```
1           ;代码清单8-2
2           ;文件名：c08.asm
3           ;文件说明：用户程序
4           ;创建日期：2011-5-5 18:17
5
6  ;===============================================================================
7  SECTION header vstart=0                     ;定义用户程序头部段
8      program_length  dd program_end          ;程序总长度[0x00]
9
10     ;用户程序入口点
11     code_entry      dw start                ;偏移地址[0x04]
12                     dd section.code_1.start ;段地址[0x06]
13
14     realloc_tbl_len dw (header_end-code_1_segment)/4
15                                             ;段重定位表项个数[0x0a]
16
17     ;段重定位表
18     code_1_segment  dd section.code_1.start ;[0x0c]
19     code_2_segment  dd section.code_2.start ;[0x10]
20     data_1_segment  dd section.data_1.start ;[0x14]
21     data_2_segment  dd section.data_2.start ;[0x18]
22     stack_segment   dd section.stack.start  ;[0x1c]
23
24     header_end:
25
26 ;===============================================================================
27 SECTION code_1 align=16 vstart=0          ;定义代码段1（16字节对齐）
28 put_string:                               ;显示串(0结尾)。
29                                           ;输入：DS:BX=串地址
30          mov cl,[bx]
31          or cl,cl                         ;cl=0 ?
32          jz .exit                         ;是的，返回主程序
33          call put_char
34          inc bx                           ;下一个字符
35          jmp put_string
36
37    .exit:
38          ret
39
40 ;-------------------------------------------------------------------------------
41 put_char:                                 ;显示一个字符
42                                           ;输入：cl=字符ascii
43          push ax
44          push bx
45          push cx
46          push dx
47          push ds
48          push es
49
50          ;以下取当前光标位置
51          mov dx,0x3d4
52          mov al,0x0e
53          out dx,al
```

```
 54          mov dx,0x3d5
 55          in al,dx                        ;高8位
 56          mov ah,al
 57
 58          mov dx,0x3d4
 59          mov al,0x0f
 60          out dx,al
 61          mov dx,0x3d5
 62          in al,dx                        ;低8位
 63          mov bx,ax                       ;BX=代表光标位置的16位数
 64
 65          cmp cl,0x0d                     ;回车符?
 66          jnz .put_0a                     ;不是。看看是不是换行等字符
 67          mov ax,bx                       ;此句略显多余，但去掉后还得改书，麻烦
 68          mov bl,80
 69          div bl
 70          mul bl
 71          mov bx,ax
 72          jmp .set_cursor
 73
 74  .put_0a:
 75          cmp cl,0x0a                     ;换行符?
 76          jnz .put_other                  ;不是，那就正常显示字符
 77          add bx,80
 78          jmp .roll_screen
 79
 80  .put_other:                             ;正常显示字符
 81          mov ax,0xb800
 82          mov es,ax
 83          shl bx,1
 84          mov [es:bx],cl
 85
 86          ;以下将光标位置推进一个字符
 87          shr bx,1
 88          add bx,1
 89
 90  .roll_screen:
 91          cmp bx,2000                     ;光标超出屏幕？滚屏
 92          jl .set_cursor
 93
 94          mov ax,0xb800
 95          mov ds,ax
 96          mov es,ax
 97          cld
 98          mov si,0xa0
 99          mov di,0x00
100          mov cx,1920
101          rep movsw
102          mov bx,3840                     ;清除屏幕最底一行
103          mov cx,80
104  .cls:
105          mov word[es:bx],0x0720
106          add bx,2
```

```
107         loop .cls
108
109         mov bx,1920
110
111 .set_cursor:
112         mov dx,0x3d4
113         mov al,0x0e
114         out dx,al
115         mov dx,0x3d5
116         mov al,bh
117         out dx,al
118         mov dx,0x3d4
119         mov al,0x0f
120         out dx,al
121         mov dx,0x3d5
122         mov al,bl
123         out dx,al
124
125         pop es
126         pop ds
127         pop dx
128         pop cx
129         pop bx
130         pop ax
131
132         ret
133
134 ;-------------------------------------------------------------------------------
135  start:
136         ;初始执行时，DS和ES指向用户程序头部段
137         mov ax,[stack_segment]            ;设置到用户程序自己的堆栈
138         mov ss,ax
139         mov sp,stack_end
140
141         mov ax,[data_1_segment]           ;设置到用户程序自己的数据段
142         mov ds,ax
143
144         mov bx,msg0
145         call put_string                   ;显示第一段信息
146
147         push word [es:code_2_segment]
148         mov ax,begin
149         push ax                           ;可以直接push begin,80386+
150
151         retf                              ;转移到代码段2执行
152
153  continue:
154         mov ax,[es:data_2_segment]        ;段寄存器DS切换到数据段2
155         mov ds,ax
156
157         mov bx,msg1
158         call put_string                   ;显示第二段信息
159
```

```
160          jmp $
161
162 ;=============================================================================
163 SECTION code_2 align=16 vstart=0          ;定义代码段2（16字节对齐）
164
165   begin:
166          push word [es:code_1_segment]
167          mov ax,continue
168          push ax                          ;可以直接push continue,80386+
169
170          retf                             ;转移到代码段1接着执行
171
172 ;=============================================================================
173 SECTION data_1 align=16 vstart=0
174
175     msg0 db '  This is NASM - the famous Netwide Assembler. '
176          db 'Back at SourceForge and in intensive development! '
177          db 'Get the current versions from http://www.nasm.us/.'
178          db 0x0d,0x0a,0x0d,0x0a
179          db '  Example code for calculate 1+2+...+1000:',0x0d,0x0a,0x0d,0x0a
180          db '     xor dx,dx',0x0d,0x0a
181          db '     xor ax,ax',0x0d,0x0a
182          db '     xor cx,cx',0x0d,0x0a
183          db '  @@:',0x0d,0x0a
184          db '     inc cx',0x0d,0x0a
185          db '     add ax,cx',0x0d,0x0a
186          db '     adc dx,0',0x0d,0x0a
187          db '     inc cx',0x0d,0x0a
188          db '     cmp cx,1000',0x0d,0x0a
189          db '     jle @@',0x0d,0x0a
190          db '     ... ...(Some other codes)',0x0d,0x0a,0x0d,0x0a
191          db 0
192
193 ;=============================================================================
194 SECTION data_2 align=16 vstart=0
195
196     msg1 db '  The above contents is written by LeeChung. '
197          db '2011-05-06'
198          db 0
199
200 ;=============================================================================
201 SECTION stack align=16 vstart=0
202
203          resb 256
204
205 stack_end:
206
207 ;=============================================================================
208 SECTION trail align=16
209 program_end:
```