

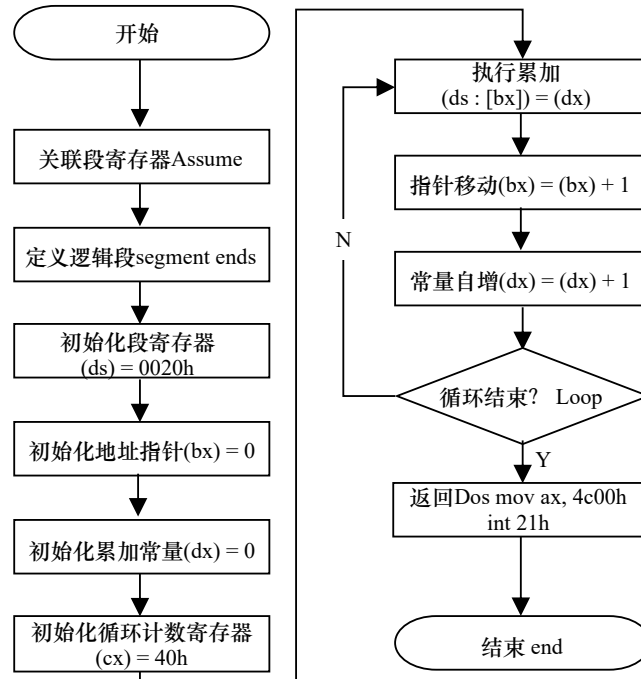
汇编语言程序设计课程作业（五）

姓名：袁昊男 学号：2018091618008

实验4 [bx]和 loop 的使用

(1) 编程，向内存 0:200~0:23F 依次传送数据 0~63(3FH)。

1) 程序流程图：



2) 代码如下：

```
assume cs:code
```

```
code segment
```

```
mov ax,0020h
mov ds,ax      ;ds 放段地址
mov bx,0       ;bx 放偏移地址，初始化为 0
mov dx,0       ;dx 放常量 0~63
mov cx,40h     ;cx 放循环次数
```

```
s: mov [bx],dx  ;向[bx]内存单元写入 dx 值
    inc bx     ;自增 bx
    inc dx     ;自增 dx
    loop s
```

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

```
code ends
```

```
end
```

3) 编译、连接：

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

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.

51770 + 464774 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
LINK : warning L4038: program has no starting address

