第一个程序

实验步骤：

一、启动 debug 程序

打开 Dosbox, 出现 Z:\>

键入 mount d: d:\, 回车

出现 Drive D is mounted as local directory d:\

Z:\>

键入 D:, 回车

出现 D:\>

键入 debug，回车

出现 debug 程序的提示符，一个短杠

二、汇编第一个程序

-a100 //将汇编的首地址设为 100

0AE9:0100 mov dl,1 //将数值 01h （系统自带的，笑脸）放进 dl 寄存器

0AE9:0102 mov ah,2 //ah 放 2，是 2 号功能,字符输出，并且要输出的字符是已经放在 dl 里面

0AE9:0104 int 21 //调用 DOS 21 号中断 2 号功能，用来逐个显示装入 DL 的字符

0AE9:0106 int 20 //调用 DOS 20 号中断，终止程序，将控制权交回给 DEBUG

0AE9:0108

三、用W命令将该程序写入（Write）磁盘中

-rbx //查看 BX 寄存器的内容

BX 0000

:

-rcx //查看 CX 寄存器的内容

CX 0000

:8 /程序的字节数为 8

-w //用W命令将该程序写入（Write）磁盘中

Writing 00008 bytes

四、退出 debug

-q

D:\>

五、运行该程序

//由于版本问题，直接执行软件显示无响应，故先保存再在 DOS 环境下打开。打开 debug，打开 smile.com 文件

键入 mount c d:\回车

键入 debug smile.com 回车

-g //go 执行

```
DOSBox 0.74, Cpu speed:   3000 cycles, Frameskip  0, Program:   DEBUG
-a100
073F:0100 mov dl,1
073F:0102 mov ah,2
073F:0104 int 21
073F:0106 int 20
073F:0108
-n smile.com
-rbx
BX 0000
:
-rcx
CX 0000
:8
-w
Writing 00008 bytes
-q

C:\>mount c d:\
Drive C already mounted with local directory d:\

C:\>debug smile.com
-g
▯
Program terminated normally
-
```

六、反汇编列出该程序

-u100 106　　//上述执行完后反汇编，会显示上述汇编的代码

```
-u100 106
075A:0100 B201          MOV     DL,01
075A:0102 B402          MOV     AH,02
075A:0104 CD21          INT     21
075A:0106 CD20          INT     20
```

七、查看寄存器的值

-r　　//read　会显示所有寄存器的值

```
-r
AX=FFFF  BX=0000  CX=0008  DX=0000  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=075A  CS=075A  IP=0100   NV UP EI PL NZ NA PO NC
075A:0100 B201          MOV     DL,01
```

