

姓名：吴涵

学号：202101000720

成绩：

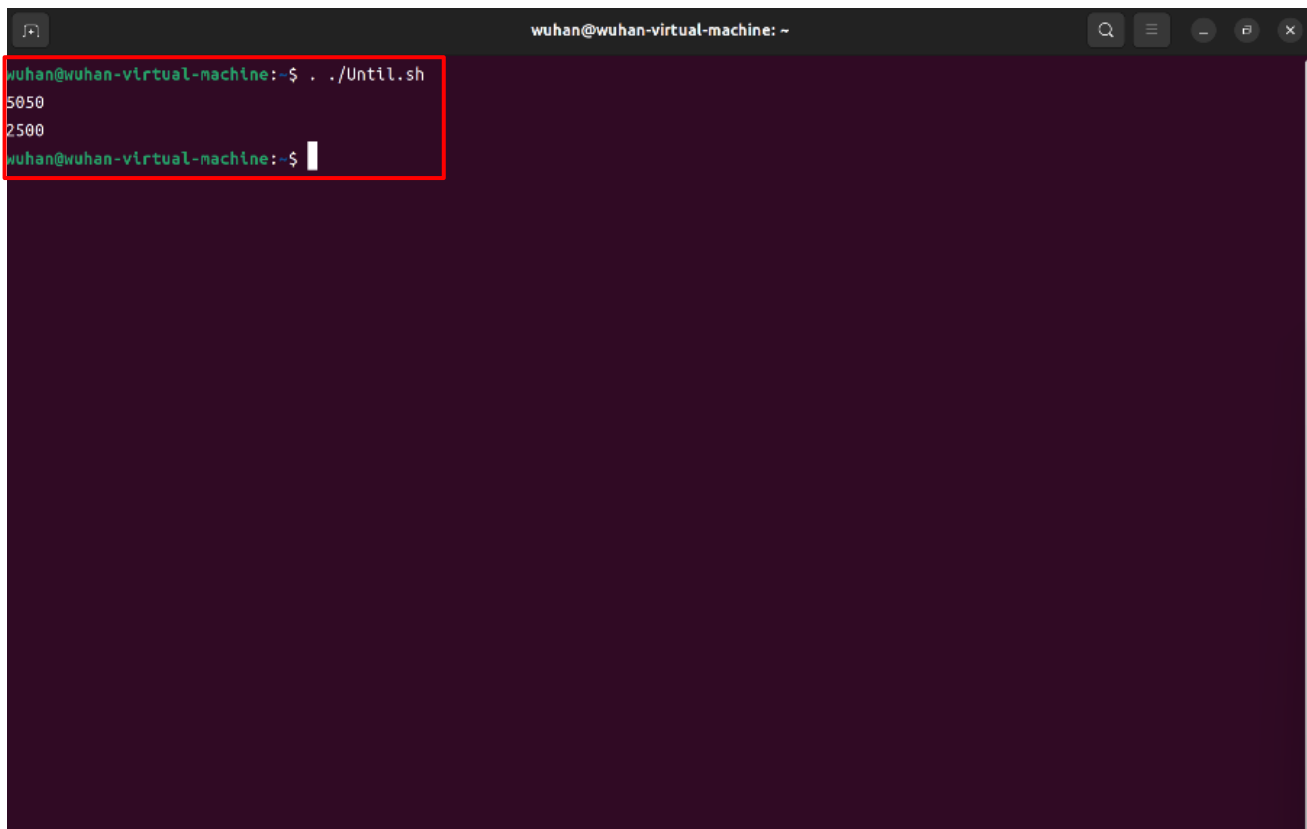
实验 17 批处理操作接口 7: until 循环与 select 循环

1、建立文件 Until.sh 实现累加和脚本：

```
#!/bin/bash
sum01=0
sum02=0
i=1
until [ $i -gt 100 ]
do
    let "sum01+=i"
    let "j=i%2"
    if [ $j -ne 0 ];then
        let "sum02+=i"
    fi
    let "i+=1"
done
echo $sum01
echo $sum02
```

命令：../Until.sh

结果：

A terminal window titled 'wuhan@wuhan-virtual-machine: ~' with a dark background. The prompt is 'wuhan@wuhan-virtual-machine:~\$'. The user enters './Until.sh'. The script outputs '5050' on the first line and '2500' on the second line. The prompt returns to 'wuhan@wuhan-virtual-machine:~\$'. A red rectangle highlights the command and its output.

