,#0
2570| 206E 0008                    MOVE.L  8(A6),A0
2574| 3028 0010                    MOVE.W  DNUMFLS(A0),D0
2578| 5240                        ADD.W   #1,D0
257A| 61**                        BSR.S   SETA1A2
257C| B06E 000C                    CMP.W   12(A6),D0               ; if dnumfiles+1>finx then
2580| 6F**                        BLE.S   @4
2582| 740D                @1      MOVE.L  #DELENG/2,D2            ;   for i:=dnumfiles downto finx do
2584| 3521                @2      MOVE.W  -(A1),-(A2)             ;     fdirt[i+1]:=fdirt[i]
2586| 5342                        SUB.W   #1,D2
2588| 66FA                        BNE.S   @2
258A| 5340                        SUB.W   #1,D0
258C| B06E 000C                    CMP.W   12(A6),D0
2590| 66F0                        BNE.S   @1
2580* 10
2592| 246E 000E            @4      MOVE.L  14(A6),A2
2596| 740D                        MOVE.L  #DELENG/2,D2
2598| 32DA                @3      MOVE.W  (A2)+,(A1)+             ; fdirt[finx]:=fentry
PAGE -  96  MONITOR   FILE: FILPRC2.TEXT


259A| 5342                        SUB.W   #1,D2
259C| 66FA                        BNE.S   @3
259E| 0668 0001 0010              ADDI.W  #1,DNUMFLS(A0)          ; dnumfiles:=dnumfiles+1
25A4| 4E5E                XITNTRY UNLK    A6
25A6| 205F                        MOVE.L  (A7)+,A0
25A8| DEFC 000A                    ADD.W   #10,A7
25AC| 4ED0                        JMP     (A0)
25AE|                       ;
25AE|                       ;       SETA1A2 -- common to INSNTRY and DELNTRY
25AE|                       ;
257A* 32
25AE| 3200                SETA1A2 MOVE.W  D0,D1
25B0| C3FC 001A                    MULS    #DELENG,D1
25B4| 2248                        MOVE.L  A0,A1
25B6| D2C1                        ADD.W   D1,A1
25B8| 2449                        MOVE.L  A1,A2
25BA| D4FC 001A                    ADD.W   #DELENG,A2
25BE| 4E75                        RTS
25C0|                       ;
25C0|                       ;       DELNTRY(FINX:DIRRANGE; FDIR:DIRP)
25C0|                       ;
25C0|                       ;       stack:
25C0|                       ;
25C0|                       ;               12      FINX
25C0|                       ;               8       FDIR
25C0|                       ;               4       Return Address
25C0|                       ;               0       Old A6
25C0|                       ;
25C0| 4E56 0000            DELNTRY LINK    A6,#0
25C4| 206E 0008                    MOVE.L  8(A6),A0
25C8| 3628 0010                    MOVE.W  DNUMFLS(A0),D3
25CC| 302E 000C                    MOVE.W  12(A6),D0
25D0| 61DC                        BSR.S   SETA1A2
25D2| B640                        CMP.W   D0,D3                   ; if finx < dnumfiles then
```

```
25D4| 6F**                          BLE.S   @3
25D6| 740D                  @1      MOVE.L  #DELENG/2,D2            ;   for i:=finx to dnumfiles-1 do
25D8| 32DA                  @2      MOVE.W  (A2)+,(A1)+             ;     fdirt[i]:=fdirt[i+1]
25DA| 5342                          SUB.W   #1,D2
25DC| 66FA                          BNE.S   @2
25DE| 5240                          ADD.W   #1,D0
25E0| B043                          CMP.W   D3,D0
25E2| 66F2                          BNE.S   @1
25D4* 0E
25E4| 4229 0006              @3      MOVE.B  #0,DTID(A1)             ; fdirt[dnumfiles].dtid:=''
25E8| 0468 0001 0010                 SUBI.W  #1,DNUMFLS(A0)          ; dnumfiles:=dnumfiles-1
25EE| 4E5E                          UNLK    A6
25F0| 205F                          MOVE.L  (A7)+,A0
25F2| 5C4F                          ADD.W   #6,A7
25F4| 4ED0                          JMP     (A0)
25F6|                       ;
25F6|                       ;       ENTRTMP(VAR F:FIB)
25F6|                       ;
25F6|                       ;       stack:
25F6|                       ;
25F6|                       ;               20      func result
PAGE -  97  MONITOR   FILE: FILPRC2.TEXT


25F6|                       ;               16      @FTID
25F6|                       ;               14      FSEGS
25F6|                       ;               12      FKIND
25F6|                       ;               8       FDIR
25F6|                       ;               4       Return Address
25F6|                       ;               0       Old A6
25F6|                       ;
25F6|                       ;       registers:
25F6|                       ;
25F6|                       ;       D0 -- temp,   CURINX      A0 -- temp
25F6|                       ;       D1 -- temp,   FRSTOPEN    A1 -- @LDE
25F6|                       ;       D2 -- temp,   NEXTUSED    A2 -- @RT11
25F6|                       ;       D3 -- temp,   FREEAREA    A3 -- FDIR
25F6|                       ;       D4 -- U:SINX, L:DINX      A4 -- @FSEGS
25F6|                       ;       D5 -- LASTI               A5 -- global ptr
25F6|                       ;       D6 -- SSEGS               A6 -- local ptr
25F6|                       ;       D7 --                     A7 -- SP
25F6|                       ;
25F6|                       ;       FINDMAX(CURINX;FRSTOPEN;NEXTUSED)
25F6|                       ;       VAR FREEAREA
25F6|                       ;
25F6| 3602                  FINDMAX MOVE.W  D2,D3                   ; freearea:=nextused-frstopen
25F8| 9641                          SUB.W   D1,D3
25FA| B654                          CMP.W   (A4),D3                 ; if freearea>fsegs then
25FC| 6F**                          BLE.S   @2
25FE| 4844                          SWAP    D4                      ;   sinx:=dinx
2600| 3C14                          MOVE.W  (A4),D6                 ;   ssegs:=fsegs
2602| 3800                          MOVE.W  D0,D4                   ;   dinx:=curinx
2604| 3883                          MOVE.W  D3,(A4)                 ;   fsegs:=freearea
2606| 4E75                  @1      RTS
25FC* 0A
2608| BC54                  @2      CMP.W   (A4),D6                 ; if freearea>ssegs then
260A| 6FFA                          BLE.S   @1
260C| 3C03                          MOVE.W  D3,D6                   ;   ssegs:=freearea
260E| 4844                          SWAP    D4
2610| 3800                          MOVE.W  D0,D4                   ;   sinx:=curinx
2612| 4844                          SWAP    D4
2614| 4E75                          RTS
2616|                       ;
2616| 0000 001C             RT11    EQU     DELENG+2
2616|                       ;
2616| 4E56 FFE4             ENTRTMP LINK    A6,#-RT11
261A| 6100 E266                     BSR     OVFCHK
261E| 244F                          MOVE.L  A7,A2                   ; @RT11
2620| 49EE 000E                     LEA     14(A6),A4               ; @FSEGS
2624| 4284                          MOVE.L  #0,D4                   ; sinx:=0, dinx:=0
2626| 266E 0008                     MOVE.L  8(A6),A3                ; FDIR
262A| 3A2B 0010                     MOVE.W  DNUMFLS(A3),D5          ; lasti:=fdirt[0].dnumfiles
262E| 204B                          MOVE.L  A3,A0
2630| 4246                          MOVE.W  #0,D6                   ; ssegs:=0
2632| 4252                          MOVE.W  #0,(A2)                 ; rt11:=false
2634| 4A54                          TST.W   (A4)                    ; if fsegs<=0 then
2636| 67**                          BEQ.S   @1
2638| 6A**                          BPL.S   @5                      ;   if fsegs<0 then rt11:=true
263A| 34BC 0001                     MOVE.W  #1,(A2)
2636* 06
PAGE -  98  MONITOR   FILE: FILPRC2.TEXT


263E| 4240                  @1      MOVE.W  #0,D0                   ;   for i:=1 to lasti do
2640| 60**                          BRA.S   @0
2642| 5240                  @2      ADD.W   #1,D0
2644| 3228 0002                     MOVE.W  LSTBLK(A0),D1           ;     findmax(i,fdirt[i-1].dlastblk,
2648| D0FC 001A                     ADD.W   #DELENG,A0              ;             fdirt[i].dfirstblk)
264C| 3410                          MOVE.W  FSTBLK(A0),D2
264E| 61A6                          BSR     FINDMAX
2640* 0E
2650| B045                  @0      CMP.W   D5,D0
2652| 66EE                          BNE.S   @2
2654| 5240                          ADD.W   #1,D0
2656| 3228 0002                     MOVE.W  LSTBLK(A0),D1           ;   findmax(lasti+1,
265A| 342B 000E                     MOVE.W  DEOVBLK(A3),D2          ;           fdirt[lasti].dlastblk,
```

```
265E| 6196                          BSR     FINDMAX                 ;           fdirt[0].deovblk)
2660| 4A52                          TST.W   (A2)                    ;   if rt11 then
2662| 67**                          BEQ.S   @4
2664| 3014                          MOVE.W  (A4),D0
2666| E248                          LSR.W   #1,D0
2668| B046                          CMP.W   D6,D0                   ;     if fsegs div 2 <= ssegs then
266A| 6E**                          BGT.S   @3
266C| 3886                          MOVE.W  D6,(A4)                 ;       fsegs:=ssegs
266E| 4844                          SWAP    D4                      ;       dinx:=sinx
2670| 60**                          BRA.S   @4
266A* 06
2672| 5254                @3        ADD.W   #1,(A4)                 ;     else
2674| E2D4                          LSR.W   (A4)                    ;       fsegs:=(fsegs+1) div 2
2670* 04
2662* 12
2676| 60**                @4        BRA.S   @9                      ; else {fsegs>0}
2638* 3E
2678| 204B                @5        MOVE.L  A3,A0
267A| 303C 0001                      MOVE.W  #1,D0                   ;   i:=1
267E| B045                @6        CMP.W   D5,D0                   ;   while i <= lasti do
2680| 6E**                          BGT.S   @8                      ;   begin
2682| 3228 0002                      MOVE.W  LSTBLK(A0),D1           ;     if fdirt[i].dfirstblk -
2686| D0FC 001A                      ADD.W   #DELENG,A0              ;     fdirt[i-1].dlastblk >= fsegs then
268A| 3410                          MOVE.W  FSTBLK(A0),D2
268C| 9441                          SUB.W   D1,D2
268E| B454                          CMP.W   (A4),D2
2690| 6D**                          BLT.S   @7
2692| 3800                          MOVE.W  D0,D4                   ;       dinx:=i
2694| 3005                          MOVE.W  D5,D0                   ;       i:=lasti
2690* 04
2696| 5240                @7        ADD.W   #1,D0                   ;     i:=i+1
2698| 60E4                          BRA.S   @6                      ;   end {while loop}
2680* 18
269A| 4A44                @8        TST.W   D4                      ;   if dinx=0 then
269C| 66**                          BNE.S   @9
269E| 342B 000E                      MOVE.W  DEOVBLK(A3),D2          ;     if fdirt[0].deovblk -
26A2| 3205                          MOVE.W  D5,D1                   ;     fdirt[i-1].dlastblk >= fsegs then
26A4| C3FC 001A                      MULS    #DELENG,D1
26A8| 9473 1802                      SUB.W   LSTBLK(A3,D1.L),D2
26AC| B454                          CMP.W   (A4),D2
26AE| 6D**                          BLT.S   @9
PAGE -  99  MONITOR   FILE: FILPRC2.TEXT


26B0| 3800                          MOVE.W  D0,D4                   ;       dinx:=lasti+1
26AE* 02
269C* 14
2676* 3A
26B2| 0C45 004D            @9        CMP.W   #MAXDIR,D5              ; if lasti = maxdir then
26B6| 6D**                          BLT.S   @10
26B8| 0C6B 0006 0002                CMP.W   #6,LSTBLK(A3)
26BE| 67**                          BEQ.S   @12
26C0| 0C45 0312                      CMP.W   #MAXDIRB,D5
26C4| 6D**                          BLT.S   @10
26BE* 06
26C6| 4244                @12       MOVE.W  #0,D4                   ;   dinx:=0
26C4* 02
26B6* 10
26C8| 3004                @10       MOVE.W  D4,D0
26CA| 67**                          BEQ.S   @11                     ; if dinx<>0 then
26CC| 43EE FFE6                      LEA     -DELENG(A6),A1
26D0| 5340                          SUB.W   #1,D0                   ;   dfirstblk:=fdirt[dinx-1].dlastblk
26D2| C1FC 001A                      MULS    #DELENG,D0
26D6| 32B3 0802                      MOVE.W  LSTBLK(A3,D0.L),FSTBLK(A1)
26DA| 3351 0002                      MOVE.W  FSTBLK(A1),LSTBLK(A1)   ;   dlastblk:=dfirstblk+fsegs
26DE| 3614                          MOVE.W  (A4),D3
26E0| D769 0002                      ADD.W   D3,LSTBLK(A1)
26E4| 336E 000C 0004                MOVE.W  12(A6),FKIND(A1)        ;   dfkind:=fkind
26EA| 206E 0010                      MOVE.L  16(A6),A0
26EE| 2F08                          MOVE.L  A0,-(A7)                ;   push source address
26F0| 3F3C 0001                      MOVE.W  #1,-(A7)                ;   push starting index
26F4| 1010                          MOVE.B  (A0),D0
26F6| 4880                          EXT.W   D0
26F8| 3F00                          MOVE.W  D0,-(A7)                ;   push length
26FA| 4869 0006                      PEA     DTID(A1)                ;   push dst address
26FE| 6100 F724                      BSR     TTLCOPY                 ;   dtid:=ftid
2702| 337C 0200 0016                MOVE.W  #FBLKSIZ,LSTBYTE(A1)
2708| 337C C800 0018                MOVE.W  #$C800,DACCESS(A1)      ;   month:=0, day:=0, year:=100
270E| 3F04                          MOVE.W  D4,-(A7)
2710| 2F09                          MOVE.L  A1,-(A7)
2712| 3F04                          MOVE.W  D4,-(A7)
2714| 2F2E 0008                      MOVE.L  8(A6),-(A7)
2718| 6100 FE52                      BSR     INSNTRY                 ;   insntry(lde,dinx,fdir)
271C| 381F                          MOVE.W  (A7)+,D4
26CA* 52
271E| 3D44 0016            @11       MOVE.W  D4,22(A6)               ; entrtmp:=dinx
2722| 4E5E                          UNLK    A6
2724| 205F                          MOVE.L  (A7)+,A0
2726| DEFC 000E                      ADD.W   #14,A7
272A| 4ED0                          JMP     (A0)
272C|                       ;
272C|                       ;       WRITEIT -- common point for unitwrites (for debugging)
272C|                       ;
272C|                       ;       registers on entry:
272C|                       ;
272C|                       ;               D0      unit#
```

```
272C|                          ;               D1      address
272C|                          ;               D2      length
272C|                          ;               D3      block
PAGE - 100  MONITOR   FILE: FILPRC2.TEXT


272C|                          ;
250C* 0220
272C|  7C02                   WRITEIT MOVE.L  #OUTBIT,D6
272E|  3F00                           MOVE.W  D0,-(A7)        ; push unit#
2730|  6100 F22A                      BSR     GETUNIT
2734|  4E90                           JSR     (A0)
2736|  6000 F298                      BRA     IOEXIT          ; note ioexit will do rts
273A|                          ;
273A|                          ;       WRTDIR -- write directory
273A|                          ;
273A|                          ;       stack:
273A|                          ;
273A|                          ;               8       UNIT#
273A|                          ;               4       LDIR
273A|                          ;               0       RETURN ADDRESS
273A|                          ;
273A|  221F                   WRTDIR  MOVE.L  (A7)+,D1
273C|  205F                           MOVE.L  (A7)+,A0                ; address of directory
273E|  3428 0010                      MOVE.W  DNUMFLS(A0),D2
2742|  5242                           ADD.W   #1,D2
2744|  C5FC 001A                      MULS    #DELENG,D2
2748|  0642 01FF                      ADD.W   #511,D2
274C|  0242 FE00                      AND.W   #$FE00,D2               ; length
2750|  C388                           EXG     D1,A0                   ; d1=address of dir, A0=return address
2752|  301F                           MOVE.W  (A7)+,D0                ; unit#
2754|  2F08                           MOVE.L  A0,-(A7)
2756|  7602                           MOVE.L  #DIRDISK,D3             ; block
2758|  60D2                           BRA     WRITEIT
275A|                          ;
275A|                          ;       FOPEN(F:FIB; TITLE:TID; OPENOLD:BOOLEAN; JUNK:FIBP)
275A|                          ;
275A|                          ;       stack
275A|                          ;
275A|                          ;               18      F
275A|                          ;               14      TITLE
275A|                          ;               12      OPENOLD
275A|                          ;               8       JUNK
275A|                          ;               4       RETURN ADDRESS
275A|                          ;               0       Old A6
275A|                          ;               -8      FREE8   8
275A|                          ;               -24     LTID    16
275A|                          ;               -26     LSEGS   word
275A|                          ;               -28     LKIND   word
275A|                          ;               -32     LDIR    ptr
275A|                          ;               -56     LVID    24
275A|                          ;               -60     Heaptop
275A|                          ;
275A|  4E56 FFC4             FOPEN   LINK    A6,#-60
275E|  6100 E122                      BSR     OVFCHK
2762|                                 GETNP   -60(A6)                 ; mark(heaptop)
2762|  2D6D FFF4 FFC4       #         MOVE.L   -12(A5),-60(A6)
2768|                          ;
2768|                          ; initialize and test if file is already open
2768|                          ;
2768|  426E FFE4                    MOVE.W  #0,-28(A6)              ; zero LKIND
PAGE - 101  MONITOR   FILE: FILPRC2.TEXT


276C|                                 GETSYSC A0
276C|  206D FFFC             #       MOVE.L   -4(A5),A0
2770|  4210                           MOVE.B  #INOERR,(A0)            ; initialize iorslt
2772|  226E 0012                      MOVE.L  18(A6),A1
2776|  4A29 000A                      TST.B   FISOPEN(A1)             ; if F.fisopen then
277A|  67**                           BEQ.S   @1
277C|  10BC 000C                      MOVE.B  #INOTCLS,(A0)           ; iorslt:=inotclosed
2780|  6000 ****                      BRA     FOPENX
2784|                          ;
2784|                          ; validate the title
2784|                          ;
277A* 08
2784|  42A7                   @1      MOVE.L  #0,-(A7)                ; func result
2786|  2F2E 000E                      MOVE.L  14(A6),-(A7)            ; push @title
278A|  486E FFC8                      PEA     -56(A6)                 ; push @lvid
278E|  486E FFE8                      PEA     -24(A6)                 ; push @ltid
2792|  486E FFE6                      PEA     -26(A6)                 ; push @lsegs
2796|  486E FFE4                      PEA     -28(A6)                 ; push @lkind
279A|  6100 F3C2                      BSR     SCANTTL
279E|  4A5F                           TST.W   (A7)+                   ; if not scantitle then
27A0|  66**                           BNE.S   @2
27A2|                                 GETSYSC A0
27A2|  206D FFFC             #       MOVE.L   -4(A5),A0
27A6|  10BC 0007                      MOVE.B  #IBADTTL,(A0)           ; iorslt:=ibadtitle
27AA|  6000 ****                      BRA     FOPENX
27AE|                          ;
27AE|                          ; scan title passed
27AE|                          ;
27A0* 0C
27AE|  42A7                   @2      MOVE.L  #0,-(A7)                ; func result
27B0|  486E FFC8                      PEA     -56(A6)                 ; push @lvid
27B4|  3F3C 0001                      MOVE.W  #1,-(A7)                ; push true
```

```
27B8| 486E FFE0                        PEA     -32(A6)                 ; push @ldir
27BC| 6100 F8F0                        BSR     VOLSRCH                 ; lunit:=volsearch
27C0| 301F                             MOVE.W  (A7)+,D0                ; get lunit
27C2| 66**                             BNE.S   @3                      ; if lunit=0 then
27C4|                                  GETSYSC A0
27C4| 206D FFFC                 #       MOVE.L   -4(A5),A0
27C8| 10BC 0009                        MOVE.B  #INOUNIT,(A0)           ; iorslt:=inounit
27CC| 6000 ****                        BRA     FOPENX
27D0|                           ;
27D0|                           ; now open the file
27D0|                           ;
27C2* 0C
27D0| 226E 0012                 @3      MOVE.L  18(A6),A1
27D4| 137C 0001 000A                     MOVE.B  #1,FISOPEN(A1)          ; fisopen:=true
27DA| 4269 001C                        MOVE.W  #0,FMODIFD(A1)          ; fmodified:=false
27DE| 3340 000C                        MOVE.W  D0,FUNIT(A1)            ; funit:=lunit
27E2| 41E9 000E                        LEA     FMACHIN(A1),A0          ; zero machine name for now
27E6| 4298                             CLR.L   (A0)+
27E8| 4298                             CLR.L   (A0)+
27EA|                           ; currently lvid is restricted to 16 bytes by volsrch (ie. no machine name)
27EA| 41EE FFC8                        LEA     -56(A6),A0              ; @lvid
27EE| 43E9 0040                        LEA     FDEVVID(A1),A1          ; @fdevvid
27F2| 22D8                             MOVE.L  (A0)+,(A1)+             ; FDEVVID:=LVID
PAGE - 102  MONITOR   FILE: FILPRC2.TEXT


27F4| 22D8                             MOVE.L  (A0)+,(A1)+
27F6| 22D8                             MOVE.L  (A0)+,(A1)+
27F8| 22D8                             MOVE.L  (A0)+,(A1)+
27FA| 226E 0012                        MOVE.L  18(A6),A1
27FE| 4269 0018                        MOVE.W  #0,FNXTBLK(A1)          ; fnxtblk:=0;
2802| 3400                             MOVE.W  D0,D2
2804| 6100 F660                        BSR     UNITISB
2808| 56C2                             SNE     D2
280A| 0202 0001                        AND.B   #1,D2
280E| 1342 000B                        MOVE.B  D2,FISBLKD(A1)
2812| 4241                             CLR.W   D1                      ; fsoftbf is a word
2814| 4A69 0008                        TST.W   FRECSZ(A1)
2818| 56C1                             SNE     D1
281A| C202                             AND.B   D2,D1
281C| 3341 0038                        MOVE.W  D1,FSOFTBF(A1)          ; fsoftbuf:=uisblkd and (frecsize<>0)
2820| 4280                             MOVE.L  #NIL,D0
2822| B0AE FFE0                        CMP.L   -32(A6),D0
2826| 6700 ****                        BEQ     @11
282A| 4A2E FFE8                        TST.B   -24(A6)                 ; if ldir<>nil and len(ltid)<>0 then
282E| 6700 ****                        BEQ     @11
2832|                           ;
2832|                           ; search the directory
2832|                           ;
2832| 42A7                             MOVE.L  #0,-(A7)                ; func result
2834| 486E FFE8                        PEA     -24(A6)                 ; push @ltid
2838| 3F2E 000C                        MOVE.W  12(A6),-(A7)            ; push fopenold
283C| 2F2E FFE0                        MOVE.L  -32(A6),-(A7)           ; push @ldir
2840| 6100 FB74                        BSR     DIRSRCH                 ; linx:=dirsearch
2844| 4A6E 000C                        TST.W   12(A6)                  ; if fopenold then
2848| 67**                             BEQ.S   @6
284A| 301F                             MOVE.W  (A7)+,D0                ; if linx=0 then
284C| 66**                             BNE.S   @4
284E|                                  GETSYSC A0
284E| 206D FFFC                 #       MOVE.L   -4(A5),A0
2852| 10BC 000A                        MOVE.B  #INOFILE,(A0)           ; iorslt:=inofile
2856| 6000 ****                        BRA     FOPEN1
285A|                           ;
285A|                           ; enter with linx * deleng in D0 !
285A|                           ;
284C* 0C
285A| 720D                     @4      MOVE.L  #DELENG/2,D1            ; else
285C| 206E FFE0                        MOVE.L  -32(A6),A0
2860| D0C0                             ADD.W   D0,A0                   ; compute @ldir+linx
2862| 226E 0012                        MOVE.L  18(A6),A1               ; with F
2866| 45E9 001E                        LEA     FHEADER(A1),A2          ; with FHEADER
286A| 264A                             MOVE.L  A2,A3
286C| 36D8                     @5      MOVE.W  (A0)+,(A3)+             ; copy directory to fheader
286E| 5341                             SUB.W   #1,D1
2870| 66FA                             BNE.S   @5
2872| 4A6E 000C                        TST.W   12(A6)                  ; if fopenold then goto FOPEN2
2876| 6600 ****                        BNE     FOPEN2
287A| 337C 0001 001C                  MOVE.W  #1,FMODIFD(A1)          ; fmodified:=true
2880| 3F29 000C                        MOVE.W  FUNIT(A1),-(A7)         ; push funit (same as lunit)
2884| 2F2E FFE0                        MOVE.L  -32(A6),-(A7)           ; push ldir
2888| 6100 FEB0                        BSR     WRTDIR                  ; writedir(lunit,ldir)
PAGE - 103  MONITOR   FILE: FILPRC2.TEXT


288C| 226E 0012                        MOVE.L  18(A6),A1
2890| 45E9 001E                        LEA     FHEADER(A1),A2
2894| 6000 ****                        BRA     FOPEN2                  ; restore registers and goto FOPEN2
2898|                           ;
2898|                           ; open new file
2898|                           ;
2848* 4E
2898| 301F                     @6      MOVE.W  (A7)+,D0                ; if linx>0 then
289A| 67**                             BEQ.S   @7
289C|                                  GETSYSC A0
289C| 206D FFFC                 #       MOVE.L   -4(A5),A0
28A0| 10BC 000B                        MOVE.B  #IDUPFIL,(A0)           ; iorslt:=idupfile
```

```
28A4| 6000 ****                    BRA     FOPEN1
289A* 0C
28A8| 43EE FFE4           @7      LEA     -28(A6),A1
28AC| 0C29 0000 0001              CMP.B   #UNTYPFL,1(A1)          ; if lkind=untypedfile then
28B2| 66**                        BNE.S   @8
28B4| 137C 0005 0001              MOVE.B  #DATAFILE,1(A1)         ; lkind:=datafile
28B2* 06
28BA| 42A7                 @8      MOVE.L  #0,-(A7)                ; func result
28BC| 486E FFE8                   PEA     -24(A6)                 ; push @ltid
28C0| 3F2E FFE6                   MOVE.W  -26(A6),-(A7)           ; push lsegs
28C4| 3F11                       MOVE.W  (A1),-(A7)              ; push lkind
28C6| 2F2E FFE0                   MOVE.L  -32(A6),-(A7)           ; push ldir
28CA| 6100 FD4A                   BSR     ENTRTMP                 ; linx:=entertemp
28CE| 301F                       MOVE.W  (A7)+,D0                ; if linx=0 then
28D0| 66**                        BNE.S   @9
28D2|                             GETSYSC A0
28D2| 206D FFFC             #      MOVE.L   -4(A5),A0
28D6| 10BC 0008                   MOVE.B  #INOROOM,(A0)           ; iorslt:=inoroom
28DA| 6000 ****                   BRA     FOPEN1
28D0* 0C
28DE| 3400                 @9      MOVE.W  D0,D2
28E0| C1FC 001A                   MULS    #DELENG,D0              ; setup D0 with linx * deleng
28E4| 43EE FFE4                   LEA     -28(A6),A1
28E8| 0C29 0003 0001              CMP.B   #TEXTFILE,1(A1)         ; if lkind=textfile then
28EE| 6600 FF6A                   BNE.S   @4
28F2| 246E FFE0                   MOVE.L  -32(A6),A2
28F6| D4C0                       ADD.W   D0,A2                   ; compute @ldir+linx
28F8| 322A 0002                   MOVE.W  LSTBLK(A2),D1
28FC| 9252                       SUB.W   FSTBLK(A2),D1
28FE| 0801 0000                   BTST    #0,D1                   ; if odd(lastblock-firstblock)
2902| 67**                        BEQ.S   @10
2904| 536A 0002                   SUB.W   #1,LSTBLK(A2)           ; lastblock:=lastblock-1
2908| 5341                       SUB.W   #1,D1
2902* 06
290A| 5941                 @10     SUB.W   #4,D1                   ; if lastblock-firstblock>=4 then
290C| 6A00 FF4C                   BPL     @4                      ;   copy fheader with linx in D0
2910| 3F02                       MOVE.W  D2,-(A7)                ; else
2912| 2F2E FFE0                   MOVE.L  -32(A6),-(A7)
2916| 6100 FCA8                   BSR     DELNTRY                 ;   delentry(linx,ldir)
291A| 4267                       MOVE.W  #0,-(A7)                ;   linx:=0
291C| 609C                       BRA     @8                      ;   go set iosrlt:=inoroom
2830* 00EE
2828* 00F6
PAGE - 104  MONITOR   FILE: FILPRC2.TEXT


291E| 45E9 001E           @11     LEA     FHEADER(A1),A2          ; with FHEADER
2922| 4252                       MOVE.W  #0,FSTBLK(A2)           ; firstblock:=0
2924| 357C 7FFF 0002              MOVE.W  #$7FFF,LSTBLK(A2)       ; lastblock:=maxint
292A| 3429 000C                   MOVE.W  FUNIT(A1),D2            ; funit (same as lunit)
292E| 6100 F536                   BSR     UNITISB                 ; if uisblkd then
2932| 67**                        BEQ.S   @12
2934| 206E FFE0                   MOVE.L  -32(A6),A0
2938| 3568 000E 0002              MOVE.W  DEOVBLK(A0),LSTBLK(A2)  ;   lastblock:=ueovblk
2932* 0A
293E| 356E FFE4 0004         @12     MOVE.W  -28(A6),FKIND(A2)       ; dfkind:=lkind
2944| 422A 0006                   MOVE.B  #0,DTID(A2)             ; dtid:=''
2948| 357C 0200 0016              MOVE.W  #FBLKSIZ,LSTBYTE(A2)    ; lastbyte:=fblksize
294E| 426A 0018                   MOVE.W  #0,DACCESS(A2)          ; set date to zero
2952|                         ;
2952|                         ;       setup file size in FMAXBLK
2952|                         ;
2896* 00BC
2878* 00DA
2952| 4240                 FOPEN2  MOVE.W  #0,D0                   ; temp:=0
2954| 4A6E 000C                   TST.W   12(A6)                  ; if openold then
2958| 67**                        BEQ.S   @1
295A| 302A 0002                   MOVE.W  LSTBLK(A2),D0
295E| 9052                       SUB.W   FSTBLK(A2),D0           ; temp:=lastblock-firstblock
2958* 06
2960| 3340 0016           @1      MOVE.W  D0,FMAXBLK(A1)          ; fmaxblk:=temp
2964|                         ;
2964|                         ; do softbuf stuff
2964|                         ;
2964| 4A69 0038                   TST.W   FSOFTBF(A1)
2968| 67**                        BEQ.S   @4
296A| 337C 0200 003C              MOVE.W  #FBLKSIZ,FNXTBYT(A1)    ; fnxtbyte:=fblksize
2970| 4269 003E                   MOVE.W  #0,FBFCHNG(A1)          ; fbufchngd:=false
2974| 303C 0200                   MOVE.W  #FBLKSIZ,D0             ; temp:=fblksize
2978| 4A6E 000C                   TST.W   12(A6)                  ; if openold then
297C| 67**                        BEQ.S   @2
297E| 302A 0016                   MOVE.W  LSTBYTE(A2),D0          ; temp:=lastbyte
297C* 04
2982| 3340 003A           @2      MOVE.W  D0,FMAXBYT(A1)          ; fmaxbyte:=temp
2986| 302A 0004                   MOVE.W  FKIND(A2),D0
298A| 0C40 0003                   CMP.W   #TEXTFILE,D0            ; if filekind = textfile then
298E| 66**                        BNE.S   @4
2990| 337C 0002 0018              MOVE.W  #2,FNXTBLK(A1)          ; fnxtblk:=2
2996| 4A6E 000C                   TST.W   12(A6)                  ; if not fopenold then
299A| 66**                        BNE.S   @4
299C|                         ;
299C|                         ; fill first two blocks of the file with zeros
299C|                         ;
299C| 303C 0100                   MOVE.W  #FBLKSIZ/2,D0
29A0| 47E9 0050                   LEA     FBUFFER(A1),A3
29A4| 220B                       MOVE.L  A3,D1                   ; address of directory buffer
```

```
29A6| 425B                    @3      MOVE.W  #0,(A3)+                ; fill fbuffer with zeroes
29A8| 5340                            SUB.W   #1,D0
29AA| 66FA                            BNE     @3
29AC| 3029 000C                       MOVE.W  FUNIT(A1),D0            ; unit #
29B0| 343C 0200                       MOVE.W  #FBLKSIZ,D2             ; length
PAGE - 105  MONITOR   FILE: FILPRC2.TEXT


