

《汇编语言》实验报告

班级	2022 秋	实验日期	2022.10.14	实验成绩	
姓名	任宇	学号	33920212204567		
实验名称	汇编语言第三次实验				
实验目的、要求	<p>实验目的： 熟练使用 Debug,理解数据在内存中的存放,并理解并练习各种寻址方式。</p> <p>实验题目：</p> <p>1) 在数据段中依次存入 10H, 11H, 12H, 13H, 14H, 15H, 16H, 17H, 将其相加, 并将结果存入 DX 寄存器。</p> <p>2) 练习使用 debug 命令破解 bios 密码, 写出自己对破解密码的理解。</p> <p>3) 在长度为 8 的字节数组 (无符号数) 中, 查找大于 42H 的无符号数的个数, 存放在字节单元 up 中; 等于 42H 的无符号数的个数, 存放在字节单元 equa 中; 小于 42H 的无符号数的个数, 存放在字节单元 down 中。程序显示 up equa down 的值。</p> <p>八个数: 31H, 21H, 42H, 52H, 87H, 23H, 98H, 01H</p>				
实验内容、步骤及结果	<p>1) 在数据段中依次存入 10H, 11H, 12H, 13H, 14H, 15H, 16H, 17H, 将其相加, 并将结果存入 DX 寄存器, 程序如下:</p> <pre>ASSUME CS:CODE, DS:DATA DATA SEGMENT NUM DW 10H, 11H, 12H, 13H, 14H, 15H, 16H, 17H DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV DS, AX MOV SI, OFFSET NUM MOV CX, 8 MOV DX, 0 SUM: ADD DX, [SI] ADD SI, 2 LOOP SUM MOV AX, 4C00H INT 21H CODE ENDS END START</pre> <p>接着在 dosbox 中编译, 连接并 debug 该程序:</p>				

```

Z:\>mount c e:\dos\asm
Drive C is mounted as local directory e:\dos\asm\

Z:\>c:

C:\>masm sum.asm
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993. All rights reserved.

    Invoking: ML.EXE /I. /Zm /c /Ta sum.asm

Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993. All rights reserved.

    Assembling: sum.asm

```

```

C:\>link sum.obj

Microsoft (R) Segmented Executable Linker Version 5.31.009 Jul 13 1992
Copyright (C) Microsoft Corp 1984-1992. All rights reserved.

Run File [sum.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:
LINK : warning L4021: no stack segment

```

利用 t 命令执行程序，并得到 dx 寄存器中的数据：

```

-t
AX=076A BX=0000 CX=0000 DX=009C SP=0000 BP=0000 SI=0010 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0015  NU UP EI PL NZ AC PO NC
076B:0015 B8004C      MOV     AX,4C00
-t
AX=4C00 BX=0000 CX=0000 DX=009C SP=0000 BP=0000 SI=0010 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0018  NU UP EI PL NZ AC PO NC
076B:0018 CD21      INT     21
- ^

```

2) 利用 Debug 破解 BIOS 密码:



DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro...

```

Welcome to DOSBox v0.74

For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

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

Z:\>mount c e:\dos\asm
Drive C is mounted as local directory e:\dos\asm\

Z:\>c:

C:\>debug
-o 70 16
-o 71 16
-q

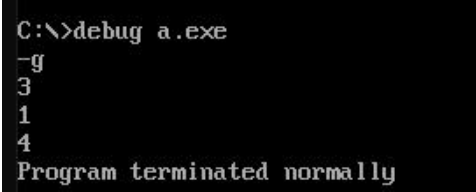
```

Ps：实际上我们是通过 BIOS 这个程序，去设置 CMOS 里的参数的。

电脑的 BIOS 设置一般是通过 70H 和 71H 两个端口进行访问和更改的，端口 70H 是一个字节的地址端口,用来设置 CMOS 中数据的地址,而端口 71H 则是用来读写 CMOS 地址中的数据单元内容，对这两个端口进行错误的赋值，使设置全部清空，恢复成出厂设置，就可以破解 BIOS 密码。

3) 编程程序如下图，利用 cmp 指令以及 jz, ja 来判断大小：

ASSUME		CS:CODE, DS:DATA
DATA	SEGMENT	
NUM	DW	31H, 21H, 42H, 52H, 87H, 23H, 98H, 01H
U	DB	0
E	DB	0
D	DB	0
DATA	ENDS	
CODE	SEGMENT	
START:		
	MOV	AX, DATA
	MOV	DS, AX
	MOV	SI, OFFSET NUM
	MOV	CX, 8
	MOV	AL, 0
COMPARE:		
	MOV	AL, [SI]
	CMP	AL, 42H
	JZ	EUQA
	JA	NEXT
	INC	D
	ADD	SI, 2
	LOOP	COMPARE
	JMP	PRINT
EUQA:	INC	E
	ADD	SI, 2
	LOOP	COMPARE
	JMP	PRINT
NEXT:	INC	U
	ADD	SI, 2
	LOOP	COMPARE
	JMP	PRINT

	<div>PRINT:</div> <div><div>MOV DL, U</div><div>ADD DL, '0'</div><div>MOV AH, 02</div><div>INT 21H</div><div>MOV DL, 10</div><div>INT 21H</div><div> </div><div>MOV DL, E</div><div>ADD DL, '0'</div><div>MOV AH, 02</div><div>INT 21H</div><div>MOV DL, 10</div><div>INT 21H</div><div> </div><div>MOV DL, D</div><div>ADD DL, '0'</div><div>MOV AH, 02</div><div>INT 21H</div><div>MOV AX, 4C00H</div><div>INT 21H</div><div> </div><div>CODE ENDS</div><div> END START</div><div>在 dosbox 里编译，连接并运行，如下图，分别对应 up, equal, down 的值：</div><div></div><div>结果正确。</div></div>
总结	<div>通过这次实验，熟练使用了 Debug，更深理解了数据在内存中的存放，但对于问题三如何将整数输出还有不懂的地方，输出的值会以 ascii 码值输出，目前只限于计数小于 10 时可以正确输出，还需要继续改进。</div>