

汇编语言程序设计课程作业（六）

姓名：袁昊男学号：2018091618008

实验 5 编写、调试具有多个段的程序

- (1) 将下面的程序编译、连接，用 Debug 加载、跟踪，然后回答问题。

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

data segment
    dw 0123h,0456h,0789h,0abch,0defh,0fedh,0cbah,0987h
data ends

stack segment
    dw 0,0,0,0,0,0,0,0
stack ends

code segment

start: mov ax,stack
        mov ss,ax
        mov sp,16

        mov ax,data
        mov ds,ax

        push ds:[0]
        push ds:[2]
        pop ds:[2]
        pop ds:[0]

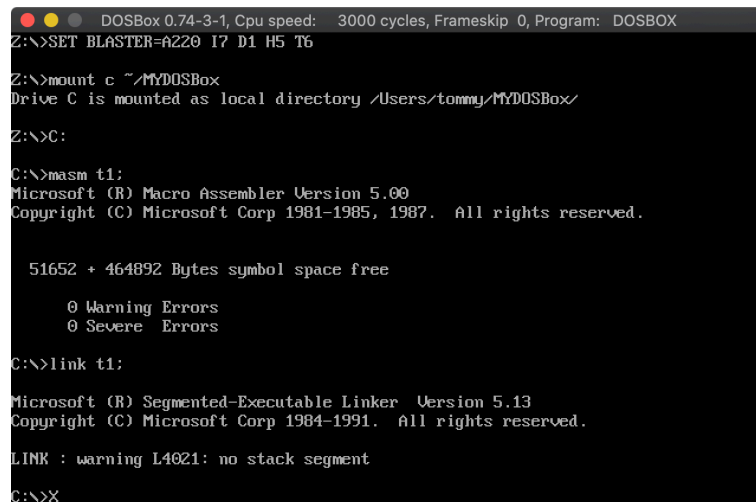
        mov ax,4c00h
        int 21h

code ends

end start
```

- ① CPU 执行程序，程序返回前，data 段中的数据为多少？

- 1) 编译、连接：



```
DOSBox 0.74-3-1, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c ~/MYDOSBox
Drive C is mounted as local directory ~/Users/tommy/MYDOSBox/

Z:\>C:

C:\>masm t1;
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

51652 + 464892 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>link t1;
Microsoft (R) Segmented-Executable Linker Version 5.13
Copyright (C) Microsoft Corp 1984-1991. All rights reserved.

LINK : warning L4021: no stack segment

C:\>X_
```

2) code 段运行前 data 段的数据:

```
DOSBox 0.74-3-1, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c ~/MYDOSBox
Drive C is mounted as local directory /Users/tommy/MYDOSBox/

Z:\>c:

C:\>Debug t1.exe

-t
AX=FFFF BX=0000 CX=0042 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=049E ES=049E SS=04AD CS=04B0 IP=0000 NU UP DI PL NZ NA PO NC
04B0:0000 B8AF04 MOV AX,04AF
-d 04ae:0
04AE:0000 23 01 56 04 89 07 BC 0A-EF 0D ED 0F BA 0C 87 09 #.U...<.o.m.:...
04AE:0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....0...#
04AE:0020 B8 AF 04 8E D0 BC 10 00-BB AE 04 8E D8 FF 36 00 8/.P<..8...X.6.
04AE:0030 00 FF 36 02 00 BF 06 02-00 BF 06 00 00 BB 00 4C ..6.....8.L
04AE:0040 CD 21 00 00 00 00 00 00 00 00 00 00 00 00 00 M!.....
04AE:0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
04AE:0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
04AE:0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

