

# chapter16

## 代码

```
1  assume cs:code,ds:data,ss:stack
2
3  stack segment
4      db 128 dup (0)
5  stack ends
6
7  data segment
8  data ends
9
10 code segment
11 start:
12
13     mov ax,stack
14     mov ss,ax
15     mov sp,128
16
17     call install_int
18
19     mov al,4
20     mov ah,3
21     int 7ch
22
23     mov ax,4c00h
24     int 21h
25
26 ;===== install_int =====
27 install_int:
28     install_int_bg:
29         mov ax,0
30         mov es,ax
31
32         cli
33         mov word ptr es:[7ch*4],200h
34         mov word ptr es:[7ch*4+2],0
35         sti
36
37
38         ;load
39         push cs
40         pop ds
41
42         mov ax,0
43         mov es,ax
44
45         mov si,offset int7ch
46         mov di,200h
47         mov cx,offset int7ch_end - offset int7ch
48         cld
49         rep movsb
```

```

50
51     install_int_end:
52         ret
53
54 ;===== int7ch =====
55 int7ch:
56     jmp int7ch_code
57
58     Mytable dw offset clear_screen - offset int7ch + 200h
59             dw offset set_front_color - offset int7ch + 200h
60             dw offset set_background_color - offset int7ch + 200h
61             dw offset move_up - offset int7ch + 200h
62 ;=====
63 int7ch_code:
64     push ax
65     push bx
66     push es
67     int7ch_bg:
68         ; call clear_screen
69         ; call set_front_color
70         ; call set_background_color
71         ; call move_up
72
73
74     cmp ah,3
75     ja int7ch_ret
76     mov bl,ah
77     mov bh,0
78     add bx,bx
79     add bx,offset Mytable - offset int7ch + 200h
80
81     call word ptr es:[bx]
82
83
84     int7ch_ret:
85         pop es
86         pop bx
87         pop ax
88         iret
89
90 ;===== clear_screen s=====
91 clear_screen:
92     push ax
93     push bx
94     push cx
95     push dx
96     push ds
97     push es
98     push si
99     push di
100
101     clear_screen_bg:
102         mov bx,0b800h
103         mov es,bx
104
105         mov bx,0
106         mov cx,2000
107     cs_lp1:

```

```

108             mov byte ptr es:[bx], ' '
109             add bx,2
110             loop cs_lp1
111         clear_screen_end:
112             pop di
113             pop si
114             pop es
115             pop ds
116             pop dx
117             pop cx
118             pop bx
119             pop ax
120             ret
121         ;===== set_front_color =====
122     set_front_color:
123         push ax
124         push bx
125         push cx
126         push dx
127         push ds
128         push es
129         push si
130         push di
131     set_front_color_bg:
132         mov bx,0b800h
133         mov es,bx
134         mov bx,1
135         mov cx,2000
136         ; mov al,2
137     sfc_lp1:
138         and byte ptr es:[bx],11111000b
139         or es:[bx],al
140         add bx,2
141         loop sfc_lp1
142     set_front_color_end:
143         pop di
144         pop si
145         pop es
146         pop ds
147         pop dx
148         pop cx
149         pop bx
150         pop ax
151         ret
152     ;===== set_background_color =====
153     set_background_color:
154         push ax
155         push bx
156         push cx
157         push dx
158         push ds
159         push es
160         push si
161         push di
162     set_background_color_bg:
163         mov cl,4
164         shl al,cl
165

