《微机原理与接口技术》实验报告

|  |  |
| --- | --- |
| 姓 名： | 郑德凯 |
| 学 号： | U202217216 |
| 专业班级： | 软件2202班 |
| 实验名称： | 第2次实验 |
| 实验日期： | 2024.10.14 |

备注：

（1）请将报告电子版发到邮箱 MrSuInterfaceWork@163.com，文件名：姓名-学号-班级-微机原理-第X次实验.docx。

（2）提交的内容：文档，实验源代码文件（有几个任务就提交几个源代码，源码文件命名方式：Test0103.asm，第1次实验的第3个任务）

（3）邮件的主题和文档名同名。

（4）文档排版统一为小四仿宋，行间距离1.5倍行距。

（5）提交日期：下一次实验之前。

## 一、实验目的

1）掌握跳转指令/循环指令，掌握分支程序和循环程序的设计

2）掌握子程序的设计和调用，掌握子程序参数传递和现场保护的概念

3）掌握DOS 21H常见系统调用的功能和应用。

4）掌握查表程序的设计思想，掌握解决类似工程数据处理的方法

## 二、实验内容

1)实验2.1 数值大小判断分支程序

2)实验2.2 字符串处理分支程序

3)实验2.3 位处理分支程序

4)实验2.4 余弦函数计算程序

5)实验2.5 字节数字显示子程序

## 三、实验过程和疑难问题解决

### 3.1 数值大小判断分支程序实验

#### 3.1.1 源代码

.MODEL SMALL

.STACK 100H

.DATA

prompt DB 'Enter 8 numbers (0-100), press Enter after each:', 13, 10, '$'

error\_msg DB 'Invalid input! Please enter a number between 0 and 100.', 13, 10, '$'

newline DB 13, 10, '$'

range1\_msg DB 'Numbers in [0-59]: $'

range2\_msg DB 'Numbers in [60-79]: $'

range3\_msg DB 'Numbers in [80-100]: $'

avg\_msg DB 'Average: $'

numbers DB 8 DUP(?)

range1 DB 0

range2 DB 0

range3 DB 0

sum DW 0

count DB 0

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

; 显示提示信息

LEA DX, prompt

MOV AH, 9

INT 21H

LEA SI, numbers

INPUT\_LOOP:

CALL READ\_NUMBER

JC INPUT\_ERROR ; 如果出错则跳转到错误处理

MOV [SI], AL

INC SI

; 累加数字和

XOR AH, AH

ADD sum, AX

; 对输入数字进行分类

CMP AL, 60

JL INCREMENT\_RANGE1

CMP AL, 80

JL INCREMENT\_RANGE2

JMP INCREMENT\_RANGE3

INCREMENT\_RANGE1:

INC range1

JMP CONTINUE\_INPUT

INCREMENT\_RANGE2:

INC range2

JMP CONTINUE\_INPUT

INCREMENT\_RANGE3:

INC range3

CONTINUE\_INPUT:

INC count

CMP count, 8

JL INPUT\_LOOP

JMP DISPLAY\_RESULTS

INPUT\_ERROR:

LEA DX, error\_msg

MOV AH, 9

INT 21H

JMP INPUT\_LOOP

DISPLAY\_RESULTS:

; 显示分类结果

CALL DISPLAY\_NEWLINE

LEA DX, range1\_msg

MOV AH, 9

INT 21H

MOV AL, range1

CALL DISPLAY\_NUMBER

CALL DISPLAY\_NEWLINE

LEA DX, range2\_msg

MOV AH, 9

INT 21H

MOV AL, range2

CALL DISPLAY\_NUMBER

CALL DISPLAY\_NEWLINE

LEA DX, range3\_msg

MOV AH, 9

INT 21H

MOV AL, range3

CALL DISPLAY\_NUMBER

; 显示平均值

CALL DISPLAY\_NEWLINE

LEA DX, avg\_msg

MOV AH, 9

INT 21H

MOV AX, sum

MOV BL, 8

DIV BL

CALL DISPLAY\_NUMBER

MOV AH, 4CH

INT 21H

MAIN ENDP

READ\_NUMBER PROC

XOR BX, BX

XOR CX, CX

READ\_DIGIT:

MOV AH, 1

INT 21H

CMP AL, 13

JE END\_READ

CMP AL, '0'

JL INVALID\_INPUT

CMP AL, '9'

JG INVALID\_INPUT

SUB AL, '0'

MOV AH, 0

PUSH AX

MOV AX, BX

MOV DX, 10

MUL DX

MOV BX, AX

POP AX

ADD BX, AX

INC CX

CMP CX, 3

JG INVALID\_INPUT

CMP BX, 100

JG INVALID\_INPUT

JMP READ\_DIGIT

INVALID\_INPUT:

STC

RET

END\_READ:

MOV AL, BL

CLC

RET

READ\_NUMBER ENDP

DISPLAY\_NUMBER PROC

XOR AH, AH

MOV BL, 10

DIV BL

PUSH AX

CMP AL, 0

JE DISPLAY\_SINGLE\_DIGIT

ADD AL, '0'

