北京师范大学2018-2019学年第一学期 “汇编程序设计”

上机作业七

姓 名	学 号	日 期
刘源	201611210134	2018.11.11

实验目的

- **ret、retf、call 指令的应用。**
- **学会子程序设计**
- **mul 乘法指令的应该用**

实验总结:

- 第一题在分析源代码时犯了一个很简单，但却容易被忽视的错误，就是复制s2的代码时，认为机器复制的是源代码，所以以为会跳转到s1，但由于机器先编译，后运行，复制的是机器码，再加上jmp是一个相对跳转指令，所以应该跳转到0000，而不是0018
- 第二题在书写代码时，发生了很多错误，像dd和dw的混淆，还有没有考虑到进位等细节。而且感觉用汇编去写一个C语言很好写的代码时，也非常难写，最后写成也是边参考C语言代码边一点点添加，还有同学提醒一些细节，最后才完成，还是写的少，对代码生疏的原因。

第 1 题

源代码

```
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
```

运行结果

```
D:\ASM>debug 9.exe
-T

AX=0000  BX=0000  CX=0023  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075C  ES=075C  SS=076B  CS=076C  IP=0008   NV UP EI PL NZ NA PO NC
076C:0008 90            NOP
- ;
```

从strat处开始执行代码

```
-T

AX=0000  BX=0000  CX=0023  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075C  ES=075C  SS=076B  CS=076C  IP=000A   NV UP EI PL NZ NA PO NC
076C:000A BF0800        MOV     DI,0008
-T

AX=0000  BX=0000  CX=0023  DX=0000  SP=0000  BP=0000  SI=0000  DI=0008
DS=075C  ES=075C  SS=076B  CS=076C  IP=000D   NV UP EI PL NZ NA PO NC
076C:000D BE2000        MOV     SI,0020
```

offset s、s2分别表示s和s2代码的地址

```
-T

AX=0000  BX=0000  CX=0023  DX=0000  SP=0000  BP=0000  SI=0020  DI=0008
DS=075C  ES=075C  SS=076B  CS=076C  IP=0010   NV UP EI PL NZ NA PO NC
076C:0010 2E            CS:
076C:0011 8B04          MOV     AX,[SI]                         CS:0020=F6EB
-T

AX=F6EB  BX=0000  CX=0023  DX=0000  SP=0000  BP=0000  SI=0020  DI=0008
DS=075C  ES=075C  SS=076B  CS=076C  IP=0013   NV UP EI PL NZ NA PO NC
076C:0013 2E            CS:
076C:0014 8905          MOV     [DI],AX                         CS:0008=9090
```

将s2处的代码复制到s中的nop处

```
-U 076C:0008
076C:0008 EBF6          JMP     0000
076C:000A BF0800        MOV     DI,0008
076C:000D BE2000        MOV     SI,0020
076C:0010 2E            CS:
076C:0011 8B04          MOV     AX,[SI]
076C:0013 2E            CS:
076C:0014 8905          MOV     [DI],AX
076C:0016 EBF0          JMP     0008
076C:0018 B80000        MOV     AX,0000
076C:001B CD21          INT     21
076C:001D B80000        MOV     AX,0000
076C:0020 EBF6          JMP     0018
076C:0022 90            NOP
076C:0023 01B82F00      ADD     [BX+SI+002F],DI
076C:0027 50            PUSH    AX
```

反汇编执行s处的代码，发现复制过来的不是JMP 0018，而是JMP 0000

```
-T

AX=F6EB  BX=0000  CX=0023  DX=0000  SP=0000  BP=0000  SI=0020  DI=0008
DS=075C  ES=075C  SS=076B  CS=076C  IP=0008   NV UP EI PL NZ NA PO NC
076C:0008 EBF6          JMP     0000
-T

AX=F6EB  BX=0000  CX=0023  DX=0000  SP=0000  BP=0000  SI=0020  DI=0008
DS=075C  ES=075C  SS=076B  CS=076C  IP=0000   NV UP EI PL NZ NA PO NC
076C:0000 B8004C        MOV     AX,4C00
-T

AX=4C00  BX=0000  CX=0023  DX=0000  SP=0000  BP=0000  SI=0020  DI=0008
DS=075C  ES=075C  SS=076B  CS=076C  IP=0003   NV UP EI PL NZ NA PO NC
076C:0003 CD21          INT     21
```