```

```

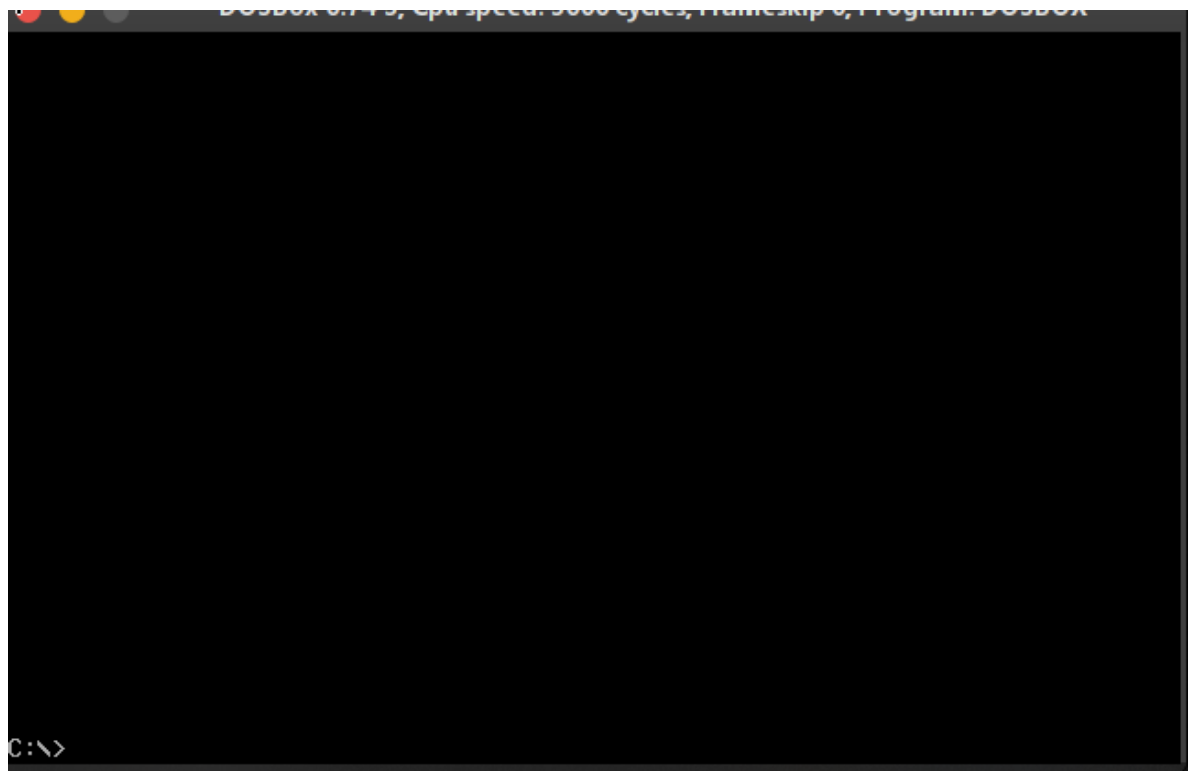
166         mov bx,0b800h
167         mov es,bx
168
169         mov bx,1
170         mov cx,2000
171         ; mov al,15
172         sbc_lp1:
173             and byte ptr es:[bx],10001111b
174             or es:[bx],al
175             add bx,2
176             loop sbc_lp1
177         set_background_color_end:
178             pop di
179             pop si
180             pop es
181             pop ds
182             pop dx
183             pop cx
184             pop bx
185             pop ax
186             ret
187
188         ;===== move up =====
189         move_up:
190             push ax
191             push bx
192             push cx
193             push dx
194             push ds
195             push es
196             push si
197             push di
198         move_up_bg:
199             mov si,0b800h
200             mov es,si
201             mov ds,si
202
203             mov si,160
204             mov di,0
205
206             cld
207             mov cx,24
208
209         mub_lp1:
210             push cx
211             mov cx,160
212             rep movsb
213             pop cx
214             loop mub_lp1
215
216             mov cx,80
217             mov si,0
218         mub_lp2:
219             mov byte ptr [160*21+si],' '
220             add si,2
221             loop mub_lp2
222         move_up_end:
223             pop di

