

学号： WA2224013 专业： 机器人工程 姓名： 郭义月

实验日期： 2024. 12. 6 指导教师： 鲍华 实验成绩：

课程目标 1 (权重__)	课程目标 2 (权重__)	课程目标 3 (权重__)	课程目标 4 (权重__)	课程目标 5 (权重__)	课程目标 6 (权重__)	综合成绩 (目标数可增删)

安徽大学人工智能学院本科实验报告

【课程名称】 微型计算机原理及接口技术

【课程目标】

提升对内存操作、条件判断和循环控制的理解与应用能力。

【实验名称】 实验二 循环结构程序设计

【实验目的】

- 1. 熟练掌握汇编语言源程序的格式，熟悉循环结构程序设计的基本方法。
- 2. 掌握用编辑软件建立、修改源程序的方法，进一步熟悉对汇编语言程序进行汇编、链接形成可执行文件的过程
- 3. 了解字符与 ASCII 码的对照关系

【实验原理及方法】

本实验旨在通过汇编语言编程，掌握字符处理和字符串比较的基本操作。实验一通过遍历字符串，比较ASCII码值，实现找出字符串中最大ASCII码字符的功能；实验二设计人机交互接口，实现密码输入、存储和校验流程，加深对数据存储、输入输出和条件判断的理解，提高编程实践能力。通过这两个实验，学生能够理解并应用汇编语言进行基本的数据处理和用户交互，为后续复杂程序设计打下坚实基础。

【实验内容及过程】

1. 假设在以 BUF 为首址的存储单元中存放一串字符,找出其中 ASCII 码值最大的字符并存入 MAX 单元中(BUF 中字符为“ABCDbdca”)

```
DATAS SEGMENT
;此处输入数据段代码
BUF      DB  'ABCDbdca'  ; 字符串 BUF, 以 null 字符结尾
COUNT EQU 8
MAX DB 0

DATAS ENDS

STACKS SEGMENT
;此处输入堆栈段代码
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX
;此处输入代码段代码
    MOV SI,  OFFSET BUF    ;
    MOV CX,COUNT
LP: MOV AL,[SI]
    CMP AL,MAX
    JB  NEXT
    MOV MAX,AL
NEXT:
    INC SI
    LOOP LP
    MOV DL,MAX
    MOV AH,2
    INT 21H
    MOV AH,4CH
    INT 21H
CODES ENDS
    END START
```

2. 设数据区定义了两个字符串 BUF1和 BUF2, 其中 BUF1区存放一个已知的登录密码(自己设置), BUF2区域用来保存用户输入的密码。要求

- (1) 设计人机接口, 提示输入登录密码。
- (2) 自行输入密码后存入 BUF2, 与 BUF1 区域的密码进行比较
- (3) 若密码正确, 输出相应提示信息;若密码不正确, 输出相应提示信息, 并要求重新输入新密码。

```

DATAS SEGMENT
;此处输入数据段代码
str1 db 'enter:$'
str2 db 'correct!'
str3 db 'wrong!enteragain:'
BUF1 db '12344'
BUF2 db 20, 0
DATAS ENDS

STACKS SEGMENT
;此处输入堆栈段代码
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX
;此处输入代码段代码
    MOV AH, 09H
    LEA dx, [str1]
    int 21h
inputloop:
    mov ah, 0Ah
    lea dx, [BUF2]
    int 21h
    lea si, [BUF2+2]
    lea di, [BUF1]
    mov cx, 5
compareloop:
    mov al, [si]
    mov bl, [di]
    cmp al, bl
    jne incorrect
    inc si
    inc di
    loop compareloop
    mov ah, 09h
    lea dx, [str2]
    int 21h
    jmp DONE
incorrect:
    mov ah, 09h
    lea dx, [str3]
    int 21h
    jmp inputloop
DONE:
; 程序结束，返回操作系统
    mov ah, 4Ch

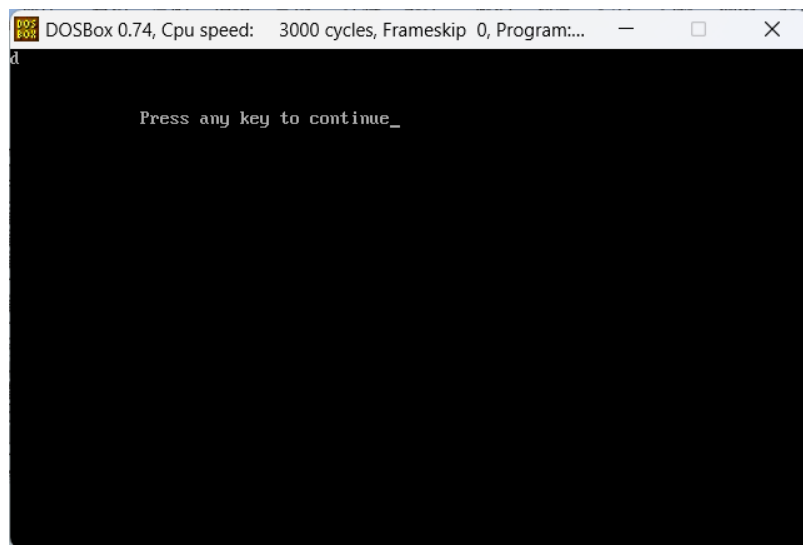
```

```
int 21h  
CODES ENDS  
END START
```

