
1 chapter10

1.1 1. 显示字符串

1.1.1 代码

```
1  assume cs:code,ds:data,ss:stack
2
3  data segment
4      db 'welcome to masn!',0
5  data ends
6
7  stack segment
8      db 128 dup(0)
9  stack ends
10
11 code segment
12
13
14     start:
15
16     mov ax,stack
17     mov ss,ax
18     mov sp,128
19
20     mov ax,data
21     mov ds,ax
22     mov si,0
23
24     mov dh,8;row
25     mov dl,3;cul
26     mov cl,2;color
27     call show_str
28
29     mov ax,4c00h
30     int 21h
31
32     show_str:
33         push ax
34         push dx
35         push cx
```

```
36     push es
37     push si
38     push di
39
40     mov ax,0b800h
41     mov es,ax
42     mov di,0
43
44     call get_row
45     add di,ax
46     call get_cul
47     add di,ax
48
49     call show
50
51     pop di
52     pop si
53     pop es
54     pop cx
55     pop dx
56     pop ax
57     ret
58
59 get_row:
60     mov al,160
61     mul dh
62     ret
63 get_cul:
64     mov al,2
65     mul dl
66     ret
67
68 show:
69     push ax
70     push bx
71     push cx
72     push di
73     push si
74
75     sub ax,ax
76     sub bx,bx
77
78
79     mov bl,cl
80 show_1:
81     mov cl,ds:[si]
82     mov al,ds:[si]
83     mov ch,0
84     jcxz ok
85     mov ah,bl
86     mov es:[di],ax
87     add di,2
```

```

88         inc si
89         jmp show_1
90
91
92     ok:
93     pop si
94     pop di
95     pop cx
96     pop bx
97     pop ax
98     ret
99
100    ret
101 code ends
102 end start

```

1.1.2 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.

Welcome to masm! tes 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:\>_

```

1.2 2. 解决除法溢出问题

1.2.1 代码

```
1 assume cs:code,ds:data,ss:stack
```

```

2
3 data segment
4 data ends
5
6 stack segment
7     db 128 dup(0)
8 stack ends
9
10 code segment
11     start:
12         mov ax,data
13         mov ds,ax
14         mov ax,stack
15         mov ss,ax
16         mov sp,128
17
18         mov ax,4240h;L
19         mov dx,000fh;H
20         mov cx,0ah;N
21
22         push ax;暂时存储ax
23         mov bp,sp;记录ax中数据的位置
24         call divdw
25
26         mov ax,4c00h
27         int 21h
28
29     divdw:
30
31         mov ax,dx
32         mov dx,0
33         div cx
34         push ax
35
36         mov ax,ss:[bp+0];
37         div cx
38         mov cx,dx
39
40         pop dx
41
42         ret
43 code ends
44 end start

```

1.2.2 截图

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
C:\>debug 1.exe
-r
AX=FFFF BX=0000 CX=00B2 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=0772 IP=0000  NV UP EI PL NZ NA PO NC
0772:0000 B86A07      MOV     AX,076A
-u
0772:0000 B86A07      MOV     AX,076A
0772:0003 8ED8        MOV     DS,AX
0772:0005 B86A07      MOV     AX,076A
0772:0008 8ED0        MOV     SS,AX
0772:000A BC8000      MOV     SP,0080
0772:000D B84042      MOV     AX,4240
0772:0010 BA0F00      MOV     DX,000F
0772:0013 B90A00      MOV     CX,000A
0772:0016 50          PUSH    AX
0772:0017 8BEC        MOV     BP,SP
0772:0019 EB0500      CALL    0021
0772:001C B8004C      MOV     AX,4C00
0772:001F CD21      INT     21
-g 1c
AX=86A0 BX=0000 CX=0000 DX=0001 SP=007E BP=007E SI=0000 DI=0000
DS=076A ES=075A SS=076A CS=0772 IP=001C  NV UP EI PL NZ NA PO NC
0772:001C B8004C      MOV     AX,4C00
```

