```
                                page    40,132
                        ;_____________________________________________________________________________
                        ;       Tiny Boot Manager. Copyright (C) 2010-21 A.C.M. Daas <http://daas.info>
                        ;       TBOOTMGR is free software: you can redistribute it and/or modify
                        ;       it under the terms of the GNU General Public License as published by
                        ;       the Free Software Foundation, either version 3 of the License,
                        ;       or any later version.
                        ;
                        ;       TBOOTMGR is distributed in the hope that it will be useful,
                        ;       but WITHOUT ANY WARRANTY; without even the implied warranty of
                        ;       MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
                        ;       See the GNU General Public License for more details.
                        ;
                        ;       You should have received a copy of the GNU General Public License
                        ;       along with this program.  If not, see <https://www.gnu.org/licenses/>.
                        ;_____________________________________________________________________________
                        ; Tiny Boot Manager Master Boot Record routine at CYL=0, HEAD=0, SECT=1
                        ; Upon execution reg values are: CS=0000h, IP=7C00h, DL=drive
0000                    tbootmgr segment
                                assume  cs:tbootmgr,ds:tbootmgr

                        ; Ascii characters used
= 0007                  bel     equ     07h     ;bel character
= 000A                  lf      equ     0Ah     ;line feed
= 000D                  cr      equ     0Dh     ;cariage return
= 001B                  escape  equ     1Bh     ;escape key

                        ; partition type list
= 0001                  fat12   equ     01h     ;11h if hidden, <32Mb
= 0004                  fat16   equ     04h     ;14h if hidden, >32Mb <500Mb
= 0005                  extend  equ     05h     ;extended
= 0006                  fat16b  equ     06h     ;16h if hidden, >32Mb <2Gb
= 0007                  ifs     equ     07h     ;17h if hidden
= 000B                  fat32   equ     0Bh     ;1Bh if hidden, <2Gb
= 000C                  f32lba  equ     0Ch     ;C/H/S=unused, relative sector=LBA
```

```
                        f16lba  equ     0Eh     ;C/H/S=unused, relative sector=LBA
                        extlba  equ     0Fh     ;C/H/S=unused, relative sector=LBA
                        hidden  equ     10h     ;single bit is set in DOS partition types
                        bootflg equ     80h

                        ; Initialize stack and relocate code from 7C00h to this address at 0600h
0000 7C00                       org     7C00h
= 7C00                  loadadr equ     $
0600                            org     600h
0600                    mbr     proc    near
0600  FC                       cld
0601  33 C9                    xor     cx,cx
0603  8E D9                    mov     ds,cx   ;set source segment
0605  BE 7C00 R                mov     si,offset loadadr       ;set offset loaded code
0608  8E C1                    mov     es,cx   ;set destination segment
060A  BF 0600 R                mov     di,offset mbr  ;and offset
060D  FA                       cli             ;prevent interrups prior to changing stack location
060E  8E D1                    mov     ss,cx   ;set stack segment
0610  8B E6                    mov     sp,si   ;set stack to before source area
0612  B5 01                    mov     ch,1h   ;256 words
0614  F3/ A5                   rep movsw       ;move this code from address 7C00h to 600h
0616  FB                       sti             ;allow interrups after string move
0617  E9 90B0 R                jmp     near ptr cont-loadadr+mbr      ;near jump will do nicely

                        ;-----------------------------------------------------------------------------
                        ; types that have hidden equivalent
061A  05                chstype db      extend
061B  01 04 06 0B       dostype db      fat12,fat16,fat16b,fat32
= 0005                  chslen  equ     $-chstype       ;types that use chs value
061F  0E 0C 07                  db      f16lba,f32lba,ifs       ;types that use lba value
= 0007                  doslen  equ     $-dostype       ;types that have hidden equivalent

                        ; Subroutine to read or write AL mod 10 sector
                        ; AL/10= function (2-read or 3-write), AH= retrycount
0622  B0 15             rd_chs: mov     al,21   ;read partition bootsector
```

```
0624  B4 08                int13:  mov     ah,100h shr 5   ;initialize retry count to 5
0626  8B F8                int13r: mov     di,ax
0628  D4 0A                        aam
062A  CD 13                        int     13h
062C  73 09                        jnc     ret13
062E  B4 00                        mov     ah,0    ;reset disk system
0630  CD 13                        int     13h
0632  97                           xchg    ax,di
0633  D0 E4                        shl     ah,1
0635  73 EF                        jnc     int13r  ;try al times
0637  C3                  ret13:  ret

                           ; Routine to write text to screen. Entrypoint is tty, or
                           ; wrtty if AL already has first character to write, or
                           ; tty_eol if AL has character and requires terminated with end of line.
