```
1           ;代码清单13-1
2           ;文件名：c13_mbr.asm
3           ;文件说明：硬盘主引导扇区代码
4           ;创建日期：2011-10-28 22:35         ;设置堆栈段和栈指针
5
6           core_base_address equ 0x00040000    ;常数，内核加载的起始内存地址
7           core_start_sector equ 0x00000001    ;常数，内核的起始逻辑扇区号
8
9           mov ax,cs
10          mov ss,ax
11          mov sp,0x7c00
12
13          ;计算GDT所在的逻辑段地址
14          mov eax,[cs:pgdt+0x7c00+0x02]       ;GDT的32位物理地址
15          xor edx,edx
16          mov ebx,16
17          div ebx                             ;分解成16位逻辑地址
18
19          mov ds,eax                          ;令DS指向该段以进行操作
20          mov ebx,edx                         ;段内起始偏移地址
21
22          ;跳过0#号描述符的槽位
23          ;创建1#描述符，这是一个数据段，对应0~4GB的线性地址空间
24          mov dword [ebx+0x08],0x0000ffff     ;基地址为0，段界限为0xFFFFF
25          mov dword [ebx+0x0c],0x00cf9200     ;粒度为4KB，存储器段描述符
26
27          ;创建保护模式下初始代码段描述符
28          mov dword [ebx+0x10],0x7c0001ff     ;基地址为0x00007c00，界限0x1FF
29          mov dword [ebx+0x14],0x00409800     ;粒度为1个字节，代码段描述符
30
31          ;建立保护模式下的堆栈段描述符       ;基地址为0x00007C00，界限0xFFFFE
32          mov dword [ebx+0x18],0x7c00fffe     ;粒度为4KB
33          mov dword [ebx+0x1c],0x00cf9600
34
35          ;建立保护模式下的显示缓冲区描述符
36          mov dword [ebx+0x20],0x80007fff     ;基地址为0x000B8000，界限0x07FFF
37          mov dword [ebx+0x24],0x0040920b     ;粒度为字节
38
39          ;初始化描述符表寄存器GDTR
40          mov word [cs: pgdt+0x7c00],39       ;描述符表的界限
41
42          lgdt [cs: pgdt+0x7c00]
43
44          in al,0x92                          ;南桥芯片内的端口
45          or al,0000_0010B
46          out 0x92,al                         ;打开A20
47
48          cli                                 ;中断机制尚未工作
49
50          mov eax,cr0
51          or eax,1
52          mov cr0,eax                         ;设置PE位
53
```

```
54          ;以下进入保护模式... ...
55          jmp dword 0x0010:flush              ;16位的描述符选择子：32位偏移
56                                              ;清流水线并串行化处理器
57          [bits 32]
58   flush:
59          mov eax,0x0008                      ;加载数据段(0..4GB)选择子
60          mov ds,eax
61
62          mov eax,0x0018                      ;加载堆栈段选择子
63          mov ss,eax
64          xor esp,esp                         ;堆栈指针 <- 0
65
66          ;以下加载系统核心程序
67          mov edi,core_base_address
68
69          mov eax,core_start_sector
70          mov ebx,edi                         ;起始地址
71          call read_hard_disk_0               ;以下读取程序的起始部分（一个扇区）
72
73          ;以下判断整个程序有多大
74          mov eax,[edi]                       ;核心程序尺寸
75          xor edx,edx
76          mov ecx,512                         ;512字节每扇区
77          div ecx
78
79          or edx,edx
80          jnz @1                              ;未除尽，因此结果比实际扇区数少1
81          dec eax                             ;已经读了一个扇区，扇区总数减1
82    @1:
83          or eax,eax                          ;考虑实际长度≤512个字节的情况
84          jz setup                            ;EAX=0 ?
