```
                                        page    40,132
                                ;_______________________________________________________________________________
                                ;       Tiny Boot Select (TBOOTSEL). Copyright (C) 2016-2021 Ton Daas
                                ;       TBOOTSEL 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.
                                ;       TBOOTSEL 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 Select is a manager that resides entirely in the master boot record.
                                ; It allows the user to choose from up to four partitions he wants to boot from,
                                ; or revert to PC-BIOS for starting ROM-Basic or a subsequent boot device.
                                ; Dos/Win partitions can be hidden, to allow for multiple primary partitions.
                                ; The selected partition will be unhidden, hiding other conflicting partitions.
                                ; It supports FAT12, FAT16, NTFS, FAT32, partitions >2GB and disks >8GB, <2TB.
                                ; It is a modified version of Tiny Boot Manager, to suit MS Windows 7-10.
                                ; This means that the master boot record will be saved with the last selected
                                ; partition marked active. Consequently the user prompt will always be shown and
                                ; unused table entries and an extended partition are treated as invalid choices.
                                ;===============================================================================
0000                            tbootsel segment
                                        assume  cs:tbootsel,ds:tbootsel
7C00                                    org     7C00h
= 7C00                          loadadr equ     $
0600                                    org     600h

                                ; Master boot record routine at CYL=0, HEAD=0, SECT=1
                                ; Upon execution reg values are: CS=0000h, IP=7C00h, DL=drive
                                ; Initialize stack and relocate code from 7C00h to this address at 0600h
```

```
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 source offset loaded code
0608  8E C1                          mov    es,cx   ;set destination segment
060A  BF 0600 R                      mov    di,offset mbr  ;and target 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 9045 R                      jmp    near ptr cont-loadadr+mbr    ;near jump will do nicely

                             ; Ascii special characters
= 0007                       bel     equ    07h
= 000A                       lf      equ    0Ah
= 000D                       cr      equ    0Dh
= 001B                       escape  equ    1Bh

                             ; Recognized 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                       ntfs    equ    07h     ;17h if hidden
= 000B                       fat32   equ    0Bh     ;1Bh if hidden, <2Gb
= 000C                       f32lba  equ    0Ch     ;C/H/S=unused, relative sector=LBA
= 000E                       f16lba  equ    0Eh     ;C/H/S=unused, relative sector=LBA
= 000F                       extlba  equ    0Fh     ;C/H/S=unused, relative sector=LBA

= 0010                       hidden  equ    10h     ;single bit is set in DOS partition types
= 0080                       bootflg equ    80h

```

```
                             ; Types that have hidden equivalent
061A  01 04 06 0B            dostype db     fat12,fat16,fat16b,fat32
= 0004                       chslen  equ    $-dostype      ;dos types that use chs method
061E  0E 0C                          db     f16lba,f32lba  ;dos types that use lba method
0620  07                             db     ntfs           ;type supports both
= 0007                       typelen equ    $-dostype

                             ; Routine to read or write 1 sector
                             ; AH= function (2-read or 3-write)
0621  B0 08                  int13:  mov    al,100h shr 5  ;initialize retry count to 5
0623  8B F8                  int13r: mov    di,ax
0625  B0 01                          mov    al,1    ;specify one sector transfer
0627  CD 13                          int    13h
0629  73 09                          jnc    ret13
062B  B4 00                          mov    ah,0    ;reset disk system
062D  CD 13                          int    13h
062F  97                             xchg   ax,di
0630  D0 E0                          shl    al,1
0632  73 EF                          jnc    int13r  ;try al shift times
0634  C3                     ret13:  ret

                             ; Routine to write text to screen; exits with ds:si pointing at 0
                             ; Entrypoint is tty (or wrtty if AL already has first character to write)
                             ; Entrypont is ttyeol to output only AL followed bij end of line
                             ; On entry ds:si points to message to write (null terminates routine)
0635  BE 07AC R              ttyeol: mov    si,offset crlf
0638  B3 07                  wrtty:  mov    bl,7    ;white
063A  B4 0E                          mov    ah,0Eh  ;write teletype to active page
063C  CD 10                          int    10h     ;AL=character, BL=foreground color
063E  AC                     tty:    lodsb
063F  3C 00                          cmp    al,0
0641  75 F5                          jnz    wrtty   ;end on nul
0643  4E                             dec    si      ;set pointer back to trailing 0
0644  C3                             ret
```

```
                             ; Look for an active partition
0645  BF 0040                cont:   mov     di,4*16 ;initialize as invalid at end of table
0648  8D 75 F0                       lea     si,[di-16]     ;set search index at last entry
064B  B8 0080                        mov     ax,bootflg     ;load valid bootflag, clear AH
064E  86 A4 07BE R           chk_bf: xchg    ah,table[si]   ;get bootflag in AH, clear in table
0652  84 E4                          test    ah,ah   ;check for empty bootflag field
0654  74 07                          jz      next    ;active flag clear?