八、退出 debug

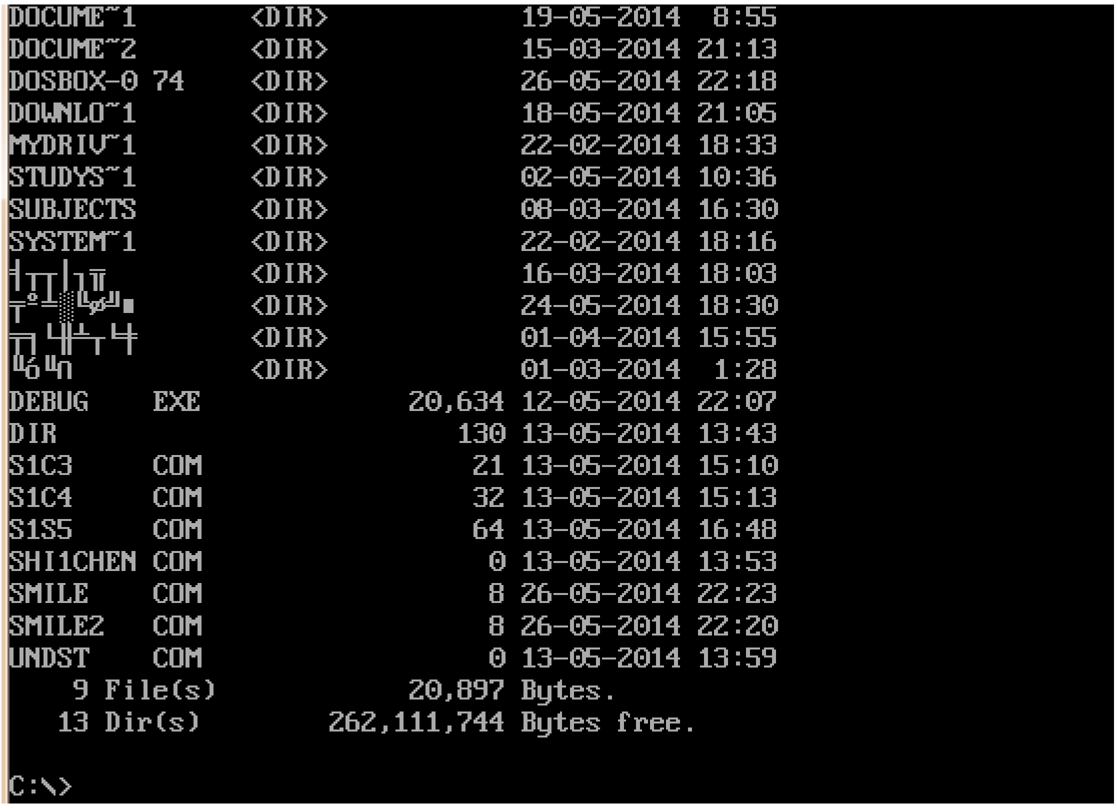
-q

D:\>

九、用 DOS 的 dir 命令查看当前目录下的文件

```
C:\>mount c d:\
Drive C already mounted with local directory d:\

C:\>dir
```



```
DOCUME~1        <DIR>             19-05-2014  8:55
DOCUME~2        <DIR>             15-03-2014 21:13
DOSBOX-0 74     <DIR>             26-05-2014 22:18
DOWNLO~1        <DIR>             18-05-2014 21:05
MYDRIV~1        <DIR>             22-02-2014 18:33
STUDYS~1        <DIR>             02-05-2014 10:36
SUBJECTS        <DIR>             08-03-2014 16:30
SYSTEM~1        <DIR>             22-02-2014 18:16
[illegible]     <DIR>             16-03-2014 18:03
[illegible]     <DIR>             24-05-2014 18:30
[illegible]     <DIR>             01-04-2014 15:55
[illegible]     <DIR>             01-03-2014  1:28
DEBUG    EXE             20,634 12-05-2014 22:07
DIR                         130 13-05-2014 13:43
S1C3     COM                 21 13-05-2014 15:10
S1C4     COM                 32 13-05-2014 15:13
S1S5     COM                 64 13-05-2014 16:48
SHI1CHEN COM                  0 13-05-2014 13:53
SMILE    COM                  8 26-05-2014 22:23
SMILE2   COM                  8 26-05-2014 22:20
UNDST    COM                  0 13-05-2014 13:59
    9 File(s)            20,897 Bytes.
   13 Dir(s)        262,111,744 Bytes free.

C:\>
```

第二个程序

——可以将所有 ASCII 码显示出来的程序

二、汇编

-a100

0AE9:0100 mov cx,0100　　//计数寄存器装入循环次数 0100，即十进制的 16^2

0AE9:0103 mov dl,00　　//将要显示的字符 00（ACSII 码）放入 dl 寄存器中

0AE9:0105 mov ah,02　　//调用 DOS 的 2 号功能,字符输出，并且要输出的字符是已经放在 dl 里面的

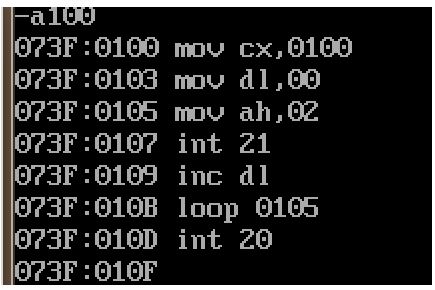
0AE9:0107 int 21　　//字符输出

0AE9:0109 inc dl　　//递增指令，dl 寄存器的值自值加一，即指向下一地址进行下一操作

0AE9:010B loop 0105　　//循环指令，每次计数器 cx 值减 1，跳到地址 0105 进行循环，，直至 cx 为 0 才结束循环，即依次输出 dl 的要显示的字符

