****

IBM-PC汇编语言

实验报告

|  |  |
| --- | --- |
| 实验内容： | Fabonacci |
| 姓 名： | 乔 楠 |
| 学 号： | 0919170210 |
| 专业班级： | 物联网工程1702班 |
| 指导教师： | 贺建飚 |

**实验五 Fibonacci**

1. **设计说明**

**1.1实验原理**

调用FIB函数求Fibonacci数,结果高位存在RESULT2H，低位存在RESULT2L，最后调用OUTPUT函数将结果以十进制的形式显示出来。

**1.2实验目的**

　　（1） 掌握循环程序的设计方法

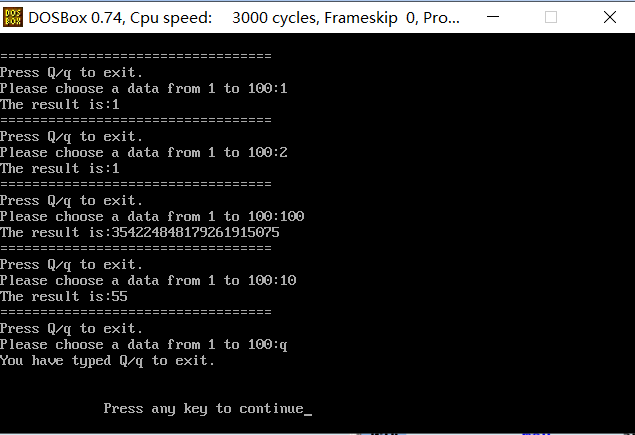
