



华南理工大学

SOUTH CHINA UNIVERSITY OF TECHNOLOGY

# 微机原理实验报告

## 实验二

-----

评阅分数： \_\_\_\_\_

## 一、实验目的

- 1、熟悉 80X86 基本指令集。
- 2、掌握汇编语言源程序的编写方法：程序结构、返回 DOS 的方法、段地址的初始化、存储单元的访问等。
- 3、掌握分支结构和循环结构的程序设计方法。
- 4、掌握 DEBUG 动态调试。

## 二、实验题目

题目 1、编程将以 '\$' 结束的字符串中的大写字母改为小写字母，显示字符串，并统计大写字母的个数。

①已知数据段的定义：

```
org 2000h
str1 db 'AeHdc123!*sLdeDI071ok',0AH,0DH,'$'
count equ $-str1
org 2030h
str2 db count dup(?) ;存放处理后的字符串
num db ? ;存放大写字母的统计个数
```

②运行显示：多次修改 str1 字符串的值，再次运行看结果，测试算法的正确性。

③debug 调试：查看内存单元，统计大写字母的个数。

源代码：

```
DATA SEGMENT
    org 2000h
    str1 db 'AeHdc123!*sLdeDI071ok',0AH,0DH,'$'
    count equ $-str1
    org 2030h
    str2 db count dup(?) ;存放处理后的字符串
    num db ? ;存放大写字母的统计个数
DATA ENDS

STACK SEGMENT PARA STACK 'STACK'
    DB 100 DUP (?)
STACK ENDS
```

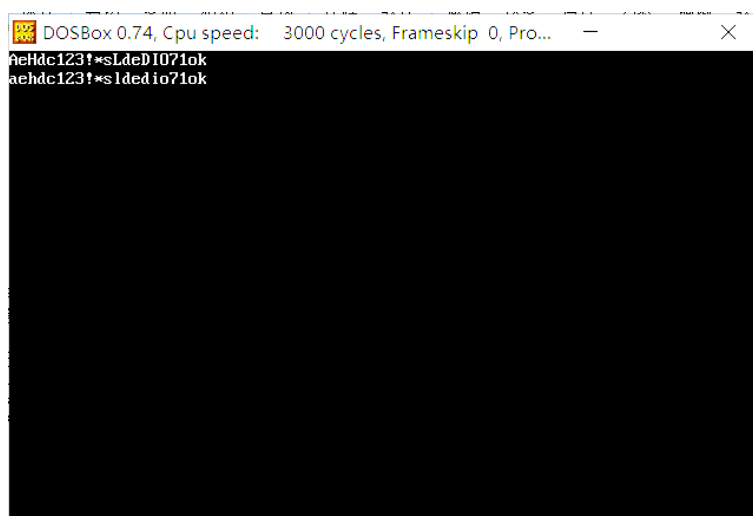
```

CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA,SS:STACK

START:  MOV AX,DATA
        MOV DS,AX
        MOV CX,count
        XOR DX,DX
        LEA SI,str1
        LEA DI,str2
NEXT:   MOV AL,[SI]
        CMP AL,'A'      ;小于A 的转跳
        JB WONTCG
        CMP AL,'Z'      ;大于Z 的转跳
        JA WONTCG
        ADD AL,20H      ;在A-Z 中的加上 20H
        INC DL           ;统计大写字母数量
WONTCG: MOV [DI],AL
        INC SI
        INC DI
        LOOP NEXT
        LEA DI,num
        MOV [DI],DL     ;存储大写字母数量
        LEA DX,str1     ;输出原字符串
        MOV AH,09H
        INT 21H
        LEA DX,str2     ;输出修改后的字符串
        MOV AH,09H
        INT 21H
CODE    ENDS
        END START

```

Debug 结果:



运行结果

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro...
0975:004E 0000 ADD [BX+SI],AL
0975:0050 0000 ADD [BX+SI],AL
0975:0052 0000 ADD [BX+SI],AL
0975:0054 0000 ADD [BX+SI],AL
0975:0056 0000 ADD [BX+SI],AL
0975:0058 0000 ADD [BX+SI],AL
0975:005A 0000 ADD [BX+SI],AL
0975:005C 0000 ADD [BX+SI],AL
0975:005E 0000 ADD [BX+SI],AL
-g=0000 003e
AeHdc123!*sLdeDI071ok
AX=0924 BX=2030 CX=0000 DX=2030 SP=0064 BP=0000 SI=2018 DI=2048
DS=0770 ES=0760 SS=0979 CS=0975 IP=003E NU UP EI PL NZ NA PE CY
0975:003E CD21 INT 21
-d 2000
0770:2000 41 65 48 64 63 31 32 33-21 2A 73 4C 64 65 44 49 AeHdc123!*sLdeDI
0770:2010 4F 37 31 6F 6B 0A 0D 24-00 00 00 00 00 00 00 00 071ok..$.
0770:2020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0770:2030 61 65 68 64 63 31 32 33-21 2A 73 6C 64 65 64 69 aeHdc123!*sLdedi
0770:2040 6F 37 31 6F 6B 0A 0D 24-06 00 00 00 00 00 00 00 o71ok..$.
0770:2050 B8 70 07 8E D8 B9 18 00-33 D2 8D 36 00 20 8D 3E .p.....3..6..>
0770:2060 30 20 8D 1E 30 20 8A 04-3C 41 72 08 3C 5A 77 04 0 ..0...<Ar.<Zw.
0770:2070 04 20 FE C2 88 05 46 47-E2 EC 8D 3E 48 20 88 15 . ...FG...>H ..
-S

```

### Debug 结果

如上图，能够正确输出修改后的字符串，在 Debug 界面中，洋红色的为原始的字符串 **str1**，青色的为修改后的字符串 **str2**，黄色部分为大写字母的数量统计，一共修改了六个大写字母。

题目 2、有一个带符号字节型数组，求最大值、最小值、总和和平均值。

①已知数据段的定义：

```

org 2000h
num db 12h,95h,0f1h,0c2h,82h,2h,10h,34h
count equ $-num
org 2010h
sum dw ?
res db 3 dup(?) ;最大值、最小值和平均值

```

②debug 调试：修改 num，运行看结果。

源代码：

```

DATA SEGMENT
org 2000h
num db 12h,95h,0f1h,0c2h,82h,2h,10h,34h
count equ $-num
org 2010h
sum dw ?
res db 3 dup(?) ;最大值、最小值和平均值
DATA ENDS

STACK SEGMENT PARA STACK 'STACK'
DB 100 DUP (?)
STACK ENDS

```

```

CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA,SS:STACK
START:  MOV AX,DATA
        MOV DS,AX
; 求最大最小值
        LEA SI,num
        MOV CX,count
        MOV BH,[SI]
        MOV BL,[SI]
        INC SI
        DEC CX
CIR:    CMP BH,[SI]
        JGE NEXT1
        MOV BH,[SI]
NEXT1:  CMP BL,[SI]
        JLE NEXT2
        MOV BL,[SI]
NEXT2:  INC SI
        LOOP CIR
; 循环完毕, 存储最大最小值
        LEA DI,res
        MOV [DI],BH
        INC DI
        MOV [DI],BL
; 求总数
        LEA SI,num
        XOR DI,DI
        XOR BX,BX
        MOV CX,count
SUMLP:  MOV AL,[SI]
        CBW
        CWD
        ADD BX,AX
        ADC DI,DX
        INC SI
        LOOP SUMLP      ; 求总数
        LEA SI,sum
        MOV [SI],BX      ; 存储总数 sum
        MOV AX,BX
        MOV DX,DI
        MOV CX,count
        IDIV CX          ; 带符号数除法
        LEA DI,res
        MOV [DI][2],AL   ; 存储平均数
        MOV AH,4CH
        INT 21H
CODE    ENDS
        END START

```

Debug 结果:

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro...
0972:004D F7F9 IDIU CX
0972:004F 8D3E1220 LEA DI,[2012]
0972:0053 8B4502 MOV [DI+02],AL
0972:0056 B44C MOV AH,4C
0972:0058 CD21 INT 21
0972:005A 0000 ADD [BX+SI],AL
0972:005C 0000 ADD [BX+SI],AL
0972:005E 0000 ADD [BX+SI],AL
0972:0060 0000 ADD [BX+SI],AL
0972:0062 0000 ADD [BX+SI],AL
-g=0000 0058
AX=4CE5 BX=FF22 CX=0000 DX=FFFA SP=0064 BP=0000 SI=2010 DI=2012
DS=0770 ES=0760 SS=0978 CS=0972 IP=0058 NU UP EI PL NZ NA PO NC
0972:0058 CD21 INT 21
-a 2000
0770:2000 12 95 F1 C2 B2 02 10 34-00 00 00 00 00 00 00 00 .....4.....
0770:2010 22 FF 34 B2 E5 00 00 00-00 00 00 00 00 00 00 00 ".4.....
0770:2020 B8 70 07 8E D8 8D 36 00-20 B9 00 00 8A 3C 8A 1C .p...6....<..
0770:2030 46 49 3A 3C 7D 02 8A 3C-3A 1C 7E 02 8A 1C 46 E2 FI:<>..<:~...F.
0770:2040 F1 8D 3E 12 20 88 3D 47-B8 1D 8D 36 00 20 33 FF ..>..=6...6..3.
0770:2050 33 DB B9 00 00 8A 04 98-99 03 D8 13 FA 46 E2 F5 3.....F..
0770:2060 8D 36 10 20 B9 1C 8B C3-BB D7 B9 00 00 F7 F9 8D .6.....
0770:2070 3E 12 20 88 45 02 B4 4C-CD 21 00 00 00 00 00 00 >..E..L.!.....
-S_