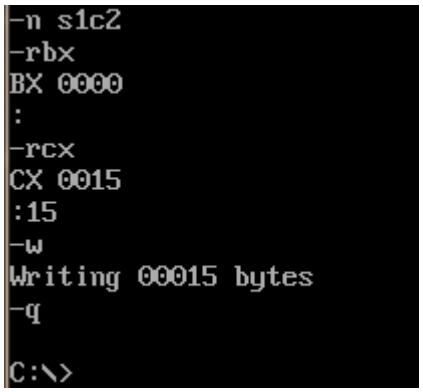
0AE9:010D int 20　　//执行中断

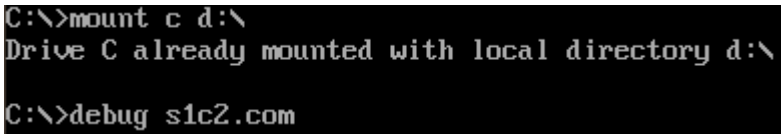
0AE9:010F



```
-a100
073F:0100 mov cx,0100
073F:0103 mov dl,00
073F:0105 mov ah,02
073F:0107 int 21
073F:0109 inc dl
073F:010B loop 0105
073F:010D int 20
073F:010F
```

三、保存执行





第三个程序

——显示字符串，如：UNDERSTAND?

一、打开 debug，汇编程序

D:\>debug

-a100

0AE9:0100 mov dx,109　　//寄存器 dx 存放的是字符串起始地址 109

0AE9:0103 mov ah,9　　//Dos 的 09h 功能调用，显示首地址存放在寄存器 dx 的字符串，字符串要求要以‘$’结尾

0AE9:0105 int 21　　//字符串输出

0AE9:0107 int 20　　//执行中断

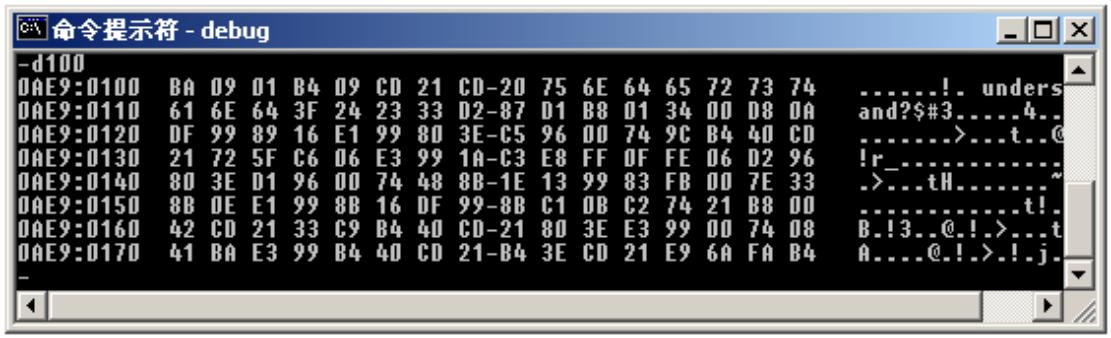
0AE9:0109 db 'understand?$'　　//定义字符串，字符串要求以‘$’结尾，将单引号内所有 ASCII 码放入内存中

0AE9:0115

// mov ah,9 的作用是显示字符串，但要求 dx 存放字符串的首地址，而字符串要以‘$’结尾，所以第一二句顺序不要求固定，另外，定义字符串也不强求一定在最后，只是放在最后最方便。

二、定义文件长度，命名文件，存盘，执行

三、查看 DB 伪指令将那些内容放入内存



第四个程序

——键盘输入任意字符串，然后显示出来

一、打开 debug，汇编程序

D:\>debug

-a100

0AE9:0100 mov dx,0116　　//将缓冲区地址 0116 的内容放入寄存器 dx，由 DB 伪指令确定缓冲区地址

0AE9:0103 mov ah,0a　　//调用 oa 功能，缓冲输入，缓冲区从存放在 dx 的地址 0116 开始，缓冲区的第一字节是该区可容纳的字节数，第二字节是实际输入长度，第三字节才开始是字符串