29B4| 45E9 001E                       LEA     FHEADER(A1),A2
29B8| 3612                            MOVE.W  FSTBLK(A2),D3           ; block
29BA| 48E7 F000                       MOVEM.L D0-D3,-(A7)
29BE| 6100 FD6C                       BSR     WRITEIT
29C2| 4CDF 000F                       MOVEM.L (A7)+,D0-D3             ; next block
29C6| 5243                            ADD.W   #1,D3
29C8| 6100 FD62                       BSR     WRITEIT
29CC|                          ;
29CC|                          ; if fopenold then freset(f) else reseter(f)
29CC|                          ;
299A* 30
298E* 3C
2968* 62
29CC| 2F2E 0012                 @4      MOVE.L  18(A6),-(A7)            ; push F
29D0| 4A6E 000C                       TST.W   12(A6)                  ; if fopenold then
29D4| 67**                            BEQ.S   @5
29D6| 6100 FA54                       BSR     FRESET                  ;   freset(F)
29DA| 60**                            BRA.S   FOPEN1                  ; else
29D4* 06
29DC| 6100 FA8E                 @5      BSR     RESETER                 ;   reseter(F)
29E0|                          ;
29E0|                          ; test iorslt, if nonzero then backup the fopen (ie. it failed)
29E0|                          ;
29E0|                          FOPEN1  GETSYSC A0
29DA* 04
28DC* 0104
28A6* 013A
2858* 0188
29E0| 206D FFFC               #       MOVE.L   -4(A5),A0
29E4| 4A10                            TST.B   (A0)                    ; if iorslt<>0 then
29E6| 67**                            BEQ.S   FOPENX
29E8| 226E 0012                       MOVE.L  18(A6),A1
29EC| 4229 000A                       MOVE.B  #0,FISOPEN(A1)          ; fisopen:=false
29F0| 4229 0005                       MOVE.B  #0,FEOF(A1)             ; feof:=false
29F4| 4229 0004                       MOVE.B  #0,FEOLN(A1)            ; feoln:=false
29F8|                          ;
29F8|                          ; cleanup and exit
29F8|                          ;
29F8|                          FOPENX  PUTNP   -60(A6)                 ; release(heaptop)
29E6* 10
27CE* 022A
27AC* 024C
2782* 0276
29F8| 2B6E FFC4 FFF4           #       MOVE.L   -60(A6),-12(A5)
29FE| 4E5E                            UNLK    A6
2A00| 205F                            MOVE.L  (A7)+,A0        ; pop return address
2A02| DEFC 000E                       ADD.W   #14,A7          ; delete parameters
2A06| 4ED0                            JMP     (A0)
2A08|                          ;
2A08|                          ;       FCLOSE(F:FIB; TITLE:TID)
2A08|                          ;
2A08|                          ;       stack
2A08|                          ;
2A08|                          ;               10      F
2A08|                          ;               8       FTYPE
PAGE - 106  MONITOR   FILE: FILPRC2.TEXT


2A08|                          ;               4       RETURN ADDRESS
2A08|                          ;               0       Old A6
2A08|                          ;               -2      FOUND
2A08|                          ;               -6      LDIR
2A08|                          ;               -8      LINX
2A08|                          ;               -32     FREE24
2A08|                          ;               -56     LVID
2A08|                          ;               -60     Heaptop
2A08|                          ;
2A08| 4E56 FFC4                 FCLOSE  LINK    A6,#-60
2A0C| 6100 DE74                       BSR     OVFCHK
2A10|                                 GETNP   -60(A6)                 ; mark(heaptop)
2A10| 2D6D FFF4 FFC4           #       MOVE.L   -12(A5),-60(A6)
2A16| 226E 000A                       MOVE.L  10(A6),A1
2A1A| 4A29 000A                       TST.B   FISOPEN(A1)             ; if fisopen then
2A1E| 6700 ****                       BEQ     FCLOSEX
2A22| 45E9 001E                       LEA     FHEADER(A1),A2
2A26| 4A29 000B                       TST.B   FISBLKD(A1)             ; if fisblkd then
2A2A| 6700 ****                       BEQ     @19
2A2E| 4A2A 0006                       TST.B   DTID(A2)                ; if dtid<>'' then
2A32| 6700 ****                       BEQ     @19
2A36| 0C6E 0003 0008                  CMP.W   #CCRUNCH,8(A6)          ; if ftype=ccrunch then
2A3C| 66**                            BNE.S   @1
2A3E| 3369 0018 0016                  MOVE.W  FNXTBLK(A1),FMAXBLK(A1) ;   fmaxblk:=fnxtblk
2A44| 026A 01FF 0018                  AND.W   #$1FF,DACCESS(A2)       ;   daccess.year:=100
2A4A| 006A C800 0018                  OR.W    #$C800,DACCESS(A2)
2A50| 3D7C 0001 0008                  MOVE.W  #CLOCK,8(A6)            ;   ftype:=clock
2A56| 4A69 0038                       TST.W   FSOFTBF(A1)             ;   if fsoftbuf then
2A5A| 67**                            BEQ.S   @1
2A5C| 3369 003C 003A                  MOVE.W  FNXTBYT(A1),FMAXBYT(A1) ;     fmaxbyte:=fnxtbyte
```

```
2A5A* 06
2A3C* 24
2A62| 2F09                    @1      MOVE.L  A1,-(A7)
2A64| 6100 FA06                       BSR     RESETER                 ; reseter(F)
2A68| 226E 000A                       MOVE.L  10(A6),A1
2A6C| 4A69 001C                       TST.W   FMODIFD(A1)             ; if fmodified or
2A70| 66**                           BNE.S   @2
2A72| 45E9 001E                       LEA     FHEADER(A1),A2
2A76| 302A 0018                       MOVE.W  DACCESS(A2),D0          ;    daccess.year=100 or
2A7A| 0240 FE00                       ANDI.W  #$FE00,D0
2A7E| 0C40 C800                       CMP.W   #$C800,D0
2A82| 67**                           BEQ.S   @2
2A84| 0C6E 0002 0008                 CMP.W   #CPURGE,8(A6)           ;    ftype=cpurge then
2A8A| 6600 ****                       BNE     @19
2A8E|                         ; currently lvid is restricted to 16 bytes by volsrch (ie. no machine name)
2A82* 0A
2A70* 1C
2A8E| 41E9 0040                @2      LEA     FDEVVID(A1),A0          ; @fdevvid
2A92| 43EE FFC8                       LEA     -56(A6),A1              ; @lvid
2A96| 22D8                           MOVE.L  (A0)+,(A1)+             ; LVID:=FDEVVID
2A98| 22D8                           MOVE.L  (A0)+,(A1)+
2A9A| 22D8                           MOVE.L  (A0)+,(A1)+
2A9C| 22D8                           MOVE.L  (A0)+,(A1)+
2A9E| 42A7                           MOVE.L  #0,-(A7)
2AA0| 486E FFC8                       PEA     -56(A6)
PAGE - 107  MONITOR   FILE: FILPRC2.TEXT


2AA4| 4267                           MOVE.W  #0,-(A7)
2AA6| 486E FFFA                       PEA     -6(A6)
2AAA| 6100 F602                       BSR     VOLSRCH         ; if funit<>volsearch(lvid,false,ldir) then
2AAE| 301F                           MOVE.W  (A7)+,D0
2AB0| 226E 000A                       MOVE.L  10(A6),A1
2AB4| 45E9 001E                       LEA     FHEADER(A1),A2
2AB8| B069 000C                       CMP.W   FUNIT(A1),D0
2ABC| 67**                           BEQ.S   @3
2ABE|                                 GETSYSC A0
2ABE| 206D FFFC                #       MOVE.L   -4(A5),A0
2AC2| 10BC 0005                       MOVE.B  #ILSTUNT,(A0)
2AC6| 6000 ****                       BRA     FCLOSE1
2ABC* 0C
2ACA| 4243                    @3      MOVE.W  #0,D3                   ; found:=false
2ACC| 7401                           MOVE.L  #1,D2                   ; linx:=1
2ACE| 206E FFFA                       MOVE.L  -6(A6),A0               ; get LDIR
2AD2| 3228 0010                       MOVE.W  DNUMFLS(A0),D1          ; get #FILES in directory
2AD6| 2648                           MOVE.L  A0,A3
2AD8| D6FC 001A                       ADD.W   #DELENG,A3              ; compute LDIR↑[1]
2ADC| B441                    @4      CMP.W   D1,D2                   ; while linx<=#FILES and not found do
2ADE| 6E**                           BGT.S   @6
2AE0| 4A43                           TST.W   D3
2AE2| 66**                           BNE.S   @6
2AE4| 3012                           MOVE.W  FSTBLK(A2),D0
2AE6| B053                           CMP.W   FSTBLK(A3),D0           ; t1:=ldir↑[linx].dfirstblk=dfirstblk
2AE8| 66**                           BNE.S   @5
2AEA| 302A 0002                       MOVE.W  LSTBLK(A2),D0
2AEE| B06B 0002                       CMP.W   LSTBLK(A3),D0           ; t2:=ldir↑[linx].dlastblk=dlastblk
2AF2| 66**                           BNE.S   @5
2AF4| 7601                           MOVE.L  #1,D3                   ; found:=t1 and t2
2AF2* 02
2AE8* 0C
2AF6| 5242                    @5      ADD.W   #1,D2                   ; linx:=linx+1
2AF8| D6FC 001A                       ADD.W   #DELENG,A3
2AFC| 60DE                           BRA.S   @4
2AE2* 1A
2ADE* 1E
2AFE| 4A43                    @6      TST.W   D3                      ; if not found then
2B00| 66**                           BNE.S   @7
2B02|                                 GETSYSC A0
2B02| 206D FFFC                #       MOVE.L   -4(A5),A0
2B06| 10BC 0006                       MOVE.B  #ILSTFIL,(A0)           ;   iorslt:=ilostfile
2B0A| 6000 ****                       BRA     FCLOSE1                 ;   goto 1
2B00* 0C
2B0E| 5342                    @7      SUB.W   #1,D2                   ; linx:=linx-1
2B10| 3D42 FFF8                       MOVE.W  D2,-8(A6)
2B14| 0C6E 0000 0008                 CMP.W   #CNORMAL,8(A6)          ; if (ftype=cnormal and
2B1A| 66**                           BNE.S   @8
2B1C| 206E FFFA                       MOVE.L  -6(A6),A0
2B20| 3602                           MOVE.W  D2,D3
2B22| C7FC 001A                       MULS    #DELENG,D3
2B26| 3030 3018                       MOVE.W  DACCESS(A0,D3),D0       ;   ldir↑[linx].daccess.year=100) or
2B2A| 0240 FE00                       AND.W   #$FE00,D0
2B2E| 0C40 C800                       CMP.W   #$C800,D0
2B32| 67**                           BEQ.S   @9
2B1A* 18
PAGE - 108  MONITOR   FILE: FILPRC2.TEXT


2B34| 0C6E 0002 0008          @8      CMP.W   #CPURGE,8(A6)           ;   ftype=cpurge then
2B3A| 66**                           BNE.S   @10
2B32* 08
2B3C| 3F02                    @9      MOVE.W  D2,-(A7)
2B3E| 2F08                           MOVE.L  A0,-(A7)
2B40| 6100 FA7E                       BSR     DELNTRY                 ;   delentry(linx,ldir)
2B44| 6000 ****                       BRA     @18
2B3A* 0C
2B48| 226E 000A                @10     MOVE.L  10(A6),A1               ; else
```

```
2B4C| 45E9 001E                    LEA     FHEADER(A1),A2
2B50| 42A7                         MOVE.L  #0,-(A7)
2B52| 486A 0006                    PEA     DTID(A2)
2B56| 3F3C 0001                    MOVE.W  #1,-(A7)
2B5A| 2F08                         MOVE.L  A0,-(A7)
2B5C| 6100 F858                    BSR     DIRSRCH                 ;   dupinx:=dirsrch(dtid,true,ldir)
2B60| 301F                         MOVE.W  (A7)+,D0
2B62| 67**                         BEQ.S   @11                     ;   if dupinx<>0 and dupinx<>linx then
2B64| 48C0                         EXT.L   D0
2B66| 81FC 001A                    DIVS    #DELENG,D0
2B6A| B06E FFF8                    CMP.W   -8(A6),D0
2B6E| 67**                         BEQ.S   @11
2B70| 3F00                         MOVE.W  D0,-(A7)
2B72| 3F00                         MOVE.W  D0,-(A7)
2B74| 2F2E FFFA                    MOVE.L  -6(A6),-(A7)
2B78| 6100 FA46                    BSR     DELNTRY                 ;     delentry(dupinx,ldir)
2B7C| 301F                         MOVE.W  (A7)+,D0
2B7E| B06E FFF8                    CMP.W   -8(A6),D0               ;     if dupinx<linx then
2B82| 6C**                         BGE.S   @11
2B84| 536E FFF8                    SUB.W   #1,-8(A6)               ;       linx:=linx-1
2B82* 04
2B6E* 18
2B62* 24
2B88| 226E 000A             @11    MOVE.L  10(A6),A1
2B8C| 45E9 001E                    LEA     FHEADER(A1),A2
2B90| 362E FFF8                    MOVE.W  -8(A6),D3
2B94| C7FC 001A                    MULS    #DELENG,D3
2B98| 206E FFFA                    MOVE.L  -6(A6),A0
2B9C| 3030 3018                    MOVE.W  DACCESS(A0,D3),D0       ;   if ldirf[linx].daccess.year=100 then
2BA0| 0240 FE00                    AND.W   #$FE00,D0
2BA4| 0C40 C800                    CMP.W   #$C800,D0
2BA8| 66**                         BNE.S   @13
2BAA| 0C6A C800 0018               CMP.W   #$C800,DACCESS(A2)      ;     if daccess.year=100 then
2BB0| 66**                         BNE.S   @12
2BB2|                        @14   ADDRTHD A3
2BB2| 47ED FFCA                 #    LEA     -54(A5),A3
2BB6| 3553 0018                    MOVE.W  (A3),DACCESS(A2)        ;       daccess:=thedate
2BB0* 08
2BBA| 60**                         BRA.S   @15                     ;     else
2BA8* 12
2BBC| 4A69 001C             @13    TST.W   FMODIFD(A1)             ;       if fmodified then
2BC0| 66F0                         BNE.S   @14                     ;         daccess:=thedate
2BC2| 3570 3018 0018               MOVE.W  DACCESS(A0,D3),DACCESS(A2)
2BBA* 0C
2BC8| 3552 0002             @15    MOVE.W  FSTBLK(A2),LSTBLK(A2)   ;   dlastblk:=dfirstblk+fmaxblk
PAGE - 109  MONITOR   FILE: FILPRC2.TEXT


2BCC| 3029 0016                    MOVE.W  FMAXBLK(A1),D0
2BD0| D16A 0002                    ADD.W   D0,LSTBLK(A2)
2BD4| 4A69 0038                    TST.W   FSOFTBF(A1)             ;   if softbf then
2BD8| 67**                         BEQ.S   @16
2BDA| 3569 003A 0016               MOVE.W  FMAXBYT(A1),LSTBYTE(A2) ;     dlastbyte:=fmaxbyte
2BD8* 06
2BE0| 026A 000F 0004         @16   ANDI.W  #$000F,FKIND(A2)        ;   fheader.filler1:=0
2BE6| 4269 001C                    MOVE.W  #0,FMODIFD(A1)          ;   fmodified:=false
2BEA| 47E9 001E                    LEA     FHEADER(A1),A3
2BEE| 49F0 3000                    LEA     0(A0,D3),A4
2BF2| 700D                         MOVE.L  #DELENG/2,D0
2BF4| 38DB                   @17   MOVE.W  (A3)+,(A4)+             ;   ldirf[linx]:=fheader
2BF6| 5340                         SUB.W   #1,D0
2BF8| 66FA                         BNE.S   @17
2B46* 00B4
2BFA| 226E 000A             @18   MOVE.L  10(A6),A1
2BFE| 3F29 000C                    MOVE.W  FUNIT(A1),-(A7)         ; push funit
2C02| 2F2E FFFA                    MOVE.L  -6(A6),-(A7)            ; push ldir
2C06| 6100 FB32                    BSR     WRTDIR                  ; writedir(lunit,ldir)
2A8C* 017E
2A34* 01D6
2A2C* 01DE
2C0A| 226E 000A             @19   MOVE.L  10(A6),A1
2C0E| 45E9 001E                    LEA     FHEADER(A1),A2
2C12| 0C6E 0002 0008               CMP.W   #CPURGE,8(A6)           ; if ftype=cpurge then
2C18| 66**                         BNE.S   FCLOSE1
2C1A| 4A2A 0006                    TST.B   DTID(A2)                ;   if dtid='' then
2C1E| 66**                         BNE.S   FCLOSE1
2C20| 4E71                         NOP                             ;     ???
2C1E* 02
2C18* 08
2B0C* 0116
2AC8* 015A
2C22| 4229 000A             FCLOSE1 MOVE.B  #0,FISOPEN(A1)         ; fisopen:=false
2C26| 137C 0001 0005               MOVE.B  #1,FEOF(A1)             ; feof:=false
2C2C| 137C 0001 0004               MOVE.B  #1,FEOLN(A1)            ; feoln:=false
2C32|                        FCLOSEX PUTNP   -60(A6)               ; release(heaptop)
2A20* 0212
2C32| 2B6E FFC4 FFF4          #     MOVE.L   -60(A6),-12(A5)
2C38| 4E5E                         UNLK    A6
2C3A| 205F                         MOVE.L  (A7)+,A0                ; pop return address
2C3C| 5C4F                         ADD.W   #6,A7                   ; delete parameters
2C3E| 4ED0                         JMP     (A0)
2C40|                        ;
2C40|
2C40|
2C40|                        ;
2C40|                               .INCLUDE BLOCKIO.TEXT
```

```
2C40|                          ;
2C40|                          ;       Filename blockio
2C40|                          ;
2C40|
2C40|                          ;
2C40|                          ;       CANTSTR -- CANTSTRETCH
2C40|                          ;
PAGE - 110  MONITOR   FILE: BLOCKIO.TEXT


2C40|                          ;       stack:
2C40|                          ;
2C40|                          ;               12      Func result
2C40|                          ;               8       File Pointer
2C40|                          ;               4       Return address
2C40|                          ;               0       Old A6
2C40|                          ;               -2      OK
2C40|                          ;               -6      LDIR
2C40|                          ;               -10     LASTAVAILBLK
2C40|                          ;               -32     FREE22
2C40|                          ;               -56     LVID
2C40|                          ;               -60     Heaptop
2C40|                          ;
2C40| 4E56 FFC4               CANTSTR LINK    A6,#-60
2C44| 6100 DC3C                       BSR     OVFCHK
2C48|                                GETNP   -60(A6)                 ; mark(heaptop)
2C48| 2D6D FFF4 FFC4          #       MOVE.L  -12(A5),-60(A6)
2C4E| 3D7C 0001 000E                 MOVE.W  #1,14(A6)       ; cantstretch:=true
2C54| 426E FFFE                       MOVE.W  #0,-2(A6)       ; ok:=false
2C58| 226E 0008                       MOVE.L  8(A6),A1
2C5C| 45E9 001E                       LEA     FHEADER(A1),A2
2C60| 4A2A 0006                       TST.B   DTID(A2)        ; if length(DTID) then
2C64| 67**                           BEQ.S   @0
2C66|                          ; currently lvid is restricted to 16 bytes by volsrch (ie. no machine name)
2C66| 41E9 0040                       LEA     FDEVVID(A1),A0          ; @fdevvid
2C6A| 43EE FFC8                       LEA     -56(A6),A1              ; @lvid
2C6E| 22D8                           MOVE.L  (A0)+,(A1)+             ; LVID:=FDEVVID
2C70| 22D8                           MOVE.L  (A0)+,(A1)+
2C72| 22D8                           MOVE.L  (A0)+,(A1)+
2C74| 22D8                           MOVE.L  (A0)+,(A1)+
2C76| 42A7                           MOVE.L  #0,-(A7)
2C78| 486E FFC8                       PEA     -56(A6)
2C7C| 4267                           MOVE.W  #0,-(A7)
2C7E| 486E FFFA                       PEA     -6(A6)
2C82| 6100 F42A                       BSR     VOLSRCH         ; if funit<>volsearch(lvid,false,ldir) then
2C86| 301F                           MOVE.W  (A7)+,D0
2C88| 226E 0008                       MOVE.L  8(A6),A1
2C8C| 45E9 001E                       LEA     FHEADER(A1),A2
2C90| B069 000C                       CMP.W   FUNIT(A1),D0
2C94| 67**                           BEQ.S   @1
2C96|                                GETSYSC A0
2C96| 206D FFFC                #       MOVE.L  -4(A5),A0
2C9A| 10BC 0005                       MOVE.B  #ILSTUNT,(A0)
2C64* 38
2C9E| 6000 ****                 @0      BRA     CANTST1
2C94* 0C
2CA2| 4243                     @1      MOVE.W  #0,D3           ; found:=false
2CA4| 7401                             MOVE.L  #1,D2           ; linx:=1
2CA6| 206E FFFA                       MOVE.L  -6(A6),A0       ; get LDIR
2CAA| 3228 0010                       MOVE.W  DNUMFLS(A0),D1  ; get #FILES in directory
2CAE| 2648                           MOVE.L  A0,A3
2CB0| D6FC 001A                       ADD.W   #DELENG,A3      ; compute LDIR↑[1]
2CB4| B441                     @2      CMP.W   D1,D2           ; while (linx<=#FILES) and (not found) do
2CB6| 6E**                           BGT.S   @4
2CB8| 4A43                           TST.W   D3
PAGE - 111  MONITOR   FILE: BLOCKIO.TEXT


2CBA| 66**                           BNE.S   @4
2CBC| 3012                           MOVE.W  FSTBLK(A2),D0
2CBE| B053                           CMP.W   FSTBLK(A3),D0   ; t1:=(ldir↑[linx].dfirstblk=dfirstblk)
2CC0| 66**                           BNE.S   @3
2CC2| 302A 0002                       MOVE.W  LSTBLK(A2),D0
2CC6| B06B 0002                       CMP.W   LSTBLK(A3),D0   ; t2:=(ldir↑[linx].dlastblk=dlastblk)
2CCA| 66**                           BNE.S   @3
2CCC| 7601                           MOVE.L  #1,D3           ; found:=t1 and t2
2CCA* 02
2CC0* 0C
2CCE| 5242                     @3      ADD.W   #1,D2           ; linx:=linx+1
2CD0| D6FC 001A                       ADD.W   #DELENG,A3
2CD4| 60DE                           BRA.S   @2
2CBA* 1A
2CB6* 1E
2CD6| 4A43                     @4      TST.W   D3              ; if not found then iorslt:=ilostfile; goto 1
2CD8| 66**                           BNE.S   @5
2CDA|                                GETSYSC A0
2CDA| 206D FFFC                #       MOVE.L  -4(A5),A0
2CDE| 10BC 0006                       MOVE.B  #ILSTFIL,(A0)
2CE2| 60**                           BRA.S   CANTST1
2CD8* 0A
2CE4| B242                     @5      CMP.W   D2,D1           ; if linx > #FILES then
2CE6| 6E**                           BGT.S   @6
2CE8| 3828 000E                       MOVE.W  DEOVBLK(A0),D4  ;   lastavailblk:=ldir↑[0].dnumfiles
2CEC| 60**                           BRA.S   @7              ; else
2CE6* 06
2CEE| 3813                     @6      MOVE.W  FSTBLK(A3),D4   ;   lastavailblk:=ldir↑[linx].dfirstblk
```

```
2CEC* 02
2CF0| 302A 0002            @7      MOVE.W  LSTBLK(A2),D0   ; if (dlstblk<lastavailblk) or
2CF4| B044                          CMP.W   D4,D0
2CF6| 6C**                          BGE.S   @8
2CF8| 302A 0016                     MOVE.W  LSTBYTE(A2),D0  ;    (dlstbyte<fblksize) then
2CFC| 0C40 0200                     CMPI.W  #FBLKSIZ,D0
2D00| 6C**                          BGE.S   @8
2D02| 96FC 001A                     SUB.W   #DELENG,A3
2D06| 3744 0002                     MOVE.W  D4,LSTBLK(A3)   ;    dlastblk:=lastavailblk;
2D0A| 377C 0200 0016                MOVE.W  #FBLKSIZ,LSTBYTE(A3)
2D10| 3D44 FFF6                     MOVE.W  D4,-10(A6)      ;    dlastbyte:=fblksize
2D14| 3F29 000C                     MOVE.W  FUNIT(A1),-(A7)
2D18| 2F2E FFFA                     MOVE.L  -6(A6),-(A7)
2D1C| 6100 FA1C                     BSR     WRTDIR          ;    writedir(funit,ldir)
2D20|                               GETSYSC A0
2D20| 206D FFFC            #       MOVE.L   -4(A5),A0
2D24| 4A10                          TST.B   (A0)            ; if iorslt<>0 then goto 1
2D26| 66**                          BNE.S   CANTST1
2D28| 226E 0008                     MOVE.L  8(A6),A1
2D2C| 45E9 001E                     LEA     FHEADER(A1),A2
2D30| 4229 0005                     MOVE.B  #0,FEOF(A1)     ; feof:=false
2D34| 137C 0001 0004                MOVE.B  #1,FEOLN(A1)    ; feoln:=false
2D3A| 356E FFF6 0002                MOVE.W  -10(A6),LSTBLK(A2)      ; dlstblk:=lastavailblk
2D40| 357C 0200 0016                MOVE.W  #FBLKSIZ,LSTBYTE(A2)    ; dlastbyte:=fblksize
2D46| 026A 01FF 0018                AND.W   #$1FF,DACCESS(A2)
2D4C| 006A C800 0018                OR.W    #$C800,DACCESS(A2)      ; daccess.year:=100
2D52| 426E 000E                     MOVE.W  #0,14(A6)       ; canstretch:=false
2D00* 54
PAGE - 112  MONITOR   FILE: BLOCKIO.TEXT


2CF6* 5E
2D56| 3D7C 0001 FFFE        @8      MOVE.W  #1,-2(A6)       ; ok:=true
2D26* 34
2CE2* 78
2CA0* 00BC
2D5C| 4A6E FFFE            CANTST1 TST.W   -2(A6)
2D60| 66**                         BNE.S   CANTSTX
2D62| 137C 0001 0005                MOVE.B  #1,FEOF(A1)
2D68| 137C 0001 0004                MOVE.B  #1,FEOLN(A1)
2D6E|                      CANTSTX PUTNP   -60(A6)                 ; release(heaptop)
2D60* 0C
2D6E| 2B6E FFC4 FFF4       #       MOVE.L   -60(A6),-12(A5)
2D74| 4E5E                         UNLK    A6
2D76| 205F                         MOVE.L  (A7)+,A0        ; pop return address
2D78| 5C4F                         ADD.W   #6,A7           ; delete parameters
2D7A| 4ED0                         JMP     (A0)
2D7C|                      ;
2D7C|                      ;       BLKIO -- BLOCKREAD AND BLOCKWRITE
2D7C|                      ;
2D7C|                      ;       stack:
2D7C|                      ;
2D7C|                      ;               22      Function result
2D7C|                      ;               18      File address
2D7C|                      ;               14      Buffer address
2D7C|                      ;               12      Number of blocks
2D7C|                      ;               10      Rel block number
2D7C|                      ;               8       Doread
2D7C|                      ;               4       Return address
2D7C|                      ;               0       Old A6
2D7C|                      ;               -2      Chunk
2D7C|                      ;               -4      Chunksize
2D7C|                      ;               -6      Numblocks
2D7C|                      ;               -8      Blocknum
2D7C|                      ;               -12     Index
2D7C|                      ;               -14     unitnum
2D7C|                      ;
2D7C|                      ;       LOOPIO
2D7C|                      ;
2D7C| 3D6E 000C FFFA        LOOPIO  MOVE.W  12(A6),-6(A6)           ; copy nblocks to numblocks
2D82| 3D6E 000A FFF8                MOVE.W  10(A6),-8(A6)           ; copy rblock to blocknum
2D88| 2D6E 000E FFF4                MOVE.L  14(A6),-12(A6)          ; copy buffer address to index
2D8E| 3D69 000C FFF2                MOVE.W  FUNIT(A1),-14(A6)       ; copy unit number
2D94| 302E 000C                     MOVE.W  12(A6),D0
2D98| 0C40 003F                     CMPI.W  #63,D0                  ; if numblocks > 63 then
2D9C| 6D**                         BLT.S   @1
2D9E| 703F                         MOVE.L  #63,D0                  ;   chunk:=63
2D9C* 02
2DA0| 3D40 FFFE            @1      MOVE.W  D0,-2(A6)               ; else chunk:=numblocks
2DA4| C1FC 0200                     MULS    #FBLKSIZ,D0
2DA8| 3D40 FFFC                     MOVE.W  D0,-4(A6)               ; chunksize:=chunk*blocksize
2DAC| 4A6E FFFA            @2      TST.W   -6(A6)                  ; while numblocks<>0 do
2DB0| 67**                         BEQ.S   @5
2DB2| 3F2E FFF2                     MOVE.W  -14(A6),-(A7)           ; unit#
2DB6| 222E FFF4                     MOVE.L  -12(A6),D1              ; address
PAGE - 113  MONITOR   FILE: BLOCKIO.TEXT


2DBA| 342E FFFC                     MOVE.W  -4(A6),D2               ; length
2DBE| 362E FFF8                     MOVE.W  -8(A6),D3               ; block#
2DC2| 4A6E 0008                     TST.W   8(A6)
2DC6| 67**                         BEQ.S   @3                      ; if doread then
2DC8| 7C01                         MOVE.L  #INBIT,D6               ;   unitread
2DCA| 60**                         BRA.S   @4                      ; else
2DC6* 04
2DCC| 7C02                @3      MOVE.L  #OUTBIT,D6              ;   unitwrite
```

```
2DCA* 02
2DCE| 6100 EB8C              @4        BSR     GETUNIT                 ; get unit# and validate, setup drvr
2DD2| 4E90                             JSR     (A0)                    ; go to driver
2DD4|                                  GETSYSC A0
2DD4| 206D FFFC              #         MOVE.L  -4(A5),A0
2DD8| 4A10                             TST.B   (A0)                    ; test iorslt
2DDA| 66**                             BNE.S   @5                      ; exit loopio if nonzero
2DDC| 302E FFFE                        MOVE.W  -2(A6),D0
2DE0| 916E FFFA                        SUB.W   D0,-6(A6)               ; sub chunk from numblocks
2DE4| 302E FFFC                        MOVE.W  -4(A6),D0
2DE8| 48C0                             EXT.L   D0
2DEA| D1AE FFF4                        ADD.L   D0,-12(A6)              ; add chunksize to index
2DEE| 302E FFFE                        MOVE.W  -2(A6),D0
2DF2| D16E FFF8                        ADD.W   D0,-8(A6)               ; add chunk to blocknum
2DF6| 302E FFFA                        MOVE.W  -6(A6),D0
2DFA| B06E FFFE                        CMP.W   -2(A6),D0               ; if numblocks < chunk then
2DFE| 6EAC                             BGT.S   @2
2E00| 3D40 FFFE                        MOVE.W  D0,-2(A6)               ; chunk:=numblocks
2E04| C1FC 0200                        MULS    #FBLKSIZ,D0
2E08| 3D40 FFFC                        MOVE.W  D0,-4(A6)               ; chunksize:=chunk*blocksize
2E0C| 609E                             BRA     @2
2DDA* 32
2DB0* 5C
2E0E| 4E75                  @5        RTS
2E10|                       ;
2E10|                       ;         BLKIO
2E10|                       ;
2E10| 4E56 FFF2             BLKIO     LINK    A6,#-14
2E14| 6100 DA6C                       BSR     OVFCHK
2E18| 42AE 0016                       MOVE.L  #0,22(A6)               ; default func result gets zero
2E1C|                                  GETSYSC A0                      ; ptr to iosrlt
2E1C| 206D FFFC              #         MOVE.L  -4(A5),A0
2E20| 4210                             MOVE.B  #INOERR,(A0)            ; initialize iorslt
2E22| 226E 0012                       MOVE.L  18(A6),A1               ; ptr to FIB
2E26| 4A29 000A                       TST.B   FISOPEN(A1)             ; file open ?
2E2A| 66**                             BNE.S   @1
2E2C| 10BC 000D              @0        MOVE.B  #INOTOPN,(A0)           ; not open or # blocks < 0
2E30| 6000 ****                       BRA     BLKRXIT
2E2A* 08
2E34| 4A6E 000C              @1        TST.W   12(A6)                  ; # blocks >=0
2E38| 6BF2                             BMI.S   @0
2E3A| 4A29 000B                       TST.B   FISBLKD(A1)             ; blocked device ?
2E3E| 6700 ****                       BEQ     @7
2E42| 45E9 001E                       LEA     FHEADER(A1),A2          ; ptr to dir entry
2E46| 4A6E 000A                       TST.W   10(A6)                  ; if rblock < 0 then rblock:=fnxtblk
2E4A| 6A**                             BPL.S   @2
2E4C| 3D69 0018 000A                  MOVE.W  FNXTBLK(A1),10(A6)
2E4A* 06
PAGE - 114  MONITOR   FILE: BLOCKIO.TEXT


