```
; ALTO MESA BOOT LOADER - MBoot.asm
; LOADED ALMOST ANYWHERE AND EXECUTED
; R. Johnsson - 20 Feb 76
; Last edited March 20, 1978  5:40 PM

        .TITL BOOT
        .TXTM B

        .ENT MBOOT
        .ENT SwatFlag

        .SREL
        MBOOT:  BOOT
        SwatFlag: BREAK

        .NREL
        .GET "Mesa-NovaDefs.asm"


;AC0 is pointer to argument list

BOOT:   MOV 0 2
        LDA 3 0,2               ; POINTER TO BOOTMAP RECORD
        LDA 0 0,3               ; SN1
        STA 0 DCB1+sn1
        STA 0 DCB2+sn1
        LDA 0 1,3
        STA 0 DCB1+sn2
        STA 0 DCB2+sn2
        LDA 0 5,3               ; first page
        STA 0 DCB2+pagenumber
        NEG 0 0                 ; -1
        COM 0 0
        STA 0 DCB1+pagenumber
        LDA 0 C6
        ADD 3 0
        STA 0 PGPTR
        LDA 0 1,2
        STA 0 DCB1+diskaddress
        LDA 0 2,2               ; initial State
        STA 0 currentState
BLTLOOP:
        LDA 0 C3
        ADD 0 2
        LDA 0 0,2
        MOV 0 0 SNR
          JMP DOIT
        LDA 1 1,2
        LDA 3 2,2
        BLT             ; do some BLTs
        JMP BLTLOOP
C3:     3

DOIT:
        DIR             ; NO MORE INTERRUPTS
        SUB 0 0         ;CLEAR SOME LOCATIONS
        LDA 2 PAGE1     ; POINTER TO PAGE 1 LOCATIONS
        STA 0 DISP,2    ;DISPLAY OFF
        STA 0 IWW,2     ;WAKEUPS WAITING

;       JSR .+1         ;RELOCATE CURSOR POINTERS
;CURFIX:        LDA 1 POCURSORM1
;       ADD 3 1
;       STA 1 POCURSORM1
;       LDA 1 PMCURSORM1
;       ADD 3 1
;       STA 1 PMCURSORM1
;
;       LDA 0 PCURSORM1
;       LDA 1 PMCURSORM1
;       LDA 3 CN16
;       BLT             ;SAVE OLD CURSOR
;       LDA 0 PMCURSORM1
;       LDA 1 CUREND
;       LDA 3 CN16
```

```
;       BLT             ;INSERT MESA CURSOR

; NOW FIXUP THE ADDRESSES IN THE DISK BLOCKS

        JSR .+1
F:      MOV 3 1         ; ADDR OF FIXUP TO AC1
        LDA 2 DCB2      ; OFFSET OF FIRST DCB IS IN SECOND
        LDA 3 DCB1      ; AND VICE VERSA
        ADD 1 2
        ADD 1 3         ; THE FIXUPS
        STA 2 DCB2
        STA 3 DCB1

; NOW FIX THE HEADER AND LABEL ADDRESSES

        LDA 0 3 2
        ADD 1 0
        STA 0 3 2
        LDA 0 4 2
        ADD 1 0
        STA 0 4 2
        LDA 0 3 3
        ADD 1 0
        STA 0 3 3
        LDA 0 4 3
        ADD 1 0
        STA 0 4 3

; START THE DISK AND JUMP INTO LOOP

        LDA 0 @PGPTR    ; pick up first data address
        STA 0 data 2            ; STORE IT IN DCB
        STA 2 @PDBLK
        JMP ADVANCE



; DCB1 AND DCB2 ARE THE DISK CONTROL BLOCKS

  ; layout of disk command block
        next = 0
        status = 1
        command = 2
        header = 3
        label = 4
        data = 5
        normalinterrupts = 6
        errorinterrupts = 7
        unused = 10
        diskaddress = 11
  ; these hang off the end but are not part of the command
        prev = 12
        blank = 13
        numchars = 14
        pagenumber = 15
        version = 16
        sn1 = 17
        sn2 = 20

DCB1:   DCB2-F  ;THIS IS THE FIRST DCB'S LINK WORD


        0       ;STATUS
        44020   ;COMMAND.  READ header and data, check label
        DHDR-F  ;HEADER
        DCB2+diskaddress-F      ;THE LABEL IS READ IN ON TOP OF THE HEADER
        0       ;DATA POINTER IS MODIFIED IN OPERATION
        0       ;INTERRUPTS ARE INACTIVE
        0
        0       ;UNUSED
        0       ;DISK ADDRESS, ALSO THE FIRST LOCATION FOR OTHER DCB'S LABEL
        0       ;prev
        0       ;blank
        0       ;numchars
        0       ;page number
        0       ;version
```

```
        0       ;SN1
        0       ;SN2

DCB2:   DCB1-F  ;THIS IS THE SECOND DCB
        0       ;STATUS
        44020   ;COMMAND
        DHDR-F  ;HEADER
        DCB1+diskaddress-F      ;LABEL
        0       ;DATA
        0
        0
        0       ;DISK ADDRESS FOR SECOND DCB, LABEL FOR FIRST DCB
        0       ;prev
        0       ;blank
        0       ;numchars
        0       ;page number
        0       ;version
        0       ;SN1
        0       ;SN2

DHDR:   .BLK 2  ;read header here

CN16:   -16.