3) Debug 查看 data 段中数据:

```
DOSBox 0.74-3-1, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
-t
AX=04AE BX=0000 CX=0042 DX=0000 SP=000E BP=0000 SI=0000 DI=0000
DS=04AE ES=049E SS=04AF CS=04B0 IP=0011 NU UP DI PL NZ NA PO NC
04B0:0011 FF360200 PUSH [0002] DS:0002=0456
-t
AX=04AE BX=0000 CX=0042 DX=0000 SP=000C BP=0000 SI=0000 DI=0000
DS=04AE ES=049E SS=04AF CS=04B0 IP=0015 NU UP DI PL NZ NA PO NC
04B0:0015 8F060200 POP [0002] DS:0002=0456
-t
AX=04AE BX=0000 CX=0042 DX=0000 SP=000E BP=0000 SI=0000 DI=0000
DS=04AE ES=049E SS=04AF CS=04B0 IP=0019 NU UP DI PL NZ NA PO NC
04B0:0019 8F060000 POP [0000] DS:0000=0123
-d ds:0
04AE:0000 23 01 56 04 89 07 BC 0A-EF 0D ED 0F BA 0C 87 09 #.U...<.o.m.:...
04AE:0010 00 00 00 00 00 00 15 00-19 00 B0 04 92 01 23 01 .....0...#
04AE:0020 B8 AF 04 8E D0 BC 10 00-BB AE 04 8E D8 FF 36 00 8/.P<..8...X.6.
04AE:0030 00 FF 36 02 00 BF 06 02-00 BF 06 00 00 BB 00 4C ..6.....8.L
04AE:0040 CD 21 00 00 00 00 00 00 00 00 00 00 00 00 00 M!.....
04AE:0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
04AE:0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
04AE:0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

4) 结论: data 段中的数据与程序运行前一致。

② CPU 执行程序, 程序返回前, cs = _____、ss = _____、ds = _____。

1) Debug 中查看 cs、ss、ds 内容:

```
DOSBox 0.74-3-1, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
-t
AX=04AE BX=0000 CX=0042 DX=0000 SP=000E BP=0000 SI=0000 DI=0000
DS=04AE ES=049E SS=04AF CS=04B0 IP=0011 NU UP DI PL NZ NA PO NC
04B0:0011 FF360200 PUSH [0002] DS:0002=0456
-t
AX=04AE BX=0000 CX=0042 DX=0000 SP=000C BP=0000 SI=0000 DI=0000
DS=04AE ES=049E SS=04AF CS=04B0 IP=0015 NU UP DI PL NZ NA PO NC
04B0:0015 8F060200 POP [0002] DS:0002=0456
-t
AX=04AE BX=0000 CX=0042 DX=0000 SP=000E BP=0000 SI=0000 DI=0000
DS=04AE ES=049E SS=04AF CS=04B0 IP=0019 NU UP DI PL NZ NA PO NC
04B0:0019 8F060000 POP [0000] DS:0000=0123
-d ds:0
04AE:0000 23 01 56 04 89 07 BC 0A-EF 0D ED 0F BA 0C 87 09 #.U...<.o.m.:...
04AE:0010 00 00 00 00 00 00 15 00-19 00 B0 04 92 01 23 01 .....0...#
04AE:0020 B8 AF 04 8E D0 BC 10 00-BB AE 04 8E D8 FF 36 00 8/.P<..8...X.6.
04AE:0030 00 FF 36 02 00 BF 06 02-00 BF 06 00 00 BB 00 4C ..6.....8.L
04AE:0040 CD 21 00 00 00 00 00 00 00 00 00 00 00 00 00 M!.....
04AE:0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
04AE:0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
04AE:0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

2) 结论: cs = 04b0h, ss = 04afh, ds = 04ach。

③ 设程序加载后，code 段的段地址为 X，则 data 段的段地址为 X-2，stack 段的段地址为 X-1。

分析：data 段与 stack 段恰好各占 16 字节，因此段地址依次相差 1。

(2) 将下面的程序编译、连接，用 Debug 加载、跟踪，然后回答问题。

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

```
data segment
    dw 0123h,0456h
data ends
```

```
stack segment
    dw 0,0
stack ends
```

```
code segment
```

```
start: mov ax,stack
        mov ss,ax
        mov sp,16
```

```
        mov ax,data
        mov ds,ax
```

```
        push ds:[0]
        push ds:[2]
        pop  ds:[2]
        pop  ds:[0]
```