2E52| 3012                  @2        MOVE.W  FSTBLK(A2),D0
2E54| D16E 000A                       ADD.W   D0,10(A6)               ; rblock:=rblock+dfirstblk
2E58| 302E 000A                       MOVE.W  10(A6),D0
2E5C| D06E 000C                       ADD.W   12(A6),D0
2E60| B06A 0002                       CMP.W   LSTBLK(A2),D0           ; if rblock+nblocks > dlstblk then
2E64| 6F**                             BLE.S   @4
2E66| 4A6E 0008                       TST.W   8(A6)                   ;   if not doread then
2E6A| 66**                             BNE.S   @3
2E6C| 42A7                             MOVE.L  #0,-(A7)
2E6E| 2F09                             MOVE.L  A1,-(A7)
2E70| 6100 FDCE                       BSR     CANTSTR                 ;     junk:=cantstretch
2E74| 4A5F                             TST.W   (A7)+
2E76|                                  GETSYSC A0
2E76| 206D FFFC              #         MOVE.L  -4(A5),A0
2E7A| 226E 0012                       MOVE.L  18(A6),A1
2E7E| 45E9 001E                       LEA     FHEADER(A1),A2
2E6A* 16
2E82| 302A 0002              @3        MOVE.W  LSTBLK(A2),D0
2E86| 906E 000A                       SUB.W   10(A6),D0
2E8A| 3D40 000C                       MOVE.W  D0,12(A6)               ;   nblocks:=dlstblk-rblock
2E64* 28
2E8E| 302E 000A              @4        MOVE.W  10(A6),D0
2E92| B06A 0002                       CMP.W   LSTBLK(A2),D0
2E96| 6D**                             BLT.S   @5
2E98| 137C 0001 0005                  MOVE.B  #1,FEOF(A1)             ; feof:=rblock>=dlstblk
2E9E| 6000 ****                       BRA     BLKRXIT
2E96* 0A
2EA2| 4229 0005              @5        MOVE.B  #0,FEOF(A1)
2EA6| 48E7 00E0                       MOVEM.L A0-A2,-(A7)
2EAA| 6100 FED0                       BSR     LOOPIO
2EAE| 4CDF 0700                       MOVEM.L (A7)+,A0-A2
2EB2| 4A10                             TST.B   (A0)                    ; if iorslt <> 0 then exit
2EB4| 6600 ****                       BNE     BLKRXIT
2EB8| 4A6E 0008                       TST.W   8(A6)
2EBC| 66**                             BNE.S   @6                      ; if not doread then fmodified:=true
2EBE| 337C 0001 001C                  MOVE.W  #1,FMODIFD(A1)
2EBC* 06
2EC4| 3D6E 000C 0018        @6        MOVE.W  12(A6),24(A6)           ; func result gets nblocks
2ECA| 302E 000C                       MOVE.W  12(A6),D0
2ECE| D16E 000A                       ADD.W   D0,10(A6)               ; rblock:=rblock+nblocks
2ED2| 302E 000A                       MOVE.W  10(A6),D0               ; feof:=rblock=dlstblk
2ED6| B06A 0002                       CMP.W   LSTBLK(A2),D0
2EDA| 57C0                             SEQ     D0
2EDC| 0200 0001                       AND.B   #1,D0
2EE0| 1340 0005                       MOVE.B  D0,FEOF(A1)
```

```
2EE4| 302E 000A                    MOVE.W  10(A6),D0
2EE8| 9052                         SUB.W   FSTBLK(A2),D0
2EEA| 3340 0018                    MOVE.W  D0,FNXTBLK(A1)  ; fnxtblk:=rblock-dfrirstblk
2EEE| B069 0016                    CMP.W   FMAXBLK(A1),D0
2EF2| 6F00 ****                    BLE     BLKRXIT         ; if fnxtblk > fmaxblk then fmaxblk:=fnxtblk
2EF6| 3369 0018 0016               MOVE.W  FNXTBLK(A1),FMAXBLK(A1)
2EFC| 6000 ****                    BRA     BLKRXIT
2E40* 00C0
2F00| 3D6E 000C 0018        @7      MOVE.W  12(A6),24(A6)   ; func result gets nblocks
PAGE - 115  MONITOR   FILE: BLOCKIO.TEXT


2F06| 48E7 00E0                    MOVEM.L A0-A2,-(A7)
2F0A| 6100 FE70                    BSR     LOOPIO
2F0E| 4CDF 0700                    MOVEM.L (A7)+,A0-A2
2F12| 4A10                         TST.B   (A0)            ; if iorslt <> 0 then
2F14| 67**                         BEQ.S   @8
2F16| 426E 0018                    MOVE.W  #0,24(A6)       ;   func result:=0 and exit
2F1A| 60**                         BRA.S   BLKRXIT
2F14* 06
2F1C| 4A6E 0008            @8      TST.W   8(A6)           ; if doread then
2F20| 66**                         BNE.S   BLKRXIT
2F22| 4E71                         NOP
2F20* 02
2F1A* 08
2EFE* 0026
2EF4* 0030
2EB6* 006E
2EA0* 0084
2E32* 00F2
2F24| 4E5E                BLKRXIT UNLK    A6
2F26| 205F                        MOVE.L  (A7)+,A0        ; pop return address
2F28| DEFC 0010                    ADD.W   #16,A7          ; delete parameters
2F2C| 4ED0                         JMP     (A0)
2F2E|                       ;
2F2E|
2F2E|
2F2E|                       ;
2F2E|                               .INCLUDE TEXTIO.TEXT
2F2E|                       ;
2F2E|                       ;       Filename textio
2F2E|                       ;
2F2E|
2F2E|                       ;
2F2E|                       ;       PROCEDURE FWRTCHAR(FILE:FIB; CH:CHAR)
2F2E|                       ;
2F2E|                       ;       stack:
2F2E|                       ;
2F2E|                       ;               10      Return address
2F2E|                       ;               6       File address
2F2E|                       ;               4       Character
2F2E|                       ;               0       Old A6
2F2E|                       ;
2F2E|                       FWRTCHAR
2F2E| 4E56 0000                         LINK    A6,#0
2F32| 206E 0006                         MOVE.L  6(A6),A0                ; with F do
2F36| 4A28 000A                         TST.B   FISOPEN(A0)             ;   if fisopen then
2F3A| 67**                             BEQ.S   @2                      ;     begin
2F3C| 2250                             MOVE.L  FWINDOW(A0),A1
2F3E| 12AE 0005                         MOVE.B  5(A6),(A1)              ;       fwindowf[0]:=ch;
2F42| 4A68 0038                         TST.W   FSOFTBF(A0)             ;       if fsoftbuf then
2F46| 67**                             BEQ.S   @1
2F48| 2F08                             MOVE.L  A0,-(A7)                ;         fput(f)
2F4A| 6100 ****                        BSR     FPUT
2F4E| 60**                             BRA.S   FWRTCHX                 ;       else
2F46* 08
2F50| 3028 000C                 @1      MOVE.W  FUNIT(A0),D0
PAGE - 116  MONITOR   FILE: TEXTIO.TEXT


2F54| 2209                             MOVE.L  A1,D1
2F56| 7401                             MOVE.L  #1,D2
2F58| 4243                             MOVE.W  #0,D3
2F5A| 6100 F7D0                        BSR     WRITEII                 ;         unitwrite(funit,fwindowf,1)
2F5E| 60**                             BRA.S   FWRTCHX                 ;     end
2F60|                               @2      GETSYSC A0                      ;   else
2F3A* 24
2F60| 206D FFFC                      #       MOVE.L   -4(A5),A0
2F64| 30BC 000D                             MOVE.W  #INOTOPN,(A0)           ;     syscomf.iorslt:=inotopen
2F5E* 08
2F4E* 18
2F68| 4E5E                        FWRTCHX UNLK    A6
2F6A| 5C4F                             ADD.W   #6,A7
2F6C| 4E75                             RTS
2F6E|                       ;
2F6E|                       ;       PROCEDURE FWRITELN(FILE:FIB)
2F6E|                       ;
2F6E|                       ;       stack:
2F6E|                       ;
2F6E|                       ;               8       Return address
2F6E|                       ;               4       File address
2F6E|                       ;               0       Old A6
2F6E|                       ;
2F6E|                       FWRITELN
2F6E| 4E56 0000                         LINK    A6,#0
2F72| 206E 0004                         MOVE.L  4(A6),A0
```

```
2F76| 2250                          MOVE.L  FWINDOW(A0),A1
2F78| 12BC 000D                      MOVE.B  #$D,(A1)                ; f.fwindow↑[0]:=chr(eol)
2F7C| 2F08                          MOVE.L  A0,-(A7)
2F7E| 6100 ****                     BSR     FPUT                    ; fput(f)
2F82| 4E5E                          UNLK    A6
2F84| 584F                          ADD.W   #4,A7
2F86| 4E75                          RTS
2F88|                       ;
2F88|                       ;       FUNCTION FREADCHR(FILE:FIB):CHAR
2F88|                       ;
2F88|                       ;       stack:
2F88|                       ;
2F88|                       ;               12      Function result (long zero)
2F88|                       ;               8       Return address
2F88|                       ;               4       File address
2F88|                       ;               0       Old A6
2F88|                       ;
2F88|                       FREADCHR
2F88| 4E56 0000                      LINK    A6,#0
2F8C|                               GETSYSC A0                      ; syscom↑.iorslt:=inoerror
2F8C| 206D FFFC               #       MOVE.L  -4(A5),A0
2F90| 4250                          MOVE.W  #INOERR,(A0)
2F92| 206E 0004                      MOVE.L  4(A6),A0                ; with f do
2F96| 0C68 0001 0006                CMP.W   #1,FSTATE(A0)           ;   if fstate=fneedchar then fget(f)
2F9C| 66**                          BNE.S   @1
2F9E| 2F08                          MOVE.L  A0,-(A7)
2FA0| 6100 ****                     BSR     FGET
2F9C* 06
2FA4| 206E 0004              @1      MOVE.L  4(A6),A0
PAGE - 117  MONITOR   FILE: TEXTIO.TEXT


2FA8| 2250                          MOVE.L  FWINDOW(A0),A1          *  ;   freadchr:=fwindow↑[0]
2FAA| 1D51 000F                      MOVE.B  (A1),15(A6)
2FAE| 317C 0001 0006                MOVE.W  #1,FSTATE(A0)           ;   fstate:=fneedchar
2FB4| 4E5E                          UNLK    A6
2FB6| 4A9F                          TST.L   (A7)+
2FB8| 205F                          MOVE.L  (A7)+,A0
2FBA| 4A5F                          TST.W   (A7)+
2FBC| 4ED0                          JMP     (A0)
2FBE|                       ;
2FBE|                       ;       PROCEDURE FREADLN(FILE:FIB)
2FBE|                       ;
2FBE|                       ;       stack:
2FBE|                       ;
2FBE|                       ;               8       Return address
2FBE|                       ;               4       File address
2FBE|                       ;               0       Old A6
2FBE|                       ;
2FBE|                       FREADLN
2FBE| 4E56 0000                      LINK    A6,#0
2FC2| 206E 0004              @1      MOVE.L  4(A6),A0
2FC6| 4A28 0004                      TST.B   FEOLN(A0)               ; while not f.eoln do fget(f)
2FCA| 66**                          BNE.S   @2
2FCC| 2F08                          MOVE.L  A0,-(A7)
2FCE| 6100 ****                     BSR     FGET
2FD2| 60EE                          BRA.S   @1
2FCA* 08
2FD4| 206E 0004              @2      MOVE.L  4(A6),A0
2FD8| 317C 0001 0006                MOVE.W  #1,FSTATE(A0)           ; f.fstate:=fneedchar
2FDE| 4228 0004                      MOVE.B  #0,FEOLN(A0)            ; f.feoln:=false
2FE2| 4E5E                          UNLK    A6
2FE4| 584F                          ADD.W   #4,A7
2FE6| 4E75                          RTS
2FE8|                       ;
2FE8|                       ;       PROCEDURE FGET(FILE:FIB)
2FE8|                       ;
2FE8|                       ;       stack
2FE8|                       ;
2FE8|                       ;               8       File Address
2FE8|                       ;               4       Return Address
2FE8|                       ;               0       Old A6
2FE8|                       ;              -2       ch
2FE8|                       ;
2FE8|                       ;       registers
2FE8|                       ;
2FE8|                       ;               D1      noeof
2FE8|                       ;               D2      device
2FE8|                       ;               D3      echo
2FE8|                       ;               D4      amount
2FE8|                       ;               D5      leftinbuf
2FE8|                       ;               D6      wininx
2FE8|                       ;               D7      leftoget
2FE8|                       ;
2FE8| 4E56 FDB0              SCANAHD LINK    A6,#-FIBSIZE
2FEC| 206E 0008                      MOVE.L  8(A6),A0                ; with f do
2FF0| 224F                          MOVE.L  A7,A1
PAGE - 118  MONITOR   FILE: TEXTIO.TEXT


2FF2| 303C 0128                      MOVE.W  #FIBSIZE/2,D0
2FF6| 2448                          MOVE.L  A0,A2
2FF8| 2649                          MOVE.L  A1,A3
2FFA| 36DA                  @0      MOVE.W  (A2)+,(A3)+             ;   currentfib:=f
2FFC| 5340                          SUB.W   #1,D0
2FFE| 66FA                          BNE.S   @0
```

```
3000| 3028 0018                    MOVE.W  FNXTBLK(A0),D0
3004| 0800 0000                    BTST    #0,D0                   ;   if odd(fnxtblk) then
3008| 67**                         BEQ.S   @1
300A| 5268 0018                    ADD.W   #1,FNXTBLK(A0)          ;     fnxtblk:=fnxtblk+1
3008* 04
300E| 317C 0200 003C        @1      MOVE.W  #FBLKSIZ,FNXTBYT(A0)    ;   fnxtbyte:=fblksize
3014| 48E7 00C0                    MOVEM.L A0/A1,-(A7)
3018| 2F08                         MOVE.L  A0,-(A7)
301A| 6100 ****                    BSR     FGET
301E| 4CDF 0300                    MOVEM.L (A7)+,A0/A1
3022| 4A28 0005                    TST.B   FEOF(A0)                ;   if feof then
3026| 67**                         BEQ.S   @3
3028| 303C 0128                    MOVE.W  #FIBSIZE/2,D0
302C| 2448                         MOVE.L  A0,A2
302E| 2649                         MOVE.L  A1,A3
3030| 34DB                 @2      MOVE.W  (A3)+,(A2)+             ;     f:=currentfib
3032| 5340                         SUB.W   #1,D0
3034| 66FA                         BNE.S   @2
3036| 117C 0001 0005               MOVE.B  #1,FEOF(A0)             ;     feof:=true
303C| 117C 0001 0004               MOVE.B  #1,FEOLN(A0)            ;     feoln:=true
3042| 5368 003C                    SUB.W   #1,FNXTBYT(A0)          ;     fnxtbyt:=fnxtbyt-1
3026* 1E
3046| 4E5E                 @3      UNLK    A6
3048| 205F                         MOVE.L  (A7)+,A0
304A| 588F                         ADD.L   #4,A7
304C| 4ED0                         JMP     (A0)
304E|                        ;
301C* 0032
2FD0* 007E
2FA2* 00AC
304E| 4E56 FFFE              FGET    LINK    A6,#-2
3052|                               GETSYSC A0                      ; syscomf.iorslt:=inoerror
3052| 206D FFFC              #       MOVE.L   -4(A5),A0
3056| 4250                         MOVE.W  #INOERR,(A0)
3058| 206E 0008                    MOVE.L  8(A6),A0                ; with f do
305C| 4A28 000A                    TST.B   FISOPEN(A0)             ; if fisopen
3060| 6700 ****                    BEQ     FGETNOP
3064| 4A68 001A                    TST.W   FREPCNT(A0)             ;   if frepcnt>0 then
3068| 6F**                         BLE.S   @1
306A| 5368 001A                    SUB.W   #1,FREPCNT(A0)          ;     frepcnt:=frepcnt-1
306E| 6E00 ****                    BGT     FGET2L                  ;     if frepcnt>0 then goto 2
3068* 08
3072| 43E8 001E              @1      LEA     FHEADER(A0),A1          ;   with fheader do
3076| 4A68 0038                    TST.W   FSOFTBF(A0)             ;   if fsoftbuf then
307A| 6700 ****                    BEQ     FGET2ND
307E| 3E28 0008                    MOVE.W  FRECSZ(A0),D7           ;     leftoget:=frecsz
3082| 4246                         MOVE.W  #0,D6                   ;     wininx:=0
3084| 3028 0018              @2      MOVE.W  FNXTBLK(A0),D0          ;     repeat
3088| B068 0016                    CMP.W   FMAXBLK(A0),D0          ;       if fnxtblk>=fmaxblk then
PAGE - 119  MONITOR   FILE: TEXTIO.TEXT


308C| 6D**                         BLT.S   @3
308E| 3028 003C                    MOVE.W  FNXTBYT(A0),D0
3092| D047                         ADD.W   D7,D0                   ;         if fnxbyte+leftoget >
3094| B068 003A                    CMP.W   FMAXBYT(A0),D0          ;           fmaxbyte then goto 1
3098| 6E00 ****                    BGT     FGET1L                  ;         else
309C| 3A29 0016                    MOVE.W  LSTBYTE(A1),D5          ;           leftinbuf:=
30A0| 60**                         BRA.S   @4                      ;               dlastbyte-fnxtbyte
308C* 14
30A2| 3A3C 0200              @3      MOVE.W  #FBLKSIZ,D5             ;       else
30A0* 04
30A6| 9A68 003C              @4      SUB.W   FNXTBYT(A0),D5          ;         leftinbuf:=fblksize-fnxtbyte
30AA| 3807                         MOVE.W  D7,D4                   ;         amount:=leftoget
30AC| B845                         CMP.W   D5,D4                   ;         if amount>leftinbuf then
30AE| 6F**                         BLE.S   @5
30B0| 3805                         MOVE.W  D5,D4                   ;           amount:=leftinbuf
30AE* 02
30B2| 4A44                  @5      TST.W   D4                      ;         if amount > 0 then
30B4| 6F**                         BLE.S   @7
30B6| 45E8 0050                    LEA     FBUFFER(A0),A2          ;           src:=fbuffer[fnxtbyte]
30BA| D4E8 003C                    ADD.W   FNXTBYT(A0),A2
30BE| 2650                         MOVE.L  FWINDOW(A0),A3
30C0| D6C6                         ADD.W   D6,A3                   ;           dst:=fwindowf[wininx]
30C2| 3004                         MOVE.W  D4,D0
30C4| 16DA                  @6      MOVE.B  (A2)+,(A3)+             ;           moveleft(src,dst,amount)
30C6| 5340                         SUB.W   #1,D0
30C8| 66FA                         BNE.S   @6
30CA| D968 003C                    ADD.W   D4,FNXTBYT(A0)          ;           fnxtbyte:=fnxtbyte+amount
30CE| DC44                         ADD.W   D4,D6                   ;           wininx:=wininx+amount
30D0| 9E44                         SUB.W   D4,D7                   ;           leftoget:=leftoget-amount
30B4* 1C
30D2| 4A47                  @7      TST.W   D7                      ;         done:=leftoget = 0
30D4| 6700 ****                    BEQ     FGET3RD                 ;         if not done then
30D8| 4A68 003E                    TST.W   FBFCHNG(A0)             ;           if fbufchngd then
30DC| 67**                         BEQ.S   @8
30DE| 4268 003E                    MOVE.W  #0,FBFCHNG(A0)          ;             fbufchngd:=false
30E2| 317C 0001 001C               MOVE.W  #1,FMODIFD(A0)          ;             fmodified:=true
30E8| 48E7 0FC0                    MOVEM.L D4-D7/A0-A1,-(A7)
30EC| 6100 ****                    BSR     LDPARMS
30F0| 5343                         SUB.W   #1,D3
30F2| 6100 F638                    BSR     WRITEIT                 ;         unitwrite(funit,fbuffer,
30F6|                                                               ;           fblksize,dfirstblk+fnxtblk-1)
30F6| 4CDF 03F0                    MOVEM.L (A7)+,D4-D7/A0-A1
30FA| 6600 ****                    BNE     FGET1L                  ;         if iorslt<>0 then goto 1
30DC* 20
```

```
30FE| 48E7 0FC0              @8       MOVEM.L D4-D7/A0-A1,-(A7)
3102| 6100 ****                       BSR     LDPARMS
3106| 3C3C 0001                       MOVE.W  #INBIT,D6
310A| 3F00                           MOVE.W  D0,-(A7)
310C| 6100 E84E                       BSR     GETUNIT                 ;         unitread(funit,fbuffer,
3110| 4E90                           JSR     (A0)                    ;           fblksize,dfirstblk+fnxtblk)
3112| 4CDF 03F0                     MOVEM.L (A7)+,D4-D7/A0-A1
3116| 6600 ****                       BNE     FGET1L                  ;         if iorslt<>0 then goto 1
311A| 5268 0018                       ADD.W   #1,FNXTBLK(A0)          ;         fnxtblk:=fnxtblk+1
311E| 4268 003C                       MOVE.W  #0,FNXTBYT(A0)          ;         fnxtbyte:=0
3122| 6000 FF60                       BRA     @2                      ;       until done {loop exit near @7}
307C* 00AA
PAGE - 120  MONITOR   FILE: TEXTIO.TEXT


3126| 0C68 0001 000C        FGET2ND CMP.W   #1,FUNIT(A0)            ;     else
312C| 57C3                           SEQ     D3                      ;       echo:=funit = 1
312E| 66**                           BNE.S   @1                      ;       if echo then
3130| 343C 0002                       MOVE.W  #2,D2                   ;         device:=2
3134| 60**                           BRA.S   @2
312E* 06
3136| 3428 000C              @1       MOVE.W  FUNIT(A0),D2            ;         device:=funit
3134* 04
313A| 323C 0001              @2       MOVE.W  #1,D1                   ;       noeof:=true
313E| 3D7C 0020 FFFE                   MOVE.W  #$0020,-2(A6)           ;       ch:=' '
3144| 4240                           MOVE.W  #0,D0                   ;       i:=0
3146| B068 0008              @3       CMP.W   FRECSZ(A0),D0           ;       while (i<frecsize) and noeof do
314A| 6C00 ****                       BGE     FGET3RD
314E| 4A41                           TST.W   D1
3150| 6700 ****                       BEQ     FGET3RD
3154| 48E7 FFC0                     MOVEM.L D0-D7/A0-A1,-(A7)
3158| 3F02                           MOVE.W  D2,-(A7)
315A| 43EE FFFE                     LEA     -2(A6),A1
315E| 2209                           MOVE.L  A1,D1
3160| 7401                           MOVE.L  #1,D2
3162| 4243                           MOVE.W  #0,D3
3164| 3C3C 0001                       MOVE.W  #INBIT,D6
3168| 6100 E7F2                       BSR     GETUNIT                 ;         unitread(device,ch,1)
316C| 4E90                           JSR     (A0)
316E| 4CDF 03FF                     MOVEM.L (A7)+,D0-D7/A0-A1
3172| 6600 ****                       BNE     FGET1L                  ;         if iorslt<>0 then goto 1
3176| 4A03                           TST.B   D3
3178| 67**                           BEQ.S   @6                      ;         if echo then
317A| 0C2E 000D FFFE                   CMP.B   #13,-2(A6)              ;           if ch=chr(13) or
3180| 67**                           BEQ.S   @4
3182| 0C2E 0020 FFFE                   CMP.B   #$20,-2(A6)             ;              ch>=' ' then
3188| 6D**                           BLT.S   @6
3180* 08
318A| 48E7 FFC0              @4       MOVEM.L D0-D7/A0-A1,-(A7)
318E| 3028 000C                       MOVE.W  FUNIT(A0),D0
3192| 43EE FFFE                     LEA     -2(A6),A1
3196| 2209                           MOVE.L  A1,D1
3198| 7401                           MOVE.L  #1,D2
319A| 4243                           MOVE.W  #0,D3
319C| 6100 F58E                       BSR     WRITEIT                 ;             unitwrite(funit,ch,1)
31A0| 4CDF 03FF                     MOVEM.L (A7)+,D0-D7/A0-A1
3188* 1A
3178* 2A
31A4| C18B                  @6       EXG     D0,A3
31A6|                           GETSYSC A2                      ;         if ch=syscomf.eof then
31A6| 246D FFFC                  #        MOVE.L  -4(A5),A2
31AA| 102A 001F                       MOVE.B  EOF(A2),D0
31AE| B06E FFFE                       CMP.W   -2(A6),D0
31B2| C18B                           EXG     D0,A3
31B4| 66**                           BNE.S   @8
31B6| 0C68 0001 000C                   CMP.W   #1,FUNIT(A0)            ;           if funit=1 then
31BC| 66**                           BNE.S   @7
31BE| 426E FFFE                       MOVE.W  #0,-2(A6)               ;             ch:=chr(0)
31BC* 04
PAGE - 121  MONITOR   FILE: TEXTIO.TEXT


31C2| 323C 0001              @7       MOVE.W  #1,D1                   ;           noeof:=false
31B4* 10
31C6| 2470 0000              @8       MOVE.L  FWINDOW(A0,D0),A2       ;         fwindowf[i]:=ch
31CA| 14AE FFFE                       MOVE.B  -2(A6),(A2)
31CE| 5240                           ADD.W   #1,D0                   ;         i:=i+1
31D0| 6000 FF74                       BRA     @3
3152* 0082
314C* 0088
30D6* 00FE
31D4| 0C68 0001 0008        FGET3RD CMP.W   #1,FRECSZ(A0)           ;   if frecsize = 1 then
31DA| 6600 ****                       BNE     FGET2L
31DE| 4228 0004                       MOVE.B  #0,FEOLN(A0)            ;     feoln:=false
31E2| 317C 0002 0006                   MOVE.W  #2,FSTATE(A0)           ;     fstate:=fgotchar
31E8| 2450                           MOVE.L  FWINDOW(A0),A2
31EA| 0C12 000D                       CMP.B   #$D,(A2)                ;     if fwindowf[0]=chr(eol) then
31EE| 66**                           BNE.S   @1
31F0| 14BC 0020                       MOVE.B  #$20,(A2)               ;       fwindowf[0]:=' '
31F4| 117C 0001 0004                   MOVE.B  #1,FEOLN(A0)            ;     feoln:=true
31FA| 6000 ****                       BRA     FGET2L
31EE* 0E
31FE| 0C12 0010              @1       CMP.B   #$10,(A2)               ;     if fwindowf[0]=chr(dle) and
3202| 66**                           BNE.S   @3
3204| 0C68 0002 000C                   CMP.W   #2,FUNIT(A0)            ;       funit > 2 then
320A| 6F**                           BLE.S   @3
```

```
320C| 2F08                          MOVE.L  A0,-(A7)
320E| 6100 FE3E                     BSR     FGET                    ;       fget(f)
3212| 206E 0008                     MOVE.L  8(A6),A0
3216| 2450                          MOVE.L  FWINDOW(A0),A2          ;       amount:=ord(fwindow↑[0])-32
3218| 1812                          MOVE.B  (A2),D4
321A| 4884                          EXT.W   D4
321C| 0444 0020                     SUB.W   #$20,D4
3220| 6F**                          BLE.S   @2                      ;       if amount > 0 and
3222| 0C44 007F                     CMP.W   #127,D4
3226| 6E**                          BGT.S   @2                      ;         amount <= 127 then
3228| 14BC 0020                     MOVE.B  #$20,(A2)               ;         fwindow↑[0]:=' '
322C| 3144 001A                     MOVE.W  D4,FREPCNT(A0)          ;         frepcnt:=amount
3230| 6000 ****                     BRA     FGET2L
3226* 0C
3220* 12
3234| 2F08                  @2      MOVE.L  A0,-(A7)
3236| 6100 FE16                     BSR     FGET                    ;       fget(f)
320A* 2E
3202* 36
323A| 206E 0008                  @3      MOVE.L  8(A6),A0
323E| 2450                          MOVE.L  FWINDOW(A0),A2
3240| 4A12                          TST.B   (A2)                    ;     if fwindow↑[0]=chr(0) then
3242| 6600 ****                     BNE     FGET2L
3246| 4A68 0038                     TST.W   FSOFTBF(A0)             ;       if fsoftbuf and
324A| 67**                          BEQ.S   @4
324C| 43E8 001E                     LEA     FHEADER(A0),A1
3250| 0C69 0003 0004                CMP.W   #TEXTFILE,FKIND(A1)     ;         fheader.dfkind=textfile then
3256| 66**                          BNE.S   @4
3258| 2F08                          MOVE.L  A0,-(A7)
325A| 6100 FD8C                     BSR     SCANAHD                 ;         scanahead(f)
325E| 6000 ****                     BRA     FGET2L                  ;       else
3256* 0A
PAGE - 122  MONITOR   FILE:  TEXTIO.TEXT


324A* 16
3262| 14BC 0020                  @4      MOVE.B  #$20,(A2)               ;         fwindow↑[0]:=' '
3266| 6000 ****                     BRA     FGET1L
326A|                         FGETNOP GETSYSC A0                      ; else
3062* 0208
326A| 206D FFFC                  #       MOVE.L   -4(A5),A0
326E| 30BC 000D                     MOVE.W  #INOTOPN,(A0)           ;   syscom↑.iorslt:=inotopen
3268* 000A
3174* 00FE
3118* 015A
30FC* 0176
309A* 01D8
3272| 206E 0008                  FGET1L  MOVE.L  8(A6),A0
3276| 117C 0001 0005                MOVE.B  #1,FEOF(A0)             ;   feof:=true
327C| 117C 0001 0004                MOVE.B  #1,FEOLN(A0)            ;   feoln:=true
3260* 0022
3244* 003E
3232* 0050
31FC* 0086
31DC* 00A6
3070* 0212
3282| 4E5E                      FGET2L  UNLK    A6
3284| 205F                          MOVE.L  (A7)+,A0
3286| 4A9F                          TST.L   (A7)+
3288| 4ED0                          JMP     (A0)
328A|                         ;
328A|                         ;       PROCEDURE FPUT(FILE:FIB)
328A|                         ;
328A|                         ;       stack
328A|                         ;
328A|                         ;               8       File Address
328A|                         ;               4       Return Address
328A|                         ;               0       Old A6
328A|                         ;
328A|                         ;       registers
328A|                         ;
328A|                         ;               D4      amount
328A|                         ;               D5      leftinbuf
328A|                         ;               D6      wininx
328A|                         ;               D7      leftoput
328A|                         ;
328A| 4E56 0000                  FPUT    LINK    A6,#0
328E|                                  GETSYSC A0                      ; syscom↑.iorslt:=inoerror
328E| 206D FFFC                  #       MOVE.L   -4(A5),A0
3292| 4250                          MOVE.W  #INOERR,(A0)
3294| 206E 0008                     MOVE.L  8(A6),A0                ; with f do
3298| 4A28 000A                     TST.B   FISOPEN(A0)             ; if fisopen
329C| 6700 ****                     BEQ     FPUTNOP
32A0| 4A68 0038                     TST.W   FSOFTBF(A0)             ;   if fsoftbuf then
32A4| 6700 ****                     BEQ     @13
32A8| 43E8 001E                     LEA     FHEADER(A0),A1          ;   with fheader do
32AC| 3E28 0008                     MOVE.W  FRECSZ(A0),D7           ;     leftoput:=frecsz
32B0| 4246                          MOVE.W  #0,D6                   ;     wininx:=0
32B2| 3028 0018                  @1      MOVE.W  FNXTBLK(A0),D0          ;     repeat
PAGE - 123  MONITOR   FILE:  TEXTIO.TEXT


32B6| D051                          ADD.W   FSTBLK(A1),D0
32B8| B069 0002                     CMP.W   LSTBLK(A1),D0           ;       if dfirstblk+fnxtblk=dlastblk then
32BC| 66**                          BNE.S   @3
32BE| 3028 003C                     MOVE.W  FNXTBYT(A0),D0
```

```
32C2| D047                          ADD.W   D7,D0                   ;          if fnxbyte+leftoput >
32C4| B069 0016                     CMP.W   LSTBYTE(A1),D0          ;            dlastbyte then
32C8| 6F**                         BLE.S   @2
32CA| 48E7 7FC0                     MOVEM.L D1-D7/A0-A1,-(A7)
32CE| 42A7                         MOVE.L  #0,-(A7)
32D0| 2F08                         MOVE.L  A0,-(A7)
32D2| 6100 F96C                     BSR     CANTSTR                 ;            if cantstretch(f) then
32D6| 301F                         MOVE.W  (A7)+,D0
32D8| 4CDF 03FE                     MOVEM.L (A7)+,D1-D7/A0-A1
32DC| 67**                         BEQ.S   @3
32DE|                              GETSYSC A0                      ;              iorslt:=inoroom; goto 1
32DE| 206D FFFC              #      MOVE.L   -4(A5),A0
32E2| 30BC 0008                     MOVE.W  #INOROOM,(A0)
32E6| 6000 ****                     BRA     FPUT1L                  ;            else
32EA|                                                              ;              leftinbuf:=
32EA|                                                              ;                fblksize-fnxtbyte
32EA|                                                              ;          else
32C8* 20
32EA| 3A29 0016              @2      MOVE.W  LSTBYTE(A1),D5          ;            leftinbuf:=
32EE| 60**                         BRA.S   @4                      ;              dlastbyte-fnxtbyte
32DC* 12
32BC* 32
32F0| 3A3C 0200              @3      MOVE.W  #FBLKSIZ,D5             ;        else
32EE* 04
32F4| 9A68 003C              @4      SUB.W   FNXTBYT(A0),D5          ;          leftinbuf:=fblksize-fnxtbyte
32F8| 3807                         MOVE.W  D7,D4                   ;          amount:=leftoput
32FA| B845                         CMP.W   D5,D4                   ;          if amount>leftinbuf then
32FC| 6F**                         BLE.S   @5
32FE| 3805                         MOVE.W  D5,D4                   ;            amount:=leftinbuf
32FC* 02
3300| 4A44                   @5      TST.W   D4                      ;          if amount > 0 then
3302| 6F**                         BLE.S   @7
3304| 317C 0001 003E               MOVE.W  #1,FBFCHNG(A0)          ;            fbufchngd:=true
330A| 47E8 0050                     LEA     FBUFFER(A0),A3          ;            dst:=fbuffer[fnxtbyte]
330E| D6E8 003C                     ADD.W   FNXTBYT(A0),A3
3312| 2450                         MOVE.L  FWINDOW(A0),A2
3314| D4C6                         ADD.W   D6,A2                   ;            src:=fwindow^[wininx]
3316| 3004                         MOVE.W  D4,D0
3318| 16DA                   @6      MOVE.B  (A2)+,(A3)+             ;            moveleft(src,dst,amount)
331A| 5340                         SUB.W   #1,D0
331C| 66FA                         BNE.S   @6
331E| D968 003C                     ADD.W   D4,FNXTBYT(A0)          ;            fnxtbyte:=fnxtbyte+amount
3322| DC44                         ADD.W   D4,D6                   ;            wininx:=wininx+amount
3324| 9E44                         SUB.W   D4,D7                   ;            leftoput:=leftoput-amount
3302* 22
3326| 4A47                   @7      TST.W   D7                      ;          done:=leftoput = 0
3328| 6700 ****                     BEQ     @14                     ;          if not done then
332C| 4A68 003E                     TST.W   FBFCHNG(A0)             ;            if fbufchngd then
3330| 67**                         BEQ.S   @8
3332| 4268 003E                     MOVE.W  #0,FBFCHNG(A0)          ;              fbufchngd:=false
3336| 317C 0001 001C               MOVE.W  #1,FMODIFD(A0)          ;              fmodified:=true
PAGE - 124  MONITOR   FILE:  TEXTIO.TEXT