1.3 3. 数值显示

1.3.1 代码

```
1  assume cs:code,ds:data,ss:stack
2
3  data segment
4      dw 123,12666,1,8,3,38,0
5  data ends
6
7  string segment
8      db 10 dup (0),0
9  string ends
10
11 stack segment
12     db 128 dup(0)
13 stack ends
14
15 code segment
16
17
18     start:
19     ;初始化
20     mov ax,stack
21     mov ss,ax
```

```
22     mov sp,128
23
24     call init_reg
25
26     mov dh,8;row
27     mov dl,3;cul
28     mov cl,2;color
29 ;显示
30     a:
31     mov ax,ds:[si]
32     mov cx,ax
33     jcxz Pro_end
34     call dtoc
35
36     inc dh
37     mov dl,3;cul
38     mov cl,2;color
39     call show_str
40
41     call zero
42
43     add si,2
44     jmp a
45
46     Pro_end:
47     mov ax,4c00h
48     int 21h
49
50 ;把string全部清零
51     zero:
52         push ax
53         push bx
54         push cx
55         push dx
56         push ds
57         push es
58         push si
59         push di
60
61     mov cx,10
62     mov ax,string
63     mov es,ax
64     mov di,0
65
66     lp:
67     mov al,0
68     mov es:[di],al
69     inc di
70     loop lp
71
72     zero_ret:
73         pop di
```

```

74             pop si
75             pop es
76             pop ds
77             pop dx
78             pop cx
79             pop bx
80             pop ax
81             ret
82             ;=====
83             dtoc:
84                 push ax
85                 push bx
86                 push cx
87                 push dx
88                 push ds
89                 push es
90                 push si
91                 push di
92
93             mov si,0
94             mov dx,0
95             dtoc_bg:
96                 mov cx,10
97                 div cx
98                 add dl,30h
99                 mov es:[si],dl
100                mov cx,ax
101                jcxz dtoc_end
102                inc si
103                mov dx,0
104                jmp dtoc_bg
105
106
107             dtoc_end:
108                 pop di
109                 pop si
110                 pop es
111                 pop ds
112                 pop dx
113                 pop cx
114                 pop bx
115                 pop ax
116             ret
117             ;=====
118             init_reg:
119                 mov ax,data
120                 mov ds,ax
121                 mov ax,string
122                 mov es,ax
123
124                 mov si,0
125

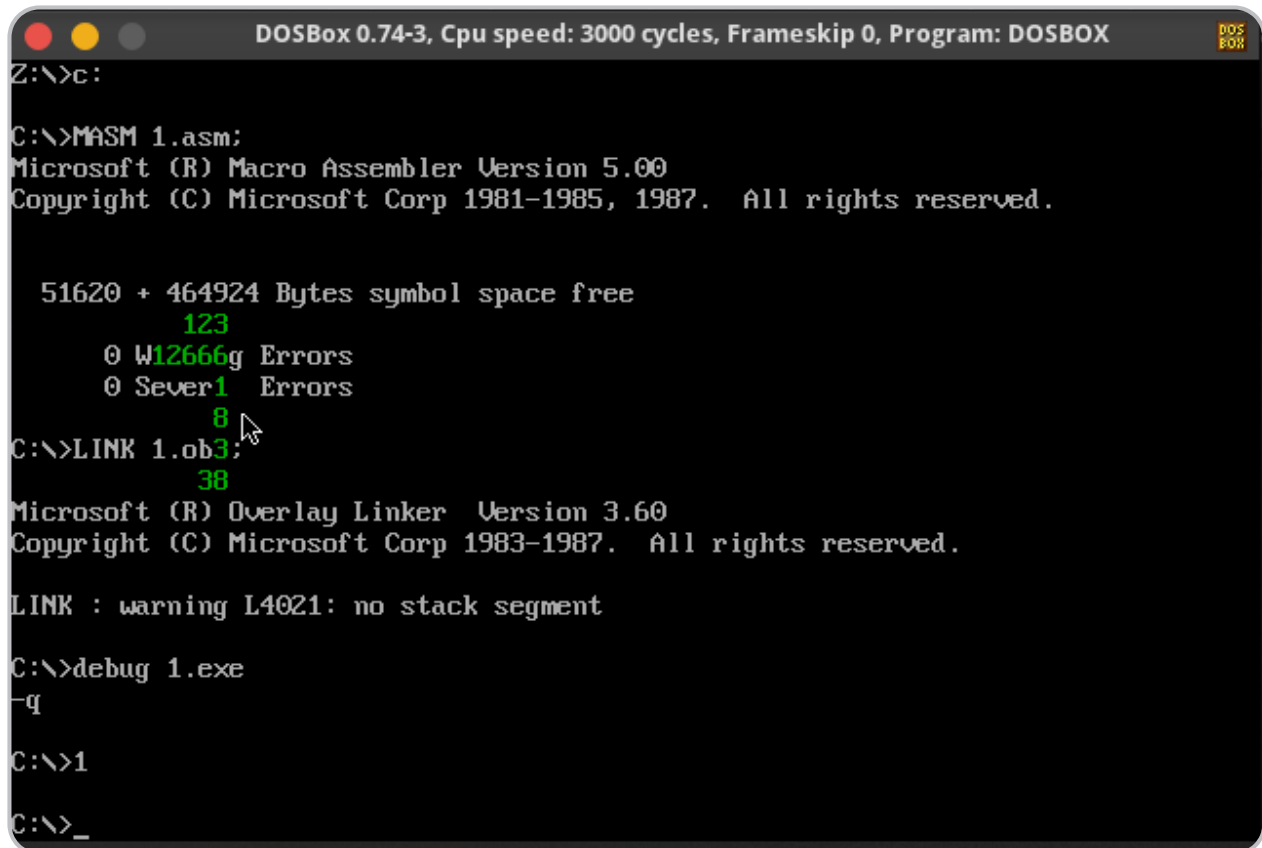
```

```
126         ret
127
128     ;=====
129     show_str:
130         push ax
131         push bx
132         push cx
133         push dx
134         push ds
135         push es
136         push si
137         push di
138
139         mov ax,string
140         mov ds,ax
141         mov si,0
142
143         mov ax,0b800h
144         mov es,ax
145         mov di,0
146
147         call get_row
148         add di,ax
149         add dl,10
150         call get_cul
151         add di,ax
152
153         call show
154
155         pop di
156         pop si
157         pop es
158         pop ds
159         pop dx
160         pop cx
161         pop bx
162         pop ax
163         ret
164
165     get_row:
166         mov al,160
167         mul dh
168         ret
169     get_cul:
170         mov al,2
171         mul dl
172         ret
173
174     show:
175         push ax
176         push bx
177         push cx
```



```
178             push dx
179             push ds
180             push es
181             push si
182             push di
183
184             sub ax,ax
185             sub bx,bx
186
187
188             mov bl,cl
189             show_1:
190                 mov cl,ds:[si]
191                 mov al,ds:[si]
192                 mov ch,0
193                 jcxz ok
194                 mov ah,bl
195                 mov es:[di],ax
196                 sub di,2
197                 inc si
198                 jmp show_1
199
200
201             ok:
202                 pop di
203                 pop si
204                 pop es
205                 pop ds
206                 pop dx
207                 pop cx
208                 pop bx
209                 pop ax
210             ret
211
212 code ends
213 end start
```