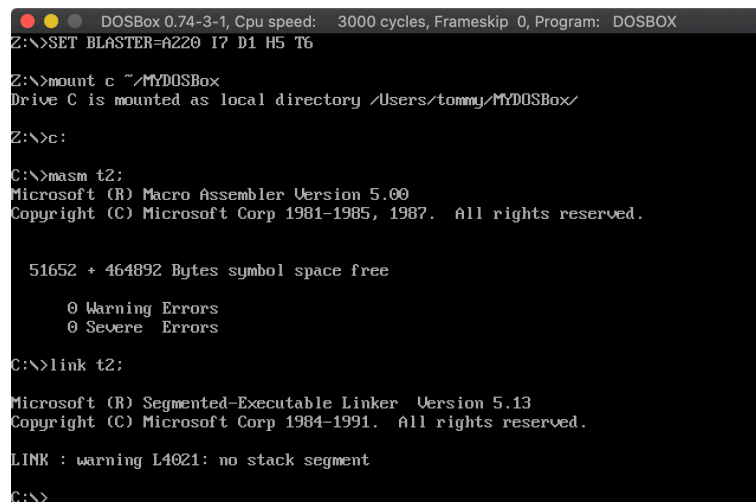
```
        mov ax,4c00h
        int 21h
```

```
code ends
```

```
end start
```

① CPU 执行程序，程序返回前，data 段中的数据为多少？

1) 编译、连接:



```
DOSBox 0.74-3-1, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c ~/MYDOSBox
Drive C is mounted as local directory /Users/tommy/MYDOSBox/

Z:\>c:

C:\>masm t2:
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

51652 + 464892 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>link t2:

Microsoft (R) Segmented-Executable Linker Version 5.13
Copyright (C) Microsoft Corp 1984-1991. All rights reserved.

LINK : warning L4021: no stack segment

C:\>
```

2) code 段运行前 data 段的数据:

```
DOSBox 0.74-3-1, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c ~/MYDOSBox
Drive C is mounted as local directory /Users/tommy/MYDOSBox/

Z:\>c:

C:\>Debug t2.exe

-r
AX=FFFF BX=0000 CX=0042 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=049E ES=049E SS=04AD CS=04B0 IP=0000 NU UP DI PL NZ NA PO NC
04B0:0000 B8AF04 MOV AX,04AF
-d 04ae:0
04AE:0000 23 01 56 04 00 00 00 00 00 00 00 00 00 00 00 00 #.U.....
04AE:0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
04AE:0020 B8 AF 04 8E D0 BC 10 00-B8 AE 04 8E D8 FF 36 00 8/.P<..8...X.6.
04AE:0030 00 FF 36 02 00 BF 06 02-00 BF 06 00 00 B8 00 4C ..6.....8.L
04AE:0040 CD 21 00 00 00 00 00 00-00 00 00 00 00 00 00 M!.....
04AE:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
04AE:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
04AE:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
```

3) Debug 查看 data 段中数据:

