

汇编语言程序设计课程作业（九）

姓名：袁昊男 学号：2018091618008

检测点 9.1

1、 程序如下。

```
assume cs:code

data segment
    ?
data ends

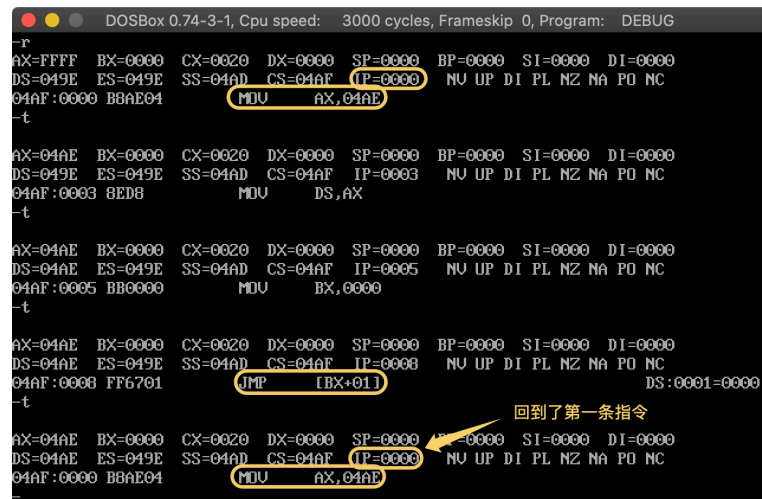
code segment
start: mov ax,data
        mov ds,ax
        mov bx,0
        jmp word ptr [bx+1]
code ends
end start
```

若要使程序中的 `jmp` 指令执行后，CS:IP 指向程序的第一条指令，在 `data` 段中应该定义哪些数据？

(1) 补全程序：

```
data segment
    db 3 dup(0)
data ends
```

(2) 跟踪、调试：



(3) 分析：执行“`jmp word ptr [bx+1]`”后， $(IP) = (ds:[1])$ ，要使 CS:IP 指向程序的第一条指令，IP 的值应该等于 0000H，因此在 `data` 段中至少第 1、2 字节为 00H，所以采用 `dup` 命令在 `data` 段放置 3 个 00H 字节，即可实现跳转到第一条指令。

2、 程序如下：


```
mov ax,2000H
mov es,ax
jmp dword ptr es:[1000H]
```

答：“`jmp dword ptr es:[1000H]`”指令是段间转移的指令，CS 和 IP 存储在一个双字的单元中，其中高 16 位存储的是 CS，低 16 位存储的是 IP。即 $IP = ([1000H])$ ， $CS = ([1002H])$ 。因此， $(CS) = 0006H$ ， $(IP) = 00BEH$ 。

补全编程，利用 `jcxz` 指令，实现在内存 2000H 段中查找第一个值为 0 的字节，找到后，将它的偏移地址存储在 `dx` 中。

```
start: mov ax,2000H
       mov ds,ax
       mov bx,0
       s: _____
          _____
          _____
          _____
          jmp short s
ok: mov dx,bx
    mov ax,4c00h
    int 21h
```

```
s:  mov cl,[bx]
    mov ch,0
    jcxz ok
    inc bx
    jmp short s
```

DOSBox 0.74-3-1, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

LINK : warning L4021: no stack segment

C:\>Debug t1.exe

-r

AX=FFFF BX=0000 CX=001B DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
 DS=049E ES=049E SS=04AD CS=04AE IP=0000 NU UP DI PL NZ NA PO NC
 04AE:0000 B80020 MOV AX,2000

-d 2000:0

2000:0000	00	CA	0F	63	61	6E	6E	6F	74	20	6F	70	65	6E	20	6F		.J.cannot open o
2000:0010	5C	64	20	76	65	72	73	69	6F	6E	00	CB	0F	6F	6C	64		ld version.K.old
2000:0020	20	76	65	72	73	69	6F	6E	20	6E	6F	74	20	73	65	67		version not seg
2000:0030	6D	65	6E	74	65	64	20	65	78	65	63	75	74	61	62	6C		mented executabl
2000:0040	65	20	66	6F	72	6D	61	74	00	CD	0F	6E	61	6D	65	20		e format.M.name
2000:0050	6F	6E	20	6F	75	74	20	6F	69	75	20	66	69	6C	65	20		of output file i
2000:0060	73	20	27	25	73	27	00	D3	0F	25	73	20	3A	20	63	61		s 'zs'.S.zs : ca
2000:0070	6E	6E	6F	74	20	66	69	6E	64	20	6C	69	62	72	61	72		mnot find librar