333C| 48E7 0FC0                     MOVEM.L D4-D7/A0-A1,-(A7)
3340| 6100 ****                     BSR     LDPARMS
3344| 5343                         SUB.W   #1,D3
3346| 6100 F3E4                     BSR     WRITEIT                 ;              unitwrite(funit,fbuffer,
334A|                                                              ;            fblksize,dfirstblk+fnxtblk-1)
334A| 4CDF 03F0                     MOVEM.L (A7)+,D4-D7/A0-A1
334E| 6600 ****                     BNE     FPUT1L                  ;          if iorslt<>0 then goto 1
3330* 20
3352| 3028 0018              @8      MOVE.W  FNXTBLK(A0),D0          ;          if fnxtblk<fmaxblk then
3356| B068 0016                     CMP.W   FMAXBLK(A0),D0
335A| 6C**                         BGE.S   @9
335C| 48E7 0FC0                     MOVEM.L D4-D7/A0-A1,-(A7)
3360| 6100 ****                     BSR     LDPARMS
3364| 3C3C 0001                     MOVE.W  #INBIT,D6
3368| 3F00                         MOVE.W  D0,-(A7)
336A| 6100 E5F0                     BSR     GETUNIT                 ;            unitread(funit,fbuffer,
336E| 4E90                         JSR     (A0)                    ;              fblksize,dfirstblk+fnxtblk)
3370| 4CDF 03F0                     MOVEM.L (A7)+,D4-D7/A0-A1
3374| 6600 ****                     BNE     FPUT1L                  ;            if iorslt<>0 then goto 1
3378| 60**                         BRA.S   @11                     ;          else
335A* 1E
337A| 45E8 0050              @9      LEA     FBUFFER(A0),A2          ;            fillchar(fbuffer,fblksize,0)
337E| 303C 0200                     MOVE.W  #FBLKSIZ,D0
3382| 421A                   @10     CLR.B   (A2)+
3384| 5340                         SUB.W   #1,D0
3386| 66FA                         BNE.S   @10
3378* 0E
3388| 5268 0018              @11     ADD.W   #1,FNXTBLK(A0)          ;        fnxtblk:=fnxtblk+1
338C| 4268 003C                     MOVE.W  #0,FNXTBYT(A0)          ;        fnxtbyte:=0
3390| 6000 FF20                     BRA     @1                      ;      until done {loop exit near @7}
332A* 006A
3394| 0C68 0001 0008         @14     CMP.W   #1,FRECSZ(A0)           ;      if frecsize=1 then
339A| 66**                         BNE.S   @12
339C| 2450                         MOVE.L  FWINDOW(A0),A2
339E| 0C12 000D                     CMP.B   #$D,(A2)                ;        if fwindow^[0]=chr(eol) then
33A2| 66**                         BNE.S   @12
33A4| 0C69 0003 0004               CMP.W   #TEXTFILE,FKIND(A1)     ;          if dfkind=textfile then
33AA| 3028 003C                     MOVE.W  FNXTBYT(A0),D0
33AE| 0C40 0181                     CMP.W   #FBLKSIZ-127,D0         ;            if fnxtbyte>=fblksize-127 and
33B2| 6D**                         BLT.S   @12
33B4| 3028 0018                     MOVE.W  FNXTBLK(A0),D0
33B8| 0800 0000                     BTST    #0,D0                   ;              not odd(fnxtblk) then
```

```
33BC| 66**                          BNE.S   @12
33BE| 317C 01FF 003C                MOVE.W  #FBLKSIZ-1,FNXTBYT(A0)  ;               fnxtbyte:=fblksize-1
33C4| 4212                          MOVE.B  #0,(A2)                 ;               fwindowt[0]:=chr(0)
33C6| 2F08                          MOVE.L  A0,-(A7)
33C8| 6100 FEC0                    BSR     FPUT                    ;       fput(f)
33BC* 0E
33B2* 18
33A2* 28
339A* 30
33CC| 6000 ****          @12       BRA     FPUT2L                  ;   else
32A6* 012A
33D0| 48E7 0FC0          @13       MOVEM.L D4-D7/A0-A1,-(A7)
33D4| 3028 000C                    MOVE.W  FUNIT(A0),D0
PAGE - 125  MONITOR   FILE: TEXTIO.TEXT


33D8| 2210                          MOVE.L  FWINDOW(A0),D1
33DA| 7401                          MOVE.L  #1,D2
33DC| 4243                          MOVE.W  #0,D3
33DE| 6100 F34C                    BSR     WRITEIT                 ;     unitwrite(funit,fwindowt,1)
33E2| 4CDF 03F0                    MOVEM.L (A7)+,D4-D7/A0-A1
33E6| 6600 ****                    BNE     FPUT1L                  ;     if iorslt<>0 then goto 1
33EA| 6000 ****                    BRA     FPUT2L
33EE|                     FPUTNOP PUTSYSC A0                      ; else
329E* 0150
33EE| 2B48 FFFC               #    MOVE.L   A0,-4(A5)
33F2| 30BC 000D                    MOVE.W  #INOTOPN,(A0)           ;   syscomt.iorslt:=inotopen
33E8* 000E
3376* 0080
3350* 00A6
32E8* 010E
33F6| 206E 0008          FPUT1L    MOVE.L  8(A6),A0
33FA| 117C 0001 0005               MOVE.B  #1,FEOF(A0)             ;   feof:=true
3400| 117C 0001 0004               MOVE.B  #1,FEOLN(A0)            ;   feoln:=true
33EC* 001A
33CE* 0038
3406| 6000 FE7A          FPUT2L    BRA     FGET2L
340A|                     ;
340A|                     ;       LDPARMS -- Load funit, fbuffer, fblocksize and dfirstblk+fnxtblk
340A|                     ;
3362* 00A8
3342* 00C8
340A| 3028 000C          LDPARMS   MOVE.W  FUNIT(A0),D0
340E| 4868 0050                    PEA     FBUFFER(A0)
3412| 221F                          MOVE.L  (A7)+,D1
3414| 243C 0000 0200               MOVE.L  #FBLKSIZ,D2
341A| 3611                          MOVE.W  FSTBLK(A1),D3
341C| D668 0018                    ADD.W   FNXTBLK(A0),D3
3420| 4E75                          RTS
3422|                     ;
3422|                     ;       PROCEDURE FSEEK(VAR F: FIB; RECNUM: INTEGER)
3422|                     ;
3422|                     ;       stack
3422|                     ;
3422|                     ;               10      File Address
3422|                     ;               8       Record Number
3422|                     ;               4       Return Address
3422|                     ;               0       Old A6
3422|                     ;
3422| 4E56 0000          FSEEK     LINK    A6,#0
3426|                               GETSYSC A0                      ; syscomt.iorslt:=inoerror
3426| 206D FFFC               #    MOVE.L   -4(A5),A0
342A| 4250                          MOVE.W  #INOERR,(A0)
342C| 206E 000A                    MOVE.L  10(A6),A0               ; with F do
3430| 4A28 000A                    TST.B   FISOPEN(A0)             ; if not fisopen then
3434| 66**                          BNE.S   @1
3436|                               GETSYSC A0
3436| 206D FFFC               #    MOVE.L   -4(A5),A0
343A| 30BC 000D                    MOVE.W  #INOTOPN,(A0)           ;    syscomt.iorslt:=inotopen
343E| 6000 ****                    BRA     FSEEK1                  ;  else
3434* 0C
PAGE - 126  MONITOR   FILE: TEXTIO.TEXT


3442| 43E8 001E          @1        LEA     FHEADER(A0),A1          ;    with fheader do
3446| 302E 0008                    MOVE.W  8(A6),D0                ;      if (recnum < 0) or
344A| 6B00 ****                    BMI     FSEEK1
344E| 4A68 0038                    TST.W   FSOFTBF(A0)             ;         (not fsoftbuf) or
3452| 6700 ****                    BEQ     FSEEK1
3456| 0C29 0003 0004               CMP.B   #TEXTFILE,FKIND(A1)     ;         ((fkind=textfile) and
345C| 66**                          BNE.S   @2
345E| 0C68 0001 0008               CMP.W   #1,FRECSZ(A0)           ;          (frecsize=1)) then
3464| 6700 ****                    BEQ     FSEEK1                  ;         goto 1
345C* 0A
3468| C1E8 0008          @2        MULS    FRECSZ(A0),D0           ;  block:=recnum*frecsize div fblksiz+1
346C| 81FC 0200                    DIVS    #FBLKSIZ,D0             ;  byte:=recnum*frecsize mod fblksiz
3470| 3200                          MOVE.W  D0,D1
3472| 5241                          ADD.W   #1,D1                   ;  D1 is now block
3474| 4840                          SWAP    D0                      ;  D0 is now byte
3476| 4A40                          TST.W   D0
3478| 66**                          BNE.S   @3                      ;  if byte = 0 then
347A| 303C 0200                    MOVE.W  #FBLKSIZ,D0             ;    byte:=fblksize
347E| 5341                          SUB.W   #1,D1                   ;    block:=block-1
3478* 06
3480| 3429 0002          @3        MOVE.W  LSTBLK(A1),D2           ;  n:=dlastblk-dfirstblk
3484| 9451                          SUB.W   FSTBLK(A1),D2
```

```
3486| B242                          CMP.W   D2,D1                   ;  if (block > n) or
3488| 6E**                          BGT.S   @4
348A| 66**                          BNE.S   @5                      ;     ((block = n) or
348C| 3629 0016                     MOVE.W  LSTBYTE(A1),D3
3490| B043                          CMP.W   D3,D0                   ;     (byte >= dlastbyte)) then
3492| 6D**                          BLT.S   @5
3488* 0A
3494| 3202                @4        MOVE.W  D2,D1                   ;       block := n
3496| 3003                          MOVE.W  D3,D0                   ;       byte := dlastbyte
3492* 04
348A* 0C
3498| B268 0018           @5        CMP.W   FNXTBLK(A0),D1          ;  if block <> fnxtblk then
349C| 67**                          BEQ.S   @9
349E| 4A68 003E                     TST.W   FBFCHNG(A0)             ;    if fbufchngd then
34A2| 67**                          BEQ.S   @8
34A4| 4268 003E                     MOVE.W  #0,FBFCHNG(A0)          ;      fbufchngd:=false
34A8| 317C 0001 001C                MOVE.W  #1,FMODIFD(A0)          ;      fmodified:=true
34AE| 48E7 FFC0                     MOVEM.L D0-D7/A0-A1,-(A7)
34B2| 6100 FF56                     BSR     LDPARMS
34B6| 5343                          SUB.W   #1,D3
34B8| 6100 F272                     BSR     WRITEIT                 ;      unitwrite(funit,fbuffer,
34BC|                                                               ;        fblksize,dfirstblk+fnxtblk-1)
34BC| 4CDF 03FF                     MOVEM.L (A7)+,D0-D7/A0-A1
34C0| 6600 ****                     BNE     FSEEK1                  ;      if iorslt<>0 then goto 1
34C4|
34A2* 20
34C4| B268 0016           @8        CMP.W   FMAXBLK(A0),D1          ;    if (block <= fmaxblk) and
34C8| 6E**                          BGT.S   @9
34CA| 0C40 0200                     CMP.W   #FBLKSIZ,D0             ;       (byte <> fblksize) then
34CE| 67**                          BEQ.S   @9
34D0| 48E7 FFC0                     MOVEM.L D0-D7/A0-A1,-(A7)
34D4| 6100 FF34                     BSR     LDPARMS
34D8| 3C3C 0001                     MOVE.W  #INBIT,D6
PAGE - 127  MONITOR   FILE: TEXTIO.TEXT


34DC| 3F00                          MOVE.W  D0,-(A7)
34DE| 6100 E47C                     BSR     GETUNIT                 ;      unitread(funit,fbuffer,
34E2| 4E90                          JSR     (A0)                    ;        fblksize,dfirstblk+fnxtblk)
34E4| 4CDF 03FF                     MOVEM.L (A7)+,D0-D7/A0-A1
34E8| 6600 ****                     BNE     FSEEK1                  ;    if iorslt<>0 then goto 1
34CE* 1C
34C8* 22
349C* 4E
34EC| 3628 0018           @9        MOVE.W  FNXTBLK(A0),D3          ;  if fnxtblk>fmaxblk then
34F0| B668 0016                     CMP.W   FMAXBLK(A0),D3
34F4| 6F**                          BLE.S   @10
34F6| 3143 0016                     MOVE.W  D3,FMAXBLK(A0)          ;    fmaxblk := fnxtblk
34FA| 3168 003C 003A                MOVE.W  FNXTBYT(A0),FMAXBYT(A0) ;    fmaxbyte := fnxtbyte
3500| 60**                          BRA.S   @11                     ;  else
34F4* 0C
3502| 66**                @10       BNE.S   @11                     ;    if (fnxtblk = fmaxblk) and
3504| 3628 003C                     MOVE.W  FNXTBYT(A0),D3
3508| B668 003A                     CMP.W   FMAXBYT(A0),D3          ;       (fnxtbyte > fmaxbyte) then
350C| 6F**                          BLE.S   @11
350E| 3143 003A                     MOVE.W  D3,FMAXBYT(A0)          ;  fmaxbyte := fnxtbyte
350C* 04
3502* 0E
3500* 10
3512| 4228 0005           @11       MOVE.B  #0,FEOF(A0)             ;  feof := false
3516| 4228 0004                     MOVE.B  #0,FEOLN(A0)            ;  feoln := false
351A| 4268 001A                     MOVE.W  #0,FREPCNT(A0)          ;  frepcnt := 0
351E| 4A68 0006                     TST.W   FSTATE(A0)              ;  if fstate <> fjandw then
3522| 67**                          BEQ.S   @12
3524| 317C 0001 0006                MOVE.W  #1,FSTATE(A0)           ;    fstate:=fneedchar
3522* 06
352A| 3141 0018           @12       MOVE.W  D1,FNXTBLK(A0)          ;  fnxtblk := block
352E| 3140 003C                     MOVE.W  D0,FNXTBYT(A0)          ;  fnxtbyte := byte
34EA* 0048
34C2* 0070
3466* 00CC
3454* 00DE
344C* 00E6
3440* 00F2
3532| 4E5E                FSEEK1    UNLK    A6
3534| 205F                          MOVE.L  (A7)+,A0
3536| 5C4F                          ADD.W   #6,A7
3538| 4ED0                          JMP     (A0)
353A|
353A|
353A|                     ;
353A|                               .INCLUDE MONLOAD.TEXT
353A|                     ;
353A|                     ;         Filename monload
353A|                     ;
353A|
353A|                     ;
353A|                     ; $$LOADIT - Memory resident segment loader
353A|                     ;
353A|                     ; Upon entry:     ST.L - procedure-to-be-called's return address
353A|                     ;                 ST.L - extra JSR's return address
PAGE - 128  MONITOR   FILE: MONLOAD.TEXT


353A|                     ;                 ST.L - pointer to ZFIRST
353A|                     ;                 D7.W - parameter to main program
```

```
353A|                          ;
353A|
353A|                          ZZLOADIT:
353A|
353A|                          ;
353A|                          ; Store the address of REMOVE1 into $FIRST-20 so that global GOTO
353A|                          ;       can unload segments as needed.
353A|                          ;
353A|
353A| 43FA ****                        LEA     REMOVE1,A1      ; Address of segment unloader
353E| 245F                             MOVE.L  (A7)+,A2        ; get @ZFIRST and save in A2
3540| 2F0D                             MOVE.L  A5,-(A7)
3542| 6100 ****                        BSR     SETUPA5
3546|                                  ADDRSAT A0
3546| 41ED FF7C                #       LEA      -132(A5),A0
354A| 48E0 0F18                        MOVEM.L A3-A4/D4-D7,-(A0)
354E|                                  RUNUTST                 ; running user for the debugger ?
354E| 4A2D FFA8                #       TST.B    -88(A5)
3552| 67**                             BEQ.S   @1
3554|                                  PUTUFST A2
3554| 2B4A FF90                #       MOVE.L   A2,-112(A5)
3558| 60**                             BRA.S   @2
355A|                           @1     PUTZFST A2
3552* 06
355A| 2B4A FFF0                #       MOVE.L   A2,-16(A5)
3558* 04
355E| 2A5F                      @2     MOVE.L  (A7)+,A5
3560| 204A                             MOVE.L  A2,A0
3562| 90FC 0014                        SUB.W   #20,A0          ; compute ZFIRST-20 in A0
3566| 2089                             MOVE.L  A1,(A0)         ; --> $FIRST-20
3568|
3568|                          ;
3568|                          ; Find which segment needs to be setup and possibly loaded.
3568|                          ;
3568|
3568| 201F                             MOVE.L  (SP)+,D0        ; (D0) = Fake call location + 6
356A| 224A                             MOVE.L  A2,A1
356C| 5949                             SUB.W   #4,A1           ; compute ZFIRST-4 in A1
356E| 2251                             MOVE.L  (A1),A1
3570| 3219                             MOVE.W  (A1)+,D1        ; D1  = Number of segments
3572| B091                     TRYNEXT:CMP.L    (A1),D0        ; Was the last segment the one?
3574| 6F**                             BLE.S   LASTSEG         ; Yes. Go get it.
3576| D2FC 0020                        ADDA.W  #32,A1          ; No. Bump A1 to next segment table.
357A| 5341                             SUBQ.W  #1,D1           ; Was that the last segment?
357C| 6AF4                             BPL.S   TRYNEXT         ; No. Try the next one.
357E| 4280                             MOVE.L  #0,D0           ; Yes. Issue error #0.
3580| 6000 ****                        BRA     ERROR           ;
3574* 0E
3584| 92FC 0020                LASTSEG:SUBA.W  #32,A1          ; Back-up A1 to first of segment table.
3588|
3588|                          ;
3588|                          ; Found the correct segment table.  Patch the return address.
3588|                          ;
PAGE - 129  MONITOR   FILE: MONLOAD.TEXT


3588|                          ;       (A1) = Segment Table
3588|                          ;       (D0) = Extra call location + 6
3588|                          ;
3588|
3588| 235F 0010                        MOVE.L  (SP)+,16(A1)    ; Save the old return address.
358C| 487A ****                        PEA     ZZUNLOAD        ; Replace it with $$UNLOAD.
3590| 4AA9 0014                        TST.L   20(A1)          ; Is this segment already in memory?
3594| 6600 ****                        BNE     PATCHIT         ; Yes. Don't need to load it.
3598|                                                          ; No. Need to load it.
3598|
3598|                          ;
3598|                          ; The required segment is not in memory.  It must be read from
3598|                          ; disk.  But first find where it will go.
3598|                          ;
3598|                          ;       (A1) = Segment Table
3598|                          ;       (D0) = Extra call location + 6
3598|                          ;
3598|
3598| 48E7 8140                        MOVEM.L D0/D7/A1,-(SP)  ; Save the necessary registers.
359C| 2078 010C                        MOVE.L  $10C,A0         ; Fetch upper memory limit.
35A0| 2808                             MOVE.L  A0,D4           ; Save a copy.
35A2| 91E9 0008                        SUBA.L  8(A1),A0        ; Subtract size of segments code.
35A6| 5988                             SUBQ.L  #4,A0           ; Leave room for pointer.
35A8| B1F8 0108                        CMPA.L  $108,A0         ; Will it fit?
35AC| 6E**                             BGT.S   IT.FITS         ; Yes.
35AE| 7001                             MOVE.L  #1,D0           ; No. Error time.
35B0| 6000 ****                        BRA     ERROR           ;
35AC* 06
35B4| 21C8 010C             IT.FITS:   MOVE.L  A0,$10C         ; Update current top of memory.
35B8| 20C4                             MOVE.L  D4,(A0)+        ; Push pointer to old top of memory.
35BA| 2348 000C                        MOVE.L  A0,12(A1)       ; Store segment address in segment table.
35BE|
35BE|                          ;
35BE|                          ; Now read the stuff into memory from disk.
35BE|                          ;
35BE|                          ;       (A0) = Where it will go.
35BE|                          ;       (A1) = This segment table.
35BE|                          ;
35BE|                          ; Register useage:
35BE|                          ;
```

```
35BE|                          ;    D0 - D2 = Scratch.
35BE|                          ;    D3      = Local.  Must be preserved by service routines.
35BE|                          ;    D4      = Address of segment table.
35BE|                          ;    D5      = Next byte to read. (INBYTE)
35BE|                          ;    D6      = Number of valid bytes in buffer. (MAXBYTES)
35BE|                          ;    D7      = First block in buffer. (INBLOCK)
35BE|                          ;    A0 - A1 = Scratch.
35BE|                          ;    A2      = Ptr to ZFIRST
35BE|                          ;    A3      = Local.  Must be preserved by service routines.
35BE|                          ;    A4      = Pointer to buffer.
35BE|                          ;
35BE|
35BE| 2809                             MOVE.L  A1,D4           ; Save address of segment table in D4.
35C0| 284A                             MOVE.L  A2,A4
35C2| 98FC 0010                        SUB.W   #16,A4          ; compute ZFIRST-16 in A4
PAGE - 130  MONITOR   FILE: MONLOAD.TEXT


35C6| 2854                             MOVE.L  (A4),A4         ; (A4) = buffer
35C8| 2A29 0004                        MOVE.L  4(A1),D5        ; Compute
35CC| 2E05                             MOVE.L  D5,D7           ;
35CE| 0285 0000 01FF                   ANDI.L  #$1FF,D5        ;   starting byte, and
35D4| E08F                             LSR.L   #8,D7           ;
35D6| E28F                             LSR.L   #1,D7           ;   starting block.
35D8| 6100 ****                        BSR     FILLBUF         ; Go fetch the first buffer full.
35DC|
35DC|                          ;
35DC|                          ; This is the main fetch loop.  Blocks are fetched one at a time
35DC|                          ; and selected blocks are processed.  Only code blocks (85),
35DC|                          ; relocation blocks (86), and end blocks (81) are examined.  The
35DC|                          ; rest are skipped.
35DC|                          ;
35DC|
35DC|                          MAINLOOP:
35DC|
35DC| 6100 ****                        BSR     NEXT.1          ; D0 := next byte.
35E0| 3600                             MOVE.W  D0,D3           ; Save it in D3.
35E2| 6100 ****                        BSR     NEXT.3          ; D0 := next 3 bytes.
35E6| 0443 0080                        SUBI.W  #$80,D3         ; Is block kind < 80?
35EA| 6D**                             BLT.S   BADBLCK         ; Yes. Issue error.
35EC| 0C43 0006                        CMP.W   #6,D3           ; No. Is it > 86?
35F0| 6E**                             BGT.S   BADBLCK         ; Yes. Issue error.
35F2| E34B                             LSL.W   #1,D3           ; No. Do a CASE
35F4| 41FA ****                        LEA     C.TABLE,A0
35F8| 3630 3000                        MOVE.W  0(A0,D3.W),D3
35FC| 4EF0 3000                        JMP     0(A0,D3.W)
3600|
35F0* 0E
35EA* 14
3600| 7002                   BADBLCK: MOVE.L  #2,D0           ; Bad Block.
3602| 6000 ****                        BRA     ERROR           ; Issue error #2.
3606|
35F6* 0010
3606| ****                   C.TABLE: .WORD   BLOCK80-C.TABLE
3608| ****                            .WORD   BLOCK81-C.TABLE
360A| ****                            .WORD   BLOCK82-C.TABLE
360C| ****                            .WORD   BLOCK83-C.TABLE
360E| ****                            .WORD   BLOCK84-C.TABLE
3610| ****                            .WORD   BLOCK85-C.TABLE
3612| ****                            .WORD   BLOCK86-C.TABLE
3614|
3606* 000E
3614|                          BLOCK80:                        ; Module Name Block
360A* 000E
3614|                          BLOCK82:                        ; Entry Point Block
360C* 000E
3614|                          BLOCK83:                        ; External Reference Block
360E* 000E
3614|                          BLOCK84:                        ; Starting Address Block
3614|
3614| 5980                             SUBQ.L  #4,D0           ; Skip blocksize - 4 bytes.
3616| DA80                             ADD.L   D0,D5
3618| BC85                             CMP.L   D5,D6           ; Is the next byte in the buffer?
PAGE - 131  MONITOR   FILE: MONLOAD.TEXT


361A| 6EC0                             BGT.S   MAINLOOP        ; Yes. Go process next block.
361C| 2C05                             MOVE.L  D5,D6           ; No. Compute
361E| 0285 0000 01FF                   ANDI.L  #$1FF,D5        ;   which byte, and
3624| E08E                             LSR.L   #8,D6           ;
3626| E28E                             LSR.L   #1,D6           ;
3628| DE86                             ADD.L   D6,D7           ;   which block its in.
362A| 6100 ****                        BSR     FILLBUF         ; Go fill the buffer.
362E| 60AC                             BRA.S   MAINLOOP        ; Go do next block.
3630|
3630|                          ;
3630|                          ; 85 - Code block.
3630|                          ;
3630|
3610* 002A
3630|                          BLOCK85:
3630| 2600                             MOVE.L  D0,D3           ; Save block size in D3.
3632| 5183                             SUBQ.L  #8,D3           ; D3 = bytes of code (BYTESLEFT).
3634| 2644                             MOVE.L  D4,A3           ; (A3) = Segment Table.
3636| 266B 000C                        MOVE.L  12(A3),A3       ; A3 = base address of segment.
363A| 6100 ****                        BSR     NEXT.4          ; Fetch relative address for code.
```

```
363E| D7C0                          ADDA.L  D0,A3           ; Relocate it to actual address.
3640| 4A83                WHILE85:  TST.L   D3              ; While BYTESLEFT > 0 DO
3642| 6F**                          BLE.S   ENDWH85
3644| 2406                          MOVE.L  D6,D2               ; BYTESINBUFFER :=
3646| 9485                          SUB.L   D5,D2               ;    MAXBYTES - INBYTE;
3648| B682                          CMP.L   D2,D3               ; IF BYTESINBUFFER < BYTESLEFT
364A| 6F**                          BLE.S   ALL.85                ; THEN
364C| 9682                          SUB.L   D2,D3                 ; BYTESLEFT -:= BYTESINBUFFER
364E| 41F4 5800                     LEA     0(A4,D5.L),A0
3652| E28A                          LSR.L   #1,D2               ; Move BYTESINBUFFER/2 words
3654| 36D8                LO.85:    MOVE.W  (A0)+,(A3)+         ;   from buffer
3656| 5382                          SUBQ.L  #1,D2               ;   to final location.
3658| 6EFA                          BGT.S   LO.85
365A| E08E                          LSR.L   #8,D6               ; See how many blocks
365C| E28E                          LSR.L   #1,D6               ;   were in buffer
365E| DE86                          ADD.L   D6,D7               ;   and add to INBLOCK.
3660| 4285                          CLR.L   D5                  ; INBYTE := 0;
3662| 6100 ****                     BSR     FILLBUF             ; Fill the buffer.
3666| 60D8                          BRA.S   WHILE85
364A* 1C
3668| 41F4 5800          ALL.85:    LEA     0(A4,D5.L),A0       ; ELSE
366C| DA83                          ADD.L   D3,D5               ; Update INBYTE.
366E| E28B                          LSR.L   #1,D3               ; Move BYTESLEFT/2 words
3670| 36D8                L1.85:    MOVE.W  (A0)+,(A3)+         ;   from buffer
3672| 5383                          SUBQ.L  #1,D3               ;   to final loction.
3674| 6EFA                          BGT.S   L1.85
3642* 32
3676| 6000 FF64          ENDWH85:   BRA     MAINLOOP        ; Go do next block.
367A|
367A|                              ;
367A|                              ; 86 - Relocation Block.
367A|                              ;
367A|                              ; Must come after the code that it relocates.
367A|                              ;
367A|
3612* 0074
PAGE - 132  MONITOR   FILE: MONLOAD.TEXT


367A|                    BLOCK86:
367A| 2600                          MOVE.L  D0,D3           ; NRELOCS :=
367C| 5983                          SUBQ.L  #4,D3           ;   (BLOCKSIZE - 4)
367E| E48B                          LSR.L   #2,D3           ;   DIV 4;
3680| 2644                          MOVE.L  D4,A3           ; Get base address
3682| 266B 000C                     MOVE.L  12(A3),A3       ;   of this segment.
3686| 6100 ****          TOP.86:    BSR     NEXT.4          ; Get next address to relocate.
368A| 220B                          MOVE.L  A3,D1           ; Put base into a D register.
368C| D3B3 0800                     ADD.L   D1,0(A3,D0.L)   ; Relocate proper place.
3690| 5383                          SUBQ.L  #1,D3           ; More to relocate?
3692| 6EF2                          BGT.S   TOP.86          ; Yes.
3694| 6000 FF46                     BRA     MAINLOOP        ; No. Go do next block.
3698|
3698|                              ;
3698|                              ; 81 - End Block.
3698|                              ;
3698|                              ; Fnished.
3698|                              ;
3698|
3608* 0092
3698|                    BLOCK81:
3698|
3698| 4CDF 0281                     MOVEM.L (SP)+,D0/D7/A1  ; Restore registers.
369C|
369C|                              ;
369C|                              ; This segment is in memory.  Change the JSR's in the descriptors
369C|                              ; into JMP's.
369C|                              ;
369C|                              ;       (A1) = Segment Table
369C|                              ;       (D0) = Extra call location + 6
369C|                              ;
369C|
3596* 0106
369C| 52A9 0014          PATCHIT:   ADDQ.L  #1,20(A1)       ; Bump segment reference counter.
36A0| 2051                          MOVE.L  (A1),A0         ; (A0) = first descriptor.
36A2| 2669 0020                     MOVE.L  32(A1),A3       ; (A3) = 1st descriptor in next segment.
36A6| 2229 000C                     MOVE.L  12(A1),D1       ;  D1  = first byte of segment in memory.
36AA| 2418                P.LOOP:    MOVE.L  (A0)+,D2        ;  D2  = segment relative address of procedure.
36AC| 30FC 4EF9                     MOVE.W  #$4EF9,(A0)+    ; Patch in JMP xxx.L opcode.
36B0| D481                          ADD.L   D1,D2           ; Compute loaded address
36B2| 20C2                          MOVE.L  D2,(A0)+        ; and drop it into JMP instrcution.
36B4| B7C8                          CMPA.L  A0,A3           ; Any more descriptors to patch?
36B6| 6EF2                          BGT.S   P.LOOP          ; Yes. Go do 'em.
36B8| 204A                          MOVE.L  A2,A0           ; No. Link this segment ...
36BA| 5148                          SUB.W   #8,A0           ; compute ZFIRST-8 in A0
36BC| 2350 0018                     MOVE.L  (A0),24(A1)     ; ... into active ...
36C0| 2089                          MOVE.L  A1,(A0)         ; ... segment list.
36C2|                              ;
36C2|                              ; Now crawl down the table of soft break points
36C2|                              ;
36C2|                              ; Each entry contains:  segmment number         2 bytes
36C2|                              ;                       offset in segment       2 bytes
36C2|                              ;                       instruction save cell   2 bytes
36C2|                              ;
PAGE - 133  MONITOR   FILE: MONLOAD.TEXT
```

```
36C2|                          ;       (A1) = Segment Table
36C2|                          ;       (D0) = Extra call location + 6
36C2|                          ;
36C2|
36C2| 2449                             MOVE.L  A1,A2           ; Compute segment number = (A1-A4)/32+1
36C4| 95CC                             SUB.L   A4,A2
36C6| 220A                             MOVE.L  A2,D1
36C8| EA81                             ASR.L   #5,D1
36CA| 5281                             ADD.L   #1,D1
36CC| 2669 000C                       MOVE.L  12(A1),A3       ; Get actual address of segment in memory
36D0| 2478 012C                       MOVE.L  $12C,A2         ; Get start of break point table
36D4|
36D4| 341A                     PATCHBP MOVE.W  (A2)+,D2        ; End of table ?
36D6| 6B**                             BMI.S   GO2USER
36D8| B242                             CMP.W   D2,D1           ; Same segment ?
36DA| 67**                             BEQ.S   SAMESEG
36DC| 4A9A                             TST.L   (A2)+           ; No, then skip seg offset and save cell
36DE| 60F4                             BRA.S   PATCHBP
36DA* 04
36E0| 341A                     SAMESEG MOVE.W  (A2)+,D2        ; Yes, get offset in segment
36E2| 34F3 2000                       MOVE.W  0(A3,D2),(A2)+  ; Save the instruction
36E6| 37BC 4E4C 2000                   MOVE.W  #$4E4C,0(A3,D2) ; Store TRAP #$C into code
36EC| 60E6                             BRA.S   PATCHBP         ; Go do next entry
36EE|                          ;
36EE|                          ; Jump to user code
36EE|                          ;
36EE|                          ;       (D0) = Extra call location + 6
36EE|                          ;
36EE|
36D6* 16
36EE| 2240                     GO2USER MOVE.L  D0,A1           ; Get extra return address.
36F0| 2F0D                             MOVE.L  A5,-(A7)        ; Save users A5
36F2| 6100 ****                       BSR     SETUPA5
36F6|                                  ADDRSAB A0
36F6| 41ED FF64                #       LEA     -156(A5),A0
36FA| 4CD8 18F0                       MOVEM.L (A0)+,A3-A4/D4-D7
36FE|                                  DBUGTST                 ; Check debug flag
36FE| 4A2D FFA9                #       TST.B    -87(A5)
3702| 67**                             BEQ.S   @1
3704|                                  DBUGCLR                 ; Clear debug flag (so we don't stop again)
3704| 422D FFA9                #       MOVE.B   #0,-87(A5)
3708| 2A5F                             MOVE.L  (A7)+,A5        ; Restore users A5
370A| 2F29 FFFC                       MOVE.L  -4(A1),-(A7)    ; Push return address
370E| 6000 ****                       BRA     FAKERED         ; Go fake red button
3702* 0E
3712| 2A5F                      @1     MOVE.L  (A7)+,A5        ; Restore users A5
3714| 4EE9 FFFA                       JMP     -6(A1)          ; Go to user !
3718|                          ;
3718|                          ; FillBuf - Fill the buffer.  This routine reads as much as
3718|                          ;               possible into the code buffer from an already
3718|                          ;               open UCSD file.
3718|                          ;
3718|                          ; Parameters: D7.L = Next block to read
3718|                          ;             A4.L = Pointer to buffer
3718|                          ;
PAGE - 134  MONITOR   FILE: MONLOAD.TEXT