【实验结果】

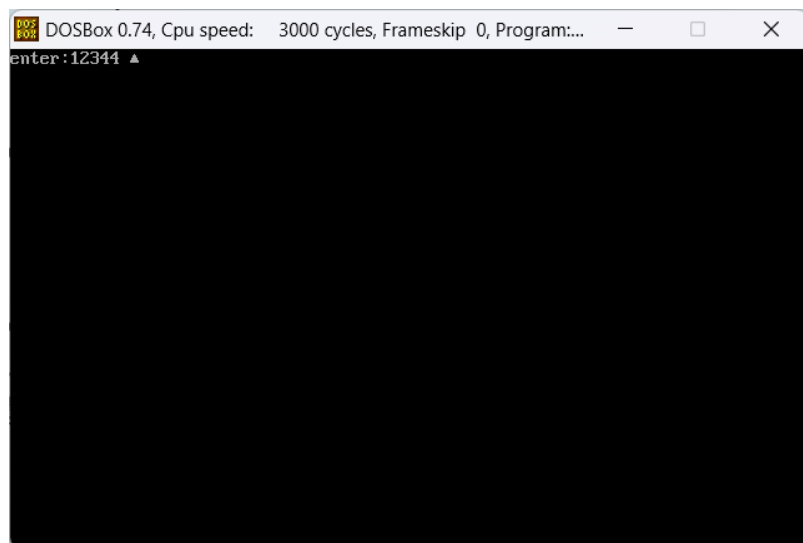
找出最大的字符存入 MAX 单元中，字符为“ABCDbdca”时，最大值位d，通过DOS指令输出再屏幕上

```
MOV DL,MAX  
MOV AH,2  
INT 21H
```

