

《汇编语言》

实验报告

实验名称：实验一 用机器指令和汇编指令程序

姓名：李**

学号：512016****

专业班级：计科卓越 1601

实验时间：2018 年 4 月 1 日

一、实验目的

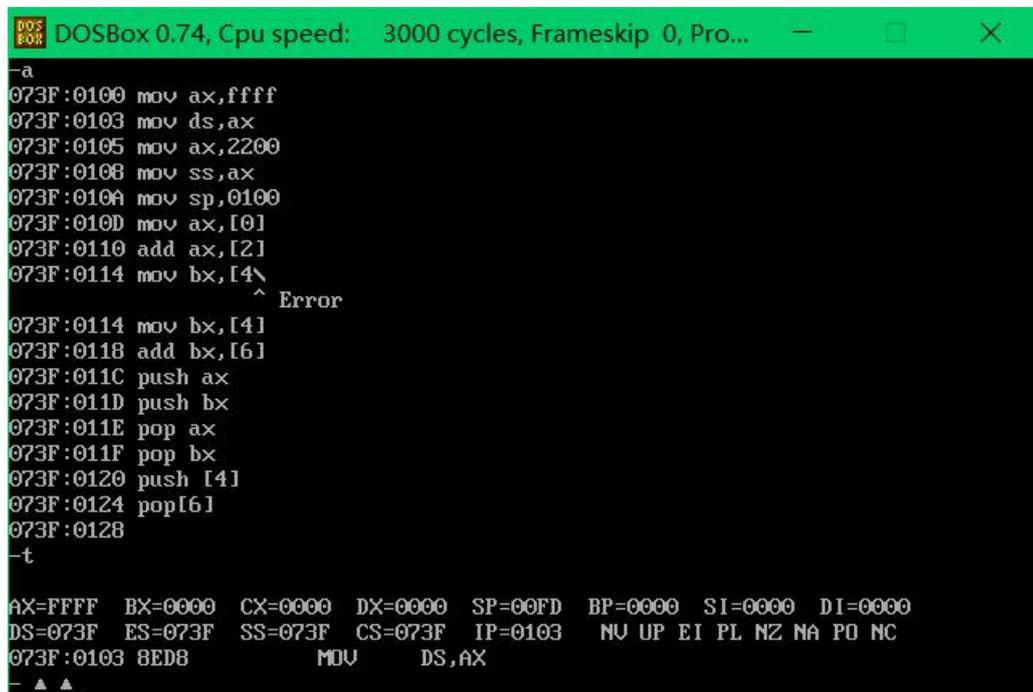
- 1、熟悉基本 debug 命令：a、r、d、e、u、t
- 2、掌握汇编指令与机器指令的对应关系
- 3、掌握利用 debug 命令查看修改内存、寄存器的方法

二、实验要求

- (1) 仔细阅读实验教程中 DEBUG 的使用部分；
- (2) 使用 DEBUG 中的 A 命令输入一段程序段；
- (3) 用 DEBUG 中的 R 命令观察寄存器中数据的存放情况，改变寄存器的值；
- (4) 用 DEBUG 中的 D 命令查看数据在内存中的表示方法；
- (5) 用 DEBUG 中的 E 命令修改内存中的数据；
- (6) 用 DEBUG 中的 T 命令执行一条语句；

三、实验步骤和实验内容

(1) 使用 debug，将下面程序段写入内存，逐条执行，根据指令执行后的实际情况填空。



```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro...
-a
073F:0100 mov ax,ffff
073F:0103 mov ds,ax
073F:0105 mov ax,2200
073F:0108 mov ss,ax
073F:010A mov sp,0100
073F:010D mov ax,[0]
073F:0110 add ax,[2]
073F:0114 mov bx,[4\
      ^ Error
073F:0114 mov bx,[4]
073F:0118 add bx,[6]
073F:011C push ax
073F:011D push bx
073F:011E pop ax
073F:011F pop bx
073F:0120 push [4]
073F:0124 pop [6]
073F:0128
-t
AX=FFFF BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0103 NU UP EI PL NZ NA PO NC
073F:0103 8ED8          MOV     DS,AX
```

