

# 上海大学



SHANGHAI UNIVERSITY

2021-2022 学年秋季学期

上海大学 计算机学院

《汇编语言程序设计》

实验 2

实验名称： 分类统计字符个数

专业： 19 级直招计科 2 班

姓名： 汪雨卿

学号： 19120191

实验名称： 分类统计字符个数

## 一、实验目的

用分支程序设计与循环程序设计的方式实现字母、数字及其它字符的分类统计。

程序接受用户键入的一行字符（字符个数不超过 80 个，该字符串用回车符结束），并按字母、数字及其它字符分类记数，然后将结果存入以 letter、digit 和 other 为名的存储单元中。

## 二、实验内容

### (1) 实验原理

#### ①从键盘读入一行字符

实验要求程序接受用户键入的一行字符，DOS 命令下可以用 `mov ah, 01H` 或者 `mov ah, 0AH`。前者是从键盘读入一个字符到 `al`，一次只能读入一个；后者是将键盘输入至缓冲区，即读入一行字符，直到回车结束，且该命令默认缓冲区首是 `DS:DX`，缓冲区最大容量是 `DS:[DX]`，输入的字符数存放在 `DS:[DX+1]`，字符串首是 `DS:DX+2`。所以从键盘键入的一行字符可以有如上两种方法进行读入与保存。

具体如下：

`mov ah, 01H`——每次读入一个字符，先与 `odh`（回车的 ASCII 码）进行比较，不是回车则进行后面的判断分类。直到读到回车，表示输入的一行字符结束，跳到输出程序进行输出。

`mov ah, 0AH`——将整行字符直接输入至缓冲区，将输入的字符数 `DS:[DX+1]` 保存在（赋值给）`c1` 寄存器，将字符串首偏移地址赋给 `si`。每次判断分类完毕后，让 `c1` 中的数减一，让 `si` 递增取下一个字符。如此循环判断直到 `c1` 为 0 跳出判断，进入输出程序输出。

本实验中，我采用了第一种方法，具体代码展示在实验记录部分。

#### ②字符的分类判断

将读入的字符分成 letter 字母，digit 数字，other 其他字符。

查询 ASCII 码表易得：letter 字母——`41H~5AH`（大写字母），`61H~7AH`（小

写字母); digit 数字——30H~39H; other——其他字符。

由此设计 cmp 命令和统计程序，具体程序在实验记录部分展示。

### ③统计数字的进制转换与输出

统计的不同种类字符的个数分别保存在 letter, digit 和 other 三个存储单元。因为计算机内部存储的是十六进制数，但是要让计算机输出一个内容是十进制的数时，需要将其每一位的数字转换成十六进制再分别输出。所以当统计的字符个数超过 9 时，计算机内部无法直接+30H 输出，需要将十六进制对应的十进制每一位都分别算出并+30H 输出。

查询资料可知，将十六进制数/0AH，得到的商是十进制的十位，余数是十进制的个位。（因为限制每行字符不超过 80D=50H，所以不考虑三位数。）由此，设计一个子程序将十六进制数转换成能够被计算机计算并输出十进制相应的字符（输出是是一个个字符，但是可以视作十进制），在输出程序中调用该子程序并返回即可。具体程序代码在实验记录部分展示。

## (2) 实验步骤

- ① 启动 MASM 6.0 或 MASM for Windows 集成编程环境。
- ② 分支指令形式编写 ASM 源程序。
- ③ 对其进行汇编及连接，产生 EXE 文件。
- ④ 作必要的调试。

## (3) 实验记录

根据实验原理，分模块进行代码设计如下：

### ①从键盘读入一行字符

本实验采用整行输入。先输出一串输入提示符，然后调用 dos 窗口，读入一整行字符。

```
038 ; 输入，输入一行字符提示
039 lea dx,input
040 mov ah,9
041 int 21h
042 ;输入一行字符
043 lea dx,line
044 mov ah,10
045 int 21h
```

