汇编语言程序设计课程实验报告

实验 8

姓名		院系	计算机工程与科学学院	学号	
实验目的：					
掌握跳跃表法实现多分支程序的方法；掌握子程序设计方法。					
实验说明					
用户菜单显示操作提示，用户按下菜单键，执行相应的选项。执行完成，重新显示用户菜单，直到用户按下退出键。					
实验内容：					
通过显示功能菜单，实现四个功能。 (1) 在屏幕上显示字符串“hello world!”.(2) 加法器，等待用户输入两个数，并返回加法结果。(3) 显示当前系统时间，以时：分：秒显示. (4) 退出程序.					

1. 程序分析

程序的基本思路是分为 4 个“大”子程序进行设计，将子程序的偏移地址存入跳跃表中，由菜单提示用户输入进行选择哪个功能 ，根据用户的输入，找到跳跃表中的位置，进而转入相应的子程序中执行，最后返回菜单栏。

难点在于处理 8 位数的加法和显示系统时间，前者需要先将用户输入存储为字符串，处理字符串变为二进制数，再将二进制数通过除 10 操作入栈弹栈进行输出。后者需要调用第 2C 号 dos 中断获取系统当时时间，再对时分秒进行二进制转十进制输出，过程较为繁琐。

2. 程序运行结果

```
************************************************************
Please input
1 print hello world!
2 adder
3 display current time
4 quit
************************************************************
3
11:16:49
```

```
D:\>D:\TEST
************************************************************
Please input
1 print hello world!
2 adder
3 display current time
4 quit
************************************************************
2
Please X:
23
Please Y:
63
86
```

3. 程序流程图

4. 实验体会

这次实验的难度适中，功能实现的内容是前几个实验中都有涉及的地方，比如二进制转十进制输出，获取系统时间（实验 2 内容），难点在于理解并使用跳跃表法进行分支结构程序设计，需要搞懂其原理在于 jmp 的无条件跳转可以改变 cs 与 ip，而且需要注意由于地址是 dw 类型，故每个子程序地址之间相隔 2 个地址单位长度。这让我增加了对高级语言中 if-elseif-else 以及 switch-case 的分支结构实现的理解。

5. 关键代码

跳跃表跳转与菜单显示

```
1.          memu_display:
2.                      sub     ax,bX
3.                      sub     bx,bX
4.                      sub     cx,cX
5.                      sub     dx,dX
6.
7.                      lea     dx, menu
8.                      mov     ah,09h
9.                      int     21h
10.
11.                     mov     ah,01h
12.                     int     21h
13.                     call    change
14.
15.                     sub     ah,ah
16.                     sub     al,'0'
17.                     sub     al,1
18.                     shl     al,1
19.
20.                     mov     si,ax
21.                     jmp     table[si]
```

字符串“十进制”转二进制

```
1.    str2num proc far
2.                      push    bx
3.                      sub     ax,ax
4.                      sub     cx,cx
5.                      sub     dx,dx
6.
7.                      mov     cl,[si]
8.                      mov     bx,10
9.          l1:
10.                     add     si,1
11.                     mov     dl,[si]
12.                     sub     dh,dh
```

```
13.                 sub     dl,'0'
14.                 mov     temp,dl
15.                 mul     bx
16.                 add     ax,word ptr temp
17.                 loop    l1
18.                 pop     bx
19.                 mov     [bx],ax
20.                 ret
21.   str2num endp
```

二进制转十进制输出

```
1.    display_result proc far
2.
3.                  MOV     AH,00H                  ;
4.                  XOR     CX,CX                   ;       CX 记录十进制位数
5.                  MOV     BL,10                   ;       除数
6.          LOOP1:
7.                  DIV     BL                  ;           出发操作，余数在 AH，商在 AL
8.                  INC     CX                  ;           位数加 1
9.                  PUSH    AX                  ;           入栈保存
10.                 MOV     AH,00H                  ;       清除余数
11.                 XOR     AL,00H                  ;       检查是否变为 0
12.                 JNZ     LOOP1                   ;       若还有的除，继续
13.
14.                 MOV     AH,02H                  ;       AH=02H 输出字符
15.         LOOP2:
16.                 POP     DX                  ;
17.                 MOV     DL,DH                   ;       DH 里是要输出的余数
18.                 ADD     DL,30H                  ;       转 ASCII 码
19.                 INT     21H                 ;           输出
20.                 LOOP    LOOP2                   ;       CX = CX-1 JNZ
21.                 ret
22.   display_result endp
```