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

-t
AX=04AE BX=0000 CX=0042 DX=0000 SP=000E BP=0000 SI=0000 DI=0000
DS=04AE ES=049E SS=04AF CS=04B0 IP=0011 NU UP DI PL NZ NA PO NC
04B0:0011 FF360200 PUSH [0002] DS:0002=0456
-t
AX=04AE BX=0000 CX=0042 DX=0000 SP=000C BP=0000 SI=0000 DI=0000
DS=04AE ES=049E SS=04AF CS=04B0 IP=0015 NU UP DI PL NZ NA PO NC
04B0:0015 8F060200 POP [0002] DS:0002=0456
-t
AX=04AE BX=0000 CX=0042 DX=0000 SP=000E BP=0000 SI=0000 DI=0000
DS=04AE ES=049E SS=04AF CS=04B0 IP=0019 NU UP DI PL NZ NA PO NC
04B0:0019 8F060000 POP [0000] DS:0000=0123
-d ds:0
04AE:0000 23 01 56 04 00 00 00 00 00 00 00 00 00 00 00 00 #.U.....
04AE:0010 00 00 00 00 00 00 15 00-19 00 B0 04 92 01 23 01 .....0...#.
04AE:0020 B8 AF 04 8E D0 BC 10 00-B8 AE 04 8E D8 FF 36 00 8/.P<..8...X.6.
04AE:0030 00 FF 36 02 00 BF 06 02-00 BF 06 00 00 B8 00 4C ..6.....8.L
04AE:0040 CD 21 81 7E F6 3E 01 74-11 81 7E F6 3F 01 74 0A M!..~>.t..~>.t.
04AE:0050 81 7E F6 3D 01 74 03 EB-27 90 B8 4A 05 8E D8 BE ..~>.t.k'.BJ..X>
04AE:0060 0D 00 EB 01 90 8C D8 89-46 B6 89 76 B4 81 7E F6 ..k...X.F6..~>
04AE:0070 3D 01 74 0C 8C 56 1E 8D-46 B4 89 46 1C EB 1E 90 =.t..U..F4.F.k..
```

4) 结论: data 段共 16 字节。执行程序后, 定义的两个数据不变, 其余用 0 补全。

② CPU 执行程序, 程序返回前, cs = _____、ss = _____、ds = _____。

1) Debug 中查看 cs、ss、ds 内容:

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

-t
AX=04AE BX=0000 CX=0042 DX=0000 SP=000E BP=0000 SI=0000 DI=0000
DS=04AE ES=049E SS=04AF CS=04B0 IP=0011 NU UP DI PL NZ NA PO NC
04B0:0011 FF360200 PUSH [0002] DS:0002=0456
-t
AX=04AE BX=0000 CX=0042 DX=0000 SP=000C BP=0000 SI=0000 DI=0000
DS=04AE ES=049E SS=04AF CS=04B0 IP=0015 NU UP DI PL NZ NA PO NC
04B0:0015 8F060200 POP [0002] DS:0002=0456
-t
AX=04AE BX=0000 CX=0042 DX=0000 SP=000E BP=0000 SI=0000 DI=0000
DS=04AE ES=049E SS=04AF CS=04B0 IP=0019 NU UP DI PL NZ NA PO NC
04B0:0019 8F060000 POP [0000] DS:0000=0123
-d ds:0
04AE:0000 23 01 56 04 00 00 00 00 00 00 00 00 00 00 00 00 #.U.....
04AE:0010 00 00 00 00 00 00 15 00-19 00 B0 04 92 01 23 01 .....0...#.
04AE:0020 B8 AF 04 8E D0 BC 10 00-B8 AE 04 8E D8 FF 36 00 8/.P<..8...X.6.
04AE:0030 00 FF 36 02 00 BF 06 02-00 BF 06 00 00 B8 00 4C ..6.....8.L
04AE:0040 CD 21 81 7E F6 3E 01 74-11 81 7E F6 3F 01 74 0A M!..~>.t..~>.t.
04AE:0050 81 7E F6 3D 01 74 03 EB-27 90 B8 4A 05 8E D8 BE ..~>.t.k'.BJ..X>
04AE:0060 0D 00 EB 01 90 8C D8 89-46 B6 89 76 B4 81 7E F6 ..k...X.F6..~>
04AE:0070 3D 01 74 0C 8C 56 1E 8D-46 B4 89 46 1C EB 1E 90 =.t..U..F4.F.k..
```

2) 结论: cs = 04b0h, ss = 04afh, ds = 04ach。

③ 设程序加载后, code 段的段地址为 X, 则 data 段的段地址为 X-2, stack 段的段地址为 X-1。

分析: data 段与 stack 段恰好各占 16 字节, 因此段地址依次相差 1。

④ 对于如下定义的段:

```
name segment
...
name ends
```

如果段中的数据占 N 个字节, 则程序加载后, 该段实际占有的空间为 $\left(\left\lceil \frac{N}{16} \right\rceil + 1\right) \times 16$