C:\>
```

4) 跟踪、调试:

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

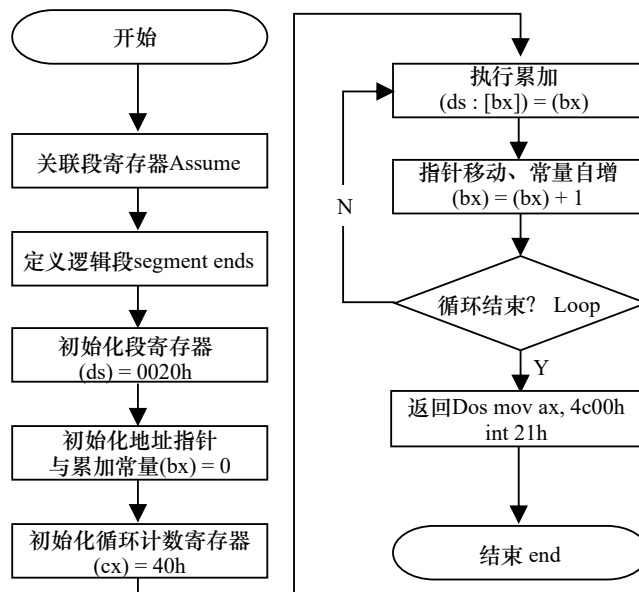
Z:\>C:

C:\>Debug t1.exe
-r
AX=FFFF BX=0000 CX=0019 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 B82000 MOV AX,0020
-u 04ae:0
04AE:0000 B82000 MOV AX,0020
04AE:0003 8ED8 MOV DS,AX
04AE:0005 BB0000 MOV BX,0000
04AE:0008 BA0000 MOV DX,0000
04AE:000B B94000 MOV CX,0040
04AE:000E 8917 MOV [BX],DX
04AE:0010 43 INC BX
04AE:0011 42 INC DX
04AE:0012 E2FA LODP 000E
04AE:0014 B8004C MOV AX,4C00
04AE:0017 CD21 INT 21
04AE:0019 0000 ADD [BX+SI],AL
04AE:001B 0000 ADD [BX+SI],AL
04AE:001D 0000 ADD [BX+SI],AL
04AE:001F 0000 ADD [BX+SI],AL
-
04AE:001D 0000 ADD [BX+SI],AL
04AE:001F 0000 ADD [BX+SI],AL
-g 0014
AX=0020 BX=0040 CX=0000 DX=0040 SP=0000 BP=0000 SI=0000 DI=0000
DS=0020 ES=049E SS=04AD CS=04AE IP=0014 NU UP DI PL NZ AC PO NC
04AE:0014 B8004C MOV AX,4C00
-t
AX=4C00 BX=0040 CX=0000 DX=0040 SP=0000 BP=0000 SI=0000 DI=0000
DS=0020 ES=049E SS=04AD CS=04AE IP=0017 NU UP DI PL NZ AC PO NC
04AE:0017 CD21 INT 21
-g
Program terminated normally
-d 0:200
0000:0200 00 01 02 03 04 05 06 07-08 09 0A 0B 0C 0D 0E 0F .....
0000:0210 10 11 12 13 14 15 16 17-18 19 1A 1B 1C 1D 1E 1F .....
0000:0220 20 21 22 23 24 25 26 27-28 29 2A 2B 2C 2D 2E 2F .....
0000:0230 30 31 32 33 34 35 36 37-38 39 3A 3B 3C 3D 3E 3F 0123456789:;<=>?
0000:0240 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:0250 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:0260 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:0270 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
```

(可以看到数据 0~63 已被正确传送到 0:200~0:23F)

- (2) 编程, 向内存 0:200~0:23F 依次传送数据 0~63(3FH), 程序中只能使用 9 条指令, 9 条指令中包括 “mov ax, 4c00h” 和 “int 21h”。

1) 程序流程图:



2) 代码如下:

```
assume cs:code
```

```
code segment
```

```
    mov ax,0020h
    mov ds,ax      ;ds 放段地址
    mov bx,0       ;bx 放偏移地址和常量数值，初始化为 0
    mov cx,40h     ;cx 放循环次数
```

```
s:  mov [bx],bx    ;向[bx]内存单元写入 bx 值
    inc bx        ;自增 bx
    loop s
```

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

```
code ends
```

```
end
```

3) 编译、连接:

```

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

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.

51770 + 464774 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
LINK : warning L4038: program has no starting address

C:\>
```