                           ; On entry ds:si points to message to write (null terminates routine)
                           ; exits with ds:si pointing at terminating 0 with AL= 0
0638  BE 07B0 R            ttyeol: mov     si,offset crlf ;AL has character to write
063B  B3 07                wrtty:  mov     bl,7    ;white
063D  B4 0E                        mov     ah,0Eh  ;write teletype to active page
063F  CD 10                        int     10h     ;AL=character, BL=foreground color
0641  AC                   tty:    lodsb
0642  3C 00                        cmp     al,0
0644  75 F5                        jnz     wrtty   ;end on nul
0646  4E                           dec     si      ;set pointer back to trailing 0
0647  C3                           ret

                           ; Exit to rom basic
0648  E8 0638 R            basic:  call    ttyeol
064B  CD 18                        int     18h     ;Return control to PC-BIOS to start ROM Basic

                           ; Prompt user to select a start partition
064D  B0 07                reask:  mov     al,bel  ;load Bel char; SI still points at terminating 0
064F  E8 063B R            rdkey:  call    wrtty
0652  B4 00                        mov     ah,0    ;read keyboard
```

```
106      0654  CD 16                       int     16h     ;AL=ascii, AH=scancode
107      0656  3C 1B                       cmp     al,escape       ;Esc char
108      0658  74 EE                       je      basic   ;esc key pressed?
109      065A  8B E8                       mov     bp,ax   ;save keystroke
110      065C  2C 31                       sub     al,'1'  ;convert numeric ascii to binairy
111      065E  A8 FC                       test    al,not 3h       ;if value any higher than 3,
112      0660  75 EB                       jnz     reask   ;reask as user response is invalid
113                                ; Unhide partition if selected partition is hidden dostype
114      0662  B4 10                       mov     ah,16
115      0664  F6 E4                       mul     ah      ;convert to table entry offset
116      0666  96                          xchg    si,ax
117      0667  BB 07C2 R                    mov     bx,offset table+4       ;point to type in partition table
118      066A  8D 30                       lea     si,[bx+si]      ;point to selected table entry
119      066C  8A 04                       mov     al,[si] ;get partition type
120      066E  84 C0                       test    al,al           ;if unused entry,
121      0670  74 0E                       jz      hide            ;then hide others
122      0672  34 10                       xor     al,hidden       ;unhide any hidden partition
123      0674  BF 061B R                   mov     di,offset dostype
124      0677  B9 0007                     mov     cx,doslen
125      067A  F2/ AE                      repne scasb
126      067C  75 2A                       jne     setact  ;selected partition other than hidden dostype?
127      067E  88 04                       mov     [si],al ;unhide selected partition
128                                ; Hide any unhidden dos partition
129      0680  3B DE                hide:  cmp     bx,si
130      0682  74 0E                       je      skphid  ;selected partition?
131      0684  8A 07                       mov     al,[bx]
132      0686  BF 061B R                   mov     di,offset dostype
133      0689  B1 07                       mov     cl,doslen
134      068B  F2/ AE                      repne scasb
135      068D  75 03                       jne     skphid  ;already hidden or non dos?