0AE9:0105 int 21　　//键盘输入缓冲区

0AE9:0107 mov dl,0a　　//由于功能 Ah 在每个字符串最后加一个归位码（0Dh 由 Enter MOV AH,02 ；产生），使光标自动回到输入行的最前端，为了使新输出的 INT 21 ；字符串不会盖掉原来输入的字符串，所以利用功能 2h 加一个换行码(OAh)，使得光标移到下一行的的最前端。

0AE9:0109 mov ah,02　　//调用 02 功能，字符输出

0AE9:010B int 21　　//输出字符

0AE9:010D mov dx,0118　　//装入字符串的起始位置，将缓冲输入的实际输入地址 0118 的内

容放入寄存器 dx

0AE9:0110 mov ah,09　　//调用 09 功能，显示字符串，输出首地址保存在 dx 中的字符串，9h 功能遇到$符号才会停止输出，故字符串最后必须加上$，否则 9h 功能会继续将内存中的无用数据胡乱显示出来

0AE9:0112 int 21　　//字符输出

0AE9:0114 int 20　　//程序终止

0AE9:0116 db 20　　//缓冲区的第一字节，定义缓冲区，赋值为该区可容纳的字节数为 20

0AE9:0117　　//缓冲区第二字节填充为实际输入的长度，可不做说明

　　//缓冲区的第三字节开始为输入的字符串

第五个程序

求 $1^2+2^2+\cdots\cdots+10^2$

```
Z:\>mount c d:\
Drive C is mounted as local directory d:\

Z:\>c:

C:\>debug
-
```

```
-a100
073F:0100 mov dx,0
073F:0103 mov al,1
073F:0105 mov bl,al
073F:0107 mul bl
073F:0109 add dx,ax
073F:010B cmp al,5
073F:010D jz 0114
073F:010F inc al
073F:0111 jmp 0105
073F:0113 int 20
073F:0115
```

DX 初值为 0,dx=0

Al 初值为 1，al=1

将 al 的值赋给 bl，bl=al

将 bl 平方放到 ax，ax=bl*bl

将 bl 的值加到 dx，dx=bl

将 al 的值与 5 比较，若小于 5 继续进行以下操作

若比较结果为否，无条件转移到地址 0114 中断程序

Al 值自值加一，al=al+1

回到地址 0105 继续循环

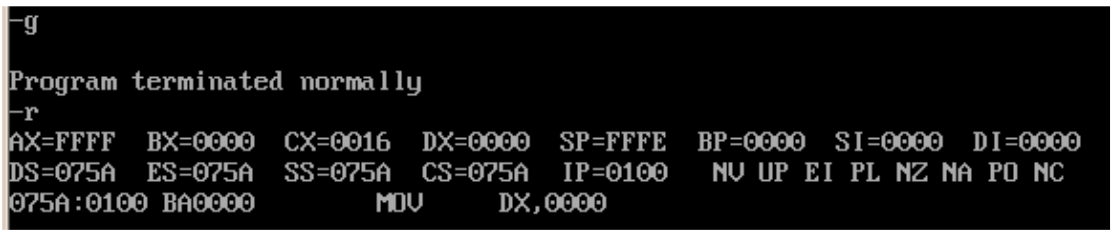
中断程序

显示不出结果

到底相乘后的结果放在哪个寄存器呢？

修改后：

```
-a100
075A:0100 mov dx,0
075A:0103 mov bx,1
075A:0106 mov ax,bx
075A:0108 mul ax
075A:010A add dx,ax
075A:010C cmp bx,a
075A:010F jz 0114
075A:0111 inc bx
075A:0112 jmp 0105
075A:0114 int 20
075A:0116
```



怎么回事？

```
Mov dx,0          //dx=0
Mov cl,0          //cx=0
Mov bl,[0118]     //bl=[0118],bl 的值等于地址 0118 的值
Int 21            //显示
Mov al,bl         //al=bl
Mul bl            //ax=bl*l
Add dx,ax         //dx=dx+ax
Int cl            //cl=cl+1
Cmp cl,4          //if cl>4 then next
Jz 011B           //转移到地址 011B
Db 10             //定义字符
Jmp 0105          //if cl<4  跳到 0105 循环
Int 20            //结束
```