4) 跟踪、调试:

```

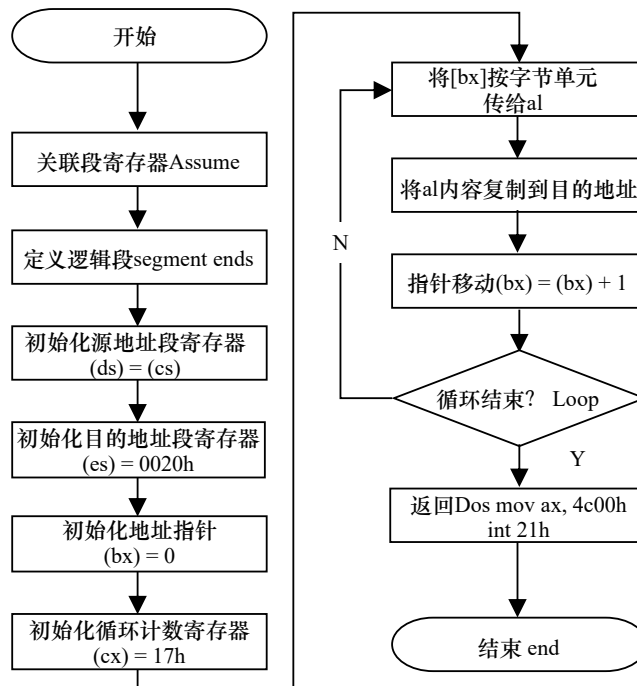
DOSBox 0.74-3-1, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
Z:\>c:
C:\>Debug t2.exe
-r
AX=FFFF BX=0000 CX=0015 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=049E ES=049E SS=04AD CS=04AE IP=0000 NU UP DI PL NZ NA PO NC
04AE:0000 B82000 MOV AX,0020
-u 04ae:0
04AE:0000 B82000 MOV AX,0020
04AE:0003 8ED8 MOV DS,AX
04AE:0005 BB0000 MOV BX,0000
04AE:0008 B94000 MOV CX,0040
04AE:000B 891F MOV [BX],BX
04AE:000D 43 INC BX
04AE:000E E2FB LOOP 000B
04AE:0010 B8004C MOV AX,4C00
04AE:0013 CD21 INT 21
04AE:0015 0000 ADD [BX+SI],AL
04AE:0017 0000 ADD [BX+SI],AL
04AE:0019 0000 ADD [BX+SI],AL
04AE:001B 0000 ADD [BX+SI],AL
04AE:001D 0000 ADD [BX+SI],AL
04AE:001F 0000 ADD [BX+SI],AL
-t
AX=0020 BX=0000 CX=0015 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=049E ES=049E SS=04AD CS=04AE IP=0003 NU UP DI PL NZ NA PO NC
04AE:0003 8ED8 MOV DS,AX
-g 0010
AX=0020 BX=0040 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0020 ES=049E SS=04AD CS=04AE IP=0010 NU UP DI PL NZ AC PO NC
04AE:0010 B8004C MOV AX,4C00
-g
Program terminated normally
-a 0:200
0000:0200 00 01 02 03 04 05 06 07-08 09 0A 0B 0C 0D 0E 0F .....
0000:0210 10 11 12 13 14 15 16 17-18 19 1A 1B 1C 1D 1E 1F .....
0000:0220 20 21 22 23 24 25 26 27-28 29 2A 2B 2C 2D 2E 2F .....
0000:0230 30 31 32 33 34 35 36 37-38 39 3A 3B 3C 3D 3E 3F .....
0000:0240 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:0250 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:0260 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:0270 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....

```

(可以看到数据 0~63 已被正确传送到 0:200~0:23F)

- (3) 下面的程序的功能是将“mov ax, 4c00h”之前的指令复制到内存 0:200 处，补全程序。
上机调试，跟踪运行结果。

1) 程序流程图:



2) 代码如下:

```
assume cs:code

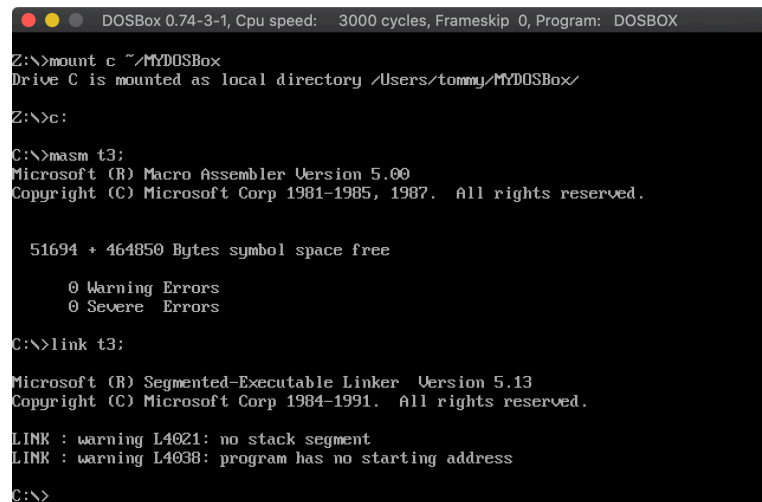
code segment

    mov ax, cs
    mov ds, ax
    mov ax, 0020h
    mov es, ax
    mov bx, 0
    mov cx, 0017h
s:  mov al, [bx]
    mov es:[bx], al
    inc bx
    loop s

    mov ax, 4c00h
    int 21h
code ends

end
```