### ②字符的分类判断

数字：30H~39H

字母：41H~5AH, 61H~7AH

其他字符：除却上面的三个范围，其余全是字符

根据上方三个范围，使用 cmp 命令和 jb（小于转移），ja（大于转移），jmp（无条件转移）将字符计数到相应的存储单元（add 加 1 计数命令）。

```
051
052 read:
053     mov al, line[si]
054     cmp al, '0'
055     jb other
056     cmp al, '9'
057     ja letterOrdigit
058     add digit_count,1
059     jmp next
060
061 letterOrdigit:
062     and al,11011111b    ;将字符串转换为大写字符
063     cmp al, 'A'
064     jb other
065     cmp al, 'Z'
066     ja other
067     add letter_count,1
068     jmp next
069
070 other:
071     add other_count,1 ;other统计器加1
072     ;输出结果
073
074 next:
075     add si,1
076     sub ah,1
077     dec cl
078     cmp cl,0
079     jz break
080     jnz read    ;没读完，继续读
081
```

### ③统计数字的进制转换与输出

统计的不同种类字符的个数分别保存在 letter\_count，digit\_count 和 other\_count 三个存储单元。输出时先输出事先设置好了的三条语句来提示之后所要输出的数字是哪个存储单元（数字/字母/其他字符）的个数，再用 call 命令调用“进制转换”disp 程序将保存在计算机内部的十六进制数所对应的十进制数输出到屏幕上。

disp 程序实现将十六进制数输出成屏幕上的“十进制数”。计算机内部存储的是十六进制数，当统计的字符个数超过 9 时，计算机内部无法直接+30H 输出，

需要将十六进制对应的十进制每一位（十位，个位）都分别算出并+30H 输出。

查询资料可知，将十六进制数/0AH，得到的商是十进制的十位，余数是十进制的个位。（因为限制每行字符不超过 80D=50H，所以不考虑三位数。）

由此，先将 letter/digit/other 的值 mov 到 al 寄存器，再将 b1 赋值为 0ah，两者相除，商保存在 al，余数保存在 ah。先输出商——直接加 30H 并输出，再输出余数——先将 ah 里面的值保存在 bh，在商成功输出后将其 mov 到 al 进行和商一样的操作。如此就可以将十六进制对应的十进制输出到屏幕上。最后加上回车换行操作再返回至主程序进行其他相同输出。

```
101     lea dx, crlf
102     mov ah, 9
103     int 21h
104     lea dx, letter_n
105     mov ah, 9
106     int 21h
107     lea dx, letter_count
108     mov al, letter_count
109     call disp
110     lea dx, crlf
111     mov ah, 9
112     int 21h
113     ;
114     lea dx, digit_n
115     mov ah, 9
116     int 21h
117     lea dx, digit_count
118     mov al, digit_count
119     call disp
120     lea dx, crlf                ;另取一行
121     mov ah, 09h
122     int 21h
123     ;
124     lea dx, other_n
125     mov ah, 9
126     int 21h
127     lea dx, other_count
128     mov al, other_count
129     call disp
130     lea dx, crlf
131     mov ah, 9
132     int 21h
133
```

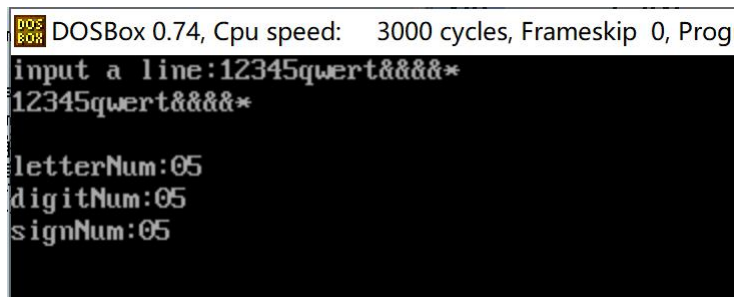
```
disp:;十六进制转成十进制，输出
mov b1,0ah;十六进制除以 0AH，得到十进制的单独位数
mov ah,0;保证 ax 高位是 0
div b1;ax/b1，商在 al--十进制高位，余数在 ah--十进制低位
add al,30h
mov dl,al
mov bh,ah;余数--十进制低位存在 bh 寄存器，防止下面的输出冲刷掉
mov ah,2
int 21h
mov al,bh;余数--十进制低位放进 al 操作
add al,30h
mov dl,al
mov ah,2
int 21h
mov dl,0ah;回车
int 21h
mov dl,0dh;换行
int 21h
ret;call 程序返回
```

