

chapter 13

1.

code

```
1  assume cs:code,ss:stack,ds:data
2
3  data segment
4      db 'welcome to masm!',0
5  data ends
6
7  stack segment
8      db 128 dup(0)
9  stack ends
10
11 code segment
12 start:
13     mov ax,stack
14     mov ss,ax
15     mov sp,128
16
17     call transfer
18     call test_application
19
20     mov ax,4c00h
21     int 21h
22
23 ;=====
24 test_application:
25     push ax
26     push bx
27     push cx
28     push dx
29     push ds
30     push es
31     push si
32     push di
33     test_application_bg:
34         mov dh,10;row
35         mov dl,10;cul
36         mov cl,2 ;color
37         mov ax,data;ds:si
38         mov ds,ax
39         mov si,0
40         int 7ch
41     test_application_end:
42         pop di
43         pop si
44         pop es
45         pop ds
46         pop dx
```

```

47         pop cx
48         pop bx
49         pop ax
50         ret
51         ;=====
52 transfer:
53     push ax
54     push bx
55     push cx
56     push dx
57     push ds
58     push es
59     push si
60     push di
61     transfer_bg:
62         ;mov 7ch
63         mov ax,cs
64         mov ds,ax
65         mov si,offset int_7ch
66
67         mov ax,0
68         mov es,ax
69         mov di,200h
70
71         mov cx,offset int_7ch_end - offset int_7ch
72
73         cld
74         rep movsb
75
76         ;set int table
77         mov ax,0
78         mov es,ax
79         mov word ptr es:[7ch*4],200h
80         mov word ptr es:[7ch*4+2],0
81     transfer_end:
82         pop di
83         pop si
84         pop es
85         pop ds
86         pop dx
87         pop cx
88         pop bx
89         pop ax
90         ret
91
92         ;=====
93 int_7ch:
94     push ax
95     push bx
96     push cx
97     push dx
98     push ds
99     push es
100    push si
101    push di
102    int_7ch_bg:
103        mov ax,0b800h
104        mov es,ax

```

```

105     mov di,0
106     call get_row
107     add di,ax
108     call get_cul
109     add di,ax
110
111
112
113     s0:
114         mov al,ds:[si]
115         cmp al,0
116         je end_0
117         mov ah,cl
118         mov es:[di],ax
119         inc si
120         inc di
121         inc di
122         jmp s0
123
124     end_0:
125         pop di
126         pop si
127         pop es
128         pop ds
129         pop dx
130         pop cx
131         pop bx
132         pop ax
133         iret
134
135     get_row:
136         mov al,160
137         mul dh
138         ret
139     get_cul:
140         mov al,2
141         mul dl
142         ret
143     int_7ch_end:
144         nop
145
146 code ends
147 end start

```

截屏

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Z:\>c:

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

51508 + 465036 Bytes symbol space free

0 Warning(s)
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.

code

```
1  assume cs:code,ss:stack,ds:data
2
3  data segment
4      db 'welcome to masm!',0
5  data ends
6
7  stack segment
8      db 128 dup(0)
9  stack ends
10
11 code segment
12 start:
13     mov ax,stack
14     mov ss,ax
15     mov sp,128
16
17     call transfer
18     call test_application
19
20     mov ax,4c00h
21     int 21h
22
23     ;=====
24 test_application:
25     push ax
26     push bx
27     push cx
28     push dx
29     push ds
30     push es
```

```

31     push si
32     push di
33     test_application_bg:
34         mov ax,0b800h
35         mov es,ax
36         mov di,160*12 ;position
37         mov bx,offset s - offset test_application_end ;length
38         mov cx,80 ;times
39         s:
40             mov byte ptr es:[di],'!'
41             mov byte ptr es:[di+1],2
42             add di,2
43             int 7ch
44     test_application_end:
45         pop di
46         pop si
47         pop es
48         pop ds
49         pop dx
50         pop cx
51         pop bx
52         pop ax
53         ret
54     ;=====
55     transfer:
56         push ax
57         push bx
58         push cx
59         push dx
60         push ds
61         push es
62         push si
63         push di
64     transfer_bg:
65         ;mov 7ch
66         mov ax,cs
67         mov ds,ax
68         mov si,offset int_7ch
69
70         mov ax,0
71         mov es,ax
72         mov di,200h
73
74         mov cx,offset int_7ch_end - offset int_7ch
75
76         cld
77         rep movsb
78
79         ;set int table
80         mov ax,0
81         mov es,ax
82         mov word ptr es:[7ch*4],200h
83         mov word ptr es:[7ch*4+2],0
84     transfer_end:
85         pop di
86         pop si
87         pop es
88         pop ds

```

```

89         pop dx
90         pop cx
91         pop bx
92         pop ax
93         ret
94
95         ;=====
96 int_7ch:
97     push ax
98     push bx
99     push dx
100    push ds
101    push es
102    push si
103    push di
104    int_7ch_bg:
105
106        push bp
107        mov bp,sp
108        dec cx
109        jcxz end_7ch
110        add [bp+2*8],bx
111
112
113    end_7ch:
114
115        pop bp
116
117        pop di
118        pop si
119        pop es
120        pop ds
121        pop dx
122        pop bx
123        pop ax
124        iret
125    int_7ch_end:
126        nop
127
128 code ends
129 end start

```

截屏

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Z:\>c:
C:\>MASM 1.asm;
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

51508 + 465036 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.

code

```
1  assume cs:code
2
3  code segment
4  s1: db 'Good,better,best','','$'
5  s2: db 'Never let it rest',''$'
6  s3: db 'Till good is better',''$'
7  s4: db 'And better,best.',''$'
8  s: dw offset s1,offset s2,offset s3,offset s4
9  row: db 2,4,6,8
10
11 start:
12
13     call clear_screen
14     call show
15
16     mov ax,4c00h
17     int 21h
18
19 show:
20     push ax
21     push bx
22     push cx
23     push dx
24     push ds
25     push es
26     push si
27     push di
28     show_bg:
29     mov ax,cs
30     mov ds,ax
```

```

31         mov bx,offset s
32         mov si,offset row
33         mov cx,4
34 ok:      mov bh,0
35         mov dh,ds:[si]
36         mov dl,0
37         mov ah,2
38         int 10h
39
40         mov dx,ds:[bx]
41         mov ah,9
42         int 21h
43
44         inc si
45         add bx,2
46
47         loop ok
48 show_end:
49         pop di
50         pop si
51         pop es
52         pop ds
53         pop dx
54         pop cx
55         pop bx
56         pop ax
57         ret
58
59 clear_screen:
60         push ax
61         push bx
62         push cx
63         push dx
64         push ds
65         push es
66         push si
67         push di
68 clear_screen_bg:
69         mov bx,0b800h
70         mov es,bx
71
72         mov bx,0
73         mov dl,0
74         mov dh,00000010b
75         mov cx,2000
76
77 clearScreen:
78         mov es:[bx],dx
79         add bx,2
80
81         loop clearScreen
82
83 clear_screen_end:
84         pop di
85         pop si
86         pop es
87         pop ds
88         pop dx

```



```
89      pop cx
90      pop bx
91      pop ax
92      ret
93
94
95 code ends
96
97 end start
```

截屏