```

上图为 debug 结果，可以看出，2000H-2008H 为原始数据，青色部分 FF22 为求和结果，也就是 sum。2013H-2015H 的为最大最小值、平均值。

题目 3、将下面数据段的数据，在屏幕上显示。

①已知数据段的定义:

```

time    db  2,0,2,0,0,4,2,1
str1    db  'Happy New Day!$'

```

②运行看输出显示结果。

源代码:

```

DATA    SEGMENT
    time    db  2,0,2,0,0,4,2,1
    str1     db  'Happy New Day!$'
DATA    ENDS

STACK   SEGMENT PARA STACK 'STACK'
    DB 100 DUP (?)
STACK   ENDS

CODE    SEGMENT
    ASSUME CS:CODE,DS:DATA,SS:STACK
START:  MOV AX,DATA
        MOV DS,AX

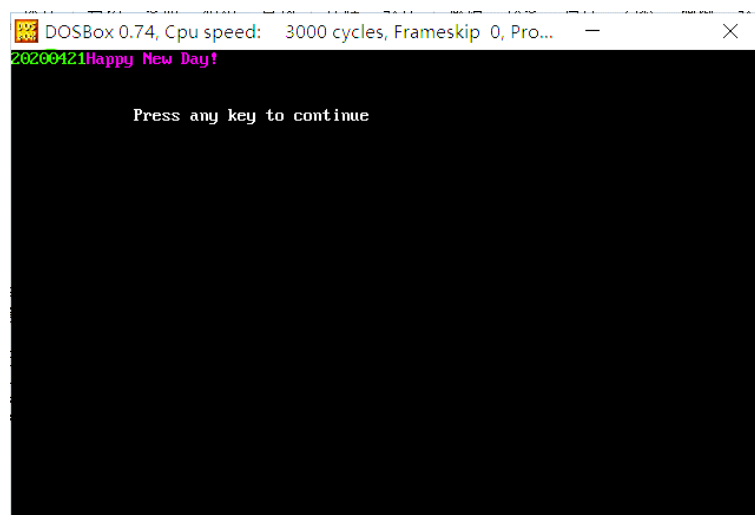
```

```

        MOV CX,8
        LEA SI,time
NEXT1:  MOV DL,[SI]
        ADD DL,30H
        MOV AH,02H
        INT 21H
        INC SI
        LOOP NEXT1
        LEA DX,str1
        MOV AH,09H
        INT 21H
        MOV AH,4CH
        INT 21H
CODE    ENDS
        END START

```

Debug 结果:



能够正确输出内容

### 三、实验总结

通过这次试验，让我对于加深了对 8060 基本指令集的印象，练习了分支结构和循环结构的程序设计方法，练习了中断的调用。但是即使做完实验，感觉对带符号数的处理还是不够熟练，第二题一开始做的也不是很顺利，后来去和同学讨论之后才做了出来。