

汇编语言斐波那契数列

汇编语言实验报告

学 院: ***

班 级: ***

姓 名: ***

学 号: ***

指导老师: **

目 录

<u>实验一 斐波那契数列</u>	5
<u>1.</u>	<u>设计要求</u>
5	
<u>2.</u>	<u>设计思想和实施方案论述</u>
5	
<u>3.</u>	<u>课程设计中遇到的问题及解决方案</u>
7	
<u>4.</u>	<u>程序流程图</u>
8	
<u>5.</u>	<u>源程序及注释</u>
8	
<u>实验二 统计学生成绩</u>	15
<u>1.</u>	<u>设计要求</u>
15	
<u>2.</u>	<u>设计思想和实施方案论述</u>
15	
<u>3.</u>	<u>典型程序模块及典型编程技巧分析</u>
15	
<u>4.</u>	<u>课程设计中遇到的问题及解决方法</u>
17	

<u>5.</u>	<u>程序流程图，程序清单和程序注释</u>	
17		
<u>6.</u>	<u>程序清单和注释</u>	
19		
<u>收获与体会</u>		24
<u>参考书籍:</u>		24

实验一 斐波那契数列

1. 设计要求

用递归的方法求斐波那契的第 24 项 (N=24), 并将结果用十进制显示出来

$$F(N) = \begin{cases} 1, & \text{当 } N=1, 2 \text{ 时} \\ F(N-1) + F(N-2), & \text{当 } N \geq 3 \text{ 时} \end{cases}$$

输出 FIB(24) 的值

2. 设计思想和实施方案论述

FIB 函数采用子程序递归调用方法, 实现求斐波那契数列

先写出高级语言中的斐波那契递归函数:

```
Int fib(int N)
{
    If (N==1 || N==0) result=1;
    Else result=fib(n-1)+fib(n-2);
    Return result;
}
```

其函数为高级语言的形式, 其中 $result=fib(n-1)+fib(n-2)$ 隐含了几个操作可把它展开为:

N=N-1;注意到这里的 N 是形参（局部变量）与调用程序的实参不是同一个变量

```
Int a=fib(N)
```

```
N=N-1
```

```
Int b=fib(N)
```

```
Result = a+b;
```

由此可知几个重要实现地方：

a) 保存现场：N 在一个 fib 函数里面是局部变量，当系统调用子程序时需要保存这个局部变量，在返回时需要恢复：

```
SUBT PROC NEAR
```

```
    PUSH AX
```

```
    PUSH BX
```

```
    .....
```

```
    .....
```

```
    POP BX
```

```
    POP AX
```

```
SUBT ENDP; 采用保存恢复寄存器的形式
```

b) 参数传递：其中的 N 在调用 fib 传参时是作为实参，在被调用函数中还需要被使用，这是参数的传递，这里采用通过数据区传递参数（同全局变量）

c) 返回值：最后的 result 在高级语言中是通过将 result 赋给一个临时变量（调用该函数的程序）然后释放 result 这个局部变量，在本程序中采用寄存器直接传递的方法，所以在 Fib 函数中将 result 赋给一个寄存器，然后不保存它，让其传递到上层调用函数中去。

3. 课程设计中遇到的问题及解决方案

● 遇到的问题

在编写斐波那契函数的过程中主要遇到了问题是保存现场的问题，在高级语言中，对子程序调用时，系统会自动为我们保存当前程序的现场指令地址，局部变量。而在汇编中除了现场指令地址保存外，其他的工作都必须我们自己完成，所以在对子程序调用时，怎么保存当前的局部变量，怎么传递参数到将要调用的函数中去使用，都是很难理顺思路的地方。

● 解决方案

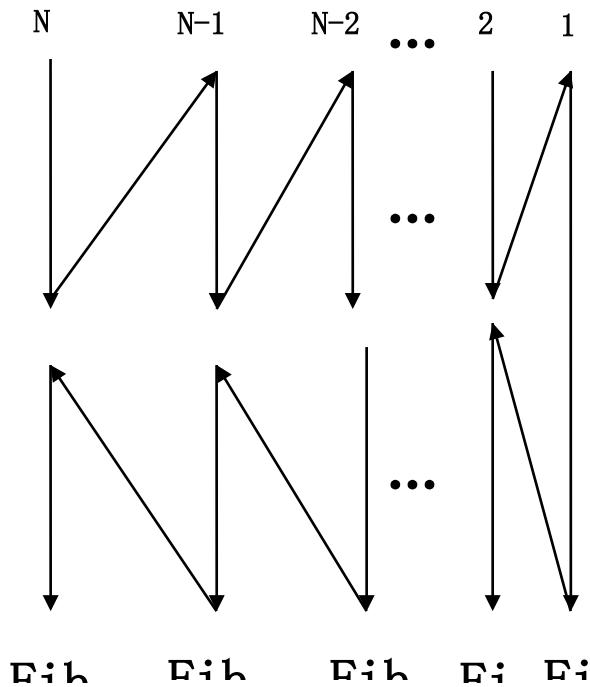
第一：明确哪些是局部变量，在调用完子程序返回时要恢复的。

第二：局部变量作为实参，与被调用的子程序中的形参的区别。

第三：确保有压栈就必须对应出栈，且系统的中断指令地址的保存与我们使用的栈是同一个栈

解决方案在上面已经写过，就没有在复述了

4. 程序流程图



5. 源程序及注释

DATAS SEGMENT

```
message1 db "input the number  
", "$"
```

```
NUM=18h ;修改此处获得第N项斐波那
```

```
契数列值
```


N dw 0 ;用于传参的N

DATAS ENDS

CODES SEGMENT

ASSUME CS:CODES,DS:DATAS

Main proc far

START:

push ds

sub ax,ax

push ax

mov ax,DATAS

mov ds,ax

mov byte ptr N,NUM

call FIB ;调用Fib子程序

mov dx,ax

;call Output_CTLF

call PrintOX ;将十进制转换为十六进制输出

出

mov ah,1

int 21h ;等待退出屏幕

ret

Main endp

; output CTLF

Output_CTLF proc Near

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。

如要下载或阅读全文，请访问：

<https://d.book118.com/617114200153006112>