C6:     6

PAGE1:  400
DISP=   420-400         ; display
DIW=    421-400         ; display vertical interrupt
IWW=    452-400         ; wakeups waiting
IACTW=  453-400         ; active interrupts

PDBLK:  521             ; disk
ERMSK:  367             ; error mask for disk transfer
PGPTR:  0               ; pointer to next Page Table entry

;CUREND:        450             ; end of cursor for BLT
;PCURSORM1: 430
;POCURSORM1: OLDCURSOR-CURFIX-1
;PMCURSORM1: MESACURSOR-CURFIX-1
;
;OLDCURSOR: .BLK 20             ; save the entry cursor
;MESACURSOR:
;       104760
;       154400
;       124400
;       104700
;       104400
;       104400
;       104760
;       000000
;       003416
;       004221
;       004021
;       003437
;       000221
;       004221
;       003421
;       000000

; MAIN LOOP OF THE LOADER.
; ENTER AT ADVANCE WITH AC2= POINTER TO DCB1, AC3= POINTER TO DCB2.
; THE DISK HAS BEEN STARTED on DCB1.

; AT SETUP AC0 HAS ADDRESS FOR NEXT TRANSFER, AC2=POINTER TO
; NEXT DCB, AC3=POINTER TO ACTIVE DCB
RETRYCOUNT: 6

SETUP:  STA 0 data 2            ; ADDRESS OF NEXT TRANSFER
        SUB 0 0
        STA 0 status 2          ; CLEAR STATUS OF NEXT
        STA 0 prev,3            ; label of next transfer cleared for check
        STA 0 blank,3
        STA 0 numchars,3
        ISZ pagenumber,3        ; bump page number
        JMP .+1
```

```
        ISZ pagenumber,3        ; by 2
        LDA 0 C6
        STA 0 RETRYCOUNT
RETRY:  LDA 1 ERMSK

KWAIT:  LDA 0 status 3          ; WAIT FOR DISK TO STOP
        MOV 0 0 SNR
          JMP KWAIT
        AND 0 1 SZR     ; CHECK FOR ERROR
          JMP DSKERR    ;   GOT ONE
;       ISZ @CUREND     ; COUNT IN CURSOR
                        ; never skip
        sta 1 diskaddress,3
        lda 0 NewDa
        sta 1 NewDa
        snz 0 0         ; if changing files
          jmp ADVANCE
        sta 0 diskaddress,2
        sta 2 @PDBLK
        sta 2 next,3
        lda 0 pagenumber,3
        neg 0 0
        com 0 0         ; subtract 1
        sta 0 pagenumber,2 ; update items for next check
        lda 1 sn1,3
        sta 1 sn1,2
        lda 1 sn2,3
        sta 1 sn2,2

ADVANCE: MOV 2 0                ; EXCHANGE CURRENT AND INACTIVE DCB POINTERS
        MOV 3 2
        MOV 0 3         ; AC3=POINTER TO ACTIVE DCB
        LDA 0 0,2               ; CHECK LINK OF DCB JUST FINISHED
        MOV 0 0 SNR     ; TEST FOR LAST PAGE
          JMP STPR
        LDA 0 diskaddress,3 ; CURRENT DISK ADDRESS
        STA 0 @PGPTR    ; REPLACE OLD CORE ADDRESS
        ISZ PGPTR
        LDA 0 @PGPTR
        MOVZR 0 0 SZC   ; TEST FOR FILE CHANGE
          JMP ChangeFile
        MOVZL 0 0 SZR   ; test for end of list
          JMP SETUP     ; MORE TO DO
        STA 0 0 3               ; CLEAR DCB CHAIN LINK
        JMP SETUP

C2:     2
NewDa:  0

ChangeFile:     ; AC0 has new file page number
        lda 1 C2
        sub 1 0
        sta 0 pagenumber,3      ; new page-2 number to label
        isz PGPTR
        sub 0 0
        sta 0 next,3            ; break the chain
        sta 0 sn1,3
        sta 0 sn2,3             ; zero the serial number
        lda 0 @PGPTR
        sta 0 NewDa             ; new disk address
        isz PGPTR
        lda 0 @PGPTR            ; first memory address for new file
        jmp SETUP

STPR:
;       LDA 0 POCURSORM1
;       LDA 1 CUREND
;       LDA 3 CN16
;       BLT             ; RESTORE CURSOR
BREAK:  JMP .+1         ; set to swat break (#77400) to break before starting Mesa
        JMP Emulate

SWATTrap:       567
DSKERR: DSZ RETRYCOUNT
          JMP TryAgain
        JSR DSKERR1
```

```
        .TXT "Disk error reading image file"
DSKERR1:        MOV 3 1
        LDA 3 @SWATTrap
        JMP 16,3        ; does a CallSwat

TryAgain:
        SUB 0 0
        STA 0 1,3       ;clear active status
        STA 3 @PDBLK    ;restart disk
        JMP RETRY

        JMP BOOT+200            ; Error if code too big


        .END;
```