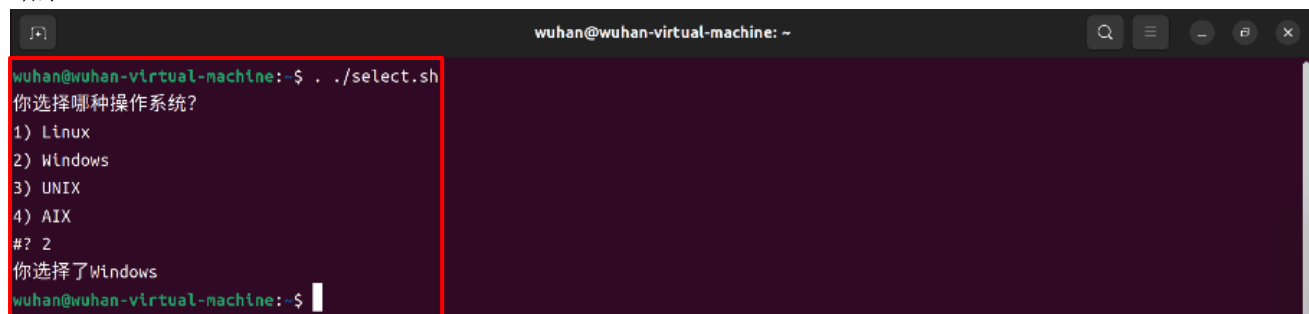
```
wuhan@wuhan-virtual-machine:~$ ./Until.sh
5050
2500
wuhan@wuhan-virtual-machine:~$
```


4、建立文件 select.sh，给出操作系统类型菜单供用户选择脚本：

```
#!/bin/bash
echo "你选择哪种操作系统？"
select OS in Linux Windows UNIX AIX
do
    break
done
echo "你选择了$OS"
```

命令： ./select.sh

结果：



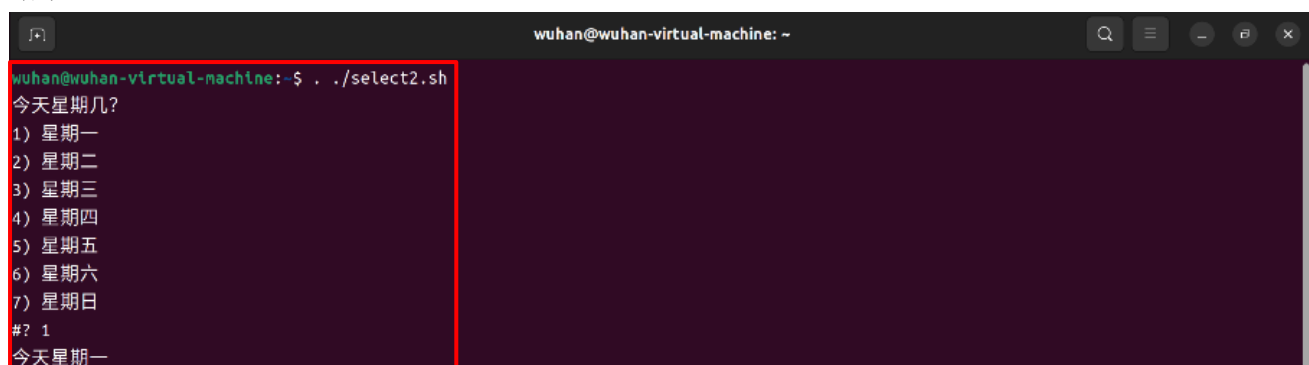
```
wuhan@wuhan-virtual-machine: ~
wuhan@wuhan-virtual-machine:~$ ./select.sh
你选择哪种操作系统?
1) Linux
2) Windows
3) UNIX
4) AIX
#? 2
你选择了Windows
wuhan@wuhan-virtual-machine:~$
```