MOV DL, AL

MOV AH, 2

INT 21H

DISPLAY\_SINGLE\_DIGIT:

POP AX

MOV AL, AH

ADD AL, '0'

MOV DL, AL

MOV AH, 2

INT 21H

RET

DISPLAY\_NUMBER ENDP

DISPLAY\_NEWLINE PROC

MOV AH, 2

MOV DL, 13

INT 21H

MOV DL, 10

INT 21H

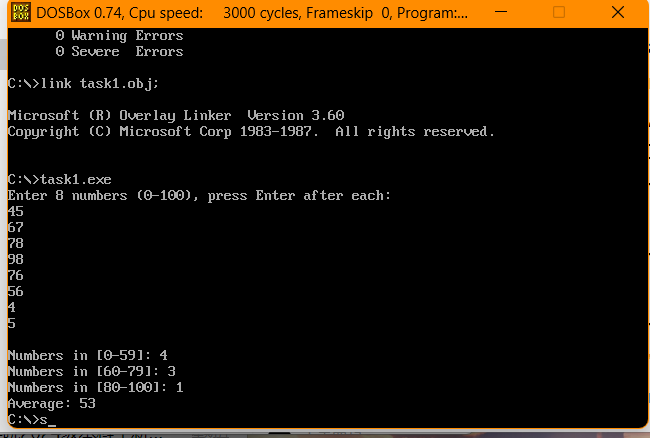
RET

DISPLAY\_NEWLINE ENDP

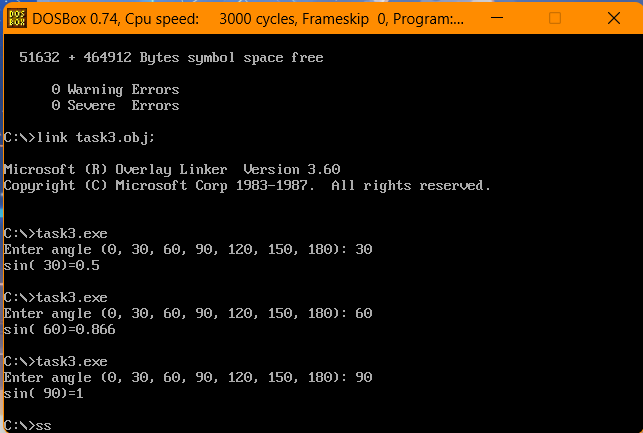
END MAIN

#### 3.1.2 观察到的现象

在运行代码后，成功输出了在各范围区间的个数以及平均数



***图1***



### 3.2 字符串处理分支程序实验

#### 3.2.1 源代码

DATA SEGMENT

PROMPT DB 'Please enter a sentence: $'

INPUT DB 80, ?, 80 DUP(' ')

OUTPUT DB 80 DUP(' '), '$'

CHANGEROW DB 0DH, 0AH, '$'

DATA ENDS

STACK SEGMENT

DW 128 DUP(0)

STACK ENDS

CODE SEGMENT

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

START:

MOV AX, DATA

MOV DS, AX

MOV AX, STACK

MOV SS, AX

MOV SP, 256

; 显示提示信息

MOV AH, 09H

LEA DX, PROMPT

INT 21H

; 读取输入的句子

LEA DX, INPUT

MOV AH, 0AH

INT 21H

; 处理句子，首字母大写，其他字母小写

LEA SI, INPUT + 2

LEA DI, OUTPUT

MOV BL, ' '

MOV CL, [INPUT+1]

PROCESS\_LOOP:

MOV AL, [SI]

; 检查是否为第一个字符

CMP SI, OFFSET INPUT + 2

JNE CHECK\_SPACE

; 将首字母转换为大写

CMP AL, 'a'

JB STORE\_CHAR

CMP AL, 'z'

JA STORE\_CHAR

SUB AL, 20H

JMP STORE\_CHAR

CHECK\_SPACE:

; 如果前一个字符不是空格，将当前字符转换为小写

CMP BL, ' '

JE STORE\_CHAR

CMP AL, 'A'

JB STORE\_CHAR

CMP AL, 'Z'

JA STORE\_CHAR

ADD AL, 20H

STORE\_CHAR:

MOV [DI], AL

INC DI

MOV BL, AL

INC SI

LOOP PROCESS\_LOOP

; 结束输出字符串

MOV BYTE PTR [DI], '$'

