

山东大学 计算机科学与技术 学院

汇编语言 课程实验报告

学号：202200130048	姓名：陈静雯	班级：6 班
实验题目：实验 2.3 统计字符串		
实验学时：2	实验日期：20241209	
<p>实验目的：巩固分支程序设计中所涉及的知识点。</p> <p>学会在自编程序中利用分支程序设计的理论与技巧。</p> <p>学会在自编程序中使用系统调用。</p>		
实验环境：Windows10、mingw-64		
源程序清单：		
1. test9.asm		
编译及运行结果：		
 <pre>Object filename [test9.OBJ]: Source listing [NUL.LST]: Cross-reference [NUL.CRF]: 51640 + 448520 Bytes symbol space free 0 Warning Errors 0 Severe Errors C:\>link test9 Microsoft (R) Overlay Linker Version 3.60 Copyright (C) Microsoft Corp 1983-1987. All rights reserved. Run File [TEST9.EXE]: List File [NUL.MAP]: Libraries [.LIB]: LINK : warning L4021: no stack segment C:\>test9 aisdhASD245J 5 4 3 C:\>_</pre>		

问题及收获:

1. 实验步骤:

(1) 数据段

```
dseg segment
input    label byte
maxlen   db 101;最大长度
actuellen db 0;实际长度
counter  db 101 dup(?);字符串
b_num    db 4 dup(0);大写字母个数
s_num    db 4 dup(0);小写字母个数
d_num    db 4 dup(0);数字个数
dseg ends
```

(2) 用 0ah 输入字符串

```
mov     ax,dseg
mov     ds,ax    ;初始化
lea     dx,input
mov     ah,0ah
int     21h      ;0ah输入字符串
mov     al,actuellen;实际长度, [0,ax-1]
xor     ah,ah
mov     si,0     ;当前字符下标
dec     ax       ;下标最大值
```

(3) 用分支判断每一位是大写小写还是数字还是什么都不是

```

start:
    cmp     counter[si], '0'
    jge     digit    ; >= 0
if_loop:
    inc     si
    cmp     si, ax
    jg      output    ; si > ax, 读到末尾了, 输出各个计数
    jmp     start     ; 否则, 接着判断当前字符是什么
digit:
    cmp     counter[si], '9'
    jle     isdigit   ; 0-9
    cmp     counter[si], 'A'
    jge     big       ; >= A
    jmp     if_loop   ; 啥都不是
isdigit:
    inc     d_num
    jmp     if_loop
big:
    cmp     counter[si], 'Z'
    jle     isbig     ; A-Z
    cmp     counter[si], 'a'
    jge     smallone  ; >= a
    jmp     if_loop   ; 啥都不是
isbig:
    inc     b_num
    jmp     if_loop
smallone:
    cmp     counter[si], 'z'
    jle     issmall   ; a-z
    jmp     if_loop   ; 啥都不是

```

(4) 输出, 需要将数字转成 asc 码

```

output:
    call    crlf
    mov     al, s_num;
    xor     ah, ah
    call    s_to_asc
    lea     dx, s_num
    mov     ah, 09h
    int     21h
    call    crlf
    mov     al, b_num; 先输出大写字母个数, 因为只有一个字节的数, 所以放到 al, ah 用 0 填充
    xor     ah, ah
    call    b_to_asc; 转成 asc 码, 输出字符串
    lea     dx, b_num
    mov     ah, 09h
    int     21h

```

(5) 如何转成 asc 码, 因为最多是 100, 三位数, 所以只需要除三次,

同时要判断 0 是首位 0 还是中间的 0 来看需不需要输出

```
b_to_asc proc near
    mov     si,0      ;b_num[0],[1],[2],最多三位数,si记录当前下标
    mov     di,0      ;di标记是否是首0
    cmp     al,0
    jz      thezero   ;al==0,说明该类字符一个都没有,0加个结束符就能输出
    mov     cx,100d    ;最多是三位数也就是100
    call    thediv_b
    mov     cx,10d
    call    thediv_b
    mov     cx,1d
    call    thediv_b
thend:
    mov     b_num[si], '$'
    ret
thezero:
    mov     b_num[0], '0'
    mov     b_num[1], '$'
    ret
b_to_asc endp
;-----
thediv_b proc near
    mov     dx,0
    div     cx          ;dx余数, ax商
    add     ax, '0'
```

2. 收获

- (1) 函数没有返回会无法输出
- (2) 小端存储
- (3) 一个四个字节的比如 b_num 要赋给 a18 位寄存器, 会直接取低位有效段
- (4) 要有结束符, 否则会输出乱码
- (5) Lea dx, input (标签处) 和 mov dx, offset maxlength 没有区别
- (6) 学会用 debug 检查问题