图 1-1 用-a 命令输入全部程序指令

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro...
AX=C0FC BX=30F0 CX=0000 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=FFFF ES=073F SS=2200 CS=073F IP=0118 NU UP EI NG NZ NA PE NC
073F:0118 031E0600 ADD BX,[0006] DS:0006=2F31
-t
AX=C0FC BX=6021 CX=0000 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=FFFF ES=073F SS=2200 CS=073F IP=011C NU UP EI PL NZ NA PE NC
073F:011C 50 PUSH AX
-t
AX=C0FC BX=6021 CX=0000 DX=0000 SP=00FE BP=0000 SI=0000 DI=0000
DS=FFFF ES=073F SS=2200 CS=073F IP=011D NU UP EI PL NZ NA PE NC
073F:011D 53 PUSH BX
-t
AX=C0FC BX=6021 CX=0000 DX=0000 SP=00FC BP=0000 SI=0000 DI=0000
DS=FFFF ES=073F SS=2200 CS=073F IP=011E NU UP EI PL NZ NA PE NC
073F:011E 58 POP AX
-t
AX=6021 BX=6021 CX=0000 DX=0000 SP=00FE BP=0000 SI=0000 DI=0000
DS=FFFF ES=073F SS=2200 CS=073F IP=011F NU UP EI PL NZ NA PE NC
073F:011F 5B POP BX
- ▲
    
```

图 1-2 用-t 命令执行每段指令

(2)重新执行上面的程序。在执行前 4 条语句后，使用-e 0FFFF:0 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8，再执行后面的语句，每条语句的执行结果会有什么变化？为什么？

在使用-e 命令后，使用-d 0ffff:0 f 查看从 ffff:0 开始的 16 个单元内容。

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro...
AX=FFFF BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0103 NU UP EI PL NZ NA PO NC
073F:0103 8ED8 MOV DS,AX
-t
AX=FFFF BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=FFFF ES=073F SS=073F CS=073F IP=0105 NU UP EI PL NZ NA PO NC
073F:0105 B80022 MOV AX,2200
-t
AX=2200 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=FFFF ES=073F SS=073F CS=073F IP=0108 NU UP EI PL NZ NA PO NC
073F:0108 8ED0 MOV SS,AX
-t
AX=2200 BX=0000 CX=0000 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=FFFF ES=073F SS=2200 CS=073F IP=010D NU UP EI PL NZ NA PO NC
073F:010D A1AEFE MOV AX,[FEAE] DS:FEAE=0789
-d ffff:0 f
FFFF:0000 EA C0 12 00 F0 30 31 2F-30 31 2F 39 32 00 FC 55 ....01/01/92..U
-e ffff:0 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8
-d ffff:0 f
FFFF:0000 EA C0 12 00 F0 30 31 2F-30 31 2F 39 32 00 FC 55 ....01/01/92..U
- ▲
    
```