（2）掌握汇编语言源程序的编辑、汇编、连接及调试过程。

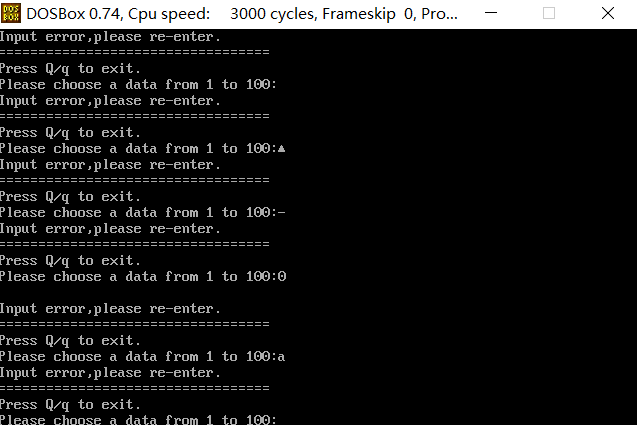
（3）进一步熟悉利用DEBUG程序修改参数的方法，并检查和验证结果的正确性。

（4） 学会针对不同的问题，选用不同的组织循环的方法。

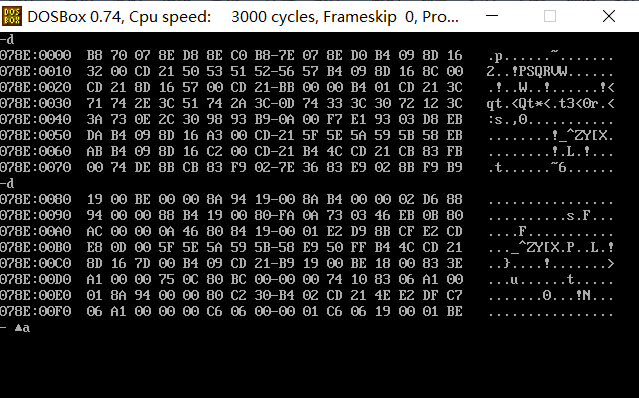
1. **调试说明**

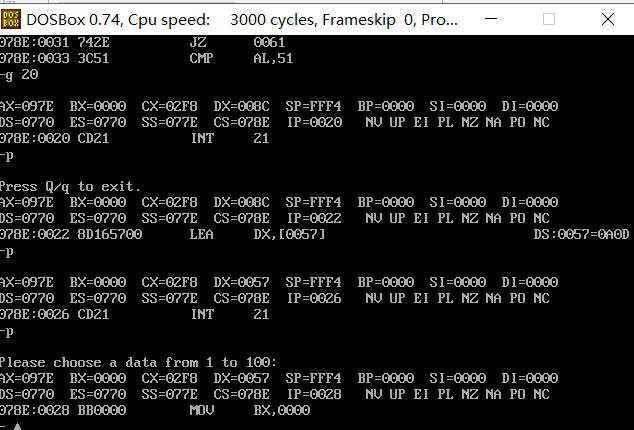
**2.1实验结果**

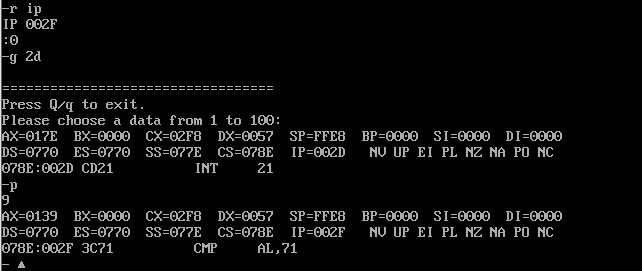


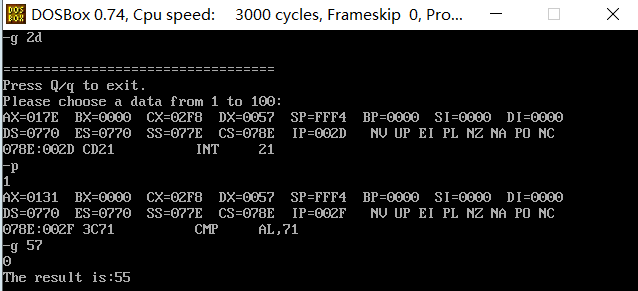


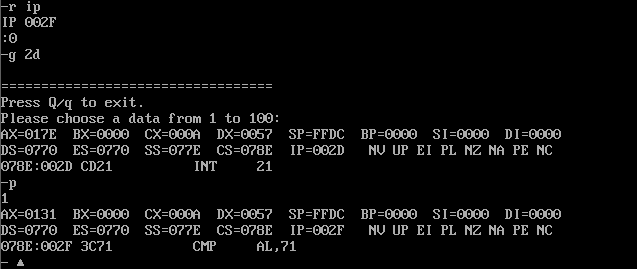