字节。

分析: 地址分配以 16 字节为分配单元。不足 16 字节的按 16 字节算; 超过 16 字节整数倍的按一个 16 字节算。综合以上。N 字节数据实际占有空间为 $\left(\left\lceil \frac{N}{16} \right\rceil + 1\right) \times 16$ 字节。

(3) 将下面的程序编译、连接, 用 Debug 加载、跟踪, 然后回答问题。

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

```
code segment
```

```
start: mov ax,stack
      mov ss,ax
      mov sp,16
```

```
      mov ax,data
      mov ds,ax
```

```
      push ds:[0]
      push ds:[2]
      pop ds:[2]
      pop ds:[0]
```

```
      mov ax,4c00h
      int 21h
```

```
code ends
```

```
data segment
      dw 0123h,0456h
data ends
```

```
stack segment
      dw 0,0
stack ends
```

```
end start
```

① CPU 执行程序, 程序返回前, data 段中的数据为多少?

1) 编译、连接:

```
DOSBox 0.74-3-1, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c ~/MYDOSBox
Drive C is mounted as local directory /Users/tommy/MYDOSBox/

Z:\>c:

C:\>masm t3;
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

51654 + 464890 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>link t3;
Microsoft (R) Segmented-Executable Linker Version 5.13
Copyright (C) Microsoft Corp 1984-1991. All rights reserved.

LINK : warning L4021: no stack segment

C:\>_
```

2) code 段运行前 data 段的数据:

```
DOSBox 0.74-3-1, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c ~/MYDOSBox
Drive C is mounted as local directory /Users/tommy/MYDOSBox/

Z:\>c:

C:\>Debug t3.exe
-r
AX=FFFF BX=0000 CX=0044 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 B8B204 MOV AX,04B2
-d 04B1:0
04B1:0000 23 01 56 04 00 00 00 00-00 00 00 00 00 00 00 00 #.U.....
04B1:0010 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
04B1:0020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
04B1:0030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
04B1:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
04B1:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
04B1:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
04B1:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
```

3) Debug 查看 data 段中数据:

```
DOSBox 0.74-3-1, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
-t
AX=04B1 BX=0000 CX=0044 DX=0000 SP=000E BP=0000 SI=0000 DI=0000
DS=04B1 ES=049E SS=04B2 CS=04AE IP=0011 NU UP DI PL NZ NA PO NC
04AE:0011 FF360200 PUSH [0002] DS:0002=0456
-t
AX=04B1 BX=0000 CX=0044 DX=0000 SP=000C BP=0000 SI=0000 DI=0000
DS=04B1 ES=049E SS=04B2 CS=04AE IP=0015 NU UP DI PL NZ NA PO NC
04AE:0015 8F060200 POP [0002] DS:0002=0456
-t
AX=04B1 BX=0000 CX=0044 DX=0000 SP=000E BP=0000 SI=0000 DI=0000
DS=04B1 ES=049E SS=04B2 CS=04AE IP=0019 NU UP DI PL NZ NA PO NC
04AE:0019 8F060000 POP [0000] DS:0000=0123
-d ds:0
04B1:0000 23 01 56 04 00 00 00 00-00 00 00 00 00 00 00 00 #.U.....
04B1:0010 00 00 00 00 F6 3E 15 00-19 00 AE 04 92 01 23 01 ...>.....#.
04B1:0020 81 7E F6 3D 01 74 03 EB-27 90 B8 4A 05 8E D8 BE ...v=.t.k'.BJ..X>
04B1:0030 0D 00 EB 01 90 8C D8 89-46 B6 89 76 B4 81 7E F6 ...k...X.F6.v4..v
04B1:0040 3D 01 74 0C 8C 56 1E 8D-46 B4 89 46 1C EB 1E 90 ...t..U..F4.F.k..
04B1:0050 8D 76 AA 8B 46 12 E8 91-00 8C 46 B2 89 76 B0 8C ...v*.F.h...F2.v0.
04B1:0060 56 1E 8D 46 B0 89 46 1C-C7 46 1A 02 00 B8 2C 05 U..F0.F.6F...8,.
04B1:0070 BE D8 83 7E F6 02 75 07-8D 36 44 00 EB 3D 90 B1 ...X..v.u..6D.k=..
```

