

```

1  org 0x7c00
2
3  BaseOfStack equ 0x7c00
4
5  BaseOfLoader equ 0x1000
6  OffsetOfLoader equ 0x00
7
8  RootDirSectors equ 14
9  SectorNumOfRootDirStart equ 19
10 SectorNumOfFAT1Start equ 1
11 SectorBalance equ 17
12
13     jmp short Label_Start
14     nop
15     BS_OEMName db 'MINEboot'
16     BPB_BytesPerSec dw 512
17     BPB_SecPerClus db 1
18     BPB_RsvdSecCnt dw 1
19     BPB_NumFATs db 2
20     BPB_RootEntCnt dw 224
21     BPB_TotSec16 dw 2880
22     BPB_Media db 0xf0
23     BPB_FATSz16 dw 9
24     BPB_SecPerTrk dw 18
25     BPB_NumHeads dw 2
26     BPB_HiddSec dd 0
27     BPB_TotSec32 dd 0
28     BS_DrvNum db 0
29     BS_Reserved1 db 0
30     BS_BootSig db 0x29
31     BS_VolID dd 0
32     BS_VolLab db 'boot loader'
33     BS_FileSysType db 'FAT12 '
34
35 Label_Start:
36
37     mov ax, cs
38     mov ds, ax
39     mov es, ax
40     mov ss, ax
41     mov sp, BaseOfStack
42
43     ;===== clear screen
44
45     mov ax, 0600h
46     mov bx, 0700h
47     mov cx, 0
48     mov dx, 0184fh
49     int 10h
50
51     ;===== set focus
52
53     mov ax, 0200h
54     mov bx, 0000h
55     mov dx, 0000h
56     int 10h
57
58     ;===== display on screen : Start Booting.....
59
60     mov ax, 1301h
61     mov bx, 000fh
62     mov dx, 0000h
63     mov cx, 10
64     push ax
65     mov ax, ds
66     mov es, ax
67     pop ax
68     mov bp, StartBootMessage
69     int 10h
70
71     ;===== reset floppy

```

```

72
73     xor ah, ah
74     xor dl, dl
75     int 13h
76
77 ;===== search loader.bin
78     mov word    [SectorNo], SectorNumOfRootDirStart
79
80 Lable_Search_In_Root_Dir_Begin:
81
82     cmp word    [RootDirSizeForLoop], 0
83     jz  Label_No_LoaderBin
84     dec word    [RootDirSizeForLoop]
85     mov ax, 00h
86     mov es, ax
87     mov bx, 8000h
88     mov ax, [SectorNo]
89     mov cl, 1
90     call Func_ReadOneSector
91     mov si, LoaderFileName
92     mov di, 8000h
93     cld
94     mov dx, 10h
95
96 Label_Search_For_LoaderBin:
97
98     cmp dx, 0
99     jz  Label_Goto_Next_Sector_In_Root_Dir
100    dec dx
101    mov cx, 11
102
103 Label_Cmp_FileName:
104
105    cmp cx, 0
106    jz  Label_FileName_Found
107    dec cx
108    lodsb
109    cmp al, byte    [es:di]
110    jz  Label_Go_On
111    jmp Label_Different
112
113 Label_Go_On:
114
115    inc di
116    jmp Label_Cmp_FileName
117
118 Label_Different:
119
120    and di, 0ffe0h
121    add di, 20h
122    mov si, LoaderFileName
123    jmp Label_Search_For_LoaderBin
124
125 Label_Goto_Next_Sector_In_Root_Dir:
126
127    add word    [SectorNo], 1
128    jmp Lable_Search_In_Root_Dir_Begin
129
130 ;===== display on screen : ERROR:No LOADER Found
131
132 Label_No_LoaderBin:
133
134    mov ax, 1301h
135    mov bx, 008ch
136    mov dx, 0100h
137    mov cx, 21
138    push ax
139    mov ax, ds
140    mov es, ax
141    pop ax
142    mov bp, NoLoaderMessage

```

```

143     int 10h
144     jmp $
145
146 ;===== found loader.bin name in root director struct
147
148 Label_FileName_Found:
149
150     mov ax, RootDirSectors
151     and di, 0ffe0h
152     add di, 01ah
153     mov cx, word [es:di]
154     push cx
155     add cx, ax
156     add cx, SectorBalance
157     mov ax, BaseOfLoader
158     mov es, ax
159     mov bx, OffsetOfLoader
160     mov ax, cx
161
162 Label_Go_On_Loading_File:
163     push ax
164     push bx
165     mov ah, 0eh
166     mov al, '.'
167     mov bl, 0fh
168     int 10h
169     pop bx
170     pop ax
171
172     mov cl, 1
173     call Func_ReadOneSector
174     pop ax
175     call Func_GetFATEntry
176     cmp ax, 0ffffh
177     jz Label_File_Loaded
178     push ax
179     mov dx, RootDirSectors
180     add ax, dx
181     add ax, SectorBalance
182     add bx, [BPB_BytesPerSec]
183     jmp Label_Go_On_Loading_File
184
185 Label_File_Loaded:
186
187     jmp BaseOfLoader:OffsetOfLoader
188
189 ;===== read one sector from floppy
190
191 Func_ReadOneSector:
192
193     push bp
194     mov bp, sp
195     sub esp, 2
196     mov byte [bp - 2], cl
197     push bx
198     mov bl, [BPB_SecPerTrk]
199     div bl
200     inc ah
201     mov cl, ah
202     mov dh, al
203     shr al, 1
204     mov ch, al
205     and dh, 1
206     pop bx
207     mov dl, [BS_DrvNum]
208 Label_Go_On_Reading:
209     mov ah, 2
210     mov al, byte [bp - 2]
211     int 13h
212     jc Label_Go_On_Reading
213     add esp, 2

```

```

214     pop bp
215     ret
216
217 ;=====    get FAT Entry
218
219 Func_GetFATEntry:
220
221     push    es
222     push    bx
223     push    ax
224     mov ax, 00
225     mov es, ax
226     pop ax
227     mov byte [Odd], 0
228     mov bx, 3
229     mul bx
230     mov bx, 2
231     div bx
232     cmp dx, 0
233     jz  Label_Even
234     mov byte [Odd], 1
235
236 Label_Even:
237
238     xor dx, dx
239     mov bx, [BPB_BytesPerSec]
240     div bx
241     push    dx
242     mov bx, 8000h
243     add ax, SectorNumOfFAT1Start
244     mov cl, 2
245     call    Func_ReadOneSector
246
247     pop dx
248     add bx, dx
249     mov ax, [es:bx]
250     cmp byte [Odd], 1
251     jnz Label_Even_2
252     shr ax, 4
253
254 Label_Even_2:
255     and ax, 0ffffh
256     pop bx
257     pop es
258     ret
259
260 ;=====    tmp variable
261
262 RootDirSizeForLoop    dw    RootDirSectors
263 SectorNo              dw    0
264 Odd                  db    0
265
266 ;=====    display messages
267
268 StartBootMessage:    db    "NKU Start"
269 NoLoaderMessage:     db    "ERROR:No LOADER Found"
270 LoaderFileName:      db    "LOADER  BIN",0
271
272 ;=====    fill zero until whole sector
273
274     times 510 - ($ - $$)    db    0
275     dw    0xaa55
276
277

```