**2.2实验调试**



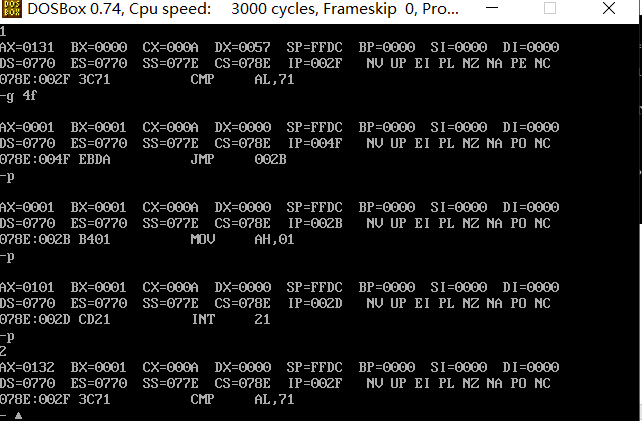




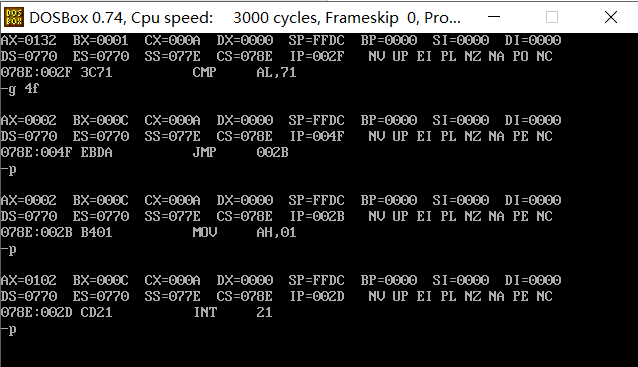




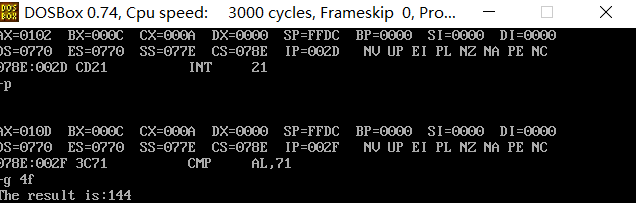
输入1



输入2

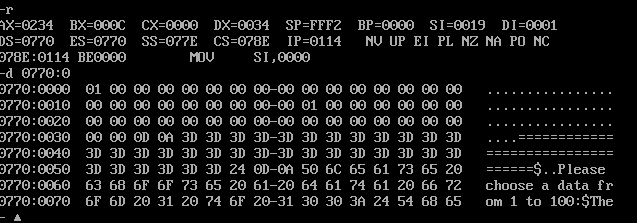


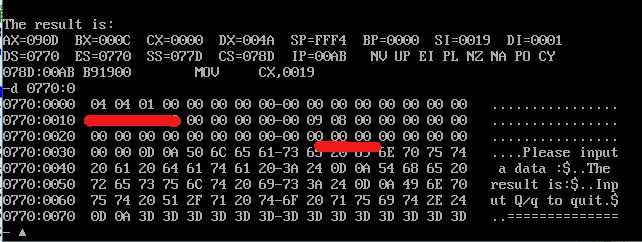
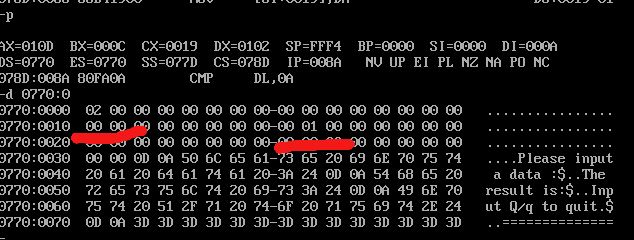
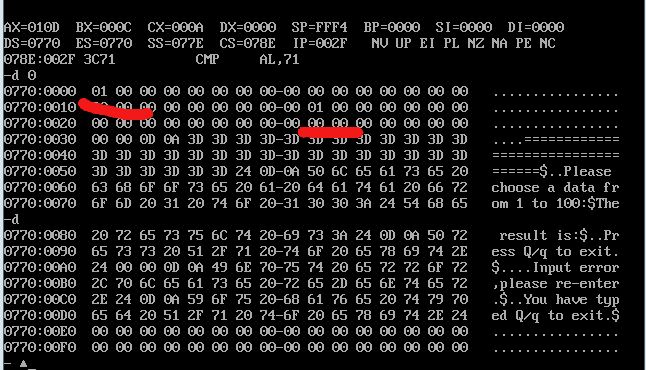
输入回车