#### (4) 数据处理

程序设计完毕，进行调试。

①（键盘输入）12345qwerty&&&&\*

（屏幕输出）5    5    5

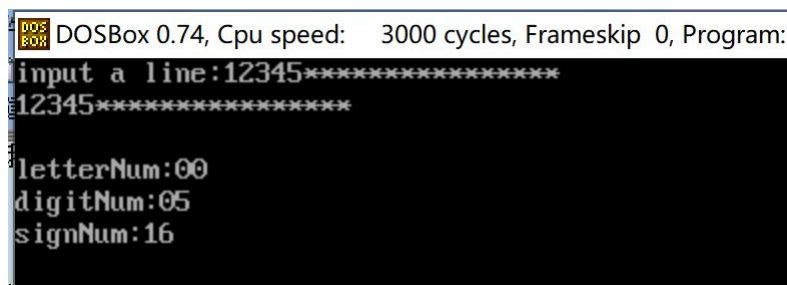


```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Prog
input a line:12345qwerty&&&&*
12345qwerty&&&&*

letterNum:05
digitNum:05
signNum:05
```

②（键盘输入）12345\*\*\*\*\*

（屏幕输出）0    05    16



```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:
input a line:12345*****
12345*****

letterNum:00
digitNum:05
signNum:16
```

### 三、实验体会

本次实验让我熟悉了如何利用汇编读入字符串。通过查找 dos 的指导命令，发现读入字符串的方法有两个方式，一种是逐个读入，另一种方式是成行读入。同时，由于读入的字符串要存放在内存中。因此，也要学习如何将指定内存中的内容输出到 dosbox 执行框中。最开始发现，直接输出字符串会出现长传乱码，研究发现从字符串中直接输出的内容需要进行转换，进而实现了最简单的输出字符串的功能。然后，在输出统计数据的过程中，又发现需要将单个输出的数据转换成 10 进制数据再输出。通过学习参考书上的案例，并且结合自己的调试，最终实现了十进制两位数结果的输出。

这个实验，一步步让我学习到了汇编语言调试和学习的乐趣。遇到困难，解决困难，最后战胜困难的过程，也让我收获颇丰。

最后，附上完整代码：

```

001 DATA SEGMENT
002 ;样例: mess db 'abcde1234%^&*','10,13','12$' ;10,13表示输出换行,字符串以$为结尾
003 ;mess db 'abcde1234%^&*$' ;10,13表示输出换行,字符串以$为结尾
004 ;一些输出提示
005 input db 'input a line:$'
006 crlf db 10,13,'$'
007 letter_n db 'letterNum:$'
008 digit_n db 'digitNum:$'
009 other_n db 'signNum:$'
010
011 ;输入的字符串
012
013 line db 80 ;预定义80字节的空间
014 db ? ;待输入完成后,自动获得输入的字符个数
015 db 80 dup(?)
016
017 len equ $ - line ;获取字符串的长度
018
019 ;设置一些计数器
020 letter_count db ?|
021 digit_count db ?
022 other_count db ?
023
024 DATA ENDS
025 CODES SEGMENT
026 ASSUME CS:CODES, DS:DATA ;代码段和数据段的名字不一定是DATA
027 START:
028 ;数据初始化,把0h放入堆栈
029 push ds
030 mov ax,0h
031 push ax
032 mov ax,DATA ;初始化ds
033 mov ds,ax
034 mov letter_count,0
035 mov digit_count,0
036 mov other_count,0

```

```

037
038 ; 输入, 输入一行字符提示
039 lea dx,input
040 mov ah,9
041 int 21h
042 ;输入一行字符
043 lea dx,line
044 mov ah,10
045 int 21h
046 ;数据统计
047 ;1. 读入数据
048 mov si,2 ;存放数组的下标要用si
049 mov cl, [line+1]
050 mov ah, len
051
052 read:
053 mov al, line[si]
054 cmp al, '0'
055 jb other
056 cmp al, '9'
057 ja letter0rdigit
058 add digit_count,1
059 jmp next
060
061 letter0rdigit:
062 and al,11011111b ;将字符串转换为大写字符
063 cmp al, 'A'
064 jb other
065 cmp al, 'Z'
066 ja other
067 add letter_count,1
068 jmp next
069
070 other:
071 add other_count,1 ;other统计器加1
072 ;输出结果

