

# 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：202120130276	姓名：王云强	班级：21.2 班
实验题目：实验九：实验 2.3		
实验学时：2	实验日期：2023.12.12	
<p>实验目的：</p> <ol style="list-style-type: none"><li>1. 巩固分支程序设计中所涉及的知识点。</li><li>2. 学会在自编程序中利用分支程序设计的理论与技巧。</li><li>3. 学会在自编程序中使用系统调用</li></ol>		
实验环境：Windows10、DOSBox-0.74、Masm64		
<p>源程序清单：</p> <ol style="list-style-type: none"><li>1. EX2_3.asm（实验 2.3 源程序）</li></ol>		
<p>编译及运行结果：</p> <p>编译结果：</p> <pre>C:\&gt;masm EX2_3 Microsoft (R) Macro Assembler Version 5.00 Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.  Object filename [EX2_3.OBJ]: Source listing [NUL.LST]: Cross-reference [NUL.CRF]:      51602 + 464942 Bytes symbol space free      0 Warning Errors     0 Severe Errors  C:\&gt;link EX2_3  Microsoft (R) Overlay Linker Version 3.60 Copyright (C) Microsoft Corp 1983-1987. All rights reserved.  Run File [EX2_3.EXE]: List File [NUL.MAP]: Libraries [.LIB]: LINK : warning L4021: no stack segment</pre>		

运行结果：

```
C:\>EX2_3
aisdhASD245J
5
4
3

C:\>EX2_3
123456ABCef
2
3
6
```

问题及收获：

问题：

一、如何使用 21h 中断表中的 0A 号功能进行输入？

这个问题其实在前面出现过很多次了，之前的实验中都有涉及，这里就简单提一下，首先 0A 号功能是读入一个字符串，主要是用到 DX 寄存器，DX 对应地址单元的地址作为开始输入的地址，DX + 1 对应地址单元存着字符串长度，DX + 2 对应地址单元开始储存实际输入的字符串，由于题目要求最多 100 个字符，所以只需要开 105 就足够。这样得到的字符串就会全部储存在 COUNTER 数组中了。

定义 COUNTER：

COUNTER	DB	105
	DB	?
	DB	105 DUP(0)

进行输入

```

LEA     DX, COUNTER
MOV     AH, 0AH
INT     21H
MOV     AL, COUNTER + 1           ;对字符串进行处理
ADD     AL, 2
MOV     AH, 0
MOV     SI, AX
MOV     COUNTER[SI], '$'

LEA     DX, CRLF                 ;另取一行
MOV     AH, 09H
INT     21H

```

## 二、如何用 21h 中断表中的 09 号功能进行输出？

这个是本程序主要需要解决的问题，因为计数的话只需要配合 LOOP 指令即可实现，那么如何输出是关键。前面的实验中都是采用 02 功能进行输出，而 09 功能主要是输出字符串，实际上也都差不多。09 输出字符串，并要求字符串以“\$”结尾，所以我们根据输入的时候字符串放在数组中，考虑在输出的时候，也将数字转换成 ASCII 型数字然后输出，存放在数组中，之后后数组的最后加一个\$即可。

于是先定义了如下数组（实际上并不需要开到 10，因为长度不超过 100，即最多输出 3 个字符，也就是 100 对应的 ASCII 码，但是开大一点主要为了防止出错）

```

9 TEP          DB          10 DUP('$')

```

之后我们依次输出用来计数的三个变量，对于每个变量，我们先考虑它是否为 100，如果为 100 的话，则采用下图方式输出，即数组 TEP 的 0 号位置为 1 的 ASCII 码 49，1 号和 2 号位置为 0 的 ASCII 码 48，之后再输出回车换行即可。

TWO:

```
MOU      [TEP + 0], 31H
MOU      [TEP + 1], 30H
MOU      [TEP + 2], 30H
MOU      [TEP + 3], 0AH
MOU      [TEP + 4], 0DH
MOU      [TEP + 5], '$'
LEA      DX, TEP
MOU      AH, 09H
INT      21H
JMP      EXITING
```

如果不为 100 但是大于 10 的话，我们采用下图方式输出，即先对十位数进行除 10，得到商和余数，之后将商和余数都转换成 ASCII 码，然后输出即可。

```
MOU      BL, 10
MOU      BH, 0
MOU      AH, 0
DIV      BL
MOU      BL, AH
ADD      BL, 30H
MOU      [TEP + 1], BL
ADD      AL, 30H
MOU      [TEP + 0], AL
MOU      [TEP + 2], 0AH
MOU      [TEP + 3], 0DH
MOU      [TEP + 4], '$'
LEA      DX, TEP
MOU      AH, 09H
INT      21H
JMP      EXITING
```

如果小于 10 的话，那就采用下图的方式输出，即直接在 TEP 数组 0 号位置存数输出即可。

ONE:

```
ADD      AL, 30H
MOV      [TEP + 0], AL
MOV      [TEP + 1], 0AH
MOV      [TEP + 2], 0DH
MOV      [TEP + 3], '$'
LEA      DX, TEP
MOV      AH, 09H
INT      21H
JMP      EXITING
```

### 三、程序的整体思路是什么？

整体思路还是比较简单、明确的。首先先进行读入，将输入的字符串读到数组中。然后利用 LOOP 循环，从数组中逐一读出字符，利用 ASCII 码，判断该字符属于小写字母、大写字母、数字中的哪一种。之后利用三个全局变量来记录小写字母、大写字母、数字出现的次数。之后利用一个子函数，将对应的出现次数放在 AX 寄存器中，传入子函数，在子函数中先特殊判断是不是 100 或者是个位数，如果是，则分别进行输出。否则的话则一定是两位数，利用除法得到商和余数，对应的就是十位数和个位数，转换一下输出即可。

下图为计数部分代码，即每次从数组中读出一个字符，然后判断它的 ASCII 码大小，之后对应到相应的全局变量加 1 即可。

COUNT:		MOU	AL,[COUNTER + BX]
		CMP	AL,65
		JB	COUNT_NUM
		CMP	AL,97
		JB	COUNT_LOW
		INC	U_NUM
		JMP	NEXT
COUNT_NUM:		INC	N_NUM
		JMP	NEXT
COUNT_LOW:		INC	L_NUM
		JMP	NEXT
NEXT:		INC	BX
		LOOP	COUNT

收获:

一、首先就是对于输入输出有了更深的认识，之前都是抄写样例程序，而这次是从零开始自己按照给定要求进行设计，印象还是很深刻的。尤其是对于 21H 中断的 09 号功能有了更新的体会，通过自己实际将定义三个全局变量以数组的形式输出，也锻炼了写汇编代码的能力。

二、同时还复习了之前所学的循环以及分支判断的相关代码书写，对分支循环以及 loop 指令也有了进一步的认识。