```
1           ;代码清单17-1
2           ;文件名：c17_mbr.asm
3           ;文件说明：硬盘主引导扇区代码
4           ;创建日期：2012-07-13 11:20         ;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  ;=============================================================================
10 SECTION  mbr  vstart=0x00007c00
11
12          mov ax,cs
13          mov ss,ax
14          mov sp,0x7c00
15
16          ;计算GDT所在的逻辑段地址
17          mov eax,[cs:pgdt+0x02]              ;GDT的32位物理地址
18          xor edx,edx
19          mov ebx,16
20          div ebx                             ;分解成16位逻辑地址
21
22          mov ds,eax                          ;令DS指向该段以进行操作
23          mov ebx,edx                         ;段内起始偏移地址
24
25          ;跳过0#号描述符的槽位
26          ;创建1#描述符，保护模式下的代码段描述符
27          mov dword [ebx+0x08],0x0000ffff     ;基地址为0，界限0xFFFFF，DPL=00
28          mov dword [ebx+0x0c],0x00cf9800     ;4KB粒度，代码段描述符，向上扩展
29
30          ;创建2#描述符，保护模式下的数据段和堆栈段描述符
31          mov dword [ebx+0x10],0x0000ffff     ;基地址为0，界限0xFFFFF，DPL=00
32          mov dword [ebx+0x14],0x00cf9200     ;4KB粒度，数据段描述符，向上扩展
33
34          ;初始化描述符表寄存器GDTR
35          mov word [cs: pgdt],23              ;描述符表的界限
36
37          lgdt [cs: pgdt]
38
39          in al,0x92                          ;南桥芯片内的端口
40          or al,0000_0010B
41          out 0x92,al                         ;打开A20
42
43          cli                                 ;中断机制尚未工作
44
45          mov eax,cr0
46          or eax,1
47          mov cr0,eax                         ;设置PE位
48
49          ;以下进入保护模式... ...
50          jmp dword 0x0008:flush              ;16位的描述符选择子：32位偏移
51                                              ;清流水线并串行化处理器
52          [bits 32]
53   flush:
```

```
54          mov eax,0x00010                         ;加载数据段(4GB)选择子
55          mov ds,eax
56          mov es,eax
57          mov fs,eax
58          mov gs,eax
59          mov ss,eax                              ;加载堆栈段(4GB)选择子
60          mov esp,0x7000                          ;堆栈指针
61
62          ;以下加载系统核心程序
63          mov edi,core_base_address
64
65          mov eax,core_start_sector
66          mov ebx,edi                             ;起始地址
67          call read_hard_disk_0                   ;以下读取程序的起始部分（一个扇区）
68
69          ;以下判断整个程序有多大
70          mov eax,[edi]                           ;核心程序尺寸
71          xor edx,edx
72          mov ecx,512                             ;512字节每扇区
73          div ecx
74
75          or edx,edx
76          jnz @1                                  ;未除尽，因此结果比实际扇区数少1
77          dec eax                                 ;已经读了一个扇区，扇区总数减1
78    @1:
79          or eax,eax                              ;考虑实际长度≤512个字节的情况
80          jz pge                                  ;EAX=0 ?
81
82          ;读取剩余的扇区
83          mov ecx,eax                             ;32位模式下的LOOP使用ECX
84          mov eax,core_start_sector
85          inc eax                                 ;从下一个逻辑扇区接着读
86    @2:
87          call read_hard_disk_0
88          inc eax
89          loop @2                                 ;循环读，直到读完整个内核
90
91    pge:
92          ;准备打开分页机制。从此，再也不用在段之间转来转去，实在晕乎~
93
94          ;创建系统内核的页目录表PDT
95          mov ebx,0x00020000                      ;页目录表PDT的物理地址
96
97          ;在页目录内创建指向页目录表自己的目录项
98          mov dword [ebx+4092],0x00020003
99
100         mov edx,0x00021003                      ;MBR空间有限，后面尽量不使用立即数
101         ;在页目录内创建与线性地址0x00000000对应的目录项
102         mov [ebx+0x000],edx                     ;写入目录项（页表的物理地址和属性）
103                                                 ;此目录项仅用于过渡。
104         ;在页目录内创建与线性地址0x80000000对应的目录项
105         mov [ebx+0x800],edx                     ;写入目录项（页表的物理地址和属性）
106
```

```
107         ;创建与上面那个目录项相对应的页表，初始化页表项
108         mov ebx,0x00021000                  ;页表的物理地址
109         xor eax,eax                         ;起始页的物理地址
110         xor esi,esi
111   .b1:
112         mov edx,eax
113         or edx,0x00000003
114         mov [ebx+esi*4],edx                 ;登记页的物理地址
115         add eax,0x1000                      ;下一个相邻页的物理地址
116         inc esi
117         cmp esi,256                         ;仅低端1MB内存对应的页才是有效的
118         jl .b1
119
120         ;令CR3寄存器指向页目录，并正式开启页功能
121         mov eax,0x00020000                  ;PCD=PWT=0
122         mov cr3,eax
123
124         ;将GDT的线性地址映射到从0x80000000开始的相同位置
125         sgdt [pgdt]
126         mov ebx,[pgdt+2]
127         add dword [pgdt+2],0x80000000       ;GDTR也用的是线性地址
128         lgdt [pgdt]
129
130         mov eax,cr0
131         or eax,0x80000000
132         mov cr0,eax                         ;开启分页机制
133
134         ;将堆栈映射到高端，这是非常容易被忽略的一件事。应当把内核的所有东西
135         ;都移到高端，否则，一定会和正在加载的用户任务局部空间里的内容冲突，
136         ;而且很难想到问题会出在这里。
137         add esp,0x80000000
138
139         jmp [0x80040004]
140
141 ;-------------------------------------------------------------------------------
142 read_hard_disk_0:                           ;从硬盘读取一个逻辑扇区
143                                             ;EAX=逻辑扇区号
144                                             ;DS:EBX=目标缓冲区地址
145                                             ;返回：EBX=EBX+512
146         push eax
147         push ecx
148         push edx
149
150         push eax
151
152         mov dx,0x1f2
153         mov al,1
154         out dx,al                           ;读取的扇区数
155
156         inc dx                              ;0x1f3
157         pop eax
158         out dx,al                           ;LBA地址7~0
159
```

```
160         inc dx                                  ;0x1f4
161         mov cl,8
162         shr eax,cl
163         out dx,al                               ;LBA地址15~8
164
165         inc dx                                  ;0x1f5
166         shr eax,cl
167         out dx,al                               ;LBA地址23~16
168
169         inc dx                                  ;0x1f6
170         shr eax,cl
171         or al,0xe0                              ;第一硬盘  LBA地址27~24
172         out dx,al
173
174         inc dx                                  ;0x1f7
175         mov al,0x20                             ;读命令
176         out dx,al
177
178   .waits:
179         in al,dx
180         and al,0x88
181         cmp al,0x08
182         jnz .waits                              ;不忙，且硬盘已准备好数据传输
183
184         mov ecx,256                             ;总共要读取的字数
185         mov dx,0x1f0
186   .readw:
187         in ax,dx
188         mov [ebx],ax
189         add ebx,2
190         loop .readw
191
192         pop edx
193         pop ecx
194         pop eax
195
196         ret
197
198 ;-------------------------------------------------------------------------------
199         pgdt             dw 0
200                          dd 0x00008000      ;GDT的物理/线性地址
201 ;-------------------------------------------------------------------------------
202         times 510-($-$$) db 0
203                          db 0x55,0xaa
```