85
86          ;读取剩余的扇区
87          mov ecx,eax                         ;32位模式下的LOOP使用ECX
88          mov eax,core_start_sector
89          inc eax                             ;从下一个逻辑扇区接着读
90    @2:
91          call read_hard_disk_0
92          inc eax
93          loop @2                             ;循环读，直到读完整个内核
94
95   setup:
96          mov esi,[0x7c00+pgdt+0x02]          ;不可以在代码段内寻址pgdt，但可以
97                                              ;通过4GB的段来访问
98          ;建立公用例程段描述符
99          mov eax,[edi+0x04]                  ;公用例程代码段起始汇编地址
100         mov ebx,[edi+0x08]                  ;核心数据段汇编地址
101         sub ebx,eax
102         dec ebx                             ;公用例程段界限
103         add eax,edi                         ;公用例程段基地址
104         mov ecx,0x00409800                  ;字节粒度的代码段描述符
105         call make_gdt_descriptor
106         mov [esi+0x28],eax
```

```
107         mov [esi+0x2c],edx
108
109         ;建立核心数据段描述符
110         mov eax,[edi+0x08]                  ;核心数据段起始汇编地址
111         mov ebx,[edi+0x0c]                  ;核心代码段汇编地址
112         sub ebx,eax
113         dec ebx                             ;核心数据段界限
114         add eax,edi                         ;核心数据段基地址
115         mov ecx,0x00409200                  ;字节粒度的数据段描述符
116         call make_gdt_descriptor
117         mov [esi+0x30],eax
118         mov [esi+0x34],edx
119
120         ;建立核心代码段描述符
121         mov eax,[edi+0x0c]                  ;核心代码段起始汇编地址
122         mov ebx,[edi+0x00]                  ;程序总长度
123         sub ebx,eax
124         dec ebx                             ;核心代码段界限
125         add eax,edi                         ;核心代码段基地址
126         mov ecx,0x00409800                  ;字节粒度的代码段描述符
127         call make_gdt_descriptor
128         mov [esi+0x38],eax
129         mov [esi+0x3c],edx
130
131         mov word [0x7c00+pgdt],63           ;描述符表的界限
132
133         lgdt [0x7c00+pgdt]
134
135         jmp far [edi+0x10]
136
137 ;-------------------------------------------------------------------------------
138 read_hard_disk_0:                           ;从硬盘读取一个逻辑扇区
139                                             ;EAX=逻辑扇区号
140                                             ;DS:EBX=目标缓冲区地址
141                                             ;返回：EBX=EBX+512
142         push eax
143         push ecx
144         push edx
145
146         push eax
147
148         mov dx,0x1f2
149         mov al,1
150         out dx,al                           ;读取的扇区数
151
152         inc dx                              ;0x1f3
153         pop eax
154         out dx,al                           ;LBA地址7~0
155
156         inc dx                              ;0x1f4
157         mov cl,8
158         shr eax,cl
159         out dx,al                           ;LBA地址15~8
```

```
160
161         inc dx                          ;0x1f5
162         shr eax,cl
163         out dx,al                       ;LBA地址23~16
164
165         inc dx                          ;0x1f6
166         shr eax,cl
167         or al,0xe0                      ;第一硬盘  LBA地址27~24
168         out dx,al
169
170         inc dx                          ;0x1f7
171         mov al,0x20                     ;读命令
172         out dx,al
173
174   .waits:
175         in al,dx
176         and al,0x88
177         cmp al,0x08
178         jnz .waits                      ;不忙，且硬盘已准备好数据传输
179
180         mov ecx,256                     ;总共要读取的字数
181         mov dx,0x1f0
182   .readw:
183         in ax,dx
184         mov [ebx],ax
185         add ebx,2
186         loop .readw
187
188         pop edx
189         pop ecx
190         pop eax
191
192         ret
193
194 ;-------------------------------------------------------------------------------
195 make_gdt_descriptor:                    ;构造描述符
196                                         ;输入：EAX=线性基地址
197                                         ;       EBX=段界限
198                                         ;       ECX=属性（各属性位都在原始
199                                         ;       位置，其它没用到的位置0）
200                                         ;返回：EDX:EAX=完整的描述符
201         mov edx,eax
202         shl eax,16
203         or ax,bx                        ;描述符前32位(EAX)构造完毕
204
205         and edx,0xffff0000              ;清除基地址中无关的位
206         rol edx,8
207         bswap edx                       ;装配基址的31~24和23~16  (80486+)
208
209         xor bx,bx
210         or edx,ebx                      ;装配段界限的高4位
211
212         or edx,ecx                      ;装配属性
```

```
213
214          ret
215
216 ;-------------------------------------------------------------------------------
217          pgdt             dw 0
218                           dd 0x00007e00      ;GDT的物理地址
219 ;-------------------------------------------------------------------------------
220          times 510-($-$$) db 0
221                           db 0x55,0xaa
```