# 《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

西南科技大学计算机科学与技术学院

## 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, Fran	neskip 0, Pro	-	×
-a 073F:0100 mov ax,ffff 073F:0103 mov ds,ax 073F:0105 mov ax,2200 073F:0108 mov ss,ax 073F:0108 mov ss,0100 073F:0100 mov ax,[0] 073F:0110 add ax,[2] 073F:0114 mov bx,[4] 073F:0114 mov bx,[4] 073F:0118 add bx,[6]				
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 D	X=0000 SP=00FD	BP=0000 SI=0000	9 DI=0000	
DS=073F ES=073F SS=073F C	S=073F IP=0103	NV UP EI PL NZ	NA PO NC	
073F:0103 8ED8 MOV	DS,AX			
- A A				

图 1-1 用-a 命令输入全部程序指令

B DOSE	3ox 0.74, 0	Cpu speed	: 3000 c	ycles, Fran	neskip 0, Pro — 🛛	×
AX=C0FC DS=FFFF 073F:011 -t	BX=30F0 ES=073F 8 031E060	CX=0000 SS=2200 0 AD	DX=0000 CS=073F D BX,	SP=0100 IP=0118 [0006]	BP=0000 SI=0000 DI=0000 NU UP EI NG NZ NA PE NC DS:000	6=2F31
AX=COFC DS=FFFF 073F:011 -t	BX=6021 ES=073F C 50	CX=0000 SS=2200 PU	DX=0000 CS=073F SH AX	SP=0100 IP=011C	BP=0000 SI=0000 DI=0000 NV UP EI PL NZ NA PE NC	
AX=C0FC DS=FFFF 073F:011 -t	BX=6021 ES=073F D 53	CX=0000 SS=2200 PU	DX=0000 CS=073F SH BX	SP=00FE IP=011D	BP=0000 SI=0000 DI=0000 NU UP EI PL NZ NA PE NC	
AX=C0FC DS=FFFF 073F:011 -t	BX=6021 ES=073F E 58	CX=0000 SS=2200 PO	DX=0000 CS=073F P AX	SP=00FC IP=011E	BP=00000 SI=00000 DI=00000 NUUPEIPLNZNAPENC	
AX=6021 DS=FFFF 073F:011 - ▲	BX=6021 ES=073F F 5B	CX=0000 SS=2200 PO	DX=0000 CS=073F P BX	SP=00FE IP=011F	BP=0000 SI=0000 DI=0000 NV UP EI PL NZ NA PE NC	

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:01234567812345678,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 Offff:0f 查看从 ffff:0开始的 16个单元内容。

DOSBox 0.74	, Cpu speed:	3000 cycles, Fra	meskip 0, Pro —	
AX=FFFFF BX=000 DS=073F ES=073 073F:0103 8ED8 -t	0 CX=0000 DX F SS=073F CS MDV	=0000 SP=00FD =073F IP=0103 DS,AX	BP=0000 SI=0000 NV UP EI PL NZ NA	DI=0000 PO NC
AX=FFFF BX=000 DS=FFFF ES=073 073F:0105 B8002 -t	0 CX=0000 DX F SS=073F CS 2 MDV	=0000 SP=00FD =073F IP=0105 AX,2200	BP=0000 SI=0000 NV UP EI PL NZ NA	DI=0000 PONC
AX=2200 BX=000 DS=FFFF ES=073 073F:0108 8ED0 -t	0 CX=0000 DX F SS=073F CS MDV	=0000 SP=00FD =073F IP=0108 SS,AX	BP=0000 SI=0000 NV UP EI PL NZ NA	DI=0000 PONC
AX=2200 BX=000 DS=FFFF ES=073 073F:010D A1AEF	0 CX=0000 DX F SS=2200 CS E MDV	=0000 SP=0100 =073F IP=010D AX,[FEAE]	BP=0000 SI=0000 NV UP EI PL NZ NA	D I = 0000 PONC DS : FEAE = 0789
-a ffff:0 f FFFF:0000 EA C -e ffff:0 1 2 3 -d ffff:0 f	01200F030 4567812	31 2F-30 31 2F 3 4 5 6 7 8	39 32 00 FC 55	01/01/92U
FFFF:0000 EA C - ▲_	0 12 00 FO 30	31 2F-30 31 2F	39 32 00 FC 55	01/01/92U

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。

🚟 DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro... × BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000 AX=FFFF IP=0105 DS=FFFF ES=073F SS=073F CS=073F NU UP EI PL NZ NA PO NC MOV AX,2200 073F:0105 B80022 -t. AX=2200 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000 SS=073F CS=073F IP=0108 DS=FFFF ES=073F NV UP EI PL NZ NA PO NC 073F:0108 8ED0 MOV SS,AX 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 NV UP EI PL NZ NA PO NC 973F:010D A1AEFE MOV AX, [FEAE] DS:FEAE=0789 d ffff:0 f FFF:0000 EA CO 12 00 FO 30 31 2F-30 31 2F 39 32 00 FC 55 .....01/01/92..U e ffff:01234567812345678 d ffff:0 f FFF:0000 EA CO 12 00 FO 30 31 2F-30 31 2F 39 32 00 FC 55 .....U/01/92..U d 1000:0 f 1000:0000 8A 45 0E 2A E4 50 9A D1-06 3B 12 83 C4 02 89 46 .E.\*.P...;....F 1000:01234567812345678 e 1000:0 f -d 1000:0000 01 02 03 04 05 06 07 08-01 02 03 04 05 06 07 08

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=) COEAH add ax,[2] ;(ax=) <u>COFCH</u> mov bx,[4] ;(bx=)\_\_\_30F0H add bx,[6] ;(bx=) 6021H ;(sp=) 00FEH ; 修改的内存单元的地址是 push ax 2200:0100 ; 内容是 COECH

#### 2017-2018-2

push bx	;(sp=)00FCH;修改的内存单元的地址是
2200:00FEH	_; 内容是 <u>6021H</u>
рор ах	;(sp=) <u>00FEH</u> ; (ax) = <u>6021H</u>
pop bx	;(sp=) <u>0100H</u> ; (ax) = <u>COFCH</u>
push [4]	;(sp=);修改的内存单元的地址是
2200:0100H	;内容是 <u>30F0H(数据段 DS:0004=30F0)</u>
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:01234567812345678,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 1000:00123456789 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。