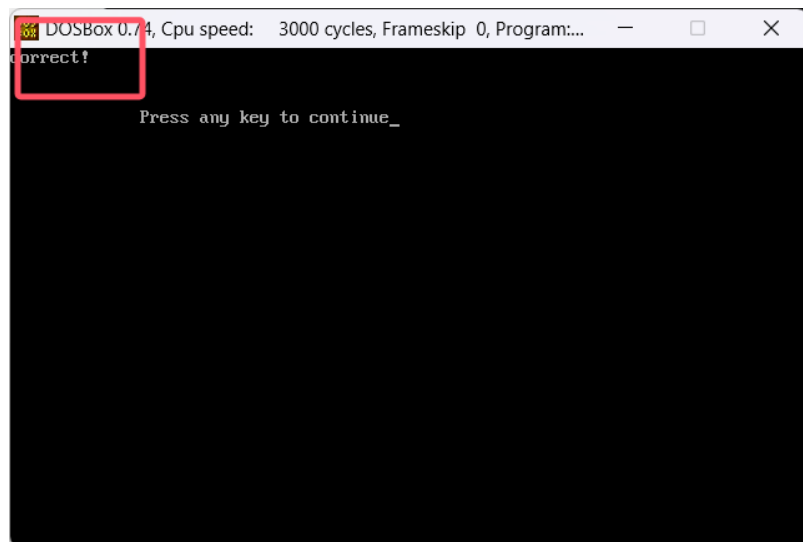


简单的密码验证功能

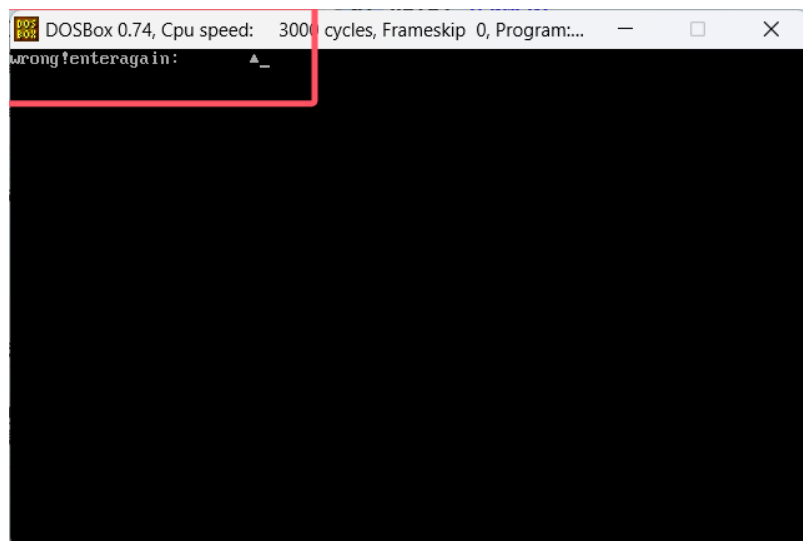
显示输入：



密码正确，程序结束



密码错误，显示重新输入：



【数据分析及处理】

任务2匹配密码是否正确时，出来用循环一个一个判断，还可以通过对cmp加上前缀repz之后实现

REPZ CMPSB

Repz是重复前缀，表示没进行一次比较cx减1，当串没有比较完且原串与目标串对应数据比较结果相等（ZF=1），则继续进行下一个数据的比较，直到两串比较完（CX=0）或比较的结果两串数不相等（ZF=0）

代码如下：

```
DATAS SEGMENT
;此处输入数据段代码
STR1 DB 'ENTER:$'
STR2 DB 'CORRECT!$'
```

```

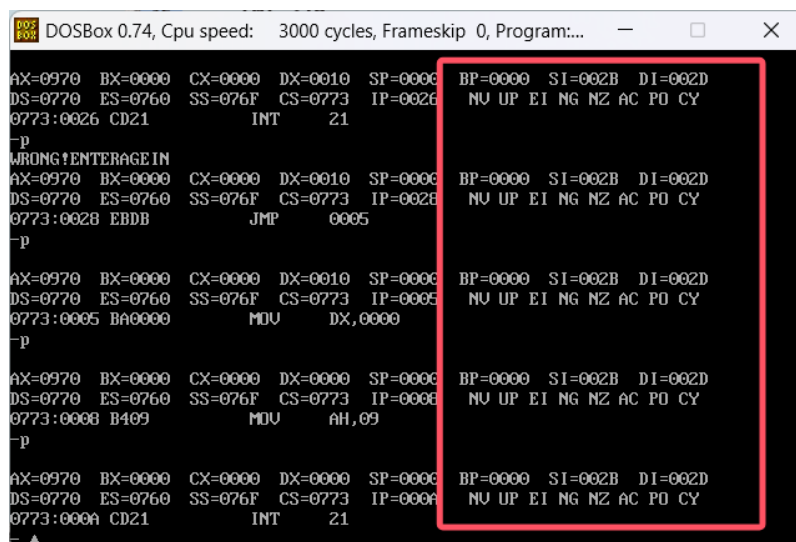
    STR3 DB 'WRONG!ENTERAGEIN$'
    BUF DB 6,0
    BUF1 DB '12344'
DATAS ENDS

STACKS SEGMENT
    ;此处输入堆栈段代码
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX
    ;此处输入代码段代码
INPUTLOOP:
    MOV DX,OFFSET STR1
    MOV AH,9
    INT 21H
    MOV DX,OFFSET BUF
    MOV AH,0AH
    INT 21H
    MOV SI,OFFSET BUF
    MOV DI,OFFSET BUF1
    CLD
    MOV CX,5
    REPNZ CMPSW
    JZ CORRECT
    MOV DX,OFFSET STR3
    MOV AH,9
    INT 21H
    JMP INPUTLOOP
    MOV AH,4CH
    INT 21H
CORRECT:
    MOV DX,OFFSET STR2
    MOV AH,9
    INT 21H
    MOV AH,4CH
    INT 21H
CODES ENDS
    END START