3) 编译、连接:



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

```
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.

51694 + 464850 Bytes symbol space free

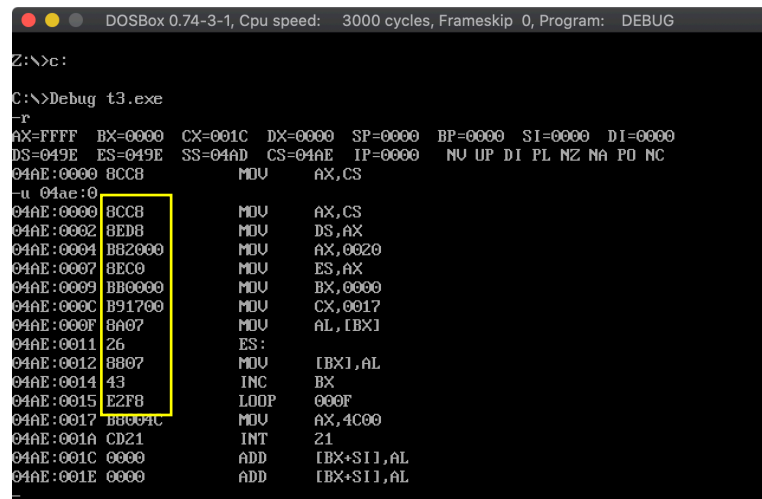
0 Warning Errors
0 Severe Errors

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
LINK : warning L4038: program has no starting address

C:\>_
```

4) 跟踪、调试:



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

```
Z:\>c:

C:\>Debug t3.exe
-r
AX=FFFF BX=0000 CX=001C 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 8CC8      MOV     AX,CS
-u 04ae:0
04AE:0000 8CC8      MOV     AX,CS
04AE:0002 8ED8      MOV     DS,AX
04AE:0004 B82000    MOV     AX,0020
04AE:0007 BEC0      MOV     ES,AX
04AE:0009 B80000    MOV     BX,0000
04AE:000C B91700    MOV     CX,0017
04AE:000F 8A07      MOV     AL,[BX]
04AE:0011 26        ES:
04AE:0012 8B07      MOV     [BX],AL
04AE:0014 43        INC     BX
04AE:0015 E2FB      LOOP    000F
04AE:0017 B8004C    MOV     AX,4C00
04AE:001A CD21      INT     21
04AE:001C 0000      ADD     [BX+SI],AL
04AE:001E 0000      ADD     [BX+SI],AL
t-
```

```
DOSBox 0.74-3-1, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
04AE:001C 0000      ADD     [BX+SI],AL
04AE:001E 0000      ADD     [BX+SI],AL
-t
AX=04AE BX=0000 CX=001C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=049E ES=049E SS=04AD CS=04AE IP=0002  NU UP DI PL NZ NA PO NC
04AE:0002 8ED8      MOV     DS,AX
-g 0017
AX=00F8 BX=0017 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=04AE ES=0020 SS=04AD CS=04AE IP=0017  NU UP DI PL NZ NA PE NC
04AE:0017 B8004C    MOV     AX,4C00
-g
Program terminated normally
-d 0:200
0000:0200 8C C8 8E D8 B8 20 00 8E C0 BB 00 00 B9 17 00 8A .H.XB ..e:..9...
0000:0210 07 26 88 07 43 E2 F8 00 00 00 00 00 00 00 00 .&...CbX.....
0000:0220 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000:0230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000:0240 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000:0250 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000:0260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000:0270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

（可以看到“mov ax, 4c00h”之前的 17 条指令已被正确传送到 0:200~0:216）

注：在程序还没有运行时，我们不知道在“mov ax, 4c00h”之前有多少条指令，因此无法确定控制循环次数的 cx 寄存器内容。可先随意对 cx 赋值，在 Debug 程序中反汇编，查看指令条数。更正 cx 内容后重新编译、连接即可。