3718|                          ; Returns:    D6.L = Number of bytes read
3718|                          ;
3718|                          ; All other registers are preserved.
3718|                          ;
3718|
3664* 00B4
362C* 00EC
35DA* 013E
3718|                          FILLBUF:
3718| 48E7 FDFE                       MOVEM.L D0-D5/D7/A0-A6,-(SP)
371C| 6100 ****                       BSR     SETUPA5
3720| 42A7                            CLR.L   -(SP)                   ; Room for function result
3722|                                 RUNUTST                         ; Running user ?
3722| 4A2D FFA8                #      TST.B    -88(A5)
3726| 67**                            BEQ.S   @1
3728|                                 GETUFIB A0                      ; Yes, Get userfib ptr
3728| 206D FFA4                #      MOVE.L   -92(A5),A0
372C| 60**                            BRA.S   @2
372E|                          @1     GETCFIB A0                      ; No, Get codefib ptr
3726* 06
372E| 206D FFDC                #      MOVE.L   -36(A5),A0
372C* 04
3732| 2F08                      @2     MOVE.L  A0,-(A7)                ; File address
3734| 2F0C                            MOVE.L  A4,-(SP)                ; Buffer address
3736| 3F3C 0004                       MOVE.W  #INBUFSZ/FBLKSIZ,-(SP)  ; # Blocks to read
373A| 3F07                            MOVE.W  D7,-(SP)                ; Block to read
373C| 3F3C 0001                       MOVE.W  #1,-(SP)                ; Read:=true
3740| 6100 F6CE                       BSR     BLKIO
3744| 4286                            CLR.L   D6              ; Fetch
3746| 3C1F                            MOVE.W  (A7)+,D6        ;   #blocks read.
3748| 4CDF 7FBF                       MOVEM.L (SP)+,D0-D5/D7/A0-A6
374C| E18E                            LSL.L   #8,D6           ; D6 = number of
374E| E38E                            LSL.L   #1,D6           ;   bytes read.
3750| 6E**                            BGT.S   READ.OK         ; Was read ok?
3752| 7003                            MOVE.L  #3,D0           ; No. Error #3.
3754| 6000 ****                       BRA     ERROR
3750* 06
3758| 4E75                     READ.OK:RTS
```

```
375A|
375A|                          ;
375A|                          ; NextBuf - Fetch the next sequential buffer full.
375A|                          ;
375A|                          ; Sets D6 = Number of bytes in buffer, and
375A|                          ;      D7 = Current first block in buffer.
375A|                          ;      D5 = Next byte to read to zero.
375A|                          ;
375A|                          ; All other registers are preserved.
375A|                          ;
375A|
375A|                          NEXTBUF:
375A|
375A| 4285                             CLR.L   D5              ; INBYTE := 0
375C| E08E                             LSR.L   #8,D6           ; See how many blocks
375E| E28E                             LSR.L   #1,D6           ;   are in this buffer.
3760| DE86                             ADD.L   D6,D7           ; Add it to INBLOCK.
PAGE - 135  MONITOR   FILE: MONLOAD.TEXT


3762| 61B4                             BSR     FILLBUF         ; Go fill the buffer.
3764| 4E75                             RTS
3766|
3766|                          ;
3766|                          ; Next_1 - Read the next byte of code.
3766|                          ;
3766|                          ; Returns:    D0.L = the byte read
3766|                          ;
3766|                          ; Preserves all registers.
3766|                          ;
3766|
35DE* 0188
3766|                          NEXT.1:
3766| BC85                             CMP.L   D5,D6           ; Next byte in buffer?
3768| 6E**                             BGT.S   OK.1            ; Yes.
376A| 61EE                             BSR     NEXTBUF         ; No. Read some more from disk.
3768* 02
376C| 4280                     OK.1:   CLR.L   D0              ; Fetch the next byte
376E| 1034 5800                       MOVE.B  0(A4,D5.L),D0   ;   from the buffer.
3772| 5285                             ADDQ.L  #1,D5           ; Update INBYTE.
3774| 4E75                             RTS
3776|
3776|                          ;
3776|                          ; Next_3 - Read the next byte of code.
3776|                          ;
3776|                          ; Returns:    D0.L = the value read
3776|                          ;
3776|
35E4* 0192
3776|                          NEXT.3:
3776| 61EE                             BSR     NEXT.1
3778| 2200                             MOVE.L  D0,D1
377A| E189                             LSL.L   #8,D1
377C| 61E8                             BSR     NEXT.1
377E| 8280                             OR.L    D0,D1
3780| E189                             LSL.L   #8,D1
3782| 61E2                             BSR     NEXT.1
3784| 8081                             OR.L    D1,D0
3786| 4E75                             RTS
3788|
3788|                          ;
3788|                          ; Next_4 - Read the next byte of code.
3788|                          ;
3788|                          ; Returns:    D0.L = the value read
3788|                          ;
3788|
3688* 0100
363C* 014C
3788|                          NEXT.4:
3788| 61EC                             BSR     NEXT.3
378A| 2200                             MOVE.L  D0,D1
378C| E189                             LSL.L   #8,D1
378E| 61D6                             BSR     NEXT.1
3790| 8081                             OR.L    D1,D0
3792| 4E75                             RTS
PAGE - 136  MONITOR   FILE: MONLOAD.TEXT


3794|                          ;
3794|                          ; $$UNLOAD - Remove a segment from memory.  This routine is returned to
3794|                          ;               when the first (and only remaining) call to a procedure
3794|                          ;               in a particular segment returns.  If that segment has
3794|                          ;               not been loaded by a specific Load Segment call (i.e.
3794|                          ;               if the reference count = 1) then this segment is removed
3794|                          ;               from memory.  In any event, this segments descriptors
3794|                          ;               are turned back into JSR $$LOADIT.L instructions and
3794|                          ;               this segment in unlinked from the active segment list.
3794|                          ;
3794|
358E* 0206
3794|                          ZZUNLOAD:
3794|
3794| 6100 ****                        BSR     REMOVE1
3798| 4ED0                             JMP     (A0)
379A|
379A|                          ;
```

```
379A|                          ; REMOVE1 - Remove the top segment and place its return address in A0.
379A|                          ;           This routine is called by the global GOTO routine as
379A|                          ;           well as the normal unloader.
379A|                          ;
379A|
3796* 0004
353C* 025E
379A|                          REMOVE1:
379A| 200B                             MOVE.L  A3,D0           ; save A3
379C| 2F0D                             MOVE.L  A5,-(A7)
379E| 6100 ****                       BSR     SETUPA5
37A2|                                  RUNUTST                 ; running user for the debugger ?
37A2| 4A2D FFA8                  #      TST.B    -88(A5)
37A6| 67**                             BEQ.S   @1
37A8|                                  GETUFST A2
37A8| 246D FF90                  #      MOVE.L   -112(A5),A2
37AC| 60**                             BRA.S   @2
37AE|                           @1     GETZFST A2
37A6* 06
37AE| 246D FFF0                  #      MOVE.L   -16(A5),A2
37AC* 04
37B2| 2A5F                      @2     MOVE.L  (A7)+,A5
37B4| 264A                             MOVE.L  A2,A3           ; copy ZFIRST into A3 for later patch
37B6| 204A                             MOVE.L  A2,A0
37B8| 5148                             SUB.W   #8,A0           ; compute ZFIRST-8 in A0
37BA| 2F08                             MOVE.L  A0,-(A7)
37BC| 2050                             MOVE.L  (A0),A0         ; (A0) = descriptor of exited segment.
37BE| 53A8 0014                       SUBQ.L  #1,20(A0)       ; Bump counter.  Still active?
37C2| 6E**                           BGT.S   STILLIN         ; Yes.  Don't free memory.
37C4| 2268 000C                       MOVE.L  12(A0),A1       ; No. Fetch old top of memory ...
37C8| 21E1 010C                       MOVE.L  -(A1),$10C      ; ... store it in current top.
37C2* 08
37CC| 245F                  STILLIN:MOVE.L  (A7)+,A2         ; Update active ...
37CE| 24A8 0018                       MOVE.L  24(A0),(A2)     ; ... segment list.
37D2| 2250                           MOVE.L  (A0),A1         ; (A1) = first descriptor
37D4| 2468 0020                       MOVE.L  32(A0),A2       ; (A2) = 1st dscriptor in next segment.
37D8| 5889                  UN.LOOP:ADDQ.L  #4,A1            ; Skip relative offset field.
PAGE - 137  MONITOR   FILE: MONLOAD.TEXT


37DA| 32FC 4EB9                       MOVE.W  #$4EB9,(A1)+    ; Store JSR xxx.L opcode.
37DE| 22CB                           MOVE.L  A3,(A1)+        ; Fill in absolute address field.
37E0| B3CA                           CMPA.L  A2,A1           ; More descriptors?
37E2| 6DF4                           BLT.S   UN.LOOP         ; Yes. Go do 'em.
37E4| 2068 0010                       MOVE.L  16(A0),A0       ; No. Load seg's return address.
37E8| 2640                           MOVE.L  D0,A3           ; Restore A3
37EA| 4E75                           RTS
37EC|                          ;
37EC| 000E                     LELPL   .WORD   14
37EE| 4C 6F 61 64 65 72 20     LEPL    .ASCII  'Loader Error  '
37F5| 45 72 72 6F 72 20 20
37FC|                          ;
37FC|                          ; ERROR - Fatal loader errors come here.
37FC|                          ;
37FC|                          ; error# is in D0
37FC|                          ;       #0 = Unknown segment.
37FC|                          ;       #1 = No room in memory.
37FC|                          ;       #2 = Bad block.
37FC|                          ;       #3 = Can't read code file.
37FC|                          ;       #4 = Jump Table Too Large.
37FC|                          ;
37FC|                          ;       stack
37FC|                          ;               0       old A6
37FC|                          ;               -2      errornumber
37FC|                          ;
37FC|
3756* 00A6
3604* 01F8
35B2* 024A
3582* 027A
37FC| 4E56 FFFE                  ERROR   LINK    A6,#-2                  ; allocate: errornumber
3800| 2F00                             MOVE.L  D0,-(A7)
3802| 43FA FFEA                       LEA     LEPL,A1                 ; address
3806| 343A FFE4                       MOVE.W  LELPL,D2                ; and length
380A| 6100 ****                       BSR     PRINT
380E| 201F                           MOVE.L  (A7)+,D0
3810| 43EE FFFE                       LEA     -2(A6),A1               ; address
3814| 7402                           MOVE.L  #2,D2                   ; and length
3816| 32BC 0030                       MOVE.W  #$30,(A1)               ; initialize error number to '0'
381A| 6000 ****                       BRA     PRNTERR
381E|
381E|
381E|                          ;
381E|                                  .INCLUDE SYMDBUG.TEXT
381E|                          ;
381E|                          ;       FILENAME: SYMDBUG
381E|                          ;
381E|
381E|                          ;
381E|                          ;       All the trap handlers for launch
381E|                          ;
381E| 5C8F                     V31A    ADD.L   #6,A7           ; discard SR and PC
3820| 60**                             BRA.S   V31X
3822| 48E7 8080                V31     MOVEM.L D0/A0,-(A7)
3826| 41FA FFF6                       LEA     V31A,A0         ; setup temporary NMI handler
PAGE - 138  MONITOR   FILE: SYMDBUG.TEXT
```

```
382A| 21C8 007C                    MOVE.L  A0,$7C          ; to wait for user to lift his finger
3820* 0C
382E| 46EF 0008          V31X     MOVE    8(A7),SR        ; allow another interrupt
3832| 203C 0001 0000              MOVE.L  #$10000,D0      ; delay about 1/4 a second
3838| 5380               @1       SUBQ.L  #1,D0
383A| 66FC                        BNE.S   @1
383C| 41FA FFE4                    LEA     V31,A0          ; fall thru to here if no NMI else goto V31A
3840| 21C8 007C                    MOVE.L  A0,$7C
3844| 41FA CFE0                    LEA     HANDLER,A0
3848| B1EF 000A                    CMP.L   10(A7),A0       ; if handler didn't get to move A6 into $140
384C| 66**                        BNE.S   @2
384E| 21CE 0140                    MOVE.L  A6,$140         ; then do it now
384C* 04
3852| 4CDF 0101          @2       MOVEM.L (A7)+,D0/A0
3856| 3F3C 001F                    MOVE.W  #31,-(A7)
385A| 60**                        BRA.S   SOFTB0
385C|
385C| 3F3C 0002          V2       MOVE.W  #2,-(A7)
3860| 60**                        BRA.S   SOFTB0
3862| 3F3C 0003          V3       MOVE.W  #3,-(A7)
3866| 60**                        BRA.S   SOFTB0
3868| 3F3C 0004          V4       MOVE.W  #4,-(A7)
386C| 60**                        BRA.S   SOFTB0
386E| 3F3C 0005          V5       MOVE.W  #5,-(A7)
3872| 60**                        BRA.S   SOFTB0
3874| 3F3C 0006          V6       MOVE.W  #6,-(A7)
3878| 60**                        BRA.S   SOFTB0
387A| 3F3C 0007          V7       MOVE.W  #7,-(A7)
387E| 60**                        BRA.S   SOFTB0
3880| 3F3C 0008          V8       MOVE.W  #8,-(A7)
3884| 60**                        BRA.S   SOFTB0
3886| 3F3C 0009          V9       MOVE.W  #9,-(A7)
388A| 60**                        BRA.S   SOFTB0
388C| 3F3C 000A          V10      MOVE.W  #10,-(A7)
3890| 60**                        BRA.S   SOFTB0
3892| 3F3C 000B          V11      MOVE.W  #11,-(A7)
3896| 60**                        BRA.S   SOFTB0
3898|
3898| 3F3C 0018          V24      MOVE.W  #24,-(A7)
389C| 60**                        BRA.S   SOFTB0
389E|
389E|                           ;
389E|                           ; Soft break point trap handler
389E|                           ;
389E|
389E| 3F3C 002C          SOFTBPT MOVE.W  #44,-(A7)       ; soft bpt exception is $B0
389C* 04
3896* 0A
3890* 10
388A* 16
3884* 1C
387E* 22
3878* 28
3872* 2E
386C* 34
3866* 3A
PAGE - 139  MONITOR   FILE: SYMDBUG.TEXT


3860* 40
385A* 46
38A2| 48E7 FFFE          SOFTB0  MOVEM.L D0-D7/A0-A6,-(A7)
38A6| 4286                        CLR.L   D6              ; D6:=nil
38A8|                           ;
38A8|                           ;       setup D4 with exception index and D5 with @user reg save area
38A8|                           ;
38A8| 382F 003C                    MOVE.W  60(A7),D4
38AC| 6100 ****                    BSR     SETUPA5
38B0|                             GETUREG D5              ; @user reg save area
38B0| 2A2D FF98              #     MOVE.L   -104(A5),D5
38B4| 0C44 0003                    CMP.W   #3,D4
38B8| 6F**                        BLE.S   @1
38BA| 206F 0040                    MOVE.L  64(A7),A0       ; get stacked pc into A0
38BE| 60**                        BRA.S   @2
38B8* 06
38C0| 41EF 003E           @1       LEA     62(A7),A0
38C4|                             GETACCA D6
38C4| 2C2D FF8C              #     MOVE.L   -116(A5),D6
38C8| 2246                        MOVE.L  D6,A1
38CA| 22D8                        MOVE.L  (A0)+,(A1)+
38CC| 22D8                        MOVE.L  (A0)+,(A1)+
38CE| 4A58                        TST.W   (A0)+           ; same as MOVE.L  72(A7),A0
38BE* 10
38D0| 0C44 002C           @2       CMP.W   #44,D4
38D4| 66**                        BNE.S   GOODPC
38D6| 5588                        SUB.L   #2,A0           ; adjust for TRAP instruction
38D8| 2F48 0040                    MOVE.L  A0,64(A7)       ; and update stacked pc
38DC|                     GOODPC  RUNUTST                 ; running user for the debugger ?
38D4* 06
38DC| 4A2D FFA8              #     TST.B    -88(A5)
38E0| 67**                        BEQ.S   @1
38E2|                             GETUFST A1
38E2| 226D FF90              #     MOVE.L   -112(A5),A1
38E6| 60**                        BRA.S   @2
```

```
38E8|                          @1       GETZFST A1
38E0* 06
38E8| 226D FFF0                #        MOVE.L   -16(A5),A1
38E6* 04
38EC| 5149                     @2       SUB.W    #8,A1           ; compute ZFIRST-8 in A0
38EE| 6100 ****                         BSR      A0TOS0
38F2| 0C44 002C                         CMP.W    #44,D4
38F6| 66**                              BNE.S    SOFTXIT
38F8| 4A41                             TST.W    D1              ; bad seg# ?
38FA| 6B**                              BMI.S    SOFTXIT
38FC| 2478 012C                         MOVE.L   $12C,A2         ; Get start of break point table
3900| 341A                     FINDBP   MOVE.W   (A2)+,D2        ; End of table ?
3902| 6B**                              BMI.S    SOFTXIT
3904| B242                             CMP.W    D2,D1           ; Same segment ?
3906| 67**                              BEQ.S    FNDSEGN
3908| 4A9A                             TST.L    (A2)+           ; No, then skip seg offset and save cell
390A| 60F4                             BRA.S    FINDBP
3906* 04
390C| B05A                     FNDSEGN  CMP.W    (A2)+,D0        ; Same segment offset ?
PAGE - 140  MONITOR   FILE: SYMDBUG.TEXT


390E| 67**                              BEQ.S    FNDSEGO
3910| 4A5A                             TST.W    (A2)+           ; No, then skip save cell
3912| 60EC                             BRA.S    FINDBP
390E* 04
3914| 3092                     FNDSEGO  MOVE.W   (A2),(A0)       ; put saved word back into code for user
3916| 41FA ****                         LEA      PUTBACK,A0
391A| 21C8 057C                         MOVE.L   A0,$57C         ; ptr to put back code
391E|                          SOFTXIT  RUNUTST                  ; Were we running the user for the sym debugger ?
3902* 1A
38FA* 22
38F6* 26
391E| 4A2D FFA8                #        TST.B    -88(A5)
3922| 67**                              BEQ.S    @1
3924|                                   GETDREG  A0              ; Yes, push parameters for sym debuggers ERROR
3924| 206D FF94                #        MOVE.L   -108(A5),A0
3928| 2268 003C                         MOVE.L   60(A0),A1       ; onto symbolic debuggers stack
392C| 3304                             MOVE.W   D4,-(A1)        ; (1) exception number
392E| 3301                             MOVE.W   D1,-(A1)        ; (2) segment number
3930| 3300                             MOVE.W   D0,-(A1)        ; (3) segment offset
3932| 2305                             MOVE.L   D5,-(A1)        ; (4) @user reg save area
3934| 2306                             MOVE.L   D6,-(A1)        ; (5) @access address record
3936| 2149 003C                         MOVE.L   A1,60(A0)       ; now update stack pointer
393A| 4A86                             TST.L    D6
393C| 4CDF 7FFF                       MOVEM.L  (A7)+,D0-D7/A0-A6
3940| 67**                              BEQ.S    RNNGUSR
3942| 504F                             ADD.W    #8,A7
3944| 60**                              BRA.S    RNNGUSR         ; didn't delete except number (for later)
3946|
3922* 22
3946| 4A86                     @1       TST.L    D6
3948| 4CDF 7FFF                         MOVEM.L  (A7)+,D0-D7/A0-A6
394C| 67**                              BEQ.S    @2
394E| 504F                             ADD.W    #8,A7
394C* 02
3950| 544F                     @2       ADD.W    #2,A7
3952| 2F38 0F7C                         MOVE.L   $F7C,-(A7)       ; No, then go to macsbug
3956| 4E75                             RTS
3958|                          ;
3958|                          ;        Exception handling for the symbolic debugger
3958|                          ;
3944* 12
3940* 16
3958| 42B8 057C                RNNGUSR  MOVE.L   #0,$57C
395C|                          ;
395C|                          ;        save the users registers
395C|                          ;
395C| 2F08                             MOVE.L   A0,-(A7)                ; save A0 on tos
395E| 2F0D                             MOVE.L   A5,-(A7)
3960| 6100 ****                         BSR      SETUPA5
3964|                                   GETUREG  A0
3964| 206D FF98                #        MOVE.L   -104(A5),A0
3968| D0FC 0040                         ADD.W    #64,A0                  ; point A0 just above reg save area
396C| 48E0 FFFF                       MOVEM.L  D0-D7/A0-A7,-(A0)       ; save Ds and As
3970| 215F 0034                         MOVE.L   (A7)+,52(A0)            ; now save A5
3974| 215F 0020                         MOVE.L   (A7)+,32(A0)            ; now save A0
PAGE - 141  MONITOR   FILE: SYMDBUG.TEXT


3978| 381F                             MOVE.W   (A7)+,D4                ; get except number out of the way
397A| 315F 0044                       MOVE.W   (A7)+,68(A0)            ; save SR
397E| 215F 0040                       MOVE.L   (A7)+,64(A0)            ; save PC
3982| 3F04                             MOVE.W   D4,-(A7)                ; push except number on tos for later
3984| 2F38 0140                       MOVE.L   $140,-(A7)              ; also push users old A6 on tos too
3988| 4268 0046                       MOVE.W   #0,70(A0)
398C| 06A8 0000 000A 003C            ADD.L    #10,60(A0)              ; adjust saved A7
3994|                          ;
3994|                          ;        switch to debugger
3994|                          ;
3994|                                   RUNUCLR                          ; runuser:=false
3994| 422D FFA8                #        MOVE.B   #0,-88(A5)
3998| 6100 ****                         BSR      RSTRXCP
399C| 6100 ****                         BSR      SWAPJT
39A0|                          ;
39A0|                          ;        load debuggers registers and call symbolic debugger
```

```
39A0|                          ;
39A0|                                  GETERRP A1                      ; load A1 with @ERROR
39A0| 226D FFA0                #       MOVE.L   -96(A5),A1
39A4|                                  GETDREG A0
39A4| 206D FF94                #       MOVE.L   -108(A5),A0
39A8| D0FC 0030                        ADD.W    #48,A0
39AC| 4CD8 F000                        MOVEM.L (A0)+,A4-A7
39B0| 4E91                            JSR      (A1)                   ; call it
39B2|                          ;
39B2|                          ;       save the debuggers registers
39B2|                          ;
39B2| 2F0D                            MOVE.L   A5,-(A7)
39B4| 6100 ****                      BSR      SETUPA5
39B8|                                  GETDREG A0
39B8| 206D FF94                #       MOVE.L   -108(A5),A0
39BC| D0FC 0040                        ADD.W    #64,A0                 ; point A0 just above reg save area
39C0| 48E0 000F                        MOVEM.L A4-A7,-(A0)
39C4|                                  GETDREG A0
39C4| 206D FF94                #       MOVE.L   -108(A5),A0
39C8| 215F 0034                        MOVE.L   (A7)+,52(A0)           ; now save A5
39CC|                          ;
39CC|                          ;       switch back to user
39CC|                          ;
39CC| 6100 ****                      BSR      SWAPJT
39D0| 6100 ****                      BSR      POKEXCP
39D4|                                  RUNUSET                         ; runuser:=true
39D4| 1B7C 0001 FFA8          #       MOVE.B   #1,-88(A5)
39DA|                          ;
39DA|                          ;       load user registers
39DA|                          ;
39DA|                                  GETUREG A0
39DA| 206D FF98                #       MOVE.L   -104(A5),A0
39DE| 4CD8 00FF                        MOVEM.L (A0)+,D0-D7
39E2| 4A98                            TST.L    (A0)+
39E4| 4CD8 FE00                        MOVEM.L (A0)+,A1-A7
39E8|                          ;
39E8|                          ;       now is later (ie time to look at the except number)
39E8|                          ;
39E8| 21DF 0140                        MOVE.L   (A7)+,$140             ; restore users old A6 (we ran debugger)
PAGE - 142  MONITOR   FILE: SYMDBUG.TEXT


39EC| 0C5F 002C                        CMP.W    #44,(A7)+
39F0| 66**                            BNE.S    NOT44                  ; do we need to putback the TRAP ?
39F2|                          ;
39F2|                          ;       yes, load user registers
39F2|                          ;
39F2| 2F18                            MOVE.L   (A0)+,-(A7)
39F4| 3F18                            MOVE.W   (A0)+,-(A7)
39F6| 91FC 0000 0026                  SUB.L    #70-32,A0
39FC| 2050                            MOVE.L   (A0),A0
39FE|                          ;
39FE|                          ;       call putback (like macsbug would)
39FE|                          ;
39FE| 2F08                            MOVE.L   A0,-(A7)
3A00|
3A00| 4E71                    BSRPUTB NOP
3A02| 4E71                            NOP
3A04|
3A04|                          ;       BSR      PUTBACK
3A04|
3A04| 205F                            MOVE.L   (A7)+,A0
3A06| 4E73                            RTE
3A08|                          ;
3A08|                          ;       no, just load user registers
3A08|                          ;
39F0* 16
3A08| 2F18                    NOT44   MOVE.L   (A0)+,-(A7)
3A0A| 3F18                            MOVE.W   (A0)+,-(A7)
3A0C| 91FC 0000 0026                  SUB.L    #70-32,A0
3A12| 2050                            MOVE.L   (A0),A0
3A14| 4E73                            RTE
3A16|                          ;
3A16|                          ;       Enter here from Macsbug
3A16|                          ;
3A16|                          ;       stack:
3A16|                          ;
3A16|                          ;               long:   PC
3A16|                          ;               word:   SR
3A16|                          ;               long:   A0 saved by Macsbug
3A16|                          ;               long:   Return address
3A16|                          ;               long:   A0 saved by this routine
3A16|                          ;
3918* 00FE
3A16| 42B8 057C              PUTBACK CLR.L    $57C           ; clear ptr to this routine
3A1A| 2F08                            MOVE.L   A0,-(A7)
3A1C| 21EF 000E 0568                  MOVE.L   14(A7),$568    ; work1 saves the pc
3A22| 21F8 0024 056C                  MOVE.L   $24,$56C       ; work2 saves the trace handler
3A28| 41FA ****                       LEA      PUTBAK2,A0
3A2C| 21C8 0024                       MOVE.L   A0,$24         ; setup new trace handler
3A30| 006F 8000 000C                  ORI.W    #$8000,12(A7)  ; turn on tracing
3A36| 205F                            MOVE.L   (A7)+,A0
3A38| 4E75                            RTS
3A3A|                          ;
3A3A|                          ;       Temp trace handler
3A3A|                          ;
```

```
3A3A|                          ;       stack:
PAGE - 143  MONITOR   FILE: SYMDBUG.TEXT


3A3A|                          ;
3A3A|                          ;               long:   PC
3A3A|                          ;               word:   SR
3A3A|                          ;               long:   A0 saved by this routine
3A3A|                          ;
3A2A* 0010
3A3A| 2F08                     PUTBAK2 MOVE.L  A0,-(A7)
3A3C| 2078 0568                         MOVE.L  $568,A0         ; put back the soft brk pt
3A40| 30BC 4E4C                         MOVE.W  #$4E4C,(A0)
3A44| 205F                             MOVE.L  (A7)+,A0
3A46| 21F8 056C 0024                    MOVE.L  $56C,$24        ; put back the trace handler
3A4C| 4A78 0550                         TST.W   $550            ; test macsbug trace flag
3A50| 66**                             BNE.S   PUTBAK3
3A52| 0257 7FFF                         ANDI.W  #$7FFF,(A7)     ; turn off tracing
3A56| 4E73                             RTE                     ; return to user code
3A50* 06
3A58| 2F38 0024                 PUTBAK3 MOVE.L  $24,-(A7)
3A5C| 4E75                             RTS                     ; go to Macsbug to handle Trace
3A5E|                          ;
3A5E|                          ;       A0TOSO -- CONVERTS A0 TO SEGNUM AND OFFSET
3A5E|                          ;
3A5E|                          ;       Enter with:     A0 = address
3A5E|                          ;                       A1 = address of the active seg list ptr
3A5E|                          ;       Exit with:      D1 = segnum
3A5E|                          ;                       D0 = offset
3A5E|                          ;
3A5E|                          ; using adjusted stacked pc find pointer to segment table entry
3A5E|                          ;
38F0* 016E
3A5E| 2251                     A0TOSO  MOVE.L  (A1),A1         ; get pointer to first segment on the list
3A60| 60**                             BRA.S   FINDLP
3A62| 2269 0018                 NEXTSEG MOVE.L  24(A1),A1       ; get ptr to next segment table
3A66| 2209                             MOVE.L  A1,D1           ; TST.L A1
3A68| 6E**                             BGT.S   FINDLP
3A6A| 72FF                             MOVE.L  #-1,D1
3A6C| 4280                             CLR.L   D0
3A6E| 4E75                             RTS
3A70|
3A68* 06
3A60* 0E
3A70| 2008                     FINDLP  MOVE.L  A0,D0
3A72| 90A9 000C                       SUB.L   12(A1),D0       ; compute offset within segment
3A76| 6BEA                             BMI.S   NEXTSEG
3A78| 2229 0008                       MOVE.L  8(A1),D1        ; get segment size
3A7C| B081                             CMP.L   D1,D0
3A7E| 6CE2                             BGE.S   NEXTSEG
3A80|                          ;
3A80|                          ; enter with pointer to segment table entry in A1
3A80|                          ;
3A80| 2209                             MOVE.L  A1,D1           ; Compute segment number = (A1-A4)/32+1
3A82| 928C                             SUB.L   A4,D1
3A84| EA81                             ASR.L   #5,D1
3A86| 5281                             ADD.L   #1,D1
3A88| 4E75                             RTS
3A8A|                          ;
PAGE - 144  MONITOR   FILE: SYMDBUG.TEXT


3A8A|                          ; PROCEDURE ADDR2SO(USER:INTEGER; ADDRESS:LONGINT; VAR SEGNUM,OFFSET:INTEGER)
3A8A|                          ;
3A8A|                          ;       stack:
3A8A|                          ;               24      flag indicating process (0=SYM DEBUG, 1=USER)
3A8A|                          ;               20      address
3A8A|                          ;               16      @segnum
3A8A|                          ;               12      @offset
3A8A|                          ;                8      Return Address
3A8A|                          ;                4      Old A5
3A8A|                          ;                0      Old A6
3A8A|                          ;
3A8A| 4E56 0000                 ADDR2SO LINK    A6,#0
3A8E| 206E 0014                         MOVE.L  20(A6),A0               ; get address to convert
3A92| 4A6E 0018                         TST.W   24(A6)                  ; running user for the debugger ?
3A96| 67**                             BEQ.S   @1
3A98| 6100 ****                       BSR     SWAPJT
3A9C|                                  GETUFST A1
3A9C| 226D FF90                #       MOVE.L   -112(A5),A1
3AA0| 60**                             BRA.S   @2
3AA2|                          @1      GETZFST A1
3A96* 0A
3AA2| 226D FFF0                #       MOVE.L   -16(A5),A1
3AA0* 04
3AA6| 5149                     @2      SUB.W   #8,A1                   ; compute ZFIRST-8 in A1
3AA8| 61B4                             BSR     A0TOSO
3AAA| 206E 0010                       MOVE.L  16(A6),A0               ; return segnum
3AAE| 3081                             MOVE.W  D1,(A0)
3AB0| 206E 000C                       MOVE.L  12(A6),A0               ; return offset
3AB4| 3080                             MOVE.W  D0,(A0)
3AB6| 4A6E 0018                       TST.W   24(A6)                  ; running user for the debugger ?
3ABA| 67**                             BEQ.S   @3
3ABC| 6100 ****                       BSR     SWAPJT
3ABA* 04
3AC0| 4E5E                     @3      UNLK    A6
```

```
3AC2| 2A5F                          MOVE.L  (A7)+,A5
3AC4| 205F                          MOVE.L  (A7)+,A0
3AC6| DEFC 000E                     ADD.W   #14,A7
3ACA| 4ED0                          JMP     (A0)
3ACC|                          ;
3ACC|                          ; PROCEDURE PMADDR2SO(USER:INTEGER; ADDRESS:LONGINT; VAR SEGNUM,OFFSET:INTEGER)
3ACC|                          ;
3ACC|                          ; PMADDR2SO converts address to segment and offset for the debugger process
3ACC|                          ; while it is running in the users environment.  Note the meaning of user
3ACC|                          ; is reversed.
3ACC|                          ;
3ACC|                          ;       stack:
3ACC|                          ;               24      flag indicating process (1=SYM DEBUG, 0=USER)
3ACC|                          ;               20      address
3ACC|                          ;               16      @segnum
3ACC|                          ;               12      @offset
3ACC|                          ;                8      Return Address
3ACC|                          ;                4      Old A5
3ACC|                          ;                0      Old A6
3ACC|                          ;
3ACC|                          PMADDR2S
PAGE - 145  MONITOR   FILE: SYMDBUG.TEXT