136      068F  80 37 10                    xor     byte ptr[bx],hidden     ;hide partition
137      0692  8D 5F 10             skphid: lea     bx,[bx+16]
138      0695  81 FB 07FE R                cmp     bx,offset table+64
139      0699  72 E5                       jb      hide
140                                ; Write changed partition table back to master boot record
```

```
141      069B  BB 0600 R                       mov     bx,offset mbr  ;buffer address
142      069E  B6 00                           mov     dh,0    ;set head=0, drive number is still in dl
143      06A0  B9 0001                         mov     cx,1    ;set sector=1 and cylinder=0
144      06A3  B0 1F                           mov     al,31   ;write back master boot record
145      06A5  E8 0624 R                       call    int13   ;continue even with error, so no need to check
146                                    ; Mark choosen partition active and display selection
147      06A8  C6 44 FC 80             setact: mov     byte ptr[si-4],bootflg ;set partition active
148      06AC  95                              xchg    ax,bp   ;reload keystroke
149      06AD  E8 0638 R                       call    ttyeol  ;display selected partition
150                                    ; Look for an active partition
151      06B0  BF 07BE R               cont:   mov     di,offset table
152      06B3  B9 0004                         mov     cx,4
153      06B6  B8 0080                         mov     ax,bootflg      ;load bootflag in AL, clear AH
154      06B9  0A 25                   check:  or      ah,[di] ;check empty bootflag field
155      06BB  74 07                           jz      next    ;active flag clear?
156      06BD  32 C4                           xor     al,ah   ;check valid bootflag and make future flags invalid
157      06BF  75 13                           jnz     inval   ;invalid bootflag field?
158      06C1  8B F7                           mov     si,di   ;save table entry offset
159      06C3  98                              cbw             ;clear AH
160      06C4  8D 7D 10                next:   lea     di,[di+16]
161      06C7  E2 F0                           loop    check   ;not all 4 partitions done?
162      06C9  A8 80                           test    al,bootflg      ;an active partition found?
163      06CB  74 0C                           jz      chktyp  ;then skip user prompt
164      06CD  BE 075B R                       mov     si,offset prompt
165      06D0  AC                              lodsb           ;load first char in AL
166      06D1  E9 064F R                       jmp     rdkey   ;go and prompt user for selection
167
168      06D4  BE 077A R               inval:  mov     si,offset inv_msg
169      06D7  EB 55                           jmp     short abend
170
171                                    ; Analize partition for CHS type
172      06D9  8A 44 04                chktyp: mov     al,[si+4]       ;get partition type, is it unused?
173      06DC  84 C0                           test    al,al
174      06DE  74 6C                           jz      noboot  ;exit, since unused partition is not bootable
175      06E0  BF 061A R                       mov     di,offset chstype
```

```
176      06E3  B1 05                        mov     cl,chslen
177      06E5  F2/ AE                       repne   scasb
178      06E7  74 4A                        je      rdchs   ;is active partition chs type?
179                                 ; For all other types check for extended int13 support
180      06E9  BB 55AA                      mov     bx,55AAh        ;fill with request signature
181      06EC  B4 41                        mov     ah,41h  ;get extended int 13 support info; DL still has drive
182      06EE  CD 13                        int     13h
183      06F0  72 41                        jc      rdchs   ;extension not found
184      06F2  81 FB AA55                   cmp     bx,0AA55h       ;signature, AH=major version, DH=extension ver.
185      06F6  75 3B                        jne     rdchs   ;requested support not installed
186      06F8  F6 C1 01                     test    cl,01h  ;bit0=1 if int13,AH=42h supported
187      06FB  74 36                        jz      rdchs   ;API subset supported
188                                 ;Read bootrecord with extended int13
189      06FD  8B DC                        mov     bx,sp   ;get bootsector load address
190      06FF  B9 0005                      mov     cx,5    ;set retrycount
191      0702  56                   retrlb: push    si      ;save si
192      0703  33 C0                        xor     ax,ax
193                                 ;build address request packet on stack
194      0705  50                           push    ax      ;sector 4th word
195      0706  50                           push    ax      ;sector 3rd word
196      0707  FF 74 0A                     push    [si+10] ;sector 2nd word
197      070A  FF 74 08                     push    [si+8]  ;sector low word
198      070D  06                           push    es      ;buffer segment
