```
1
          ;代码清单8-1
          ;文件名: c08 mbr.asm
2
          ;文件说明: 硬盘主引导扇区代码(加载程序)
3
4
          ;创建日期: 2011-5-5 18:17
5
                                      ;声明常数(用户程序起始逻辑扇区号)
6
          app_lba_start equ 100
 7
                                       ;常数的声明不会占用汇编地址
8
9 SECTION mbr align=16 vstart=0x7c00
10
          ;设置堆栈段和栈指针
11
          mov ax, 0
12
13
          mov ss,ax
14
          mov sp,ax
15
                                       ;计算用于加载用户程序的逻辑段地址
16
          mov ax,[cs:phy base]
17
          mov dx, [cs:phy base+0x02]
          mov bx,16
18
19
          div bx
20
                                       ;令DS和ES指向该段以进行操作
          mov ds, ax
21
          mov es,ax
22
23
          ;以下读取程序的起始部分
          xor di, di
24
                                       ;程序在硬盘上的起始逻辑扇区号
25
          mov si, app lba start
                                       ;加载到DS:0x0000处
26
          xor bx,bx
          call read hard disk 0
27
28
          ;以下判断整个程序有多大
29
30
          mov dx, [2]
                                       ;曾经把dx写成了ds,花了二十分钟排错
          mov ax, [0]
31
32
          mov bx, 512
                                       ;512字节每扇区
33
          div bx
34
          cmp dx, 0
35
          jnz @1
                                       ;未除尽,因此结果比实际扇区数少1
                                       ;已经读了一个扇区,扇区总数减1
36
          dec ax
     @1:
37
                                       ;考虑实际长度小于等于512个字节的情况
38
          cmp ax, 0
39
          jz direct
40
41
          ;读取剩余的扇区
                                       ;以下要用到并改变DS寄存器
42
          push ds
43
44
                                       ;循环次数(剩余扇区数)
          mov cx, ax
45
     @2:
46
          mov ax, ds
47
          add ax,0x20
                                       ;得到下一个以512字节为边界的段地址
48
          mov ds, ax
49
50
          xor bx,bx
                                       ;每次读时,偏移地址始终为0x0000
51
                                       ;下一个逻辑扇区
          inc si
          call read hard disk 0
52
                                       ;循环读,直到读完整个功能程序
53
          loop @2
```

```
54
 55
                                          ;恢复数据段基址到用户程序头部段
           pop ds
 56
 57
           ;计算入口点代码段基址
 58
      direct:
 59
           mov dx, [0x08]
 60
           mov ax, [0x06]
           call calc_segment_base
 61
                                          ;回填修正后的入口点代码段基址
 62
           mov [0x06], ax
 63
 64
           ;开始处理段重定位表
                                          ;需要重定位的项目数量
 65
           mov cx, [0x0a]
 66
                                          ;重定位表首地址
           mov bx,0x0c
 67
 68 realloc:
                                          ;32位地址的高16位
 69
          mov dx, [bx+0x02]
70
          mov ax, [bx]
 71
           call calc segment base
 72
                                          ;回填段的基址
           mov [bx],ax
73
                                          ;下一个重定位项(每项占4个字节)
           add bx, 4
 74
           loop realloc
 75
76
                                          ;转移到用户程序
           jmp far [0x04]
77
                                          ;从硬盘读取一个逻辑扇区
79 read hard disk 0:
80
                                          ;输入: DI:SI=起始逻辑扇区号
81
                                          ; DS:BX=目标缓冲区地址
 82
           push ax
 83
           push bx
 84
           push cx
 85
           push dx
 86
 87
           mov dx,0x1f2
 88
           mov al, 1
           out dx,al
                                          ;读取的扇区数
 89
 90
 91
           inc dx
                                          ;0x1f3
           mov ax, si
 92
 93
                                          ;LBA地址7~0
           out dx, al
 94
           inc dx
 95
                                          ;0x1f4
           mov al, ah
 96
 97
                                          ;LBA地址15~8
           out dx, al
98
99
           inc dx
                                          ;0x1f5
100
           mov ax, di
101
           out dx, al
                                          ;LBA地址23~16
102
103
           inc dx
                                          ;0x1f6
104
                                          ;LBA28模式,主盘
           mov al, 0xe0
                                          ;LBA地址27~24
105
           or al, ah
106
           out dx, al
```

```
107
108
           inc dx
                                        ;0x1f7
109
           mov al, 0x20
                                        ;读命令
110
           out dx, al
111
112 .waits:
113
           in al, dx
           and al, 0x88
114
           cmp al, 0x08
115
                                        ;不忙,且硬盘已准备好数据传输
116
           jnz .waits
117
                                        ;总共要读取的字数
118
           mov cx, 256
119
           mov dx,0x1f0
120 .readw:
121
           in ax,dx
122
           mov [bx],ax
123
           add bx, 2
124
           loop .readw
125
126
           pop dx
127
           pop cx
128
           pop bx
129
           pop ax
130
131
           ret
132
                                       ;计算16位段地址
134 calc segment base:
                                        ;输入: DX:AX=32位物理地址
135
136
                                        ;返回: AX=16位段基地址
137
           push dx
138
139
           add ax, [cs:phy base]
140
           adc dx,[cs:phy_base+0x02]
141
           shr ax,4
142
           ror dx,4
           and dx,0xf000
143
           or ax, dx
144
145
146
           pop dx
147
148
           ret
149
150 ;-----
151
                                       ;用户程序被加载的物理起始地址
           phy_base_dd 0x10000
152
153 times 510-(\$-\$\$) db 0
154
                   db 0x55,0xaa
```