0656  32 C4                          xor     al,ah   ;check valid bootflag and make future flags invalid
0658  75 0E                          jnz     inval   ;invalid bootflag field?
065A  8B FE                          mov     di,si   ;save current active table entry offset
065C  98                             cbw             ;clear ah
065D  83 EE 10               next:   sub     si,16
0660  73 EC                          jnc     chk_bf  ;not all 4 partitions done?
0662  BE 0757 R                      mov     si,offset prompt
0665  AC                             lodsb
0666  EB 10                          jmp     short rdkey    ;go and prompt user for selection

0668  BE 0776 R              inval:  mov     si,offset inv_msg
066B  E9 072A R                      jmp     abend

                             ; Exit to BIOS to start ROM Basic or subsequent boot device
066E  E8 0635 R              basic:  call    ttyeol  ;show entered keystroke in AL to user
0671  CD 18                          int     18h     ;start ROM Basic

                             ; Prompt user to select a partition
0673  BE 07AE R              reask:  mov     si,offset null
0676  B0 07                          mov     al,bel  ;load bel char
0678  E8 0638 R              rdkey:  call    wrtty
067B  B4 00                          mov     ah,0    ;read keyboard
067D  CD 16                          int     16h     ;AL=ascii, AH=scancode
067F  3C 1B                          cmp     al,escape      ;Esc char
0681  74 EB                          je      basic   ;esc key pressed?
0683  8B E8                          mov     bp,ax   ;save keystroke
0685  2C 31                          sub     al,'1'  ;convert numeric 1 base ascii to binairy 0 base
0687  A8 FC                          test    al,not 3h      ;if not within 0 to 3,
0689  75 E8                          jnz     reask   ;invalid choice, reask user
068B  B4 10                          mov     ah,16
068D  F6 E4                          mul     ah      ;convert to table entry offset
068F  96                             xchg    si,ax   ;transfer offset to SI

                             ; Is selection unused or an extended partition?
0690  BB 07BE R                      mov     bx,offset table
0693  8A 40 04                       mov     al,[si+bx+4]    ;get partition type
0696  84 C0                          test    al,al           ;if it is an unused entry?
0698  74 D9                          jz      reask           ;then reask
069A  3C 05                          cmp     al,extend       ;if extended dos partition choosen?
069C  74 D5                          je      reask   ;then reask
069E  3C 0F                          cmp     al,extlba       ;if extended lba dos partition choosen?
06A0  74 D1                          je      reask   ;then reask

                             ; Mark selected partition active
06A2  C6 00 80                       mov     byte ptr [si+bx],bootflg
06A5  3B F7                          cmp     si,di   ;if selection is same as prevous,
06A7  8D 30                          lea     si,[si+bx]
06A9  74 34                          je      no_chg  ;then no need to update MBR

                             ; Unhide partition if selected partition is hidden dostype
06AB  34 10                          xor     al,hidden       ;unhide any hidden partition
06AD  BF 061A R                      mov     di,offset dostype
06B0  B9 0007                        mov     cx,typelen
06B3  F2/ AE                         repne scasb
06B5  75 1C                          jne     wrtchg  ;selected partition other than hidden dostype?

                             ; Hide any other unhidden dos partition
06B7  3B DE                  hide:   cmp     bx,si
06B9  74 0C                          je      unhide  ;selected partition?
06BB  8A 47 04                       mov     al,[bx+4]
06BE  BF 061A R                      mov     di,offset dostype
06C1  B1 07                          mov     cl,typelen
06C3  F2/ AE                         repne scasb
```

```
06C5  75 04                          jne     skphid  ;already hidden or non dos?
06C7  80 77 04 10            unhide: xor     byte ptr[bx+4],hidden  ;hide partition
06CB  8D 5F 10               skphid: lea     bx,[bx+16]
06CE  80 FB FE                       cmp     bl,low offset table+64
06D1  72 E4                          jb      hide

                             ; Write changed partition table back to master boot record
06D3  BB 0600 R              wrtchg: mov     bx,offset mbr  ;buffer address
06D6  B6 00                          mov     dh,0    ;set head=0, drive number is still in DL
06D8  B1 01                          mov     cl,1    ;set sector=1 and cylinder=0 (assume CH=0)
06DA  B4 03                          mov     ah,3    ;write back master boot record
06DC  E8 0621 R                      call    int13   ;continue even with error

                             ; Output selection to user
06DF  56                     no_chg: push    si      ;save si
06E0  95                             xchg    ax,bp   ;reload keystroke and save table entry in BP
06E1  E8 0635 R                      call    ttyeol  ;show selection choice
06E4  5E                             pop     si      ;restore si

                             ; Analize partition type
                             ;       mov     al,[si+4]       ;get partition type
                             ;       mov     di,offset dostype
                             ;       mov     cl,chslen
                             ;       repne   scasb   ;is active partition chs type?
