

## 《微机原理与接口技术》实验报告

姓 名：	
学 号：	
专业班级：	
实验名称：	第二次实验
实验日期：	2023 年 10 月 17 日

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（有几个任务就提交几个源代码）
- (3) 邮件的主题和文件名同名。
- (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) 掌握典型的跳转指令和循环指令的原理，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DOS21H常见系统调用的功能和应用。
- 4) 掌握查表程序的设计思想，解决一类工程数据处理问题。

## 二、实验内容

- 1) 数值大小判断分支程序(2023.10)
- 2) 字符串处理分支程序(2023.10)
- 3) 位处理分支程序(2023.10)
- 4) 余弦函数计算程序(2023.10)
- 5) 字节数字显示子程序(2023.10)

## 三、实验过程和结果

### 3.1 字符串处理分支程序

#### 3.1.1 源代码

```
1  DATA SEGMENT
2      PROMPT      DB 'Please enter a sentence: $'
3      INPUT       DB 80, ?, 80 DUP(' ')
4      OUTPUT      DB 80 DUP(' '), '$'
5      CHANGEROW   DB 0DH, 0AH, '$'
6  DATA ENDS
7
8  STACK SEGMENT
9      DW 128 DUP(0)
10 STACK ENDS
11
12 CODE SEGMENT
13             ASSUME CS:CODE, DS:DATA, SS:STACK
14
15  START:
16             MOV     AX, DATA
17             MOV     DS, AX
18             MOV     AX, STACK
19             MOV     SS, AX
20             MOV     SP, 256
21
22             ; Display prompt
23             MOV     AH, 09H
24             LEA     DX, PROMPT
25             INT     21H
26
27             ; Read input sentence
28             LEA     DX, INPUT
29             MOV     AH, 0AH
```

```

30             INT     21H
31
32     ; Process the sentence
33             LEA     SI, INPUT + 2             ; Skip the length
byte
34             LEA     DI, OUTPUT
35             MOV     BL, ' '                 ; Initialize BL
with space character
36
37     PROCESS_LOOP:
38             MOV     AL, [SI]
39
40     ; Convert to uppercase if previous character is space
41             CMP     BL, ' '
42             JNE     CHECK_LOWERCASE
43             CMP     AL, 'a'
44             JB      STORE_CHAR
45             CMP     AL, 'z'
46             JA      STORE_CHAR
47             SUB     AL, 20H
48             JMP     STORE_CHAR
49
50     CHECK_LOWERCASE:
51     ; Convert to lowercase
52             CMP     AL, 'A'
53             JB      STORE_CHAR
54             CMP     AL, 'Z'
55             JA      STORE_CHAR
56             ADD     AL, 20H
57
58     STORE_CHAR:
59             MOV     [DI], AL
60             INC     DI
61             MOV     BL, AL
62             INC     SI
63             DEC     BYTE PTR [INPUT+1]       ; Decrement the
length counter
64             JNZ     PROCESS_LOOP
65
66     ; Terminate the output string
67             MOV     BYTE PTR [DI], '$'
68
69             MOV     DX, OFFSET CHANGEROW
70             MOV     AH, 09H
71             INT     21H
72
73     ; Display the processed sentence
74             MOV     AH, 09H
75             LEA     DX, OUTPUT
76             INT     21H
77
78     ; End program
79             MOV     AH, 4CH
80             INT     21H
81
82     CODE ENDS

```

83 END START

84

### 3.1.2 观察到的现象

```
Drive D is mounted as local directory ./code/

Z:\>d:

D:\>set PATH=C:\TASM

D:\>TASM D:\test.asm
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file: D:\test.asm to test.OBJ
*Warning* D:\test.asm(8) Reserved word used as symbol: STACK
Error messages: None
Warning messages: 1
Passes: 1
Remaining memory: 467k

D:\>TLINK D:\test
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International
Warning: No stack

D:\>D:\test
Please enter a sentence: hello2 world
Hello2 World
D:\>_
```