3ACC| 4E56 0000                       LINK    A6,#0
3AD0| 206E 0014                       MOVE.L  20(A6),A0               ; get address to convert
3AD4| 4A6E 0018                       TST.W   24(A6)                  ; running user for the debugger ?
3AD8| 67**                            BEQ.S   @1
3ADA| 6100 ****                       BSR     SWAPJT
3ADE|                                 GETZFST A1
3ADE| 226D FFF0                 #       MOVE.L  -16(A5),A1
3AE2| 60**                            BRA.S   @2
3AE4|                           @1      GETUFST A1
3AD8* 0A
3AE4| 226D FF90                 #       MOVE.L  -112(A5),A1
3AE2* 04
3AE8| 5149                        @2      SUB.W   #8,A1                   ; compute ZFIRST-8 in A1
3AEA| 6100 FF72                       BSR     A0TOSO
3AEE| 206E 0010                       MOVE.L  16(A6),A0               ; return segnum
3AF2| 3081                           MOVE.W  D1,(A0)
3AF4| 206E 000C                       MOVE.L  12(A6),A0               ; return offset
3AF8| 3080                           MOVE.W  D0,(A0)
3AFA| 4A6E 0018                       TST.W   24(A6)                  ; running user for the debugger ?
3AFE| 67**                            BEQ.S   @3
3B00| 6100 ****                       BSR     SWAPJT
3AFE* 04
3B04| 4E5E                        @3      UNLK    A6
3B06| 2A5F                          MOVE.L  (A7)+,A5
3B08| 205F                          MOVE.L  (A7)+,A0
3B0A| DEFC 000E                     ADD.W   #14,A7
3B0E| 4ED0                          JMP     (A0)
3B10|                          ;
3B10|                          ;       LAUNCH -- Called by symbolic debugger to launch user process
3B10|                          ;
3B10|                          ;       stack:
3B10|                          ;
3B10|                          ;               24      address object file
3B10|                          ;               20      address error proc
3B10|                          ;               16      address of parms: J.T. load addr, size, data size
3B10|                          ;               12      address of jump table
3B10|                          ;               8       return address
3B10|                          ;               4       saved A5
3B10|                          ;               0       old A6
3B10|                          ;
3B10| 4E56 0000                LAUNCH  LINK    A6,#0
3B14|                          ;
3B14|                          ;       save the debuggers registers
3B14|                          ;
3B14| 2F08                          MOVE.L  A0,-(A7)                ; save A0 on tos
3B16|                                 GETDREG A0
3B16| 206D FF94                 #       MOVE.L   -108(A5),A0
3B1A| D0FC 0040                       ADD.W   #64,A0                  ; point A0 just above reg save area
3B1E| 48E0 FFFF                       MOVEM.L D0-D7/A0-A7,-(A0)       ; save Ds and As
3B22| 215F 0020                       MOVE.L  (A7)+,32(A0)            ; now save A0
3B26| 216E 0004 0034                 MOVE.L  4(A6),52(A0)            ; and real A5
3B2C| 216E 0008 0040                 MOVE.L  8(A6),64(A0)            ; save return PC
3B32| 40E8 0044                       MOVE    SR,68(A0)               ; and slightly modified SR
3B36| 0268 7FFF 0044                 ANDI.W  #$7FFF,68(A0)           ; make sure tracing is turned off
3B3C| 4268 0046                       MOVE.W  #0,70(A0)
PAGE - 146  MONITOR   FILE: SYMDBUG.TEXT


3B40|                          ;
3B40|                          ;       save ptr to fib
3B40|                          ;
3B40| 206E 0018                       MOVE.L  24(A6),A0
3B44|                                 PUTUFIB A0
3B44| 2B48 FFA4                 #       MOVE.L   A0,-92(A5)
3B48|                          ;
3B48|                          ;       save ptr to error proc
3B48|                          ;
3B48| 206E 0014                       MOVE.L  20(A6),A0
3B4C|                                 PUTERRP A0
3B4C| 2B48 FFA0                 #       MOVE.L   A0,-96(A5)
3B50|                          ;
3B50|                          ;       range check jump table size
```

```
3B50|                           ;
3B50|  206E 0010                         MOVE.L  16(A6),A0
3B54|  0C68 1800 0004                   CMP.W   #UJMPTOP-UJMPTBL,4(A0)
3B5A|  63**                             BLS.S   @1
3B5C|  7004                             MOVE.L  #4,D0
3B5E|  6000 FC9C                        BRA     ERROR
3B62|                           ;
3B62|                           ;       swap jump tables (just swap 6k brute force)
3B62|                           ;
3B5A* 06
3B62|  206E 000C                 @1      MOVE.L  12(A6),A0
3B66|                                   PUTUJPT A0
3B66|  2B48 FF9C                 #       MOVE.L   A0,-100(A5)
3B6A|  6100 ****                        BSR     SWAPJT
3B6E|                           ;
3B6E|                           ;       setup user state
3B6E|                           ;
3B6E|  9FFC 0000 2000                     SUB.L   #$2000,A7               ; leave 8k bytes for debugger
3B74|  6100 ****                        BSR     POKEXCP
3B78|                                   RUNUSET                         ; runuser:=true
3B78|  1B7C 0001 FFA8            #       MOVE.B   #1,-88(A5)
3B7E|                           ;
3B7E|                           ;       address, jump table size and commonsize
3B7E|                           ;
3B7E|  266E 0010                         MOVE.L  16(A6),A3
3B82|  205B                             MOVE.L  (A3)+,A0                ; get address as baseloc in A0
3B84|  4A9B                             TST.L   (A3)+                   ; skip bytesleft
3B86|  2E1B                             MOVE.L  (A3)+,D7                ; get commonsize in D7
3B88|                           ;
3B88|                           ; startit up, A0 contains baseloc (ie. $$TOP)
3B88|                           ;
3B88|  3410                             MOVE.W  (A0),D2         ; D2 = number of segments
3B8A|  E84A                             LSL.W   #5,D2           ;       *32
3B8C|  2270 2002                        MOVE.L  2(A0,D2.W),A1   ; A1 = address of $$FIRST-20
3B90|                                   GETIBF  A2
3B90|  246D FFAE                #       MOVE.L   -82(A5),A2
3B94|  234A 0004                        MOVE.L  A2,4(A1)        ; Move address of buffer to $$FIRST-16
3B98|                                   GETUFIB A2
3B98|  246D FFA4                #       MOVE.L   -92(A5),A2
3B9C|  234A 0008                        MOVE.L  A2,8(A1)        ; Move address of objfile to $$FIRST-12
3BA0|  4869 0014                        PEA     20(A1)          ; Push pointer to $$FIRST
PAGE - 147  MONITOR   FILE: SYMDBUG.TEXT


3BA4|  4267                             MOVE.W  #0,-(A7)        ; Dummy push for Segment Bank Word
3BA6|  48E7 1F3E                        MOVEM.L D3-D7/A2-A6,-(SP)
3BAA|                                   GETOFIB -(A7)           ; +12(A5) = @OUTPUT
3BAA|  2F2D FFE8                #       MOVE.L   -24(A5),-(A7)
3BAE|                                   GETIFIB -(A7)           ;  +8(A5) = @INPUT
3BAE|  2F2D FFEC                #       MOVE.L   -20(A5),-(A7)
3BB2|  4BEF FFF8                        LEA     -8(SP),A5       ; Set A5 so 16(A5) = P-regs.
3BB6|  2068 0002                        MOVE.L  2(A0),A0        ; A0 = Main Program Entry - 4
3BBA|  4EA8 0004                        JSR     4(A0)           ; Start her up!
3BBE|  508F                             ADDQ.L  #8,SP           ; Restore the registers
3BC0|  4CDF 7CF8                        MOVEM.L (SP)+,D3-D7/A2-A6
3BC4|  4A5F                             TST.W   (SP)+           ; Throw away dummy Segment Bank Word
3BC6|  588F                             ADDQ.L  #4,SP           ; Eliminate pointer to $$FIRST
3BC8|                           ;
3BC8|                           ; get back to debugger
3BC8|                           ;
3BC8|  6100 ****                        BSR     SWAPJT
3BCC|                                   RUNUCLR
3BCC|  422D FFA8                #       MOVE.B   #0,-88(A5)
3BD0|                           ;
3BD0|                           ;       exit
3BD0|                           ;
3BD0|  4E5E                             UNLK    A6
3BD2|  2A5F                             MOVE.L  (A7)+,A5                ; restore global ptr
3BD4|  205F                             MOVE.L  (A7)+,A0                ; pop return address
3BD6|  DEFC 0010                        ADD.W   #16,A7                  ; delete parameters
3BDA|  4ED0                             JMP     (A0)
3BDC|                           ;
3BDC|                           ;       POKEXCP -- poke exception vectors for the symbolic debugger
3BDC|                           ;
3B76* 0066
39D2* 020A
3BDC|  2F08                     POKEXCP MOVE.L  A0,-(A7)
3BDE|  41FA FC7C                        LEA     V2,A0
3BE2|  21C8 0008                        MOVE.L  A0,$8
3BE6|  41FA FC7A                        LEA     V3,A0
3BEA|  21C8 000C                        MOVE.L  A0,$C
3BEE|  41FA FC78                        LEA     V4,A0
3BF2|  21C8 0010                        MOVE.L  A0,$10
3BF6|  41FA FC76                        LEA     V5,A0
3BFA|  21C8 0014                        MOVE.L  A0,$14
3BFE|  41FA FC74                        LEA     V6,A0
3C02|  21C8 0018                        MOVE.L  A0,$18
3C06|  41FA FC72                        LEA     V7,A0
3C0A|  21C8 001C                        MOVE.L  A0,$1C
3C0E|  41FA FC70                        LEA     V8,A0
3C12|  21C8 0020                        MOVE.L  A0,$20
3C16|  41FA FC6E                        LEA     V9,A0
3C1A|  21C8 0024                        MOVE.L  A0,$24
3C1E|  41FA FC6C                        LEA     V10,A0
3C22|  21C8 0028                        MOVE.L  A0,$28
3C26|  41FA FC6A                        LEA     V11,A0
```

```
3C2A| 21C8 002C                    MOVE.L  A0,$2C
3C2E| 41FA FC68                    LEA     V24,A0
3C32| 21C8 0060                    MOVE.L  A0,$60
PAGE - 148  MONITOR   FILE: SYMDBUG.TEXT


3C36| 41FA FBEA                    LEA     V31,A0
3C3A| 21C8 007C                    MOVE.L  A0,$7C
3C3E| 205F                         MOVE.L  (A7)+,A0
3C40| 4E75                         RTS
3C42|                         ;
3C42|                         ;       RSTRXCP -- restore exception vectors for macsbug
3C42|                         ;
3C42| 93C9                    RSTRXCP MOVE.L  #$00000000,A1
3C44| 207C 0000 0F00                 MOVE.L  #$00000F00,A0
3C4A| 303C 003F                       MOVE.W  #63,D0                  ; zero page length is 64 longs
3C4E| 22D8                    @1      MOVE.L  (A0)+,(A1)+
3C50| 51C8 FFFC                       DBF     D0,@1
3C54| 4E75                         RTS
3C56|                         ;
3C56|                         ;       SWAPJT -- swap user and debugger jump tables
3C56|                         ;
3BCA* 008C
3B6C* 00EA
3B02* 0154
3ADC* 017A
3ABE* 0198
3A9A* 01BC
3C56| 303C 0C00              SWAPJT  MOVE.W  #UJMPTOP-UJMPTBL/2,D0
3C5A| 224C                         MOVE.L  A4,A1
3C5C|                              GETUJPT A2
3C5C| 246D FF9C               #       MOVE.L   -100(A5),A2
3C60| 3211                     @1      MOVE.W  (A1),D1
3C62| 32D2                             MOVE.W  (A2),(A1)+
3C64| 34C1                             MOVE.W  D1,(A2)+
3C66| 5340                             SUB.W   #1,D0
3C68| 66F6                             BNE.S   @1
3C6A| 4E75                         RTS
3C6C|
3C6C|
3C6C|                         ;
3C6C|                              .INCLUDE COMMAND.TEXT
3C6C|                         ;
3C6C|                         ;       FILENAME: COMMAND
3C6C|                         ;
3C6C|
3C6C|                         ;
3C6C|                         ;       INITPRG
3C6C|                         ;
3C6C|                          INITPRG RUNUCLR
3C6C| 422D FFA8               #       MOVE.B   #0,-88(A5)
3C70| 41EF 0004                       LEA     4(A7),A0                ; setup end of break point table
3C74| 30BC FFFF                       MOVE.W  #$FFFF,(A0)
3C78| 21C8 012C                       MOVE.L  A0,$12C
3C7C|                              GETLOGN A0
3C7C| 206D FF60               #       MOVE.L   -160(A5),A0
3C80| 303C 0007                       MOVE.W  #MAXDEV,D0
3C84| 4298                     @1      CLR.L   (A0)+                   ; clear LOGN bit for each unit each dev
3C86| 51C8 FFFC                       DBF     D0,@1
3C8A| 4E75                         RTS
3C8C|                         ;
PAGE - 149  MONITOR   FILE: COMMAND.TEXT


3C8C|                         ;       SETUPA5
3C8C|                         ;
3C8C| 2A7C 0000 0D00            SETUPA5 MOVE.L  #$00000D00,A5           ; ptr to system globals
3C92| 4E75                         RTS
3C94|                         ;
3C94|                         ;       GETREGS
3C94|                         ;
3C94| 245F                    GETREGS MOVE.L  (A7)+,A2
3C96| 61F4                            BSR     SETUPA5                 ; get base of globals
3C98| 2038 0144                       MOVE.L  $144,D0                 ; get base of memory volume
3C9C| 67**                           BEQ.S   @1
3C9E| 2040                           MOVE.L  D0,A0
3CA0| D0FC 0400                       ADD.W   #2*FBLKSIZ,A0
3CA4| 3228 000E                       MOVE.W  DEOVBLK(A0),D1
3CA8| C3FC 0200                       MULS    #FBLKSIZ,D1
3CAC| D081                           ADD.L   D1,D0                   ; compute empty heap
3CAE|                              PUTEMTH D0                      ; save as empty heap
3CAE| 2B40 FFF8               #       MOVE.L   D0,-8(A5)
3CB2|                              PUTNP   D0                      ; cut back heap
3CB2| 2B40 FFF4               #       MOVE.L   D0,-12(A5)
3CB6| 60**                           BRA.S   @2
3CB8|                          @1      GETEMTH A0                      ; get empty heap
3C9C* 1A
3CB8| 206D FFF8               #       MOVE.L   -8(A5),A0
3CBC|                              PUTNP   A0                      ; cut back heap
3CBC| 2B48 FFF4               #       MOVE.L   A0,-12(A5)
3CC0|                           @2      GETSPTR A7                      ; get Default SP into A7
3CB6* 08
3CC0| 2E6D FFB2               #       MOVE.L   -78(A5),A7
3CC4| 207C 0000 0108                 MOVE.L  #$00000108,A0
3CCA| 20CF                           MOVE.L  A7,(A0)+                ; set asm lang top
3CCC| 43FA CA22                       LEA     UNITBL,A1
```

```
3CD0| 20C9                          MOVE.L  A1,(A0)+                ; set buffer bottom
3CD2| 4ED2                          JMP     (A2)                    ; return
3CD4|                       ;
3CD4| 000A                  ILPL    .WORD   10
3CD6| 49 2F 4F 20 45 72 72  IPL     .ASCII  'I/O Error '
3CDD| 6F 72 20
3CE0|                       ;
3CE0| 001A                  TLPL    .WORD   26
3CE2| 20 74 79 70 65 3A 20  TPL     .ASCII  ' type: <space> to continue'
3CE9| 3C 73 70 61 63 65 3E
3CF0| 20 74 6F 20 63 6F 6E
3CF7| 74 69 6E 75 65
3CFC|                       ;
3CFC|                       ;       IOCHK
3CFC|                       ;
3CFC| 4E56 FFFC            IOCHK   LINK    A6,#-4                  ; allocate: iorsltcopy & errornumber
3D00|                               GETSYSC A0                      ; @iorslt
3D00| 206D FFFC              #       MOVE.L   -4(A5),A0
3D04| 1D50 FFFE                      MOVE.B  (A0),-2(A6)             ; save iorslt
3D08| 67**                          BEQ.S   IOCHKX
3D0A| 43FA FFCA                      LEA     IPL,A1                  ; address
3D0E| 343A FFC4                      MOVE.W  ILPL,D2                 ; and length
3D12| 6100 ****                      BSR     PRINT
PAGE - 150  MONITOR   FILE: COMMAND.TEXT


3D16| 43EE FFFC                      LEA     -4(A6),A1               ; address
3D1A| 7402                          MOVE.L  #2,D2                   ; and length
3D1C| 32BC 0030                      MOVE.W  #$30,(A1)               ; initialize error number to '0'
3D20| 4280                          CLR.L   D0
3D22| 102E FFFE                      MOVE.B  -2(A6),D0               ; get iorslt as 32 bit integer
3D26| 80FC 000A            PRNTERR DIVU    #10,D0                  ; divide by 10
3D2A| 4840                          SWAP    D0                      ; get remainder
3D2C| 0200 0007                      AND.B   #7,D0
3D30| D129 0001                      ADD.B   D0,1(A1)                ; least sig digit
3D34| 4840                          SWAP    D0                      ; get quotient
3D36| 0200 0007                      AND.B   #7,D0                   ; most sig digit
3D3A| 67**                          BEQ.S   @1
3D3C| 0600 0030                      ADD.B   #$30,D0                 ; if most sig digit <> 0 then add '0'
3D40| 1280                          MOVE.B  D0,(A1)
3D3A* 06
3D42| 6100 ****              @1      BSR     PRINT
3D46| 43FA FF9A                      LEA     TPL,A1                  ; address
3D4A| 343A FF94                      MOVE.W  TLPL,D2                 ; and length
3D4E| 6100 ****                      BSR     PRINT
3D52| 6100 ****              @2      BSR     GETCHAR                 ; get next character
3D56| 101F                          MOVE.B  (A7)+,D0
3D58| 0C00 0020                      CMPI.B  #$20,D0                 ; loop until <space> is read
3D5C| 66F4                          BNE.S   @2
3D5E| 6000 ****                      BRA     RESTART
3D62|                       ;
3D08* 58
3D62| 4E5E                  IOCHKX  UNLK    A6
3D64| 4E75                          RTS
3D66|                       ;
3D66|                       ;       UPSHFT -- upshift the char in D0
3D66|                       ;
3D66| 0C00 0061            UPSHFT  CMPI.B  #'a',D0                 ; less than 'a' ?
3D6A| 6D**                          BLT.S   UPSHFTX
3D6C| 0C00 007A                      CMPI.B  #'z',D0                 ; greater than 'z' ?
3D70| 6E**                          BGT.S   UPSHFTX
3D72| 0400 0020                      SUB.B   #32,D0                  ; D0 in 'a'..'z' so upshift
3D70* 04
3D6A* 0A
3D76| 4E75                  UPSHFTX RTS
3D78|                       ;
3D78|                       ;       DELAY FOR SLOW TERMINALS
3D78|                       ;
3D78| 0000 0000 0000 0000   WAIT    .WORD   0,0,0,0,0
3D80| 0000
3D82|                       ;
3D82|                       ;       PUTPRFX -- PUT PREFIXED CRTCTRL CHAR
3D82|                       ;
3D82|                       ;       ENTER WITH:
3D82|                       ;
3D82|                       ;               A0      PTR TO SYSCOM
3D82|                       ;               D0      WHICH CRTCTRL CHAR
3D82|                       ;               A1      @COMMAND CHAR
3D82|                       ;
3D82| 4A11                  PUTPRFX TST.B   (A1)                    ; exit if command char = null
3D84| 67**                          BEQ.S   PUTPRF2
PAGE - 151  MONITOR   FILE: COMMAND.TEXT


3D86| 3F09                          MOVE.W  A1,-(A7)                ; save @command char
3D88| 3228 0014                      MOVE.W  CRTESC+10(A0),D1
3D8C| 0101                          BTST    D0,D1                   ; prefixed ?
3D8E| 67**                          BEQ.S   PUTPRF1
3D90| 43E8 000B                      LEA     CRTESC+1(A0),A1         ; @esc char
3D94| 7401                          MOVE.L  #1,D2                   ; length of 1
3D96| 6100 ****                      BSR     PRINT                   ; print esc char
3D9A| 43FA FFDC                      LEA     WAIT,A1                 ; address
3D9E| 740A                          MOVE.L  #10,D2                  ; and length
3DA0| 6100 ****                      BSR     PRINT
3D8E* 14
3DA4| 325F                  PUTPRF1 MOVE.W  (A7)+,A1                ; restore @command char
```

```
3DA6| 7401                          MOVE.L  #1,D2                   ; length of 1
3DA8| 6100 ****                     BSR     PRINT                   ; print command char
3DAC| 43FA FFCA                     LEA     WAIT,A1                 ; address
3DB0| 740A                          MOVE.L  #10,D2                  ; and length
3DB2| 6100 ****                     BSR     PRINT
3D84* 30
3DB6| 4E75                  PUTPRF2 RTS
3DB8|                       ;
3DB8|                       ;       HOMCRSR -- HOME CURSOR
3DB8|                       ;
3DB8|                       HOMCRSR GETSYSC A0
3DB8| 206D FFFC             #       MOVE.L   -4(A5),A0
3DBC| 7004                          MOVE.L  #P_HOME,D0              ; index to prefix array
3DBE| 43E8 000A                     LEA     CRTESC+0(A0),A1         ; @home char
3DC2| 60BE                          BRA     PUTPRFX
3DC4|                       ;
3DC4|                       ;       CLRLN -- CLEAR LINE
3DC4|                       ;
3DC4|                       CLRLN   GETSYSC A0
3DC4| 206D FFFC             #       MOVE.L   -4(A5),A0
3DC8| 43E8 000C                     LEA     CRTESC+2(A0),A1         ; @erase eol char
3DCC| 4A11                          TST.B   (A1)
3DCE| 67**                          BEQ.S   CLRLN2
3DD0| 7002                          MOVE.L  #P_ERSEL,D0             ; index to prefix array
3DD2| 60AE                          BRA     PUTPRFX
3DCE* 04
3DD4| 43E8 0013              CLRLN2  LEA     CLRLINE(A0),A1          ; @clear line char
3DD8| 7007                          MOVE.L  #P_CLRLN,D0             ; index to prefix array
3DDA| 60A6                          BRA     PUTPRFX
3DDC|                       ;
3DDC|                       ;       CLRSCR -- CLEAR SCREEN
3DDC|                       ;
3DDC|                       CLRSCR  GETSYSC A0
3DDC| 206D FFFC             #       MOVE.L   -4(A5),A0
3DE0| 43E8 000D                     LEA     CRTESC+3(A0),A1         ; @erase eos char
3DE4| 4A11                          TST.B   (A1)
3DE6| 67**                          BEQ.S   CLRSCR2
3DE8| 7003                          MOVE.L  #P_ERSES,D0             ; index to prefix array
3DEA| 6096                          BRA     PUTPRFX
3DE6* 04
3DEC| 43E8 0012              CLRSCR2 LEA     CLRSCRN(A0),A1          ; @clear screen char
3DF0| 7006                          MOVE.L  #P_CLRSC,D0             ; index to prefix array
3DF2| 608E                          BRA     PUTPRFX
PAGE - 152  MONITOR   FILE: COMMAND.TEXT


3DF4|                       ;
3DF4| 004C                  LPL2    .WORD   76
3DF6| 4D 28 61 63 73 62 75 PL2     .ASCII  'M(acsbug, U(csd editor, G(enerate code, '
3DFD| 67 2C 20 55 28 63 73
3E04| 64 20 65 64 69 74 6F
3E0B| 72 2C 20 47 28 65 6E
3E12| 65 72 61 74 65 20 63
3E19| 6F 64 65 2C 20
3E1E| 44 28 65 62 75 67 20         .ASCII  'D(ebug                         '
3E25| 20 20 20 20 20 20 20
3E2C| 20 20 20 20 20 20 20
3E33| 20 20 20 20 20 20 20
3E3A| 20 20 20 20 20 20 20
3E41| 20
3E42|                       ;
3E42|                       ;       PROMPT2 -- PROMPT THE USER FOR A COMMAND
3E42|                       ;
3E42| 6100 FF74             PROMPT2 BSR     HOMCRSR
3E46| 6100 FF7C                     BSR     CLRLN
3E4A| 43FA FFAA                     LEA     PL2,A1                  ; PL address
3E4E| 343A FFA4                     MOVE.W  LPL2,D2                 ; and length
3E52| 60**                          BRA.S   PRINT
3E54|                       ;
3E54|                       ;
3E54|                       ;
3E54|                       REVISION
3E54| 3F7C 006E 0004                MOVE.W  #110,4(A7)      ; high byte = 0, low byte = 11.0
3E5A| 4E75                          RTS
3E5C|                       ;
3E5C| 004F                  LPL     .WORD   79
3E5E| 4D 6F 6E 69 74 6F 72 PL      .ASCII  'Monitor: E(dit, C(ompile, F(ile, L(ink, '
3E65| 3A 20 45 28 64 69 74
3E6C| 2C 20 43 28 6F 6D 70
3E73| 69 6C 65 2C 20 46 28
3E7A| 69 6C 65 2C 20 4C 28
3E81| 69 6E 6B 2C 20
3E86| 58 28 65 63 75 74 65         .ASCII  'X(ecute, A(ssemble, S(ysmgr, ? [0.11.0] '
3E8D| 2C 20 41 28 73 73 65
3E94| 6D 62 6C 65 2C 20 53
3E9B| 28 79 73 6D 67 72 2C
3EA2| 20 3F 20 5B 30 2E 31
3EA9| 31 2E 30 5D 20
3EAE|                       ;       PROMPT -- PROMPT THE USER FOR A COMMAND
3EAE|                       ;
3EAE| 6100 FF08             PROMPT  BSR     HOMCRSR
3EB2| 6100 FF10                     BSR     CLRLN
3EB6| 43FA FFA6                     LEA     PL,A1                   ; PL address
3EBA| 343A FFA0                     MOVE.W  LPL,D2                  ; and length
3EBE|                       ;
3EBE|                       ;       PRINT -- PRINT AN ARRAY OF BYTES
```

```
3EBE|                          ;
3EBE|                          ;       ENTER WITH:
3EBE|                          ;
3EBE|                          ;               A1      address
3EBE|                          ;               D2      length
PAGE - 153  MONITOR   FILE: COMMAND.TEXT


3EBE|                          ;
3E52* 6A
3DB4* 010A
3DAA* 0114
3DA2* 011C
3D98* 0126
3D50* 016E
3D44* 017A
3D14* 01AA
3EBE| 3F3C 0001              PRINT   MOVE.W  #CONSOLE,-(A7)          ; unit#
3EC2| 2209                          MOVE.L  A1,D1                   ; address
3EC4| 4243                          MOVE.W  #0,D3                   ; block#
3EC6| 7C02                          MOVE.L  #OUTBIT,D6
3EC8| 6100 DA92                     BSR     GETUNIT                 ; get unit# and validate, setup drvr
3ECC| 4E90                          JSR     (A0)                    ; go to driver
3ECE| 4E75                          RTS
3ED0|                          ;
3ED0|                          ;       GETCHAR -- GET A CHAR FROM THE CONSOLE
3ED0|                          ;
3ED0|                          ;       RETURNS THE CHAR ON TOS IN THE MS BYTE OF WORD
3ED0|                          ;
3D54* 017C
3ED0| 205F                    GETCHAR MOVE.L  (A7)+,A0                ; save return address
3ED2| 4267                          MOVE.W  #0,-(A7)                ; allocate word for result
3ED4| 220F                          MOVE.L  A7,D1                   ; address
3ED6| 2F08                          MOVE.L  A0,-(A7)                ; restore return address
3ED8| 7401                          MOVE.L  #1,D2                   ; length
3EDA| 3F3C 0002                     MOVE.W  #SYSTERM,-(A7)          ; unit#
3EDE| 4243                          MOVE.W  #0,D3                   ; block#
3EE0| 7C01                          MOVE.L  #INBIT,D6
3EE2| 6100 DA78                     BSR     GETUNIT                 ; get unit# and validate, setup drvr
3EE6| 4E90                          JSR     (A0)                    ; go to driver
3EE8| 41EF 0004                     LEA     4(A7),A0                ; address of char read
3EEC| 1010                          MOVE.B  (A0),D0
3EEE| 0C00 000D                     CMPI.B  #13,D0                  ; if char = cr then echo crlf
3EF2| 67**                          BEQ.S   WCRLF
3EF4| 0C00 0020                     CMPI.B  #32,D0                  ; if char >= ' ' then echo char
3EF8| 6D**                          BLT.S   GETCHRX
3EFA| 2208                          MOVE.L  A0,D1                   ; address
3EFC| 7401                          MOVE.L  #1,D2                   ; length
3EFE| 3F3C 0002                     MOVE.W  #SYSTERM,-(A7)          ; unit#
3F02| 4243                          MOVE.W  #0,D3                   ; block#
3F04| 7C02                          MOVE.L  #OUTBIT,D6
3F06| 6100 DA54                     BSR     GETUNIT                 ; get unit# and validate, setup drvr
3F0A| 4E90                          JSR     (A0)                    ; go to driver
3EF8* 12
3F0C| 4E75                    GETCHRX RTS
3F0E|                          ;
3F0E| 0D00                    CRLF    .WORD   $0D00
3F10|                          ;
3F10|                          ;       WCRLF -- write CRLF to CONSOLE
3F10|                          ;
3EF2* 1C
3F10| 43FA FFFC              WCRLF   LEA     CRLF,A1
3F14| 7401                          MOVE.L  #1,D2
PAGE - 154  MONITOR   FILE: COMMAND.TEXT


3F16| 60A6                          BRA     PRINT
3F18|                          ;
3F18|                          ;       XEQUTE -- execute a file
3F18|                          ;
3F18| 000C                    XLPL    .WORD   12
3F1A| 57 68 61 74 20 66 69    XPL     .ASCII  'What file ? '
3F21| 6C 65 20 3F 20
3F26| 0820 0800                BSSPBS  .WORD   $0820,$0800
3F2A|                          ;
3F2A|                          ; initialize & prompt user for filename
3F2A|                          ;
3F2A|                          XEQUTE  GETSYSC A0
3F2A| 206D FFFC               #        MOVE.L   -4(A5),A0
3F2E| 4210                          MOVE.B  #INOERR,(A0)            ; initialize iorslt
3F30| 6100 FE86                XEQUTE0 BSR     HOMCRSR
3F34| 6100 FEA6                     BSR     CLRSCR
3F38| 43FA FFE0                     LEA     XPL,A1                  ; PL address
3F3C| 343A FFDA                     MOVE.W  XLPL,D2                 ; and length
3F40| 6100 FF7C                     BSR     PRINT
3F44|                               GETSTRB A2                      ; @string
3F44| 246D FFAA               #        MOVE.L   -86(A5),A2
3F48| 264A                          MOVE.L  A2,A3
3F4A| 421A                          MOVE.B  #0,(A2)+                ; initialize string
3F4C|                          ;
3F4C|                          ; read the filename into strbuf
3F4C|                          ;
3F4C| 6182                    XEQUTE1 BSR     GETCHAR                 ; get next character
3F4E| 101F                          MOVE.B  (A7)+,D0
3F50| 6100 FE14                     BSR     UPSHFT                  ; and upshift it
3F54| 0C00 0008                     CMPI.B  #8,D0                   ; char = <bs> ?
```

```
3F58| 67**                          BEQ.S   XEQUTE4
3F5A|                               GETSYSC A0
3F5A| 206D FFFC              #       MOVE.L   -4(A5),A0
3F5E| B028 0025                     CMP.B   LINDEL(A0),D0           ; char = <linedel> ?
3F62| 67CC                          BEQ.S   XEQUTE0
3F64| 0C00 000D                     CMPI.B  #13,D0                  ; char = <cr> ?
3F68| 67**                          BEQ.S   XEQUTE2
3F6A| 5213                         ADD.B   #1,(A3)                 ; no then incr string length
3F6C| 67**                          BEQ.S   XEQUTE2                 ; string overflow ?
3F6E| 14C0                         MOVE.B  D0,(A2)+                ; no then append the character
3F70| 60DA                         BRA.S   XEQUTE1                 ; go get next character
3F58* 18
3F72| 4A13                XEQUTE4 TST.B   (A3)                    ; any chars ?
3F74| 67D6                         BEQ.S   XEQUTE1
3F76| 5313                         SUB.B   #1,(A3)                 ; yes decr string length and
3F78| 4A22                         TST.B   -(A2)                   ;      discard the last char
3F7A| 43FA FFAA                    LEA     BSSPBS,A1
3F7E| 7403                         MOVE.L  #3,D2
3F80| 6100 FF3C                     BSR     PRINT                   ; erase 1 char
3F84| 60C6                         BRA.S   XEQUTE1
3F86|                       ;
3F86|                       ; exit
3F86|                       ;
3F6C* 18
3F68* 1C
PAGE - 155  MONITOR   FILE: COMMAND.TEXT