4) 结论: data 段共 16 字节。执行程序后, 定义的两个数据不变, 其余用 0 补全。

② CPU 执行程序, 程序返回前, cs = _____、ss = _____、ds = _____。

1) Debug 中查看 cs、ss、ds 内容:

```

DOSBox 0.74-3-1, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
-t
AX=04B1 BX=0000 CX=0044 DX=0000 SP=000E BP=0000 SI=0000 DI=0000
DS=04B1 ES=049E SS=04B2 CS=04AE IP=0011 NU UP DI PL NZ NA PO NC
04AE:0011 FF360200 PUSH [0002] DS:0002=0456
-t
AX=04B1 BX=0000 CX=0044 DX=0000 SP=000C BP=0000 SI=0000 DI=0000
DS=04B1 ES=049E SS=04B2 CS=04AE IP=0015 NU UP DI PL NZ NA PO NC
04AE:0015 8F060200 POP [0002] DS:0002=0456
-t
AX=04B1 BX=0000 CX=0044 DX=0000 SP=000E BP=0000 SI=0000 DI=0000
DS=04B1 ES=049E SS=04B2 CS=04AE IP=0019 NU UP DI PL NZ NA PO NC
04AE:0019 8F060000 POP [0000] DS:0000=0123
-d ds:0
04B1:0000 23 01 56 04 00 00 00-00 00 00 00 00 00 00 #.U.....
04B1:0010 00 00 00 00 F6 3E 15 00-19 00 AE 04 92 01 23 01 ...>.....#.
04B1:0020 81 7E F6 3D 01 74 03 EB-27 90 B8 4A 05 8E D8 BE ..v=.t.k'.BJ..X>
04B1:0030 0D 00 EB 01 90 8C D8 89-46 B6 89 76 B4 81 7E F6 ..k...X.F6.v4..v
04B1:0040 3D 01 74 0C 8C 56 1E 8D-46 B4 89 46 1C EB 1E 90 =.t...U...F4.F.k..
04B1:0050 8D 76 AA 8B 46 12 E8 91-00 8C 46 B2 89 76 B0 8C ..v*.F.h...F2.v0.
04B1:0060 56 1E 8D 46 B0 89 46 1C-C7 46 1A 02 00 B8 2C 05 U..F0.F.GF...8,.
04B1:0070 BE D8 83 7E F6 02 75 07-8D 36 44 00 EB 3D 90 81 .X..v.u...6D.k=..

```

2) 结论: cs = 04aeh, ss = 04b2h, ds = 04b1h。

③ 设程序加载后, code 段的段地址为 X, 则 data 段的段地址为 X+3, stack 段的段地址为 X+4。

分析: 通过 Debug 查看 code 段数据长度:

```

DOSBox 0.74-3-1, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
Z:\>SET BLASTER=A220 I7 D1 H5 T6
Z:\>mount c ~\MyDOSBox
Drive C is mounted as local directory /Users/tommy/MyDOSBox/
Z:\>c:
C:\>Debug t3.exe
-r
AX=FFFF BX=0000 CX=0044 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 B8B204 MOV AX,04B2
-u 04ae:0
04AE:0000 B8B204 MOV AX,04B2
04AE:0003 ED09 MOV SS,AX
04AE:0005 BC1000 MOV SP,0010
04AE:0008 B8B104 MOV AX,04B1
04AE:000B ED08 MOV DS,AX
04AE:000D FF360000 PUSH [0000]
04AE:0011 FF360200 PUSH [0002]
04AE:0015 8F060200 POP [0002]
04AE:0019 8F060000 POP [0000]
04AE:001D B8004C MOV AX,4C00
04AE:0020 CD21 INT 21
04AE:0022 0000 ADD [BX+SI],AL
04AE:0024 0000 ADD [BX+SI],AL