MOV AH, 09H

LEA DX, CHANGEROW

INT 21H

; 显示处理后的句子

MOV AH, 09H

LEA DX, OUTPUT

INT 21H

MOV AH, 4CH

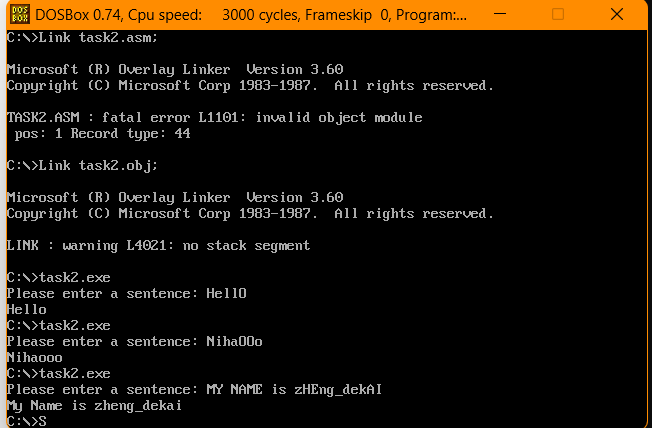
INT 21H

CODE ENDS

END START

#### 3.2.2 观察到的现象

可以看到在图二中，输入的句子均完成功能



***图二***

### 3.3 字符串处理实验

#### 3.3.1 源代码

.MODEL SMALL

.STACK 100H

.DATA

prompt DB 'Enter angle (0, 30, 60, 90, 120, 150, 180): $'

error\_msg DB 'Invalid input! Please enter a valid angle.', 13, 10, '$'

newline DB 13, 10, '$'

angles DW 0, 30, 60, 90, 120, 150, 180

sine\_values DB 'sin( 0)=0 $'

DB 'sin( 30)=0.5 $'

DB 'sin( 60)=0.866$'

DB 'sin( 90)=1 $'

DB 'sin(120)=0.866$'

DB 'sin(150)=0.5 $'

DB 'sin(180)=0 $'

value\_length DB 15 ; 每个正弦值字符串的长度（包括$符号）

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

INPUT:

; 显示提示信息

LEA DX, prompt

MOV AH, 9

INT 21H

; 读取角度

CALL READ\_NUMBER

; 验证输入

LEA SI, angles

MOV CX, 7 ; 有效角度的数量

MOV BX, AX ; 将输入存储在BX中

VALIDATE:

CMP BX, [SI]

JE VALID\_INPUT

ADD SI, 2

LOOP VALIDATE

; 无效输入

LEA DX, error\_msg

MOV AH, 9

INT 21H

JMP INPUT

VALID\_INPUT:

; 计算正弦值的索引

SUB SI, OFFSET angles

MOV AX, SI

MOV BL, 2

DIV BL

; 计算sine\_values中的偏移量

MUL value\_length

LEA SI, sine\_values

ADD SI, AX

; 显示正弦值

MOV DX, SI

MOV AH, 9

INT 21H

; 打印换行

LEA DX, newline

INT 21H

; 退出程序

MOV AH, 4CH

INT 21H

MAIN ENDP

READ\_NUMBER PROC

XOR BX, BX ; 使用BX存储数字

MOV CX, 3 ; 最多3位数字

READ\_DIGIT:

MOV AH, 1

INT 21H

CMP AL, 13 ; 检查回车键

JE END\_READ

CMP AL, '0'

JL INVALID\_INPUT

CMP AL, '9'

JG INVALID\_INPUT

SUB AL, '0'

MOV AH, 0

PUSH AX

MOV AX, 10

MUL BX

MOV BX, AX

POP AX

ADD BX, AX

LOOP READ\_DIGIT

JMP END\_READ

INVALID\_INPUT:

MOV BX, 1000 ; 设置一个无效值

END\_READ:

MOV AX, BX ; 将结果移动到AX

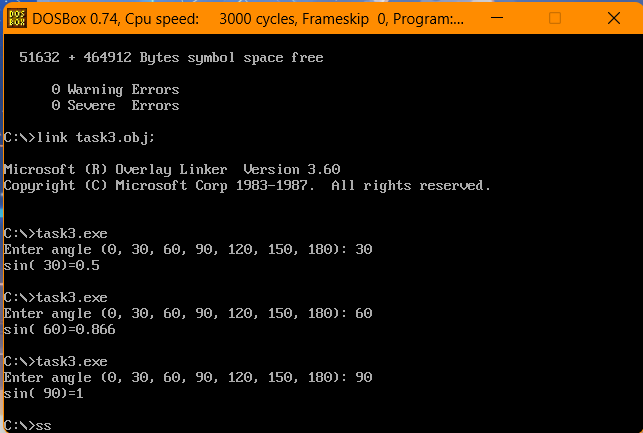
RET

READ\_NUMBER ENDP

END MAIN

#### 3.3.2 观察到的现象

可以看到在图三中，成功输出了正弦值



*图三*

## 四、收获

通过第二次汇编实验的学习，我加深了对汇编语言中分支和循环程序设计的理解。在数值大小判断程序中，掌握了如何通过条件跳转指令进行数据范围的统计；在字符串处理程序中，学会了如何判断字符的类型并进行格式化处理；而位处理程序则让我更加熟悉了位操作与条件判断的结合使用。这些实验帮助我全面理解了汇编语言在数据处理和程序控制方面的应用，为今后的编程打下了坚实的基础。