运行后也可以看出它跳向了codesg的开头，结束了该程序

第 2 题

源代码

```
assume cs:codesg
data segment
    dw 1, 2, 3, 4, 5, 6, 7, 8, 9
    dw 9, 8, 7, 6, 5, 4, 3, 2, 1
data ends

result segment
    dd 0, 0, 0, 0, 0, 0, 0, 0, 0
result ends

codesg segment
    start:
        mov ax, data
        mov ds, ax          ;将a,b矩阵移入ds
        mov ax, result
        mov es, ax
```

```
        call matrix_mul

        mov ax, 4c00h
        int 21h

    matrix_mul:
        mov bx, 0               ;bx = i
        mov cx, 3
        s1:
            push cx
            mov cx, 3
            mov di, 0           ;di = j
        s2:
            push cx
            mov cx, 3
            mov si, 0           ;si = k
        s3:
            mov ax, 3
            mul bx
            mov bp, ax
            mov ax, ds:[bp + si]    ;ax = a[i][k]
            push ax

            mov ax, 3
            mul si
            mov bp, ax
            pop ax
            mul word ptr ds:[bp + 18 + di]  ;ax = a[i][k] * b[k][j]
            push ax

            mov ax, 3
            mul bx
            add ax, di
            mov bp, ax          ;bp = 3 * i + j
            add bp, bp
            pop ax
            add es:[bp], ax
            add es:[bp + 2], dx

            add si, 2
            loop s3

            pop cx
            add di, 2
            loop s2

            pop cx
            add bx, 2
            loop s1
            ret
codesg ends

end start
```

运行结果

```
-t

AX=0009  BX=0006  CX=0000  DX=0000  SP=0000  BP=0020  SI=0006  DI=0006
DS=076C  ES=076F  SS=076B  CS=0772  IP=000D   NV UP EI PL NZ NA PE NC
0772:000D B8004C        MOV     AX,4C00
-t

AX=4C00  BX=0006  CX=0000  DX=0000  SP=0000  BP=0020  SI=0006  DI=0006
DS=076C  ES=076F  SS=076B  CS=0772  IP=0010   NV UP EI PL NZ NA PE NC
0772:0010 CD21          INT     21
-t

AX=4C00  BX=0006  CX=0000  DX=0000  SP=FFFA  BP=0020  SI=0006  DI=0006
DS=076C  ES=076F  SS=076B  CS=F000  IP=14A0   NV UP DI PL NZ NA PE NC
F000:14A0 FB            STI
-D 076F:0000
076F:0000  1E 00 00 00 18 00 00 00-12 00 00 00 54 00 00 00   ............T...
076F:0010  45 00 00 00 36 00 00 00-8A 00 00 00 72 00 00 00   E...6.......r...
076F:0020  5A 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   Z...............
076F:0030  B8 6C 07 8E D8 B8 6F 07-8E C0 E8 05 00 B8 00 4C   .l....o........L
076F:0040  CD 21 BB 00 00 B9 03 00-51 B9 03 00 BF 00 00 51   .!......Q......Q
076F:0050  B9 03 00 BE 00 00 B8 03-00 F7 E3 8B E8 3E 8B 02   .............>..
076F:0060  50 B8 03 00 F7 E6 8B E8-58 3E F7 63 12 50 B8 03   P.......X>.c.P..
076F:0070  00 F7 E3 03 C7 8B E8 03-ED 58 26 01 46 00 26 01   .........X&.F.&.
```

图中076F:0000~076F:0024为所求结果

由于:

$$\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}*\begin{bmatrix}9&8&7\\6&5&4\\3&2&1\end{bmatrix}=\begin{bmatrix}30&24&18\\84&69&54\\138&114&90\end{bmatrix}$$

可知结果正确