1.3.2 截图



```
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.

51620 + 464924 Bytes symbol space free
123
0 W12666g Errors
0 Sever1 Errors
C:\>LINK 1.obj
38
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 课程设计1

2.0.3 代码

```
1  assume cs:code,ds:data,ss:stack
2
3  data segment
4      ;0,0
5      db '1975','1976','1977','1978','1979','1980','1981','1982','1983'
6      db '1984','1985','1986','1987','1988','1989','1990','1991','1992'
7      db '1993','1994','1995'
8      ;以上是表示21年的21个字符串 year
9
10     ;84,54h
11     dd 16,22,382,1356,2390,8000,16000,24486,50065,97479,140417,197514
12     dd 345980,590827,803530,1183000,1843000,2759000,3753000,4649000,5937000
13     ;以上是表示21年公司总收入的21个dword数据 sum
```

```

14
15         ;168,a8
16         dw 3,7,9,13,28,38,130,220,476,778,1001,1442,2258,2793,4037,5635,8226
17         dw 11542,14430,15257,17800
18 data ends
19
20
21
22 stack segment
23         db 128 dup(0)
24 stack ends
25
26
27 table segment ;0123456789ABCDEF
28         db 21 dup ('year summe ?? ')
29 table ends
30
31 string segment ;0123456789ABCDEF
32         db 21 dup (0000000000000000)
33 string ends
34
35
36 code segment
37 start:
38         mov ax,stack;stack
39         mov ss,ax
40         mov sp,128
41
42         call clear_screen;清空屏幕
43
44         call init_reg;初始化寄存器组,把数据导入
45
46         call input_table;把数据格式化存储
47
48         call output_table;输出格式化数据
49
50         mov ax,4C00H
51         int 21H
52
53 ;=====;output_table
54 output_table:
55         push ax
56         push bx
57         push cx
58         push dx
59         push es
60         push ds
61         push di
62         push si
63
64 output_table_bg:
65         ;分别调用显示子程序

```