```

```

074 next:
075     add si,1
076     sub ah,1
077     dec cl
078     cmp cl,0
079     jz break
080     jnz read    ;没读完,继续读
081
082 break:
083     ;输出输入的字符串
084     LEA DX, crlf                ;另取一行
085     MOV AH, 09H
086     INT 21H
087     MOV AL, line+1              ;对字符串进行处理
088     ADD AL, 2
089     MOV AH, 0
090     MOV SI, AX
091     MOV line[SI], '$'
092     LEA DX, crlf                ;另取一行
093     MOV AH, 09H
094     LEA DX, line+2              ;输出输入的字符串
095     MOV AH, 09H
096     INT 21H
097     LEA DX, crlf                ;另取一行
098     MOV AH, 09H
099     INT 21H
100 ;
101     lea dx, crlf
102     mov ah, 9
103     int 21h
104     lea dx, letter_n
105     mov ah, 9
106     int 21h
107     lea dx, letter_count
108     mov al, letter_count
109     call disp
110     lea dx, crlf
111     mov ah, 9
112     int 21h

```

```

112     int 21h
113 ;
114     lea dx, digit_n
115     mov ah, 9
116     int 21h
117     lea dx, digit_count
118     mov al, digit_count
119     call disp
120     LEA DX, crlf                ;另取一行
121     MOV AH, 09H
122     INT 21H
123 ;
124     lea dx, other_n
125     mov ah, 9
126     int 21h
127     lea dx, other_count
128     mov al, other_count
129     call disp
130     lea dx, crlf
131     mov ah, 9
132     int 21h
133
134 exit:
135     MOV AH,4CH
136     INT 21H
137
138 disp:    ;十进制数形式显示AL中的内容.
139     mov ah, 0
140     mov bl, 10
141     div bl    ;div 无符号:div src 16位操作:商ax=(dx,ax)/src,余数dx
142     add al, 30h    ;比如说al=15h,即21,表示letter数量,然后,这个过程就是, ax=0015h(
143     mov dl, al    ;则除完了的结果为2余1, 则ah=01,al=02,即ax=0102h;,那么 al+30h即为表
144     mov bh, ah    ;则ax=0132h,d1=32,bh=01;
145     mov ah, 02h    ;显示输出dx,则显示32码对应的数字, 2
146     int 21h
147     mov al, bh    ;把01给al,然后算出ASCII码,然后给dx, 然后显示
148     add al, 30h
149     mov dl, al    ;除数有 8位和16位两种, 保存在一个reg寄存器里 或是内存单元中。
150
151     ;除数(除数)左放在ax中 (16位以内) 或 dx和ax中 (32位) ax存放

```



```

138 disp:      ;十进制数形式显示AL中的内容.
139      mov    ah, 0
140      mov    bl, 10
141      div    bl      ;div 无符号:div src 16位操作:商ax=(dx,ax)/src,余数dx
142      add    al, 30h  ;比如说al=15h,即21,表示letter数量,然后,这个过程就是,ax=0015h(
143      mov    dl, al   ;则除完了的结果为2余1,则ah=01,a1=02,即ax=0102h;,那么al+30h即为表
144      mov    bh, ah   ;则ax=0132h,d1=32,bh=01;
145      mov    ah, 02h  ;显示输出dx,则显示32码对应的数字, 2
146      int    21h
147      mov    al, bh   ;把01给a1,然后算出ASCII码,然后给dx,然后显示
148      add    al, 30h
149      mov    dl, al   ;除数B有 8位和16位两种,保存在一个reg寄存器里 或是内存单元中。
150
151      ;被除数A 默认存放在AX中(16位以内) 或 AX和DX中(32位, DX存放高
152
153      ;结果: 如果除数B是8位,那么除法的结果AL保存商,AH保存余数,
154      ;如果除数B是16位,那么除法的结果 AX保存商, DX保存余数。
155      mov    ah, 02h
156      int    21h
157      ret
158
159 CODES  ENDS
160      END  START
161
162
163
164
165
166
167

```