3F86| 4A13                XEQUTE2 TST.B   (A3)                    ; string length zero ?
3F88| 66**                          BNE.S   XEQUTE3                 ; yes, then no file name
3F8A|                               GETSYSC A0
3F8A| 206D FFFC              #       MOVE.L   -4(A5),A0
3F8E| 10BC 000A                     MOVE.B  #INOFILE,(A0)
3F88* 08
3F92| 4E75                XEQUTE3 RTS
3F94|                       ;
3F94|                       ;       FROMEXEC -- get a character from the exec file
3F94|                       ;
3F94|                       FROMEXEC
3F94| 48E7 7FFE                     MOVEM.L D1-D7/A0-A6,-(A7)
3F98|                               GETNXCH D0
3F98| 102D FF86              #       MOVE.B   -122(A5),D0
3F9C| 66**                          BNE.S   @1
3F9E| 42A7                         CLR.L   -(A7)
3FA0| 487A ****                     PEA     @3
3FA4|                               GETXFIB -(A7)
3FA4| 2F2D FF82              #       MOVE.L   -126(A5),-(A7)
3FA8| 6000 EFDE                     BRA     FREADCHR
3FA2* 000A
3FAC| 301F                 @3       MOVE.W  (A7)+,D0
3FAE|                               PUTNXCH D0
3FAE| 1B40 FF86              #       MOVE.B   D0,-122(A5)
3F9C* 14
3FB2| 42A7                 @1       CLR.L   -(A7)
3FB4| 487A ****                     PEA     @4
3FB8|                               GETXFIB -(A7)
3FB8| 2F2D FF82              #       MOVE.L   -126(A5),-(A7)
3FBC| 6000 EFCA                     BRA     FREADCHR
3FC0|                       @4      GETNXCH D0
3FB6* 000A
3FC0| 102D FF86              #       MOVE.B   -122(A5),D0
3FC4| 321F                         MOVE.W  (A7)+,D1
3FC6|                               GETXFIB A0
3FC6| 206D FF82              #       MOVE.L   -126(A5),A0
3FCA| 4A28 0004                     TST.B   FEOLN(A0)
3FCE| 67**                          BEQ.S   @5
3FD0| 123C 000D                     MOVE.B  #$D,D1
3FD4| 4228 0004                     MOVE.B  #0,FEOLN(A0)
3FD8|                       @5      PUTNXCH D1
3FCE* 08
3FD8| 1B41 FF86              #       MOVE.B   D1,-122(A5)
3FDC|                               GETEBST D2
3FDC| 142D FF87              #       MOVE.B   -121(A5),D2
3FE0| B002                         CMP.B   D2,D0
3FE2| 66**                          BNE.S   @2
3FE4| B202                         CMP.B   D2,D1
3FE6| 66**                          BNE.S   @2
3FE8|                               PUTEBST #0
3FE8| 422D FF87              #       MOVE.B   #0,-121(A5)
3FEC|                               PUTNXCH #0
3FEC| 422D FF86              #       MOVE.B   #0,-122(A5)
3FF0| 6100 FEDE                     BSR     GETCHAR                 ; get next character from console
3FF4| 101F                         MOVE.B  (A7)+,D0
3FE6* 0E
PAGE - 156  MONITOR   FILE: COMMAND.TEXT


3FE2* 12
3FF6| 4CDF 7FFE             @2      MOVEM.L (A7)+,D1-D7/A0-A6
3FFA| 4E75                         RTS
3FFC|                       ;
3FFC|                       ;       OPNEXEC -- open exec file
3FFC|                       ;
3FFC|                       OPNEXEC GETXFIB -(A7)                   ; @file
3FFC| 2F2D FF82              #       MOVE.L   -126(A5),-(A7)
4000| 42A7                         CLR.L   -(A7)
```

```
4002| 3F3C FFFE                    MOVE.W  #-2,-(A7)               ; text file
4006| 6100 DB04                    BSR     FINIT
400A|                              GETXFIB -(A7)                   ; @file
400A| 2F2D FF82              #     MOVE.L   -126(A5),-(A7)
400E|                              GETSTRB -(A7)                   ; @title
400E| 2F2D FFAA              #     MOVE.L   -86(A5),-(A7)
4012| 3F3C 0001                    MOVE.W  #1,-(A7)                ; openold:=true
4016| 42A7                         CLR.L   -(A7)                   ; junk
4018| 6100 E740                    BSR     FOPEN                   ; open exec file
401C| 4E75                         RTS
401E|                         ;
401E|                         ;     APNDTXT -- appends '.TEXT' to the code file name
401E|                         ;
401E|                         APNDTXT GETSTRB A0
401E| 206D FFAA              #     MOVE.L   -86(A5),A0
4022| 2248                         MOVE.L  A0,A1
4024| 4240                         CLR.W   D0
4026| 1018                         MOVE.B  (A0)+,D0
4028| D0C0                         ADD.W   D0,A0
402A| 10FC 002E                    MOVE.B  #'.',(A0)+
402E| 10FC 0054                    MOVE.B  #'T',(A0)+
4032| 10FC 0045                    MOVE.B  #'E',(A0)+
4036| 10FC 0058                    MOVE.B  #'X',(A0)+
403A| 10FC 0054                    MOVE.B  #'T',(A0)+
403E| 5A11                         ADD.B   #5,(A1)
4040| 4E75                         RTS
4042|                         ;
4042|                         ;     TRYEXEC -- looks for EXEC/ in strbuf
4042|                         ;
4042| 3F01                    TRYEXEC MOVE.W  D1,-(A7)
4044|                              GETEBST D0                      ; already in an exec ?
4044| 102D FF87              #     MOVE.B   -121(A5),D0
4048| 6600 ****                    BNE     @1
404C| 4A57                         TST.W   (A7)
404E| 67**                         BEQ.S   @4
4050|                              GETSTRB A0
4050| 206D FFAA              #     MOVE.L   -86(A5),A0
4054| 1018                         MOVE.B  (A0)+,D0
4056| 0C00 0005                    CMP.B   #5,D0
405A| 6D00 ****                    BLT     @1
405E| 1018                         MOVE.B  (A0)+,D0                ; look for EXEC/
4060| 0C00 0045                    CMP.B   #$45,D0
4064| 6600 ****                    BNE     @1
4068| 2010                         MOVE.L  (A0),D0
406A| 0C80 5845 432F               CMP.L   #$5845432F,D0
PAGE - 157  MONITOR   FILE: COMMAND.TEXT


4070| 6600 ****                    BNE     @1
4074|                              GETSTRB A0                      ; delete the EXEC/
4074| 206D FFAA              #     MOVE.L   -86(A5),A0
4078| 7201                         MOVE.L  #1,D1
407A| 363C 0005                    MOVE.W  #5,D3
407E| 6100 DD7A               @2    BSR     DEL1CH
4082| 5343                         SUB.W   #1,D3
4084| 66F8                         BNE.S   @2
404E* 36
4086| 6100 FF74               @4    BSR     OPNEXEC                 ; go open it
408A|                              GETSYSC A0
408A| 206D FFFC              #     MOVE.L   -4(A5),A0
408E| 4A10                         TST.B   (A0)                    ; test iorslt
4090| 67**                         BEQ.S   @3
4092| 6100 ****                    BSR     INSSTAR
4096| 6100 FF64                    BSR     OPNEXEC
409A|                              GETSYSC A0
409A| 206D FFFC              #     MOVE.L   -4(A5),A0
409E| 4A10                         TST.B   (A0)
40A0| 67**                         BEQ.S   @3
40A2|                              GETSTRB A0
40A2| 206D FFAA              #     MOVE.L   -86(A5),A0
40A6| 7201                         MOVE.L  #1,D1
40A8| 6100 DD50                    BSR     DEL1CH
40AC| 6100 FF70                    BSR     APNDTXT
40B0| 6100 FF4A                    BSR     OPNEXEC                 ; go open it
40B4|                              GETSYSC A0
40B4| 206D FFFC              #     MOVE.L   -4(A5),A0
40B8| 4A10                         TST.B   (A0)                    ; test iorslt
40BA| 67**                         BEQ.S   @3
40BC| 6100 ****                    BSR     INSSTAR
40C0| 6100 FF3A                    BSR     OPNEXEC
40C4|                              GETSYSC A0
40C4| 206D FFFC              #     MOVE.L   -4(A5),A0
40C8| 4A10                         TST.B   (A0)                    ; test iorslt
40CA| 66**                         BNE.S   @1
40BA* 10
40A0* 2A
4090* 3A
40CC| 6100 FEC6               @3    BSR     FROMEXEC
40D0| 4A57                         TST.W   (A7)
40D2| 66**                         BNE.S   @6
40D4| 0C00 0025                    CMP.B   #'%',D0
40D8| 66**                         BNE.S   @1
40DA|                          @6    PUTEBST D0                      ; set ebstop to first char
40D2* 06
40DA| 1B40 FF87              #     MOVE.B   D0,-121(A5)
40DE| 303C 0001                    MOVE.W  #1,D0
```

```
40E2| 60**                          BRA.S   @5
40D8* 0A
40CA* 18
4072* 0072
4066* 007E
405C* 0088
404A* 009A
PAGE - 158  MONITOR   FILE: COMMAND.TEXT


40E4| 4240                  @1      MOVE.W  #0,D0
40E2* 02
40E6| 4A5F                  @5      TST.W   (A7)+                   ; discard D1
40E8| 4E75                          RTS
40EA|                       ;
40EA|                       ;       OPNCODE -- open code file
40EA|                       ;
40EA|                       OPNCODE GETCFIB -(A7)                   ; @file
40EA| 2F2D FFDC             #       MOVE.L   -36(A5),-(A7)
40EE| 42A7                          CLR.L   -(A7)
40F0| 3F3C FFFF                      MOVE.W  #-1,-(A7)               ; untyped file
40F4| 6100 DA16                      BSR     FINIT                   ; initialize codefib
40F8|                               GETCFIB -(A7)                   ; @file
40F8| 2F2D FFDC             #       MOVE.L   -36(A5),-(A7)
40FC|                               GETSTRB -(A7)                   ; @title
40FC| 2F2D FFAA             #       MOVE.L   -86(A5),-(A7)
4100| 3F3C 0001                      MOVE.W  #1,-(A7)                ; openold:=true
4104| 42A7                          CLR.L   -(A7)                   ; junk
4106| 6100 E652                      BSR     FOPEN                   ; open code file
410A| 4E75                          RTS
410C|                       ;
410C|                       ;       APNDOBJ -- appends '.OBJ' to the code file name
410C|                       ;
410C|                       APNDOBJ GETSTRB A0
410C| 206D FFAA             #       MOVE.L   -86(A5),A0
4110| 2248                          MOVE.L  A0,A1
4112| 4240                          CLR.W   D0
4114| 1018                          MOVE.B  (A0)+,D0
4116| D0C0                          ADD.W   D0,A0
4118| 10FC 002E                     MOVE.B  #'.',(A0)+
411C| 10FC 004F                     MOVE.B  #'O',(A0)+
4120| 10FC 0042                     MOVE.B  #'B',(A0)+
4124| 10FC 004A                     MOVE.B  #'J',(A0)+
4128| 5811                          ADD.B   #4,(A1)
412A| 4E75                          RTS
412C|                       ;
412C|                       ;
412C|                       ;
412C|                       INSSTAR GETSTRB A0
40BE* 006E
4094* 0098
412C| 206D FFAA             #       MOVE.L   -86(A5),A0
4130| 4240                          CLR.W   D0
4132| 1018                          MOVE.B  (A0)+,D0
4134| D0C0                          ADD.W   D0,A0
4136| 43E8 0001                     LEA     1(A0),A1
413A| 1320                  @1      MOVE.B  -(A0),-(A1)
413C| 5300                          SUB.B   #1,D0
413E| 66FA                          BNE.S   @1
4140| 133C 002A                     MOVE.B  #'*',-(A1)
4144|                               GETSTRB A0
4144| 206D FFAA             #       MOVE.L   -86(A5),A0
4148| 5210                          ADD.B   #1,(A0)
414A| 4E75                          RTS
414C|                       ;
PAGE - 159  MONITOR   FILE: COMMAND.TEXT


414C|                       ;       LOADOBJ -- loads object file into buffer and verifies header
414C|                       ;
414C|                       ;       first read code file into buffer
414C|                       ;
414C|                       ;       note: the address of the buffer is propagated from here
414C|                       ;
414C|                       LOADOBJ GETIBF  A4                      ; set A4 to buffer start
414C| 286D FFAE             #       MOVE.L   -82(A5),A4
4150| 4287                          MOVE.L  #0,D7                   ; next block to read is 0
4152| 4285                          MOVE.L  #0,D5                   ; set INBYTE to zero
4154| 6100 F5C2                     BSR     FILLBUF                 ; go fill the buffer
4158|                       ;
4158|                       ; verify it is an executable file
4158|                       ;
4158| 6100 F60C                     BSR     NEXT.1                  ; get first byte
415C| 0C40 008F                     CMP.W   #$8F,D0                 ; executable ?
4160| 67**                          BEQ.S   @1
4162|                               GETSYSC A0
4162| 206D FFFC             #       MOVE.L   -4(A5),A0
4166| 10BC 000E                     MOVE.B  #IBADFMT,(A0)           ; set iorslt
4160* 08
416A| 4E75                  @1      RTS
416C|                       ;
416C|                       ;       STRTOBJ
416C|                       ;
416C| 000C                  RLPL    .WORD   12
416E| 52 75 6E 6E 69 6E 67  RPL     .ASCII  'Running ... '
4175| 20 2E 2E 2E 20
```

```
417A|                          ;
417A|                          ; executable file, get size, address, jump table size and commonsize
417A|                          ;
417A| 4287                     STRTOBJ MOVE.L  #0,D7                   ; restore D7 (next block is still 0)
417C| 6100 F5F8                        BSR     NEXT.3                  ; get blocksize
4180| 6100 F606                        BSR     NEXT.4                  ; get address
4184| 2240                            MOVE.L  D0,A1
4186| 2049                            MOVE.L  A1,A0                   ; save address as baseloc in A0
4188| 6100 F5FE                      BSR     NEXT.4                  ; get bytesleft
418C| 2400                            MOVE.L  D0,D2
418E| 6100 F5F8                      BSR     NEXT.4                  ; get commonsize
4192| 2F00                            MOVE.L  D0,-(A7)
4194| 303C 1800                      MOVE.W  #UJMPTBL,D0             ; compute heapbase-jumptablestart
4198| 48C0                            EXT.L   D0
419A| 4480                            NEG.L   D0
419C| D0B8 0138                      ADD.L   $138,D0
41A0| B440                            CMP.W   D0,D2                   ; user jump table too large ?
41A2| 63**                            BLS.S   @1
41A4| 7004                            MOVE.L  #4,D0
41A6| 6000 F654                      BRA     ERROR
41AA|                          ;
41AA|                          ; copy the jump table into memory
41AA|                          ;
41A2* 06
41AA| 6100 F5BA                @1      BSR     NEXT.1                  ; copy next byte of jump table
41AE| 12C0                            MOVE.B  D0,(A1)+
41B0| 5342                            SUB.W   #1,D2
PAGE - 160  MONITOR   FILE: COMMAND.TEXT


41B2| 66F6                            BNE.S   @1
41B4|                          ;
41B4|                          ; startit up
41B4|                          ;
41B4|                          ; A0 contains Baseloc (ie. $$TOP)
41B4|                          ;
41B4| 3410                            MOVE.W  (A0),D2         ; D2 = number of segments
41B6| EB4A                            LSL.W   #5,D2           ;        *32
41B8| 2270 2002                      MOVE.L  2(A0,D2.W),A1   ; A1 = address of $$FIRST-20
41BC| 234C 0004                      MOVE.L  A4,4(A1)        ; Move address of buffer to $$FIRST-16
41C0|                                GETCFIB A2
41C0| 246D FFDC                #     MOVE.L   -36(A5),A2
41C4| 234A 0008                      MOVE.L  A2,8(A1)        ; Move address of objfile to $$FIRST-12
41C8| 201F                            MOVE.L  (A7)+,D0        ; Fetch common size
41CA| 4869 0014                      PEA     20(A1)          ; Push pointer to $$FIRST
41CE| 4267                            MOVE.W  #0,-(A7)        ; Dummy push for Segment Bank Word
41D0| 48E7 1F3E                      MOVEM.L D3-D7/A2-A6,-(SP)
41D4| 48E7 8080                      MOVEM.L D0/A0,-(A7)     ; Save Baseloc and Common Size
41D8| 43FA FF94                      LEA     RPL,A1          ; PL address
41DC| 343A FF8E                      MOVE.W  RLPL,D2         ; and length
41E0| 6100 FCDC                      BSR     PRINT
41E4| 6100 FD2A                      BSR     WCRLF
41E8| 4CDF 0180                      MOVEM.L (A7)+,D7/A0     ; Restore Baseloc and Common Size (into D7)
41EC| 2848                            MOVE.L  A0,A4           ; Setup A4 Jump Table Base Address
41EE|                                GETOFIB -(A7)           ; +12(A5) = @OUTPUT
41EE| 2F2D FFE8                #     MOVE.L   -24(A5),-(A7)
41F2|                                GETIFIB -(A7)           ;  +8(A5) = @INPUT
41F2| 2F2D FFEC                #     MOVE.L   -20(A5),-(A7)
41F6| 4BEF FFF8                      LEA     -8(SP),A5       ; Set A5 so 16(A5) = P-regs.
41FA| 2068 0002                      MOVE.L  2(A0),A0        ; A0 = Main Program Entry - 4
41FE| 4EA8 0004                      JSR     4(A0)           ; Start her up!
4202| 508F                            ADDQ.L  #8,SP           ; Restore the registers
4204| 4CDF 7CF8                      MOVEM.L (SP)+,D3-D7/A2-A6
4208| 4A5F                            TST.W   (SP)+           ; Throw away dummy Segment Bank Word
420A| 588F                            ADDQ.L  #4,SP           ; Eliminate pointer to $$FIRST
420C| 4E75                            RTS
420E|                          ;
420E|                          ;  Main prompt loop starts here
420E|                          ;
420E| 6100 FA84                RESTART BSR     GETREGS                 ; restore A5, SP and NP
4212| 6100 FBA4                        BSR     HOMCRSR
4216| 6100 FBC4                        BSR     CLRSCR
421A| 6100 FA78                CMDLOOP BSR     GETREGS                 ; restore buf ptrs, A5, SP and NP
421E| 6100 FA22                        BSR     RSTRXCP
4222| 6100 FA48                        BSR     INITPRG
4226| 6100 FC86                        BSR     PROMPT
422A|                          CMDLOOP2
422A| 6100 FCA4                        BSR     GETCHAR                 ; returns char on tos
422E| 101F                            MOVE.B  (A7)+,D0
4230| 6100 FB34                        BSR     UPSHFT
4234| 0C00 0058                      CMPI.B  #'X',D0                 ; X(ecute ?
4238| 6600 ****                      BNE     TRYDBUG
423C|                                DBUGCLR
423C| 422D FFA9                #     MOVE.B   #0,-87(A5)
4240| 6100 FCE8                      BSR     XEQUTE                  ; get file to execute
PAGE - 161  MONITOR   FILE: COMMAND.TEXT


4244|                                GETSYSC A0
4244| 206D FFFC                #     MOVE.L   -4(A5),A0
4248| 4A10                            TST.B   (A0)                    ; test iorslt
424A| 66C2                            BNE     RESTART
424C| 323C 0001                      MOVE.W  #1,D1
4250| 6100 FDF0                      BSR     TRYEXEC                 ; insist on EXEC/
4254| 4A40                            TST.W   D0
4256| 66C2                            BNE     CMDLOOP
```

```
4258| 6100 FE90              TRYXEQT BSR     OPNCODE                 ; open object file
425C|                               GETSYSC A0
425C| 206D FFFC            #         MOVE.L  -4(A5),A0
4260| 4A10                          TST.B   (A0)                    ; test iorslt
4262| 67**                          BEQ.S   FOPENOK
4264|                               GETSTRB A0
4264| 206D FFAA            #         MOVE.L  -86(A5),A0
4268| 4A10                          TST.B   (A0)                    ; long string ?
426A| 6BA2                          BMI.S   RESTART
426C| 6100 FEBE                     BSR     INSSTAR
4270| 6100 FE78                     BSR     OPNCODE
4274|                               GETSYSC A0
4274| 206D FFFC            #         MOVE.L  -4(A5),A0
4278| 4A10                          TST.B   (A0)
427A| 67**                          BEQ.S   FOPENOK
427C| 7201                          MOVE.L  #1,D1
427E|                               GETSTRB A0
427E| 206D FFAA            #         MOVE.L  -86(A5),A0
4282| 6100 DB76                     BSR     DEL1CH
4286| 6100 FE84                     BSR     APNDOBJ                 ; no, then append '.OBJ'
428A| 6100 FE5E                     BSR     OPNCODE                 ; try to open again
428E|                               GETSYSC A0
428E| 206D FFFC            #         MOVE.L  -4(A5),A0
4292| 4A10                          TST.B   (A0)                    ; test iorslt
4294| 67**                          BEQ.S   FOPENOK
4296| 6100 FE94                     BSR     INSSTAR
429A| 6100 FE4E                     BSR     OPNCODE
429E|                               GETSYSC A0
429E| 206D FFFC            #         MOVE.L  -4(A5),A0
42A2| 4A10                          TST.B   (A0)                    ; test iorslt
42A4| 67**                          BEQ.S   FOPENOK
42A6| 7201                          MOVE.L  #1,D1
42A8|                               GETSTRB A0
42A8| 206D FFAA            #         MOVE.L  -86(A5),A0
42AC| 6100 DB4C                     BSR     DEL1CH
42B0|                               GETSTRB A0
42B0| 206D FFAA            #         MOVE.L  -86(A5),A0
42B4| 5910                          SUB.B   #4,(A0)                 ; delete '.OBJ'
42B6| 4241                          CLR.W   D1
42B8| 6100 FD88                     BSR     TRYEXEC                 ; don't insist on EXEC/
42BC| 4A40                          TST.W   D0
42BE| 6600 FF5A                     BNE     CMDLOOP
42C2| 4AB8 0154           CALLLDR   TST.L   $154                    ; check to see if loader is in memory
42C6| 67**                          BEQ.S   NOLOADR
42C8| 2078 0154                     MOVE.L  $154,A0
42CC|                               GETSTRB -(A7)                   ; @title
42CC| 2F2D FFAA            #         MOVE.L  -86(A5),-(A7)
PAGE - 162  MONITOR   FILE: COMMAND.TEXT


42D0|                               GETOFIB -(A7)                   ; @OUTPUT
42D0| 2F2D FFE8            #         MOVE.L  -24(A5),-(A7)
42D4|                               GETIFIB -(A7)                   ; @INPUT
42D4| 2F2D FFEC            #         MOVE.L  -20(A5),-(A7)
42D8| 4E90                          JSR     (A0)                    ; call the intrinsic unit loader
42DA| 6000 FF3E                     BRA     CMDLOOP
42C6* 16
42DE| 6100 FA1C           NOLOADR   BSR     IOCHK
42A4* 3C
4294* 4C
427A* 66
4262* 7E
42E2| 6100 FE68           FOPENOK   BSR     LOADOBJ                 ; load the object file
42E6|                               GETSYSC A0
42E6| 206D FFFC            #         MOVE.L  -4(A5),A0
42EA| 4A10                          TST.B   (A0)                    ; test iorslt
42EC| 66D4                          BNE.S   CALLLDR                 ; if ioresult<>0 then try iu loader
42EE| 6100 FE8A                     BSR     STRTOBJ                 ; start it
42F2| 6000 FF26                     BRA     CMDLOOP
423A* 00BC
42F6| 0C00 004D           TRYDBUG   CMPI.B  #'M',D0                 ; Macsbug ?
42FA| 66**                          BNE.S   TRYQ
42FC|                               DBUGSET
42FC| 1B7C 0001 FFA9      #         MOVE.B  #1,-87(A5)
4302| 6100 FC26                     BSR     XEQUTE                  ; get file to debug
4306|                               GETSYSC A0
4306| 206D FFFC            #         MOVE.L  -4(A5),A0
430A| 4A10                          TST.B   (A0)                    ; file name present ?
430C| 6700 FF4A                     BEQ     TRYXEQT                 ; yes, try to execute the file
4310| 41FA FEFC                     LEA     RESTART,A0              ; no, then fake red button
4314| 2F08                          MOVE.L  A0,-(A7)
4316| 40E7                 FAKERED  MOVE    SR,-(A7)
4318| 4E4D                          TRAP    #$D
42FA* 1E
431A| 0C00 003F           TRYQ      CMPI.B  #'?',D0
431E| 66**                          BNE.S   ACEFL
4320| 6100 F972                     BSR     GETREGS
4324| 6100 F91C                     BSR     RSTRXCP
4328| 6100 F942                     BSR     INITPRG
432C| 6100 FB14                     BSR     PROMPT2
4330| 6000 FEF8                     BRA     CMDLOOP2
431E* 14
4334| 0C00 0041           ACEFL     CMPI.B  #'A',D0                 ; A(ssemble ?
4338| 66**                          BNE.S   @1
433A| 4267                          MOVE.W  #0*24,-(A7)
433C| 60**                          BRA.S   SYSPROG
```

```
4338* 04
433E| 0C00 0043              @1      CMPI.B  #'C',D0                 ; C(ompile ?
4342| 66**                            BNE.S   @2
4344| 3F3C 0018                      MOVE.W  #1*24,-(A7)
4348| 60**                            BRA.S   SYSPROG
4342* 06
434A| 0C00 0045              @2      CMPI.B  #'E',D0                 ; E(dit ?
434E| 66**                            BNE.S   @3
4350| 3F3C 0030                      MOVE.W  #2*24,-(A7)
PAGE - 163  MONITOR   FILE: COMMAND.TEXT


4354| 60**                            BRA.S   SYSPROG
434E* 06
4356| 0C00 0046              @3      CMPI.B  #'F',D0                 ; F(ile ?
435A| 66**                            BNE.S   @4
435C| 3F3C 0048                      MOVE.W  #3*24,-(A7)
4360| 60**                            BRA.S   SYSPROG
435A* 06
4362| 0C00 004C              @4      CMPI.B  #'L',D0                 ; L(ink ?
4366| 66**                            BNE.S   @5
4368| 3F3C 0060                      MOVE.W  #4*24,-(A7)
436C| 60**                            BRA.S   SYSPROG
4366* 06
436E| 0C00 0044              @5      CMPI.B  #'D',D0                 ; D(ebug ?
4372| 66**                            BNE.S   @6
4374| 3F3C 0078                      MOVE.W  #5*24,-(A7)
4378| 60**                            BRA.S   SYSPROG
4372* 06
437A| 0C00 0055              @6      CMPI.B  #'U',D0                 ; U(csdeditor ?
437E| 66**                            BNE.S   @7
4380| 3F3C 0090                      MOVE.W  #6*24,-(A7)
4384| 60**                            BRA.S   SYSPROG
437E* 06
4386| 0C00 0047              @7      CMPI.B  #'G',D0                 ; G(enerate Code ?
438A| 66**                            BNE.S   @8
438C| 3F3C 00A8                      MOVE.W  #7*24,-(A7)
4390| 60**                            BRA.S   SYSPROG
438A* 06
4392| 0C00 0053              @8      CMPI.B  #'S',D0                 ; S(ysmgr ?
4396| 6600 FE76                      BNE.S   RESTART
439A| 3F3C 00C0                      MOVE.W  #8*24,-(A7)
439E|                        SYSPROG DBUGCLR
4390* 0C
4384* 18
4378* 24
436C* 30
4360* 3C
4354* 48
4348* 54
433C* 60
439E| 422D FFA9              #       MOVE.B   #0,-87(A5)
43A2| 6100 FA14                      BSR     HOMCRSR
43A6| 6100 FA34                      BSR     CLRSCR
43AA| 6100 FB64                      BSR     WCRLF
43AE| 301F                          MOVE.W  (A7)+,D0
43B0|                               GETSTBL A0
43B0| 206D FFB6              #       MOVE.L   -74(A5),A0
43B4| 41F0 0000                      LEA     0(A0,D0),A0             ; @filename
43B8| 2F08                          MOVE.L  A0,-(A7)                ; save it
43BA|                        ;
43BA|                        ; first try to find the system file on #4:
43BA|                        ;
43BA|                               GETSTRB A1
43BA| 226D FFAA              #       MOVE.L   -86(A5),A1
43BE| 7006                          MOVE.L  #6,D0
43C0| 22D8                    @1      MOVE.L  (A0)+,(A1)+             ; copy filename to strbuf
PAGE - 164  MONITOR   FILE: COMMAND.TEXT


43C2| 5340                          SUB.W   #1,D0
43C4| 66FA                          BNE.S   @1
43C6|                               GETSTRB A0
43C6| 206D FFAA              #       MOVE.L   -86(A5),A0
43CA| 0C28 002A 0001                CMPI.B  #42,1(A0)               ; if filename[1]='*' then
43D0| 66**                          BNE.S   @4
43D2| 2257                          MOVE.L  (A7),A1
43D4| 1019                          MOVE.B  (A1)+,D0
43D6| 1080                          MOVE.B  D0,(A0)
43D8| 5418                          ADD.B   #2,(A0)+
43DA| 10FC 0023                    MOVE.B  #35,(A0)+               ;   insert('#4:',filename,1)
43DE| 10FC 0034                    MOVE.B  #'4',(A0)+
43E2| 10FC 003A                    MOVE.B  #58,(A0)+
43E6| 4A19                          TST.B   (A1)+                   ;   skip over '*'
43E8| 10D9                    @3      MOVE.B  (A1)+,(A0)+             ;   and copy filename
43EA| 5300                          SUB.B   #1,D0
43EC| 66FA                          BNE.S   @3
43EE| 6000 ****                    BRA     FNDSYS2
43D0* 20
43F2| 0C28 003A 0003        @4      CMPI.B  #58,3(A0)               ; if filename[3]=':' then
43F8| 66**                          BNE.S   @5
43FA| 117C 0034 0002                MOVE.B  #'4',2(A0)              ;   filename[2]:='4'
4400| 6000 ****                    BRA     FNDSYS2
43F8* 0A
4404| 0C28 003A 0004        @5      CMPI.B  #58,4(A0)               ; if filename[4]=':' then
440A| 66**                          BNE.S   FNDSYS3
```

```
440C| 117C 0030 0002              MOVE.B  #'0',2(A0)              ;   filename[2]:='0'
4412| 117C 0034 0003              MOVE.B  #'4',3(A0)              ;   filename[3]:='4'
4402* 0016
43F0* 0028
4418| 6100 FCD0           FNDSYS2 BSR     OPNCODE                 ; open object file
441C|                            GETSYSC A0
441C| 206D FFFC           #       MOVE.L  -4(A5),A0
4420| 4A10                        TST.B   (A0)                    ; test iorslt, ok ?
4422| 66**                        BNE.S   FNDSYS3
4424| 4A9F                        TST.L   (A7)+                   ; yes discard @filename
4426| 6000 FEBA                   BRA     FOPENOK
4422* 06
440A* 1E
442A| 2057                FNDSYS3 MOVE.L  (A7),A0
442C|                       ;
442C|                       ; enter here with @filename in A0 and a copy on tos
442C|                       ;
442C|                       FINDSYS GETSTRB A1
442C| 226D FFAA           #       MOVE.L  -86(A5),A1
4430| 7006                        MOVE.L  #6,D0
4432| 22D8                 @1     MOVE.L  (A0)+,(A1)+             ; copy filename to strbuf
4434| 5340                        SUB.W   #1,D0
4436| 66FA                        BNE.S   @1
4438| 6100 FCB0                   BSR     OPNCODE                 ; open object file
443C|                            GETSYSC A0
443C| 206D FFFC           #       MOVE.L  -4(A5),A0
4440| 4A10                        TST.B   (A0)                    ; test iorslt, ok ?
4442| 66**                        BNE.S   @2
4444| 4A9F                        TST.L   (A7)+                   ; yes discard @filename
PAGE - 165  MONITOR   FILE: COMMAND.TEXT


4446| 6000 FE9A                   BRA     FOPENOK
4442* 06
444A| 2057                 @2     MOVE.L  (A7),A0
444C| 0C28 002A 0001              CMPI.B  #42,1(A0)               ; if filename[1]='*' then
4452| 66**                        BNE.S   @4
4454|                            GETSTRB A1
4454| 226D FFAA           #       MOVE.L  -86(A5),A1
4458| 1019                        MOVE.B  (A1)+,D0
445A| 1080                        MOVE.B  D0,(A0)
445C| 5418                        ADD.B   #2,(A0)+
445E| 10FC 0023                   MOVE.B  #35,(A0)+               ;   insert('#4:',filename,1)
4462| 10FC 0034                   MOVE.B  #'4',(A0)+
4466| 10FC 003A                   MOVE.B  #58,(A0)+
446A| 4A19                        TST.B   (A1)+                   ;   skip over '*'
446C| 10D9                 @3     MOVE.B  (A1)+,(A0)+             ;   and copy filename
446E| 5300                        SUB.B   #1,D0
4470| 66FA                        BNE.S   @3
4472| 2057                        MOVE.L  (A7),A0
4474| 60B6                        BRA     FINDSYS
4452* 22
4476| 0C28 0034 0002         @4   CMPI.B  #'4',2(A0)              ; if filename[2]='4' then
447C| 66**                        BNE.S   @5
447E| 117C 0035 0002              MOVE.B  #'5',2(A0)              ;   filename[2]:='5'
4484| 60A6                        BRA     FINDSYS
447C* 08
4486| 0C28 0035 0002         @5   CMPI.B  #'5',2(A0)              ; if filename[2]='5' then
448C| 66**                        BNE.S   @6
448E| 117C 0039 0002              MOVE.B  #'9',2(A0)              ;   filename[2]:='9'
4494| 6096                        BRA     FINDSYS
448C* 08
4496| 0C28 0039 0002         @6   CMPI.B  #'9',2(A0)              ; if filename[2]='9' then
449C| 66**                        BNE.S   @8
449E|                            GETSTRB A1
449E| 226D FFAA           #       MOVE.L  -86(A5),A1
44A2| 137C 0031 0002              MOVE.B  #'1',2(A1)              ;   filename[2]:='1'
44A8| 1019                        MOVE.B  (A1)+,D0
44AA| 1080                        MOVE.B  D0,(A0)
44AC| 5218                        ADD.B   #1,(A0)+
44AE| 10D9                        MOVE.B  (A1)+,(A0)+
44B0| 10D9                        MOVE.B  (A1)+,(A0)+
44B2| 10FC 0030                   MOVE.B  #'0',(A0)+              ;   insert('0',filename,3)
44B6| 10D9                 @7     MOVE.B  (A1)+,(A0)+
44B8| 5300                        SUB.B   #1,D0
44BA| 66FA                        BNE.S   @7
44BC| 2057                        MOVE.L  (A7),A0
44BE| 6000 FF6C                   BRA     FINDSYS
449C* 24
44C2| 0C28 0030 0003         @8   CMPI.B  #'0',3(A0)              ; if filename[3]='0' then
44C8| 66**                        BNE.S   @9
44CA| 117C 0031 0003              MOVE.B  #'1',3(A0)              ;   filename[3]:='1'
44D0| 6000 FF5A                   BRA     FINDSYS
44C8* 0A
44D4| 0C28 0031 0003         @9   CMPI.B  #'1',3(A0)              ; if filename[3]='1' then
44DA| 66**                        BNE.S   @10
44DC| 117C 0032 0003              MOVE.B  #'2',3(A0)              ;   filename[2]:='2'
PAGE - 166  MONITOR   FILE: COMMAND.TEXT