```

66         call print_year;年
67         call print_income;总收入
68         call print_employee;员工数量
69         call print_avg;人均收入
70
71     output_table_end:
72         pop si
73         pop di
74         pop ds
75         pop es
76         pop dx
77         pop cx
78         pop bx
79         pop ax
80         ret
81     ;=====
82     print_avg:
83         push ax
84         push bx
85         push cx
86         push dx
87         push es
88         push ds
89         push di
90         push si
91
92         mov ax,table
93         mov ds,ax
94         mov ax,string
95         mov es,ax
96
97         mov si,0
98
99     print_avg_bg:
100
101         mov cx,21
102         mov di,160*3+2*40;在屏幕中的显示位置
103     pa_lp1:
104         push cx
105
106         mov ax,ds:[si+13]
107         mov dx,0
108         mov cx,10
109
110         call transfer;改进后的子程序3
111
112         add si,16
113         add di,160
114         pop cx
115         loop pa_lp1
116
117         pop si

```

```
118         pop di
119         pop ds
120         pop es
121         pop dx
122         pop cx
123         pop bx
124         pop ax
125         ret
126
127         ;=====
128     print_employee:
129         push ax
130         push bx
131         push cx
132         push dx
133         push es
134         push ds
135         push di
136         push si
137
138         mov ax,table
139         mov ds,ax
140         mov ax,string
141         mov es,ax
142
143         mov si,0
144
145     print_employee_bg:
146         mov cx,21
147         mov di,160*3+2*30
148
149     peb_lp1:
150         push cx
151
152         mov ax,ds:[si+10]
153         mov dx,0
154         mov cx,10
155
156         call transfer
157
158         add si,16
159         add di,160
160         pop cx
161         loop peb_lp1
162
163     print_employee_end:
164         pop si
165         pop di
166         pop ds
167         pop es
168         pop dx
169         pop cx
```

```
170         pop bx
171         pop ax
172         ret
173
174         ;=====
175     print_income:
176         push ax
177         push bx
178         push cx
179         push dx
180         push es
181         push ds
182         push di
183         push si
184
185         mov ax,table
186         mov ds,ax
187         mov ax,string
188         mov es,ax
189
190         mov si,0
191
192
193     print_income_bg:
194
195         mov cx,21
196         mov di,160*3+2*20
197     pib_lp1:
198         push cx
199
200         mov ax,ds:[si+5]
201         mov dx,ds:[si+7]
202         mov cx,10
203
204
205         call transfer
206
207         add si,16
208         ;add di,16
209         add di,160
210         pop cx
211         loop pib_lp1
212
213
214     print_income_end:
215         pop si
216         pop di
217         pop ds
218         pop es
219         pop dx
220         pop cx
221         pop bx
```

```

222         pop ax
223         ret
224
225         ;=====
226     transfer:
227         push ax
228         push bx
229         push cx
230         push dx
231         push es
232         push ds
233         push di
234         push si
235
236         mov bx,15
237
238     transfer_bg:    ;判断是否大于一个字节
239         mov cx,dx
240         jcxz short_div;不大于一个字节
241
242         mov cx,10
243         push ax
244         mov bp,sp
245         call divdw
246         add sp,2
247
248         add cl,30h
249         mov es:[bx],cl
250
251         dec bx
252         jmp transfer_bg
253     sn:
254         call show_number;输出结果
255
256     transfer_end:
257         pop si
258         pop di
259         pop ds
260         pop es
261         pop dx
262         pop cx
263         pop bx
264         pop ax
265         ret
266         ;=====
267     show_number:
268         push ax
269         push bx
270         push cx
271         push dx
272         push es
273         push ds

```