输入一个带阿拉伯数字的英语句子，会自动将首字符大写，别的字母小写，其他数字不变。

## 3.2 正弦函数计算程序

### 3.2.1 源代码

```
1 DATA SEGMENT
2     PROMPT DB 'Please enter the degree (0,30,60,90,120,150,180): $'
3     OUTPUT DB 'sin(', 3 DUP(' '), ')=','$'
4     DEGREES DW 0, 30, 60, 90, 120, 150, 180
5     VALUES DB '0 $', '0.5 $', '0.866$', '1 $', '0.866$', '0.5 $',
6     '0 $'
7     CHANGEROW DB 0DH, 0AH, '$',
8
9 DATA ENDS
10
11 STACK SEGMENT
12     DW 128 DUP(0)
13 STACK ENDS
14
15 CODE SEGMENT
16     ASSUME CS:CODE, DS:DATA, SS:STACK
17
18 START:
19     MOV AX, DATA
20     MOV DS, AX
21     MOV AX, STACK
22     MOV SS, AX
23     MOV SP, 256
```

```

23 ; Display prompt
24         MOV     AH, 09H
25         LEA     DX, PROMPT
26         INT     21H
27
28 ; Read input
29         MOV     AH, 01H
30         INT     21H
31         MOV     [OUTPUT+4], AL
32         CMP     AL, '9'
33         JA      SINGLE_DIGIT
34         INT     21H
35         MOV     [OUTPUT+5], AL
36
37 SINGLE_DIGIT:
38 ; Convert input to number
39         MOV     AH, 0
40         MOV     AL, [OUTPUT+4]
41         SUB     AL, '0'
42         MOV     BH, 10
43         MUL     BH
44         MOV     BH, [OUTPUT+5]
45         SUB     BH, '0'
46         ADD     AL, BH
47         MOV     AH, 0
48
49 ; Find sin value
50         LEA     BX, DEGREES
51         LEA     SI, VALUES
52         MOV     CX, 7
53
54 FIND_VALUE:
55         CMP     AX, [BX]
56         JE      DISPLAY_VALUE
57         ADD     BX, 2
58         ADD     SI, 6
59         DEC     CX
60         JNZ     FIND_VALUE
61
62 ; Degree not found
63         INT     20H
64
65 DISPLAY_VALUE:
66
67         MOV     DX, OFFSET CHANGEROW
68         MOV     AH, 09H
69         INT     21H
70
71         MOV     AH, 09H
72         LEA     DX, OUTPUT
73         INT     21H
74         MOV     AH, 09H
75         MOV     DX, SI
76         INT     21H
77
78 ; End program

```

```

79      MOV     AH, 4CH
80      INT     21H
81
82  CODE ENDS
83  END START
84

```

### 3.2.2 观察到的现象

运行程序，可以观察窗口内输出了对应的正弦值：

```

Z:\>d:
D:\>set PATH=C:\TASM
D:\>TASM D:\test.asm
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file:  D:\test.asm to test.OBJ
*Warning* D:\test.asm(6) Missing operand - trailing ? assumed
*Warning* D:\test.asm(9) Reserved word used as symbol: STACK
Error messages:   None
Warning messages: 2
Passes:           1
Remaining memory: 466k

D:\>TLINK D:\test
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International
Warning: No stack

D:\>D:\test
Please enter the degree (0,30,60,90,120,150,180): 60
sin(60 )=0.866
D:\>

```

## 3.3 字节数字显示子程序

### 3.3.1 源代码

```

1  DATA SEGMENT
2      DIWEI      DW 0
3      GAOWEI     DW 0
4      COUNT      DW 0
5      SAVE       DW 5 DUP(0)
6      SAVE2      DW 5 DUP(0)
7      INPUTWARN  DB 'INPUT YOUR NUMBER,END WITH ENTER: ','$'
8      OUTPUTWARN DB 'TRANSFORED NUMBER: ','$'
9      ERRORWARN  DB 'INPUT ERROR!','$'
10     SHUWEI      DW 0,1,10,100,1000,10000
11     MOD         DW 16
12 DATA ENDS
13
14 STACK SEGMENT
15     DW 100 DUP(0)
16 STACK ENDS
17
18 CODE SEGMENT
19     ASSUME CS:CODE,DS:DATA,ES:DATA

```

20	START:			
21		MOV	AX, DATA	
22		MOV	DS, AX	
23		MOV	ES, AX	
24		MOV	AX, STACK	
25		MOV	SS, AX	
26				
27		LEA	SI, SAVE	
28		LEA	DX, INPUTWARN	
29		MOV	AH, 9H	
30		INT	21H	
31				
32		;调用子程序进行处理		
33		CALL	NumToHex	
34				
35	OVER:			
36		MOV	AH, 4CH	
37		INT	21H	
38				
39				
40	NumToHex PROC			
41	L0:			
42		MOV	AH, 1H	
43		INT	21H	
44		CMP	AL, 0DH	
45		JZ	L1	;输入回车，输入过程结束
46		CMP	AL, 30H	
47		JB	L2	;不是数字字符提示输入错误
48		CMP	AL, 39H	
49		JA	L2	
50		JMP	L3	;输入为数字字符
51				
52	L1:			
53		MOV	DL, 0AH	;结束输入，令下一次输出换行
54		MOV	AH, 02H	
55		INT	21H	
56		JMP	L4	
57				
58	L2:			
59		LEA	DX, ERRORWARN	;错误警告，结束程序
60		MOV	AH, 9H	
61		INT	21H	
62		JMP	OVER	
63	L3:			
64		SUB	AL, 30H	;转换为数字
65		MOV	AH, 0	
66		MOV	[SI], AX	;将每一位数字存储在SAVE中
67		INC	SI	
68		INC	SI	
69		INC	COUNT	;统计输入数字的位数
70		JMP	L0	
71				
72	L4:			