图 1-3 修改并查看 ffff:0~ffff:f 的内容

由于地址单元信息不能随便修改，所以这里实验失败。这里我们使用 1000:0 到 1000:f 修改成功。

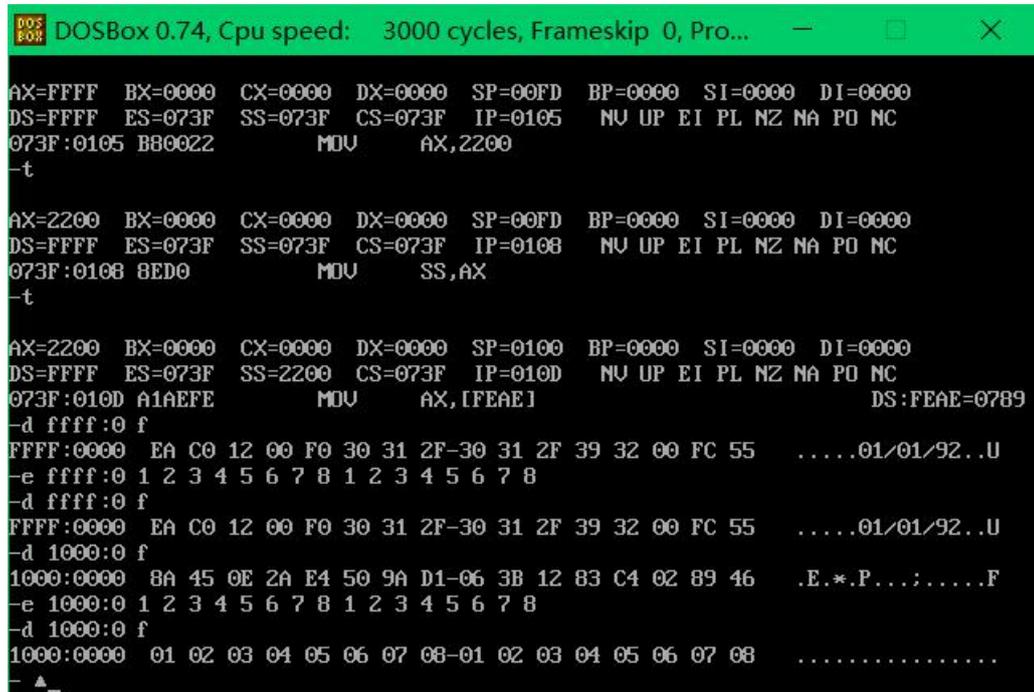


图 1-4 修改并查看 1000:0~1000:f 的内容

四、实验结果

(1)使用 debug，将下面程序段写入内存，逐条执行，根据指令执行后的实际情况填空。

```

mov ax,ffffh

mov ds,ax

mov ax,2200h

mov ss,ax

mov sp,0100

mov ax,[0]          ;(ax=)___ C0EAH ___

add ax,[2]          ;(ax=)___ C0FCH ___

mov bx,[4]          ;(bx=)___ 30F0H ___

add bx,[6]          ;(bx=)___ 6021H ___

push ax             ;(sp=)___ 00FEH ___; 修改的内存单元的地址是
2200:0100 ___; 内容是 COECH ___

```

push bx ;(sp=) 00FCH ; 修改的内存单元的地址是 2200:00FEH ; 内容是 6021H

pop ax ;(sp=) 00FEH ; (ax) = 6021H

pop bx ;(sp=) 0100H ; (ax) = C0FCH

push [4] ;(sp=) 00FEH ; 修改的内存单元的地址是 2200:0100H ; 内容是 30F0H (数据段 DS:0004=30F0)

pop [6] ;(sp=) 0100H ; 修改的内存单元的地址是 2200:00FEH ; 内容是 2F31H (数据段 DS:0004=2F31)

(2)重新执行上面的程序。在执行前4条语句后，使用-e 0FFFF:0 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8，再执行后面的语句，每条语句的执行结果会有什么变化？为什么？

在使用-e命令后，使用-d 0ffff:0 f 查看从ffff:0开始的16个单元内容。但是因为有些内容内的关键信息不能被随意修改，所以这里的实验结果并没有什么变化（如图1-3）。这里我再尝试了一下，如果修改1000:0到1000:f的内容，就修改成功，（如图1-4）。

五、实验心得

①给数据段ds、堆栈段ss添加地址时，需要先把数据mov到ax中，在用mov移动到数据段和堆栈段中。

②-a与-t命令并用，-a输入一段程序，-t执行下一条指令

③-r命令查看或修改单个寄存器的值

格式：r/rax

④-d与-e命令并用，-d查看一段地址的内容，-e修改一段地址的内容

格式：d 1000:0 f 查看10000到1000f的地址内容

e 1000:0 0 1 2 3 4 5 6 7 8 9 依次修改1000:0开始的10个内容单元

这次实验比较简单，帮助我理解和熟悉了debug的命令下，每条指令执行了什么，改变了什么。