                             ;       je      rdchs   ;read partition traditional chs style

                             ; For all other types check for extended int 13h support
06E5  BB 55AA                        mov     bx,55AAh        ;fill with request signature
06E8  B4 41                          mov     ah,41h  ;get extended int 13 support info; DL still has drive
06EA  CD 13                          int     13h
06EC  72 41                          jc      rdchs   ;extension not found
06EE  81 FB AA55                     cmp     bx,0AA55h       ;signature, AH=major version, DH=extension ver.
06F2  75 3B                          jne     rdchs   ;requested support not installed
06F4  F6 C1 01                       test    cl,01h  ;bit0=1 if int13,AH=42h supported
06F7  74 36                          jz      rdchs   ;API subset not supported
```

```
                             ; Read bootrecord with extended interrupt 13h
06F9  8B DC                          mov    bx,sp  ;get bootsector load address
06FB  B9 0005                        mov    cx,5   ;set retrycount
06FE  56                     retrlb: push   si     ;save si
06FF  33 C0                          xor    ax,ax
                             ; Build address request packet
0701  50                             push   ax     ;sector 4th word
0702  50                             push   ax     ;sector 3rd word
0703  FF 74 0A                       push   [si+10];sector 2nd word
0706  FF 74 08                       push   [si+8] ;sector low word
0709  06                             push   es     ;buffer segment
070A  53                             push   bx     ;buffer offset
070B  40                             inc    ax
070C  50                             push   ax     ;number of sectors (max.(7F)
070D  B0 10                          mov    al,10h ;packet size
070F  50                             push   ax     ;high byte reserved (=0)
0710  8B F4                          mov    si,sp  ;DS:SI points to request address packet
0712  B4 42                          mov    ah,42h ;extended disk read; DL has drive number
0714  CD 13                          int    13h
0716  72 04                          jc     skp_ck ;if CF then AH=errorcode else AH=0
                             ; Check sector count read, as CF is not set if sector not found error
0718  83 7C 02 01                    cmp    word ptr[si+2],1      ;also need to check actual count
071C  8D 64 0E               skp_ck: lea    sp,[si+14]     ;purge address request packet from stack
071F  58                             pop    ax     ;and last word of package into AX
0720  5E                             pop    si     ;restore si
0721  73 1B                          jnc    readok
0723  CD 13                          int    13h    ;reset disk system
0725  E2 D7                          loop   retrlb
0727  BE 078E R              rdfail: mov    si,offset err_msg
072A  E8 063E R              abend:  call   tty
072D  EB FB                          jmp    short abend    ;loop on last 0

                             ; Read bootrecord of active partition
072F  8B DC                  rdchs:  mov    bx,sp  ;set ES:BX to buffer address 7C00h
```

```
0731  8A 74 01                      mov     dh,[si+1]      ;set head number, DL still has drive number
0734  8B 4C 02                      mov     cx,[si+2]      ;set sector & cyl
0737  B4 02                         mov     ah,2    ;read partition bootsector
0739  E8 0621 R                     call    int13
073C  72 E9                         jc      rdfail
073E  81 BF 01FE AA55        readok: cmp     word ptr [bx+(bootid-mbr)],0AA55h
0744  75 02                         jne     noboot
0746  FF E3                         jmp     bx      ;execute partitions boot record
                             ; Upon exit DS:SI points to booted partition table entry

0748  BE 0794 R              noboot: mov     si,offset mis_msg
074B  8B FE                         mov     di,si
074D  B8 694D                       mov     ax,'iM'
0750  AB                            stosw           ;replace first characters with 'Miss'
0751  B8 7373                       mov     ax,'ss'
0754  AB                            stosw
0755  EB D3                         jmp     short abend

0757  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
0776  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
078E  45 72 72 6F 72 20      err_msg db      'Error '       ;loading operating system'
0794  6C 6F 61 64 69 6E      mis_msg db      'loading operating system' ;first 4 characters could be patched
      67 20 6F 70 65 72
      61 74 69 6E 67 20
      73 79 73 74 65 6D
07AC  0D 0A                  crlf    db      cr,lf   ;should be terminated by zero
07AE     07 [                null    db      1B5h-($-mbr) dup (0)   ;fill unused space
```

```
               00
                   ]

07B5  76 8E 94                       db     low offset inv_msg, low offset err_msg, low offset mis_msg
07B8     06 [                        db     6 dup (0)      ;Windows NT signature
               00
                   ]

07BE     40 [                table   db     64 dup (0)
               00
                   ]

07FE  AA55                   bootid  dw     0AA55h
0800                         mbr     endp
0800                         tbootsel ends
                                     end
```