73		LEA	SI,SAVE	
74		MOV	CX,COUNT	
75		MOV	DI,COUNT	
76		ADD	DI,COUNT	
77				
78	L5:			;将输入的数字按位乘以1,
			10, 100, 1000, 10000, 并相加放入GAOWEI,DIWEI中	
79		MOV	AX,[SI]	
80		MUL	SHUWEI[DI]	
81		INC	SI	
82		INC	SI	
83		DEC	DI	
84		DEC	DI	
85		ADD	DIWEI,AX	
86		ADC	GAOWEI,DX	
87		MOV	DX,0	
88		LOOP	L5	
89				
90		LEA	SI,SAVE2	;对16取余,得到十六进制
	数			
91		MOV	COUNT,0	
92		MOV	AX,DIWEI	
93		MOV	DX,GAOWEI	
94	L6:			
95		DIV	MOD	
96		MOV	[SI],DX	
97		INC	COUNT	
98		INC	SI	
99		INC	SI	
100		MOV	DX,0	
101		CMP	AX,0	
102		JNZ	L6	
103				
104		LEA	DX,OUTPUTWARN	
105		MOV	AH,9H	
106		INT	21H	
107				;SAVE2中存储单元地址从低到高存储数据的高位到低位
108		LEA	SI,SAVE2	;从高位到低位检索,直至
	第一个不为0的数			
109		ADD	SI,8	
110	L7:			
111		LEA	DI,SAVE2	
112		CMP	SI,DI	;检查当前SI的指向是否越
	界,越界输出0			
113		JB	ZERO	
114		CMP	[SI],0	
115		JNZ	L9	;找到不为零的数
116				
117	L8:			
118		DEC	SI	
119		DEC	SI	
120		JMP	L7	
121				
122	L9:			
123		LEA	DI,SAVE2	
124		CMP	SI,DI	



```
125          JB      OUTPUTH          ;检查当前SI的指向是否越
      界，越界输出H
126          CMP     [SI],9          ;与9比较，若为数字，加
      30H，为字母，加37H，转换为ASCII码
127          JA      L10
128          MOV     AX,[SI]
129          ADD     AL,30H
130          MOV     DL,AL
131          MOV     AH,02H
132          INT     21H          ;直接输出显示为十六进制
      位
133          DEC     SI
134          DEC     SI
135          JMP     L9
136
137      L10:
138          MOV     AX,[SI]
139          ADD     AL,37H
140          MOV     AH,02H
141          MOV     DL,AL
142          INT     21H
143          DEC     SI
144          DEC     SI
145          JMP     L9
146
147      ZERO:
148          MOV     DL,'0'
149          MOV     AH,02H
150          INT     21H
151
152      OUTPUTH:          ;H结尾，说明16进制
153          MOV     DL,'H'
154          MOV     AH,02H
155          INT     21H
156          RET
157 NumToHex ENDP
158 CODE ENDS
159 END START
```

### 3.3.2 观察到的现象

输入十进制的数字，屏幕上会自动显示其十六进制的数值：

```
D:\>set PATH=C:\TASM

D:\>TASM D:\ntest.asm
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file: D:\ntest.asm to test.OBJ
*Warning* D:\ntest.asm(11) Reserved word used as symbol: MOD
*Warning* D:\ntest.asm(14) Reserved word used as symbol: STACK
*Warning* D:\ntest.asm(114) Argument needs type override
*Warning* D:\ntest.asm(126) Argument needs type override
Error messages: None
Warning messages: 4
Passes: 1
Remaining memory: 465k

D:\>TLINK D:\ntest
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International
Warning: No stack

D:\>D:\ntest
INPUT YOUR NUMBER,END WITH ENTER: 21

TRANSFORMED NUMBER: 15H
D:\>_
```

## 四、收获

---

通过微机原理实验二，我深入理解了跳转指令和循环指令的工作原理，以及如何设计分支和循环程序。我掌握了子程序的设计和调用方法，特别是参数传递和现场保护的重要概念。此外，我还学习了如何使用DOS 21H系统调用来实现各种功能，并通过查表程序设计思想解决了实际的工程数据处理问题。