```

274         push di
275         push si
276
277         mov ax,string
278         mov ds,ax
279         mov ax,0b800h
280         mov es,ax
281
282
283     sn_lp1:
284         mov cx,0
285         mov cl,ds:[bx]
286         jcxz show_number_end
287         mov ch,00000111b
288         mov es:[di],cx
289
290         inc bx
291         add di,2
292
293         jmp sn_lp1
294
295     show_number_end:
296         pop si
297         pop di
298         pop ds
299         pop es
300         pop dx
301         pop cx
302         pop bx
303         pop ax
304         ret
305
306     ;=====
307     divdw:
308
309     mov ax,dx
310     mov dx,0
311
312     div cx
313     push ax
314     mov ax,ss:[bp+0]
315     div cx
316
317     mov cx,dx
318     pop dx
319
320     ret
321
322     ;=====
323     short_div:
324     mov cx,10
325     div cx

```



```

326         add dl,30h
327         mov es:[bx],dl
328         mov cx,ax
329         jcxz sn
330         dec bx
331         mov dx,0
332         jmp short_div
333
334         ;=====
335     print_year:
336         push ax
337         push bx
338         push cx
339         push dx
340         push es
341         push ds
342         push di
343         push si
344
345         ;0123456789ABCDEF
346         ;year summe ??
347
348     print_year_bg:
349         mov ax,table
350         mov ds,ax
351         mov ax,0b800h
352         mov es,ax
353         mov si,0
354         mov di,160*3+2*10
355
356         mov cx,21
357     pyb_lp1:
358         push cx
359         mov cx,4
360         mov bx,0
361     pyb_lp2:
362         mov al,ds:[si+bx]
363         ;mov ah,00000111b
364         mov ah,00000111b
365         push bx
366         add bx,bx
367         mov es:[di+bx],ax
368         pop bx
369
370         inc bx
371         loop pyb_lp2
372
373         add si,16
374         add di,160
375         pop cx
376         loop pyb_lp1
377

```

```

378         print_year_end:
379             pop si
380             pop di
381             pop ds
382             pop es
383             pop dx
384             pop cx
385             pop bx
386             pop ax
387             ret
388
389         ;=====;input_table
390     input_table:
391         push ax
392         push bx
393         push cx
394         push dx
395         push es
396         push ds
397         push di
398         push si
399
400         mov si,0
401         mov di,0
402         mov bx,21*4*2
403
404         mov cx,21
405
406
407         ;0123456789ABCDEF
408         ;year summ ne ??
409     input_table_bg:
410         ;year
411         push ds:[si+0]
412         pop es:[di+0]
413         push ds:[si+2]
414         pop es:[di+2]
415         ;income
416         mov ax,ds:[si+21*4+0]
417         mov dx,ds:[si+21*4+2]
418         mov es:[di+5],ax
419         mov es:[di+7],dx
420         ;employee
421         push ds:[bx]
422         pop es:[di+10]
423         ;avg
424         div word ptr ds:[bx]
425         mov es:[di+13],ax
426
427         add si,4
428         add di,16
429         add bx,2

```

```

430
431         loop input_table_bg
432     input_table_end:
433         pop si
434         pop di
435         pop ds
436         pop es
437         pop dx
438         pop cx
439         pop bx
440         pop ax
441         ret
442
443
444
445 ;=====初始化寄存器组，把数据导入
446 init_reg:
447         mov bx,data;data in
448         mov ds,bx
449         mov bx,table;data out
450         mov es,bx
451         ret
452
453 ;=====清空屏幕
454 clear_screen:
455         mov bx,0b800h
456         mov es,bx
457
458         mov bx,0
459         mov dx,0000h
460         mov cx,2000
461
462     clearScreen:
463         mov es:[bx],dx
464         add bx,2
465
466         loop clearScreen
467
468         ret
469 code ends
470
471 end start

```

2.0.4 截图

1975	16	3	5
1976	22	7	3
1977	382	9	42
1978	1356	13	104
1979	2390	28	85
1980	8000	38	210
1981	16000	130	123
1982	24486	220	111
1983	50065	476	105
1984	97479	778	125
1985	140417	1001	140
1986	197514	1442	136
1987	345980	2258	153
1988	590827	2793	211
1989	803530	4037	199
1990	1183000	5635	209
1991	1843000	8226	224
1992	2759000	11542	239
1993	3753000	14430	260
1994	4649000	15257	304
1995	5937000	17800	333

C:\>_