结果为144

**以下为数据段中result，x处数据的变化**





**2.3观察现象**

1. 在程序运行前，数据段的变量并没有进入数据段。

2. 在0770：0000为result，0770：0019为x

3. 几次循环之后，数据存入内存。

**2.4实验心得**

我认为本次实验主要是要清楚各分支程序。什么时候该跳转到哪个分支，并在分支中做好各项处理工作是本次实验的重点。

并且此次实验也让我们明白了，汇编的输出都是“字符”，也就是说要想输出两位数字必须做一点处理，将它转换为相应的字符“一个一个”地输出，从而解决了实验中的疑问。

还有一点就是关于输入的问题，我采用的是单字符输入，边输入边处理。因为用字符串的话，输入之后，仍然要“一个一个”字符取出来去做判断，还不如直接就判断了。

1. **程序框图**



1. **程序清单**

**实验代码如下：**

;fibonicci

;南有乔木

datas segment

result db 1d,24 dup(0)

x db 1d,24 dup(0)

mess0 db 0dh,0ah,'==================================','$'

mess1 db 0dh,0ah,'Please choose a data from 1 to 100:','$'

mess2 db 'The result is:','$'

mess3 db 0dh,0ah,'Press Q/q to exit.','$'

flag dw ?

messerror db 0dh,0ah,'Input error,please re-enter.','$'

quit db 0dh,0ah,'You have typed Q/q to exit.','$'

datas ends

stacks segment

dw 128 dup(?)

stacks ends

codes segment

main proc far

assume cs:codes,ds:datas,ss:stacks,es:datas

start::

mov ax,datas ;ds寄存器设置到当前的数据段

mov ds,ax

mov es,ax

mov ax,stacks ;ss寄存器设置到当前堆栈段

mov ss,ax

reinput:

mov ah,09

lea dx,mess0

int 21h

push ax ;进栈

push bx

push cx

push dx

push si

push di

mov ah,09 ;显示操作符

lea dx,mess3

int 21h

lea dx,mess1

int 21h

;输入规模n，并且转换为十进制数

mov bx,0

input:

mov ah,01

int 21h

cmp al,'q' ;判断输入是否为q

jz end\_ ;跳转到end\_

cmp al,'Q'

jz end\_

cmp al,0dh

jz cont

cmp al,'0' ;输入是否为数字

jb error ;跳转到error

cmp al,':'

jnb error

sub al,30h

cbw

xchg ax,bx

mov cx,10d

mul cx

xchg ax,bx

add bx,ax

jmp input

error: ;提示出错信息

mov ah,09

lea dx,messerror

int 21h

pop di ;出栈

pop si

pop dx

pop cx

pop bx

pop ax

jmp reinput ;跳转回开始状态

end\_: ;提示已退出程序信息

mov ah,09

lea dx,quit

int 21h

mov ah,4ch

int 21h

ret

cont: cmp bx,0

jz error

mov cx,bx ;输入的数放到cx中

cmp cx,2

jle call\_f

;如果小于等于2，就直接输出结果

sub cx,2 ;否则以cx-2作为外循环，把两个数相加

next: mov di,cx

mov cx,25

mov si,0

adde: mov dl,x[si]

mov dh,result[si]

add dl,dh ;将两个存储单元的数中进行相加

mov result[si],dl

mov x[si],dh ;将上次的相加结果放入X中

cmp dl,10d

jae great ;如果大于10D，则转移

inc si

jmp goon

great: sub result[si],10d ;将尾数存入本单元

inc si

add x[si],1 ;高位加1

goon: loop adde

mov cx,di

loop next

call\_f:

call fibp

pop di ;出栈

pop si

pop dx

pop cx

pop bx

pop ax

jmp reinput ;跳转回开始状态，进行下一次计算

mov ah,4ch

int 21h

main endp

fibp proc near

print: lea dx,mess2 ;输出显示提示

mov ah,09

int 21h

mov cx,25

mov si,24

display1: cmp flag,0 ;标志位判断输出的高位是否为0

jnz n2

cmp result[si],0

jz n3

add flag,1

n2: mov dl,result[si] ;以十进制输出

add dl,30h

mov ah,02h

int 21h

n3: dec si

loop display1

mov flag,0

mov result[0],1d

mov x[0],1d

mov si,1

mov cx,24

initial: mov result[si],0

mov x[si],0

add si,1

loop initial

mov si,0

ret

fibp endp

codes ends

end start