199      070E  53                           push    bx      ;buffer offset
200      070F  40                           inc     ax      ;sector count to 1
201      0710  50                           push    ax      ;number of sectors (max.7F)
202      0711  B0 10                        mov     al,10h  ;packet size
203      0713  50                           push    ax      ;high byte reserved (=0)
204      0714  8B F4                        mov     si,sp   ;DS:SI points to request address packet
205      0716  B4 42                        mov     ah,42h  ;extended disk read; DL has drive number
206      0718  CD 13                        int     13h
207      071A  72 04                        jc      skp_ck  ;if CF then AH=errorcode else AH=0
208                                 ; Check sector count read, as C is not set if sector not found error
209      071C  83 7C 02 01                 cmp     word ptr[si+2],1        ;also need to check actual count
210      0720  8D 64 0E             skp_ck: lea     sp,[si+14]      ;purge address request packet-1w from stack
```

```
0723  58                        pop     ax      ;restore AX (=0) with last word of packet
0724  5E                        pop     si      ;restore initial SI
0725  73 19                     jnc     readok  ;if count < 1 then return C=1, else C=0
0727  CD 13                     int     13h     ;reset drive
0729  E2 D7                     loop    retrlb  ;try again if count not exhausted
072B  BE 0792 R           rdfail: mov     si,offset err_msg
072E  E8 0641 R           abend:  call    tty
0731  EB FB                       jmp     short abend     ;loop on last 0

                          ; Read bootrecord of active partition using CHS
0733  8B DC               rdchs:  mov     bx,sp   ;set ES:BX to buffer address 7C00h
0735  8A 74 01                    mov     dh,[si+1]       ;set head number, DL still has drive number
0738  8B 4C 02                    mov     cx,[si+2]       ;set sector & cyl
073B  E8 0622 R                    call    rd_chs
073E  72 EB                        jc      rdfail
0740  81 BF 01FE AA55     readok: cmp     word ptr[bx+(bootid-mbr)],0AA55h
0746  75 04                        jne     noboot
0748  FF D3                        call    bx      ;execute partitions boot record
                          ; Upon exit DS:SI points to booted partition table entry, DL= disk
                          ; If returned from extended, entry is replaced with data partition
074A  EB 8D                        jmp     chktyp  ;so try to load this new entry

074C  BE 0798 R           noboot: mov     si,offset mis_msg
074F  8B FE                       mov     di,si
0751  B8 694D                     mov     ax,'iM' ;modify message text
0754  AB                          stosw
0755  B8 7373                     mov     ax,'ss'
0758  AB                          stosw
0759  EB D3                       jmp     short abend

075B  53 74 61 72 74 20   prompt  db      'Start partition (1-4 or Esc)?:',0
      70 61 72 74 69 74
      69 6F 6E 20 28 31
      2D 34 20 6F 72 20
      45 73 63 29 3F 3A
```

```
                00
077A  49 6E 76 61 6C 69      inv_msg db      'Invalid partition table',0
      64 20 70 61 72 74
      69 74 69 6F 6E 20
      74 61 62 6C 65 00
0792  45 72 72 6F 72 20      err_msg db      'Error '        ;'loading operating system'
0798  6C 6F 61 64 69 6E      mis_msg db      'loading operating system' ;first 4 chars replaced with 'Miss'
      67 20 6F 70 65 72
      61 74 69 6E 67 20
      73 79 73 74 65 6D
07B0  0D 0A                  crlf    db      cr,lf
                             ;fill unused space, Windows NT signature, etc.
07B2     03 [                null    db      (mbr+1B5h-$) dup (0)
                   00
                      ]
07B5  7A 92 98                       db      low offset inv_msg, low offset err_msg, low offset mis_msg
07B8     06 [                        db      6 dup (0)       ;NT-signature
                   00
                      ]
07BE     40 [                table   db      64 dup (0)      ;Partition table
                   00
                      ]
07FE  AA55                   bootid  dw      0AA55h
0800                         mbr     endp
```

```
                                        page
                                ;_____________________________________________________________________________
                                ; Tiny Boot Manager primairy EBR code. Upon execution reg values are:
                                ;       CS=0000h, IP=7C00h, DL=drive,
                                ;       DS:[SI] points at our Primairy Partition Table entry
                                ; It will replace entry DS:[SI] with first entry in EBR partition table.
                                ; First sector LBA value is converted to absolute value
                                ; and partition type converted to LBA type if this EBR is LBA type.
                                ; It then will return from call to MBR boot code with initial SI and DL values.
                                ; On return AX, DI are destroyed.