```

处理的过程中发现，无论密码是否正确，都无法跳转到correct函数，调试结果为：



无论怎么输入密码，这里列的标志位中，ZF始终不为1，目前找不到解决方法

【总结或讨论】

由于调试的过程需要观察标志位的值，这是找到的标志位的符号和其对应的含义，可以通过下表对应调试。

-r

然后看右下角： NV UP EI PL NZ NA PO NC

这个是符号值对应表：

溢出标志 OF(Over flow flag) OV(1) NV(0)

方向标志 DF(Direction flag) DN(1) UP(0)

中断标志 IF(Interrupt flag) EI(1) DI(0)

符号标志 SF(Sign flag) NG(1) PL(0)

零标志 ZF(Zero flag) ZR(1) NZ(0)

辅助标志 AF(Auxiliary carry flag) AC(1) NA(0)

奇偶标志 PF(Parity flag) PE(1) PO(0)

进位标志 CF(Carry flag) CY(1) NC(0)

实验过程中遇到的问题有：

Dos端无法输出中文，每次输出都是乱码，所以提示文字只能通过英文输入

DATAS SEGMENT

;此处输入数据段代码

STR1 DB 'ENTER:\$'

STR2 DB 'CORRECT!\$'

STR3 DB 'WRONG!ENTERAGEIN\$'

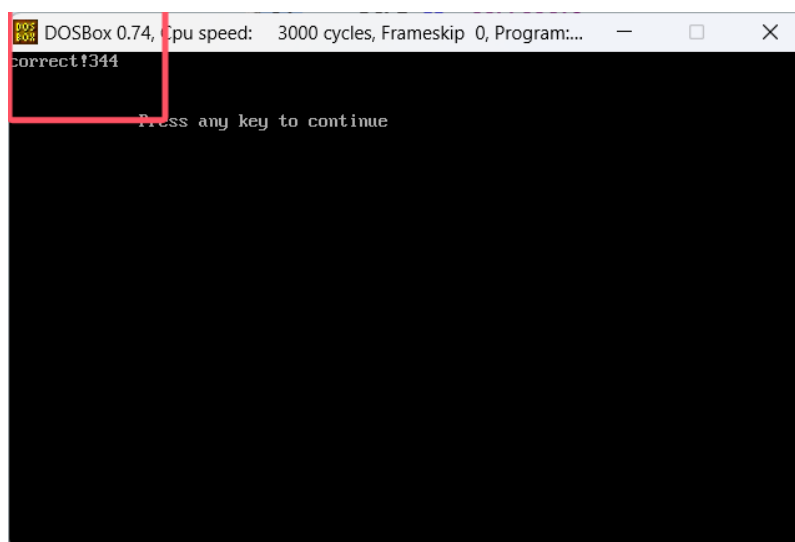
BUF DB 6,0

BUF1 DB '12344'

DATAS ENDS

Dos输出时的不会自动清屏，而会覆盖掉之前的内容，当之前的内容大于新的内容时，之前没

有被覆盖的部分依然会显示到屏幕上，如下图：



Correct之后没有被覆盖的密码依然会显示出来，解决此问题有一个很简单的方法就是在输入的字符串后面补足足够的空格，保证可以覆盖掉之前输入的密码，如以下代码所示，或者通过dos的清屏操作，把清屏指令加入到每次输入密码的循环前面

```
DATAS SEGMENT
;此处输入数据段代码
str1 db 'enter:$'
str2 db 'correct!'
str3 db 'wrong!enteragain: $'
BUF1 db '12344'
BUF2 db 20, 0
DATAS ENDS
```