```

简单计算可得: code 段数据长度为 34 字节, 根据公式 $\left(\left\lceil \frac{N}{16} \right\rceil + 1\right) \times 16$, 34 字节数据实际占有 3 个 16 字节长度单元。因此, 若 code 段的段地址为 X, 则 data 段的段地址为 X+3, stack 段的段地址为 X+4。

(4) 如果将(1)、(2)、(3)题中的最后一条伪指令“end start”改为“end”(也就是说, 不指明程序的入口), 则哪个程序仍然可以正确执行? 请说明原因。

答: 都能执行, 但只有(3)可以正确执行。因为如果不指明程序入口, 程序将从头开始执行。那么(1)、(2)中一开始定义的数据将会被当作指令执行, 而(3)由于就是以 code 段开头, 因此不受影响, 可以正确执行。

- (5) 程序如下，编写 code 段中的代码，将 a 段和 b 段中的数据依次相加，将结果存到 c 段中。

```

assume cs:code

a segment
    db 1,2,3,4,5,6,7,8
a ends

b segment
    db 1,2,3,4,5,6,7,8
b ends

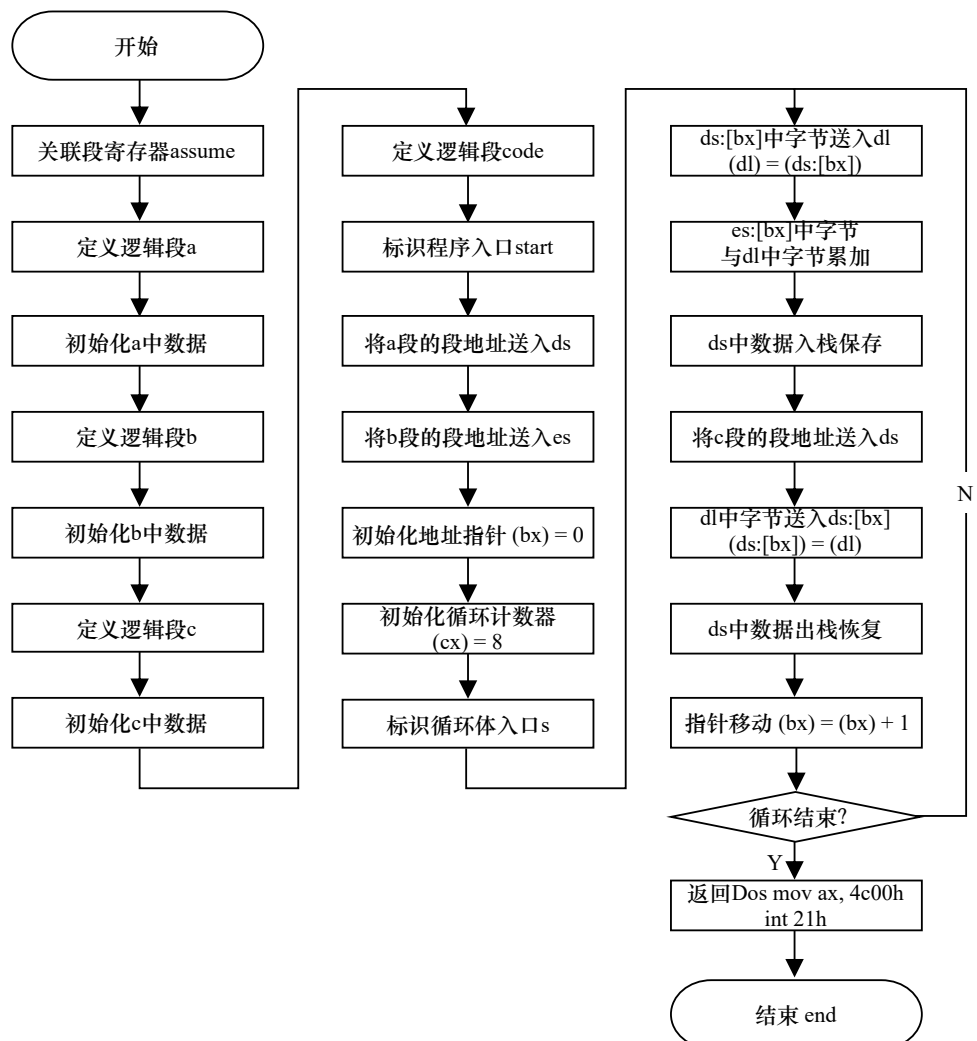
c segment
    db 0,0,0,0,0,0,0,0
c ends

code segment
start:
    ?
code ends

end start