Program terminated normally

-r

AX=FFFF BX=0000 CX=001B DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
 DS=049E ES=049E SS=04AD CS=04AE IP=0000 NU UP DI PL NZ NA PO NC
 04AE:0000 B80020 MOV AX,2000

中，程序退出。

实验 8 分析一个奇怪的程序

分析下面的程序，在运行前思考：这个程序可以正确返回吗？

运行后再思考：为什么是这种结果？

通过这个程序加深对相关内容的理解。

```
assume cs:codesg
codesg segment

    mov ax,4c00h
    int 21h

start: mov ax,0
      s: nop
        nop

        mov di,offset s
        mov si,offset s2
        mov ax,cs:[si]
        mov cs:[di],ax

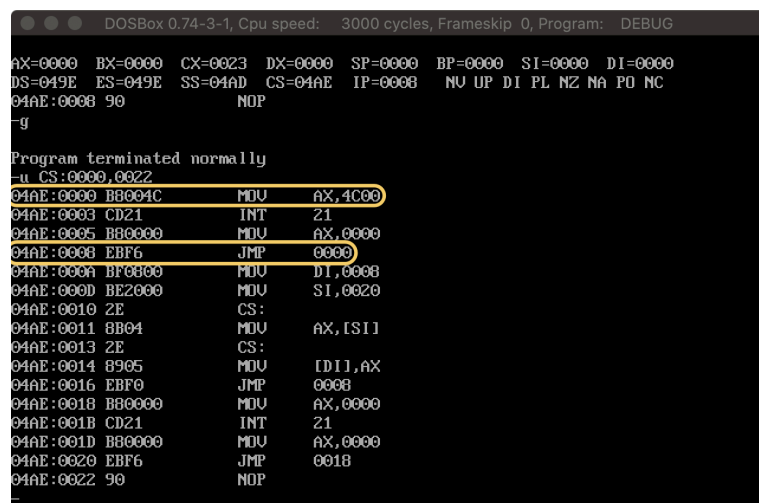
s0: jmp short s

s1: mov ax,0
    int 21h
    mov ax,0

s2: jmp short s1
    nop

codesg ends
end start
```

1、跟踪、调试：



```
DOSBox 0.74-3-1, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
AX=0000 BX=0000 CX=0023 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=049E ES=049E SS=04AD CS=04AE IP=000B NU UP DI PL NZ NA PO NC
04AE:000B 90      NOP
-g
Program terminated normally
-u CS:0000,0022
04AE:0000 B8004C    MOV     AX,4C00
04AE:0003 CD21    INT     21
04AE:0005 B80000    MOV     AX,0000
04AE:0008 EBF6    JMP     0000
04AE:000A BF0300    MOV     DI,0000
04AE:000D BE2000    MOV     SI,0020
04AE:0010 2E      CS:
04AE:0011 B804    MOV     AX,[SI]
04AE:0013 2E      CS:
04AE:0014 B905    MOV     [DI],AX
04AE:0016 EBF0    JMP     0000
04AE:0018 B80000    MOV     AX,0000
04AE:001B CD21    INT     21
04AE:001D B80000    MOV     AX,0000
04AE:0020 EBF6    JMP     0018
04AE:0022 90      NOP
```

可以看到程序正确返回。

2、分析：

首先 s 段中 4 条 mov 指令的作用是将标号 s2 中的一条指令复制到标号 s 处。被复制的指

令“`jmp short s1`”所对应的机器码是“EBF6”，“F6H”为-10d的补码，即从s2处跳转到s1处需要将IP向前移动10个字节（此位移量也由指令“`jmp short s1`”处的偏移地址18H减去指令“`jmp short s1`”后一个字节的偏移地址22H得出）。将指令复制到s处后，程序继续向下执行，来到标号s0处。s0为跳转指令，跳转至标号s处，此时如1中图所示， $(IP) = 0008H$ ，CPU从CS:[0008]取到指令EBF6， $(IP) = (IP) + 2$ （指令长度）后 $(IP) = 000AH$ ，CPU执行EBF6指令（作用是使IP向前移动10字节）后， $(IP) = 0000H$ ，即代码段第一行指令。故程序可以正确返回。