44E2| 6000 FF48                   BRA     FINDSYS
44DA* 0A
44E6| 2057                 @10    MOVE.L  (A7),A0
44E8| 0C28 003A 0001              CMPI.B  #58,1(A0)               ; while filename[1]<>':' do
44EE| 67**                        BEQ.S   @11
44F0| 7201                        MOVE.L  #1,D1
```

```
44F2| 6100 D906                    BSR     DEL1CH                  ;   delete(ftitle,1,1)
44F6| 60EE                        BRA.S   @10
44EE* 08
44F8| 117C 002A 0001        @11     MOVE.B  #42,1(A0)               ; filename[1]:='*'
44FE| 4A9F                        TST.L   (A7)+                   ; delete @filename
4500| 487A FD0C                    PEA     RESTART
4504| 6000 F7F6                    BRA     IOCHK
4508|                         ;
4508|                         ; Handler for Trap #D
4508|                         ;
4508| 301F                    TRP2MAX MOVE.W  (A7)+,D0                ; get SR
450A| 4A9F                        TST.L   (A7)+                   ; delete PC
450C| 0800 000D                  BTST    #13,D0                  ; trap from user mode ?
4510| 66**                        BNE.S   GOTOABT
4512| 5D8F                        SUB.L   #6,A7                   ; make space to copy SR and PC
4514| 4E68                        MOVE    USP,A0
4516| 3E98                        MOVE.W  (A0)+,(A7)              ; copy SR
4518| 2F58 0002                  MOVE.L  (A0)+,2(A7)             ; copy PC
451C| 4E60                        MOVE    A0,USP
4510* 0C
451E| 4A38 014C              GOTOABT TST.B   $14C
4522| 67**                        BEQ.S   USELVL7                 ; if little machine then
4524| 2078 0078                  MOVE.L  $78,A0                  ;   goto ABORTB in MACSBUG via Level 6
4528| 4ED0                        JMP     (A0)                    ; else
4522* 06
452A| 2078 007C              USELVL7 MOVE.L  $7C,A0                  ;   goto ABORTB in MACSBUG via Level 7
452E| 4ED0                        JMP     (A0)
4530|                         ;
4530| 6000 FCDC                    BRA     RESTART                 ; Last instruction of monitor
4534|                         ;
4534| 0000                    THEEND  .WORD   0
4536|
4536|
4536|                         ;
4536|                                 .END
PAGE - 167  MONITOR   FILE:MONITOR.TEXT  SYMBOLTABLE DUMP


AB - Absolute     LB - Label     UD - Undefined     MC - Macro
RF - Ref          DF - Def       PR - Proc          FC - Func
PB - Public       PV - Private   CS - Consts


A0TOS0   LB 00003A5E|  ACEFL    LB 00004334|  ACR      AB 00000058|  ACTNTBL  LB 00001A46|  ADDR2S0  LB 00003A8A
ADDRDRVR LB 000019EA|  ADDRFLR  MC --------|  ADDRNP   MC --------|  ADDRSAB  MC --------|  ADDRSAT  MC --------
ADDRTHD  MC --------|  ALL.85   LB 00003668|  ALLBIT  AB 0000000F|  ALPHLCK  AB 00000028|  ALTMOD   AB 00000024
APNDOBJ  LB 0000410C|  APNDTXT  LB 0000401E|  BADBLCK LB 00003600|  BADCHR   AB 00000022|  BLKH     AB FFFFFFFB
BLKIO    LB 00002E10|  BLKL     AB FFFFFFFD|  BLKM   AB FFFFFFFC|  BLKRXIT  LB 00002F24|  BLKSIZE  AB 000000FF
BLOCK80  LB 00003614|  BLOCK81  LB 00003698|  BLOCK82 LB 00003614|  BLOCK83  LB 00003614|  BLOCK84  LB 00003614
BLOCK85  LB 00003630|  BLOCK86  LB 0000367A|  BLOCKH  AB FFFFFFFF|  BLOCKL   AB FFFFFFFE|  BOOTDVC  AB 000001B3
BOTSYSC  LB 00000030|  BOTSYSF  LB 00000208|  BOTUTBL LB 0000012E|  BREAK    AB 00000021|  BREINI2  LB 00000BAC
BREINIT  LB 00000B62|  BRVOLSX  LB 00002164|  BSRPUTB LB 00003A00|  BSSPBS   LB 00003F26|  BSYBIT   AB 00000008
C.TABLE  LB 00003606|  CALLLDR  LB 000042C2|  CANTST1 LB 00002D5C|  CANTSTR  LB 00002C40|  CANTSTX  LB 00002D6E
CCRUNCH  AB 00000003|  CDBUSY   LB 00001720|  CDCLEAR LB 00001726|  CDCPT    AB 00000004|  CDERROR  LB 00001714
CDEVVOL  LB 0000234C|  CDOVRUN  AB 00000001|  CDREAD  LB 0000172A|  CDRWXIT  LB 000017B0|  CDSKCSZ  LB 00000E94
CDSKINIT LB 00000E6E|  CDSKRD   LB 00000F86|  CDSKREAD LB 00000E84|  CDSKWR   LB 00001018|  CDSKWRT  LB 00000E8C
CDWRITE  LB 000017E8|  CHECKCD  LB 0000169C|  CHKBLK7 LB 00000AFA|  CHKDEV   LB 00002168|  CHKERR   LB 000008D8
CHKMTBL  LB 00000B64|  CHRDEL   AB 00000023|  CJMPTBL LB 00000E5E|  CLOCK    AB 00000001|  CLRBIT   AB 00000004
CLRLINE  AB 00000013|  CLRLN    LB 00003DC4|  CLRLN2  LB 00003DD4|  CLRSCR   LB 00003DDC|  CLRSCR2  LB 00003DEC
CLRSCRN  AB 00000012|  CMDLOOP  LB 0000421A|  CMDLOOP2 LB 0000422A|  CMDSIZE  AB 00000004|  CMPSFX   LB 00001E14
CMPSFXX  LB 00001E20|  CNORMAL  AB 00000000|  CODEFILE AB 00000002|  CONSOLE  AB 00000001|  COPY6    LB 00001178
COPY6LP  LB 00001186|  CPURGE   AB 00000002|  CPYDEVE  LB 00000BD0|  CPYDEVN  LB 00000BCA|  CPYLOOP  LB 0000061A
CRLF     LB 00003F0E|  CRTESC   AB 0000000A|  CRTINFO  AB 00000016|  CRTTYP   AB 00000008|  CSLASH   LB 000020EE
CSZEXIT  LB 00000E7E|  DACCESS  AB 00000018|  DATAFILE AB 00000005|  DBUGCLR  MC --------|  DBUGSET  MC --------
DBUGTST  MC --------|  DDRA     AB 00000018|  DDRB     AB 00000010|  DECDTOW  MC --------|  DECRTOW  LB 00000998
DEL1CH   LB 00001DFA|  DELDEV   LB 00002182|  DELENG   AB 0000001A|  DELNTRY  LB 000025C0|  DEOVBLK  AB 0000000E
DEVBASE  AB 00000010|  DEVGLBS  AB 0000001C|  DEVJMPT  LB 00000558|  DEVJTBL  AB 00000018|  DEVLOOP  LB 0000054E
DEVMTBL  AB 00000014|  DEVNAME  AB 00000000|  DEVNSHF  AB 00000005|  DEVNUM0  LB 00000578|  DEVNUM1  LB 00000588
DEVNUM2  LB 00000598|  DEVNUM3  LB 000005A2|  DEVNUM4  LB 000005B2|  DEVNUM5  LB 000005CC|  DEVNUM6  LB 000005E6
DEVNUM7  LB 00000600|  DEVNUMB  LB 0000219E|  DIRDISK  AB 00000002|  DIRSRCH  LB 000023B6|  DIRSRCX  LB 00002422
DIRSRLP  LB 000023D2|  DIRSZ    AB 00000800|  DISKERR  LB 00000E34|  DISKIO   LB 00000EBE|  DISKIOX  LB 00000F7E
DLASTBT  AB 00000014|  DLOADTM  AB 00000012|  DNTRYSZ  AB 00000020|  DNUMFLS  AB 00000010|  DOCOPY   LB 00000618
DOSFX    LB 00001D34|  DOSUFIX  LB 00001D1E|  DRVR     LB 0000193C|  DRVRTBL  LB 00001838|  DSKBLK   AB 00000200
DSKERR   LB 000012C6|  DSKRD1   LB 00000FBC|  DSKRD2   LB 00000FCC|  DSKRD3   LB 00000FD0|  DSKRD4   LB 00000FE0
DSKRD5   LB 00000FF6|  DSKRD6   LB 00001000|  DSKREAD  LB 00000E28|  DSKWRT   LB 00000E40|  DSKWRT2  LB 0000104E
DSKWRT3  LB 0000105C|  DSKWRT4  LB 0000106C|  DSKWRT5  LB 00001086|  DSKWRT6  LB 0000108A|  DSKWRT7  LB 0000109A
DTBLSIZ  AB 00000100|  DTID     AB 00000006|  DTRQST_  AB 00000007|  DVID     AB 00000006|  ENDWH85  LB 00003676
ENTRTMP  LB 00002616|  EOF      AB 0000001F|  ERROR    LB 000037FC|  ETX      AB 00000026|  EXECDRVR LB 00001666
FAIRTTL  LB 00001BC8|  FAKERED  LB 00004316|  FBFCHNG  AB 0000003E|  FBLKSIZ  AB 00000200|  FBUFFER  AB 00000050
FCLOSE   LB 00002A08|  FCLOSE1  LB 00002C22|  FCLOSEX  LB 00002C32|  FDEVVID  AB 00000040|  FEOF     AB 00000005
FEOLN    AB 00000004|  FGET     LB 0000304E|  FGET1L   LB 00003272|  FGET2L   LB 00003282|  FGET2ND  LB 00003126
FGET3RD  LB 000031D4|  FGETNOP  LB 0000326A|  FHEADER  AB 0000001E|  FIBSIZE  AB 00000250|  FIBSZ    AB 00000050
FIL1STR  LB 000003EC|  FIL2STR  LB 000003F6|  FILCNT   AB 00000011|  FILLBUF  LB 00003718|  FINDBP   LB 00003900
FINDD2   LB 0000112A|  FINDLP   LB 00003A70|  FINDMAX  LB 000025F6|  FINDSYS  LB 0000442C|  FINISH   LB 000012CE
FINIT    LB 00001B0C|  FINITX   LB 00001B5C|  FISBLKD  AB 0000000B|  FISOPEN  AB 0000000A|  FIXFVID  LB 0000232E
FKIND    AB 00000004|  FLUSH    AB 0000001E|  FMACHIN  AB 0000000E|  FMAXBLK  AB 00000016|  FMAXBYT  AB 0000003A
FMODIFD  AB 0000001C|  FNDSEGN  LB 0000390C|  FNDSEGO  LB 00003914|  FNDSYS2  LB 00004418|  FNDSYS3  LB 0000442A
FNXTBLK  AB 00000018|  FNXTBYT  AB 0000003C|  FOPEN    LB 0000275A|  FOPEN1   LB 000029E0|  FOPEN2   LB 00002952
FOPENOK  LB 000042E2|  FOPENX   LB 000029F8|  FOTOFILE AB 00000007|  FPUT     LB 0000328A|  FPUT1L   LB 000033F6
FPUT2L   LB 00003406|  FPUTNOP  LB 000033EE|  FREADCHR LB 00002F88|  FREADLN  LB 00002FBE|  FRECSZ   AB 00000008
FREEBYT  AB 00000008|  FREEPROC LB 00001E64|  FREPCNT  AB 0000001A|  FRESET   LB 0000242C|  FRESETX  LB 00002464
FROMEXEC LB 00003F94|  FSEEK    LB 00003422|  FSEEK1   LB 00003532|  FSOFTBF  AB 00000038|  FSTATE   AB 00000006
FSTBLK   AB 00000000|  FTBLSZ   AB 000000D8|  FTCHDIR  LB 00001EA2|  FTCHDRX  LB 000020A6|  FUNIT    AB 0000000C
PAGE - 168  MONITOR   FILE:MONITOR.TEXT  SYMBOLTABLE DUMP
```

```
FWINDOW  AB 00000000|  FWRITELN LB 00002F6E|  FWRTCHAR LB 00002F2E|  FWRTCHX  LB 00002F68|  GET      LB 0000185C
GETACCA  MC --------|  GETBASE  LB 00000972|  GETCFIB  MC --------|  GETCHAR  LB 00003ED0|  GETCHRX  LB 00003F0C
GETDKV   MC --------|  GETDREG  MC --------|  GETDTBL  MC --------|  GETEBST  MC --------|  GETEMTH  MC --------
GETERRP  MC --------|  GETIBF   MC --------|  GETIFIB  MC --------|  GETINDX  LB 00000834|  GETINFO  LB 000016B0
GETJTBL  LB 00000E02|  GETLOGN  MC --------|  GETMTBL  LB 00000956|  GETNP   MC --------|  GETNXCH  MC --------
GETOFIB  MC --------|  GETPRBA  MC --------|  GETPRNT  MC --------|  GETRDIR  LB 00000CD2|  GETREGS  LB 00003C94
GETRSLT  LB 00001878|  GETSFIB  MC --------|  GETSPTR  MC --------|  GETSTAT  LB 000011B8|  GETSTBL  MC --------
GETSTRB  MC --------|  GETSYSC  MC --------|  GETSYV   MC --------|  GETTFIB  MC --------|  GETUFIB  MC --------
GETUFST  MC --------|  GETUJPT  MC --------|  GETUNIT  LB 0000195C|  GETUREG  MC --------|  GETUT17  MC --------
GETUTBL  MC --------|  GETXFIB  MC --------|  GETZFST  MC --------|  GO2USER  LB 000036EE|  GOODPC   LB 000038DC
GOTOABT  LB 0000451E|  GRAFFILE AB 00000006|  HANDLER  LB 00000826|  HDRLOOP  LB 00001338|  HDRSIZE  AB 00000009
HDSKCSZ  LB 00000E4E|  HDSKINIT LB 00000E18|  HITIME   AB 00000002|  HLPL     LB 000008A4|  HOMCRSR  LB 00003DB8
HPL      LB 000008A6|  IBADBLK  AB 00000001|  IBADFMT  AB 0000000E|  IBADMOD  AB 00000003|  IBADTTL  AB 00000007
IBADUNT  AB 00000002|  IDFIELD  AB 000000AB|  IDUPFIL  AB 0000000B|  IER      AB 00000070|  IFR      AB 00000068
ILPL     LB 00003CD4|  ILSTFIL  AB 00000006|  ILSTUNT  AB 00000005|  IMTBUSY  AB 00000005|  INBIT    AB 00000001
INBUFSZ  AB 00000800|  INCSR    AB 00000004|  INDATA   AB 00000000|  INDXERO  LB 000008B2|  INDXERR  LB 000008B4
INFOFILE AB 00000004|  INITDEV  LB 0000052E|  INITFLS  LB 00000400|  INITIAL  LB 0000020A|  INITMI   LB 00000476
INITMIL  LB 000004D0|  INITMIX  LB 000004D6|  INITPRG  LB 00003C6C|  INITSYS  LB 0000023C|  INITSYSF LB 0000036E
INITUTBL LB 0000035C|  INITXIT  LB 00000E7C|  INOERR   AB 00000000|  INOFILE  AB 0000000A|  INOROOM  AB 00000008
INOTCLS  AB 0000000C|  INOTOPN  AB 0000000D|  INOUNIT  AB 00000009|  INSNTRY  LB 0000256C|  INSSTAR  LB 0000412C
INTRLV   LB 0000110C|  IOCHK    LB 00003CFC|  IOCHKX   LB 00003D62|  IOCMD    AB FFFFFFFC|  IODRV    AB FFFFFFFD
IOEXIT   LB 000019D0|  IORSLT   AB 00000000|  IPL      LB 00003CD6|  IRA      AB 00000008|  IRB      AB 00000000
ISTROVF  AB 0000000F|  IT.FITS  LB 000035B4|  ITIMOUT  AB 00000004|  JMPTBL  LB 0000076E|  L0.85    LB 00003654
L1.85    LB 00003670|  LASTSEG  LB 00003584|  LAUNCH   LB 00003B10|  LDPARMS  LB 0000340A|  LDSKRD   LB 00001224
LDSKWR   LB 0000137E|  LELPL    LB 000037EC|  LEPL     LB 000037EE|  LINDEL   AB 00000025|  LNFVID0  LB 00002282
LOADOBJ  LB 0000414C|  LOGENSZ  AB 00000004|  LOGNSZ   AB 00000020|  LOOPEND  LB 00000BB0|  LOOPIO   LB 00002D7C
LOOPTBL  LB 00000B7C|  LOOPVOL  LB 00000B1C|  LOTIME   AB 00000004|  LPL      LB 00003E5C|  LPL2     LB 00003DF4
LSLASH   LB 000020DC|  LSTBLK   AB 00000002|  LSTBYTE  AB 00000016|  MAINLOOP LB 00003D5C|  MAXDEV   AB 00000007
MAXDIR   AB 0000004D|  MAXDIRB  AB 00000312|  MAXINDX  AB 000000B8|  MAXU     AB 00000014|  MDSKCSZ  LB 00001448
MDSKINIT LB 00001426|  MDSKRD   LB 00001452|  MDSKREAD LB 00001438|  MDSKRES  LB 0000140A|  MDSKWR   LB 0000155C
MDSKWRT  LB 00001440|  MEMA     LB 00001A26|  MEMREAD  LB 000017B4|  MEMWRITE LB 000017E2|  MGOTOXY  LB 000008F8
MHALT    LB 00000850|  MIOERR   LB 000008F4|  MISCBUF  AB 00000250|  MISCFIB  AB 00000050|  MJMPTBL  LB 000013FA
MMRK     LB 00001A16|  MNEW     LB 00001A02|  MONBASE  LB 00000000|  MONITOR  PR --------|  MOVEOP   LB 0000034A
MOVEMEM  LB 000017B8|  MOVFAST  LB 000017CC|  MOVSLOW  LB 000017BE|  MRKRES   AB 00FC7001|  MRLS     LB 00001A1E
MSCNFO   AB 00000006|  MTBLSIZ   AB 00000054|  MTBLSZ5  AB 0000024C|  MVMTABL  LB 00000C8E|  NDSKRD   LB 00001222
NDSKREAD LB 000010F2|  NDSKWR   LB 0000137C|  NDSKWRT  LB 000010FA|  NEWTMP1  LB 0000098E|  NEXT.1   LB 00003766
NEXT.3   LB 00003776|  NEXT.4   LB 00003788|  NEXTBUF  LB 0000375A|  NEXTSEG  LB 00003A62|  NHS      AB 00000078
NIL      AB 00000000|  NJMPTBL  LB 000010BE|  NOAPPL2  LB 0000186C|  NOAPPLE  LB 00001866|  NOCARD   LB 00000CC8
NOLOADR  LB 000042DE|  NOT44    LB 00003A08|  NOT60HZ  LB 000009A6|  NOTDISK  LB 00001706|  NOTOKLH  LB 000022BE
NOTSTAR  LB 00001BEE|  NOTWGDR  LB 0000039A|  OFSMISC  LB 0000000A|  OK.1     LB 0000376C|  OPNCODE  LB 000040EA
OPNEXEC  LB 00003FFC|  ORA      AB 00000008|  ORB      AB 00000000|  OUTBIT   AB 00000002|  OUTCSR   AB 00000006
OUTDATA  AB 00000002|  OVFCHK   LB 00000882|  OVFCHKX  LB 0000089E|  P.LOOP   LB 000036AA|  PATCHBP  LB 000036D4
PATCHIT  LB 0000369C|  PCMND    AB FFFFFFFA|  PCMNDSZ  AB 00000005|  PCR      AB 00000060|  PDSKCSZ  LB 00001102
PDSKINIT LB 000010CE|  PDSKRD   LB 0000121C|  PDSKREAD LB 000010E2|  PDSKWR   LB 00001376|  PDSKWRT  LB 000010EA
PJMPTBL  LB 000010AE|  PL       LB 00003E5E|  PL2      LB 00003DF6|  PMADDR2S LB 00003ACC|  POKEXCP  LB 00003BDC
PREFIX   AB 00000027|  PRINT    LB 00003EBE|  PRNTERR  LB 00003D26|  PROMPT   LB 00003EAE|  PROMPT2  LB 00003E42
PRTBUSY  LB 00001A4E|  PRTCLR   LB 00001A54|  PRTDRVR  LB 00001A3E|  PRTEXIT  LB 00001AE2|  PRTINIT  LB 00000E9C
PRTREAD  LB 00001A50|  PRTSEND  LB 00001AE6|  PRTSETUP  LB 00001A6A|  PRTWRT   LB 00001A84|  PUT      LB 0000184C
PUTACCA  MC --------|  PUTBACK  LB 00003A16|  PUTBAK2  LB 00003A3A|  PUTBAK3  LB 00003A58|  PUTCFIB  MC --------
PUTDKV   MC --------|  PUTDREG  MC --------|  PUTDTBL  MC --------|  PUTDTOW  MC --------|  PUTEBST  MC --------
PUTEMTH  MC --------|  PUTERRP  MC --------|  PUTIBF   MC --------|  PUTIFIB  MC --------|  PUTLOGN  MC --------
PUTNP    MC --------|  PUTNXCH  MC --------|  PUTOFIB  MC --------|  PUTPRBA  MC --------|  PUTPRF1  LB 00003DA4
PUTPRF2  LB 00003DB6|  PUTPRFX  LB 00003D82|  PUTPRNT  MC --------|  PUTSFIB  MC --------|  PUTSPTR  MC --------
PUTSTBL  MC --------|  PUTSTRB  MC --------|  PUTSYSC  MC --------|  PUTSYV   MC --------|  PUTTFIB  MC --------
PUTUFIB  MC --------|  PUTUFST  MC --------|  PUTUJPT  MC --------|  PUTUREG  MC --------|  PUTUT17  MC --------
PAGE - 169  MONITOR   FILE:MONITOR.TEXT  SYMBOLTABLE DUMP


PUTUTBL  MC --------|  PUTXFIB  MC --------|  PUTZFST  MC --------|  P_BCKSP  AB 00000005|  P_CLRLN  AB 00000007
P_CLRSC  AB 00000006|  P_ERSEL  AB 00000002|  P_ERSES  AB 00000003|  P_HOME   AB 00000004|  P_NDFS   AB 00000001
P_RLF    AB 00000000|  QUITDSK  LB 000009AA|  RCERR    LB 00000856|  RDDATA   LB 000018E2|  RDMTABL  LB 000009FE
RDMTINIT LB 00000A6E|  RDMTLP   LB 00000A74|  RDNRES   LB 0000126E|  READ.OK  LB 00003758|  READHDR  LB 0000128C
READLP   LB 000012A4|  REG0     AB 00000000|  REG1     AB 00000008|  REG2     AB 00000010|  REG3     AB 00000018
REG4     AB 00000020|  REG5     AB 00000028|  REG6     AB 00000030|  REG7     AB 00000038|  REG8     AB 00000040
REG9     AB 00000048|  REGA     AB 00000050|  REGB     AB 00000058|  REGC     AB 00000060|  REGD     AB 00000068
REGE     AB 00000070|  REGF     AB 00000078|  REINIT   LB 00000C5E|  REINITJ  LB 00000C72|  REMAP    LB 0000111C
REMOVE1  LB 0000379A|  RESETER  LB 0000246C|  RESETRX  LB 00002564|  RESTART  LB 0000420E|  RETRY    AB FFFFFFFE
REVISION LB 00003E54|  RLPL     LB 0000416C|  RNNGUSR  LB 00003958|  RPL      LB 0000416E|  RSKPHDR  LB 0000129C
RSTRXCP  LB 00003C42|  RT11     AB 0000001C|  RUNUCLR  MC --------|  RUNUSET  MC --------|  RUNUTST  MC --------
SAMESEG  LB 000036E0|  SCANAHD  LB 00002FE8|  SCANTTL  LB 00001B5E|  SCANTTLX LB 00001DCC|  SCERR    LB 0000086C
SCNSTRC  LB 00001B98|  SECURDIR AB 00000008|  SENDCMD  LB 00001880|  SENDHDR  LB 00001888|  SETA1A2  LB 000025AE
SETDISK  LB 00000DF0|  SETHDSK  LB 00000DB4|  SETTWGY  LB 00000BE6|  SETUPA5  LB 00003C8C|  SFXBACK  LB 00001DE2
SFXCODE  LB 00001DDC|  SFXFOTO  LB 00001DF4|  SFXGRAF  LB 00001DEE|  SFXINFO  LB 00001DE8|  SFXTEXT  LB 00001DD6
SMSCNFO  LB 00000468|  SNDR1    LB 00001146|  SOFTBO   LB 000038A2|  SOFTBPT  LB 0000389E|  SOFTXIT  LB 0000391E
SSTOP    AB 00000020|  STARTUP  LB 0000069E|  STAT01   LB 00001166|  STATJMP  LB 00000DD8|  STILLIN  LB 000037CC
STRTOBJ  LB 0000417A|  STRTRD   LB 000011D2|  STRTWRT  LB 000012D8|  SWAPJT   LB 00003C56|  SYSCMSZ  AB 0000002C
SYSPROG  LB 0000439E|  SYSTERM  AB 00000002|  SYSUNT   AB 00000001|  TEXTFILE AB 00000003|  THEEND   LB 00004534
THRESH   AB FFFFFFFF|  TLPL     LB 00003CE0|  TOP.86   LB 00003686|  TOPSYSC  LB 00000004|  TOPSYSF  LB 00000130
TOPUTBL  LB 00000032|  TPL      LB 00003CE2|  TRAPTO   MC --------|  TRP2MAX  LB 00004508|  TRYCRV   LB 00000DEC
TRYDBUG  LB 000042F6|  TRYEXEC  LB 00004042|  TRYMRK   LB 00000DE0|  TRYNEXT  LB 00003572|  TRYNJMP  LB 00000C52
TRYPIP   LB 00000DE6|  TRYQ     LB 0000431A|  TRYXEQT  LB 00004258|  TSTFVID  LB 00001C52|  TSTSTAT  LB 00000DCC
TTLCOPY  LB 00001E24|  UBUSY    LB 000019D8|  UCLR     LB 000019B4|  UEOVBLK  AB 0000000A|  UIO      LB 000019C2
UISBLKD  AB 00000009|  UJMPTBL  AB 00001800|  UJMPTOP  AB 00003000|  UN.LOOP  LB 000037D8|  UNITBL   LB 000006F0
UNITBSY  LB 000018B2|  UNITCLR  LB 000018CE|  UNITDEV  LB 0000167A|  UNITISB  LB 00001E66|  UNTYPFL  AB 00000000
UPSHFT   LB 00003D66|  UPSHFTX  LB 00003D76|  UREAD    LB 000019C0|  USELVL7  LB 0000452A|  UT17IDX  LB 00001E8C
UT17SIZ  AB 00000498|  UT17TSZ  AB 000000A8|  UT17VID  AB 00000000|  UT17VSZ  AB 00000008|  UTABLSZ  AB 000000FC
UTBLSIZ  AB 0000000C|  UVID     AB 00000000|  UWRITE   LB 000019BC|  V10      LB 0000388C|  V11      LB 00003892
V2       LB 0000385C|  V24      LB 00003898|  V3       LB 00003862|  V31      LB 00003822|  V31A     LB 0000381E
V31X     LB 0000382E|  V4       LB 00003868|  V5       LB 0000386E|  V6       LB 00003874|  V7       LB 0000387A
V8       LB 00003880|  V9       LB 00003886|  VALIDB7  LB 00000BB8|  VOLSRCH  LB 000020AE|  VOLSRCX  LB 000023A2
VOLSRXT  LB 000023AC|  W1LPL    LB 0000064C|  W1PL     LB 0000064E|  W2LPL    LB 00000656|  W2PL     LB 00000658
WAIT     LB 00003D78|  WCRLF    LB 00003F10|  WELCOM2  LB 000006BA|  WELCOME  LB 000006AA|  WFB1     LB 00001132
WFNB1    LB 00001156|  WHILE85  LB 00003640|  WRDATA   LB 0000190A|  WRITEIT  LB 0000272C|  WRTDIR   LB 0000273A
WRTLOOP  LB 00001354|  WRTNRES  LB 000013EC|  WRTPRT   LB 0000171A|  WSKPHDR  LB 0000134C|  XDSKFILE AB 00000001
XEQUTE   LB 00003F2A|  XEQUTE0  LB 00003F30|  XEQUTE1  LB 00003F4C|  XEQUTE2  LB 00003F86|  XEQUTE3  LB 00003F92
XEQUTE4  LB 00003F72|  XITNTRY  LB 000025A4|  XLPL     LB 00003F18|  XPBUSY   AB 00000003|  XPCPT    AB 00000002
```

```
XPL      LB 00003F1A|  XPOVRUN  AB 00000000|  XSTRTUP  LB 00000668|  Y.ERROR  LB 00001E60|  Y.LEAVE  LB 00001E50
Y.LOOP   LB 00001E4A|  Y.TEST   LB 00001E4C|  ZEROMEM  LB 000004DA|  ZZIORES  LB 000019F4|  ZZLOADIT LB 0000353A
ZZUNLOAD LB 00003794|
PAGE - 170  MONITOR   FILE:MONITOR.TEXT


03E6* 0282
03EA* 02C0
021A* 04D6
0320* 0506
0338* 05A0
03D4* 062A
03C8* 0636
03BC* 0642
03B0* 064E
03A4* 065A
0398* 0666
038C* 0672
09FC* 0276
03DA* 08F8
0818* 059C
0A7C* 0386
0B02* 0316
080C* 060C
0AC6* 0362
0810* 0618
0814* 062C
0B0E* 0340
081C* 0632
0A56* 0418
0DE8* 02C6
0C6A* 0444
09D0* 06DE
0C54* 046A
09DC* 06E2
0DE2* 0618
0A86* 0974
0A5E* 09AC
07A4* 1210
0950* 106C
07AC* 1210
07A8* 1218
07B0* 1228
0808* 11E2
07B4* 1240
0794* 126E
0798* 127E
079C* 1282
07A0* 1286
0318* 1726
0788* 1384
0486* 1686
0452* 16BA
0430* 16DC
040E* 16FE
07E4* 137A
07E8* 167C
0B7E* 12E8
0CE8* 11BA
07EC* 18C2
07F0* 18C6
PAGE - 171  MONITOR   FILE:MONITOR.TEXT


07F4* 1C38
07F8* 1D74
2050* 0570
07FC* 1DC4
207C* 06BE
078C* 1FCE
0498* 22C2
0464* 22F6
0442* 2318
0420* 233A
07BC* 224C
0790* 2680
04B8* 2958
0778* 27B6
077C* 27F2
0780* 2808
0784* 283A
245A* 0BF4
07D8* 2876
2F80* 030A
2F4C* 033E
07DC* 2AAE
3104* 0306
30EE* 031C
07E0* 2C42
0770* 2DCA
0774* 3020
07B8* 2FE2
0330* 356E
08EC* 2FB6
087A* 3028
0864* 303E
```

```
0800* 328A
0804* 32C8
07D4* 333C
399A* 02A8
39CE* 0288
399E* 02B8
068E* 35DE
39B6* 02D6
3962* 032A
38AE* 03DE
37A0* 04EC
371E* 056E
36F4* 0598
3544* 0748
08DC* 33B0
0844* 3448
0226* 3A66
06AC* 35E8
068A* 360A
08F6* 3406
381C* 050A
08D6* 3450
1BBE* 21A8
PAGE - 172  MONITOR   FILE:MONITOR.TEXT


06B0* 3708
0692* 3726
06B4* 3728
0696* 3746
0824* 3630
380C* 06B2
08C2* 35FC
06E6* 37D8
06DA* 37E4
06CE* 37F0
06EA* 3826
06BC* 3854
1670* 2924
0680* 39C2
3D60* 04AE
06EE* 382C
069A* 3B80
3710* 0C06
0328* 41E0
0220* 4314

  Pass 1 complete:       8345 lines
     0   Warnings
     0   Errors
ASSEMBLY COMPLETE.
 Et↑■MON11.TEXTDRIVE2/TPRTPR:MON11.TEXTZ","■0x@","""@X/ "01",10 ",hdLISA:MPPSW.FLISA:MPFPW.FLISA:MFRW.FLISA:MFIW.F"■H0x@TPR:
ON11.TEXTZ","■0x@","""■X/ "01",10 ",hdLISA:MPPSW.FLISA:MPFPW.FLISA:MFRW.FLISA:MFIW.F
```