```

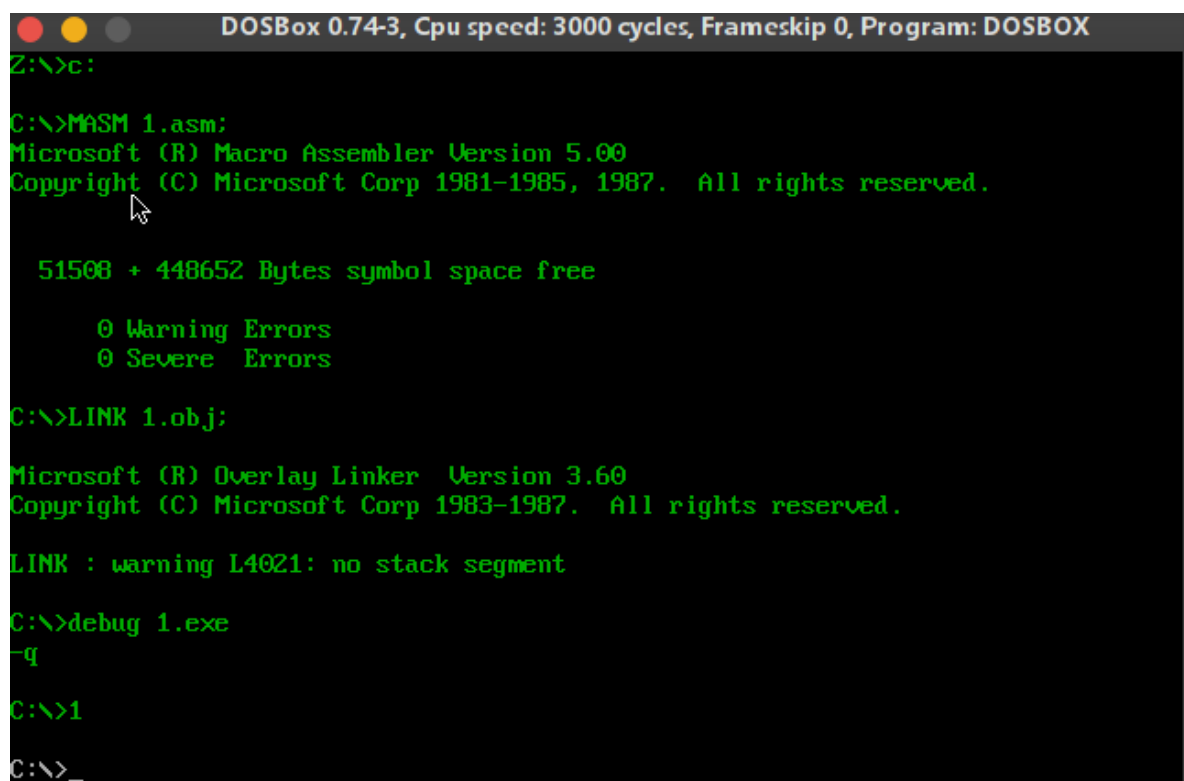
```
224         pop si
225         pop es
226         pop ds
227         pop dx
228         pop cx
229         pop bx
230         pop ax
231         ret
232
233     ;=====
234     int7ch_end:
235         nop
236
237
238 code ends
239 end start
```

## 截屏

### 0 清屏



### 1 设置前景色

A screenshot of a DOSBox window with a black background. The window title bar shows 'DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX'. The command prompt shows the following sequence of commands and output:

```
Z:\>c:
C:\>MASM 1.asm;
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

51508 + 448652 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>LINK 1.obj;
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

LINK : warning L4021: no stack segment

C:\>debug 1.exe
-q
C:\>1
C:\>_
```

## 2 设置背景色

A screenshot of a DOSBox window with a green background. The window title bar shows 'DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX'. The command prompt shows the following sequence of commands and output:

```
Z:\>c:
C:\>MASM 1.asm;
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

51508 + 448652 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>LINK 1.obj;
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

LINK : warning L4021: no stack segment

C:\>debug 1.exe
-q
C:\>1
C:\>
```

## 3 向上滚动一行

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

C:\>MASM 1.asm;
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

51508 + 448652 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>LINK 1.obj;
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

LINK : warning L4021: no stack segment

C:\>debug 1.exe
-q

C:\>1

C:\>
```