                                ; Check if we received a call from TBOOTMGR MBR code
 7C00                                   org     loadadr
 7C00                           ebr     proc    near
 7C00  33 C0                            xor     ax,ax
 7C02  81 FC 7BFE R                     cmp     sp,offset loadadr-2 ;if stack holds a return address,
 7C06  74 15                            je      eptcpy  ;then continue with chainboot
                                ; else write boot failure text to screen. DS may have unexpected value
 7C08  8E D8                            mov     ds,ax   ;make DS zero
 7C0A  BE 7C58 R                        mov     si,offset ebrmsg        ;get message offset
 7C0D  B3 07                            mov     bl,7    ;white
 7C0F  AC                       lp_tty: lodsb
 7C10  84 C0                            test    al,al   ;if not end of message,
 7C12  75 03                            jnz     wr_tty  ;then write character
 7C14  4E                       no_tty: dec     si      ;else move pointer back to trailing 0
 7C15  EB F8                            jmp     short lp_tty    ;loop on last 0

 7C17  B4 0E                    wr_tty: mov     ah,0Eh  ;write teletype to active page
 7C19  CD 10                            int     10h     ;AL=character, BL=foreground color
 7C1B  EB F2                            jmp     short lp_tty

                                ; Copy first ept entry into ppt table at DS:[SI] and return
 7C1D  56                       eptcpy: push    si      ;save SI
 7C1E  06                               push    es      ;save ES
 7C1F  1E                               push    ds      ;copy DS
```

```
308      7C20  07                          pop     es      ;to ES
309      7C21  8E D8                       mov     ds,ax   ;set DS to 0
310      7C23  8D 7C 01                    lea     di,[si+1]       ;skip bootflag
311      7C26  BE 7DBF R                   mov     si,offset ept+1 ;our load address + table offset +1
312      7C29  A4                          movsb           ;copy first head
313      7C2A  A5                          movsw           ;copy first cyl/sec
314      7C2B  AC                          lodsb           ;load type in AL
315      7C2C  26: 80 3D 0F                cmp     byte ptr es:[di],extlba       ;if our type is not LBA,
316      7C30  75 12                       jne     ebrnlb  ;then continue with copy
317      7C32  3C 06                       cmp     al,fat16b       ;else replace CHS types to LBA
318      7C34  75 02                       jne     ebrn16
319      7C36  B0 0E                       mov     al,f16lba
320      7C38  3C 0B                ebrn16: cmp    al,fat32
321      7C3A  75 02                       jne     ebrn32
322      7C3C  B0 0C                       mov     al,f32lba
323      7C3E  3C 05                ebrn32: cmp    al,extend
324      7C40  75 02                       jne     ebrnlb
325      7C42  B0 0F                       mov     al,extlba
326      7C44  AA                   ebrnlb: stosb          ;save type
327      7C45  A4                          movsb           ;copy last head
328      7C46  A5                          movsw           ;copy last cyl/sec
329      7C47  AD                          lodsw           ;get first sector LBA low word
330      7C48  26: 03 05                   add     ax,es:[di]      ;add to parent LBA
331      7C4B  AB                          stosw           ;store result
332      7C4C  AD                          lodsw           ;get first sector LBA high word
333      7C4D  26: 13 05                   adc     ax,es:[di]      ;add to parent LBA high word
334      7C50  AB                          stosw           ;store result
335      7C51  A5                          movsw           ;copy sector amount low word
336      7C52  A5                          movsw           ;copy sector amount high word
337      7C53  06                          push    es      ;move ES to DS
338      7C54  1F                          pop     ds
339      7C55  07                          pop     es      ;restore ES
340      7C56  5E                          pop     si      ;restore SI
341      7C57  C3                          ret             ;return control to TBOOTMGR MBR
342
```

```
343      7C58  45 78 74 65 6E 64      ebrmsg  db      'Extended partition not bootable',0
344            65 64 20 70 61 72
345            74 69 74 69 6F 6E
346            20 6E 6F 74 20 62
347            6F 6F 74 61 62 6C
348            65 00
349
350      7DBE                                  org     loadadr+1BEh
351      7DBE     40 [                 ept     db      64 dup (0)
352                    00
353                        ]
354
355      7DFE  AA55                            dw      0AA55h
356      7E00                          ebr     endp
357      7E00                          tbootmgr ends
358                                            end
```