```

1) 程序流程图:



2) 补全程序:

```
start: mov ax,a
      mov ds,ax
      mov ax,b
      mov es,ax
      mov bx,0
      mov cx,8
      s: mov dl,[bx]
         add dl,es:[bx]
         push ds
```

```
      mov ax,c
      mov ds,ax
      mov [bx],dl
      pop ds
      inc bx
      loop s

      mov ax,4c00h
      int 21h
```

(6) 程序如下, 编写 code 段中的代码, 用 push 指令将 a 段中的前 8 个字型数据逆序存储到 b 段中。

```
assume cs:code
```

```
a segment
```

```
    dw 1,2,3,4,5,6,7,8,9,0ah,0bh,0ch,0dh,0eh,0fh,0ffh
```

```
a ends
```

```
b segment
```

```
    dw 0,0,0,0,0,0,0,0
```

```
b ends
```

```
code segment
```

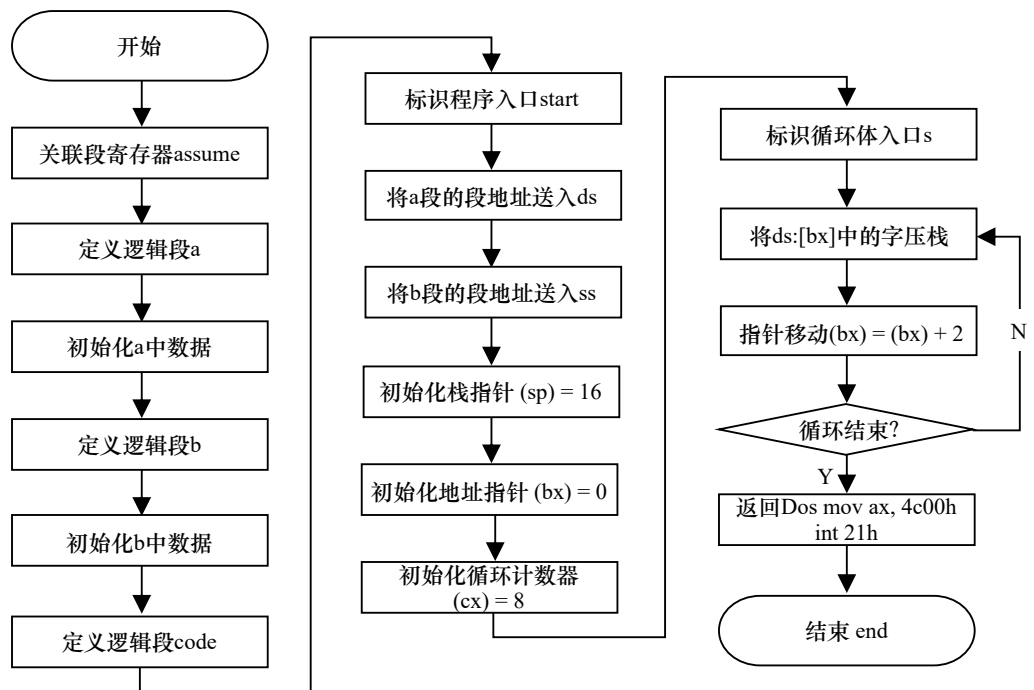
```
start:
```

```
    ?
```

```
code ends
```

```
end start
```

1) 程序流程图:



2) 补全程序:

```
start: mov ax,a
      mov ds,ax
      mov ax,b
      mov ss,ax
      mov sp,16
      mov bx,0
      mov cx,8
s:    push [bx]
```

```
      add bx,2
      loop s

      mov ax,4c00h
      int 21h
```