5、建立文件 select2.sh，显示星期数供用户选择脚本：

```
#!/bin/bash
echo "今天星期几？"
select DAY in 星期一 星期二 星期三 星期四 星期五 星期六 星期日
do
    case $DAY in
        星期一) echo "今天星期一";;
        星期二) echo "今天星期二";;
        星期三) echo "今天星期三";;
        星期四) echo "今天星期四";;
        星期五) echo "今天星期五";;
        星期六|星期日) echo "今天双休日";;
        *) echo "未知输入，程序退出" && break;;
    esac
done
```

命令： ./select2.sh

结果：



```
wuhan@wuhan-virtual-machine: ~
wuhan@wuhan-virtual-machine:~$ ./select2.sh
今天星期几?
1) 星期一
2) 星期二
3) 星期三
4) 星期四
5) 星期五
6) 星期六
7) 星期日
#? 1
今天星期一
```

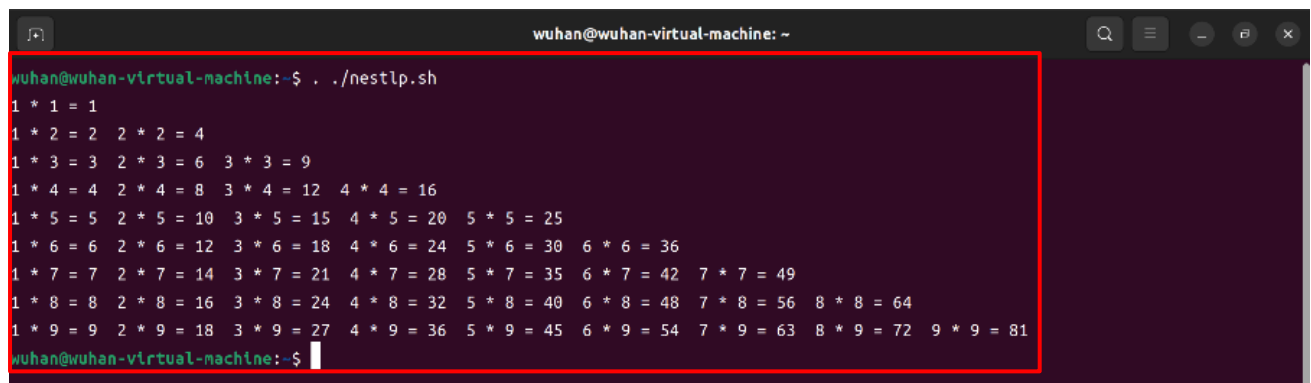
6、使用多重循环打印乘法表

脚本:

```
#!/bin/bash
for ((i=1;i<=9;i++))
do
    for ((j=1;j<=i;j++))
    do
        let "multi = $j * $i"
        echo -n "$j * $i = $multi  "
    done
    echo
done
```

命令: `./nestlp.sh`

结果:

A terminal window titled 'wuhan@wuhan-virtual-machine: ~' showing the execution of the script './nestlp.sh'. The output is a 9x9 multiplication table. The first row is '1 * 1 = 1'. The second row is '1 * 2 = 2 2 * 2 = 4'. The third row is '1 * 3 = 3 2 * 3 = 6 3 * 3 = 9'. The fourth row is '1 * 4 = 4 2 * 4 = 8 3 * 4 = 12 4 * 4 = 16'. The fifth row is '1 * 5 = 5 2 * 5 = 10 3 * 5 = 15 4 * 5 = 20 5 * 5 = 25'. The sixth row is '1 * 6 = 6 2 * 6 = 12 3 * 6 = 18 4 * 6 = 24 5 * 6 = 30 6 * 6 = 36'. The seventh row is '1 * 7 = 7 2 * 7 = 14 3 * 7 = 21 4 * 7 = 28 5 * 7 = 35 6 * 7 = 42 7 * 7 = 49'. The eighth row is '1 * 8 = 8 2 * 8 = 16 3 * 8 = 24 4 * 8 = 32 5 * 8 = 40 6 * 8 = 48 7 * 8 = 56 8 * 8 = 64'. The ninth row is '1 * 9 = 9 2 * 9 = 18 3 * 9 = 27 4 * 9 = 36 5 * 9 = 45 6 * 9 = 54 7 * 9 = 63 8 * 9 = 72 9 * 9 = 81'. The prompt 'wuhan@wuhan-virtual-machine:~\$' is visible at the bottom.

```
wuhan@wuhan-virtual-machine:~$ ./nestlp.sh
1 * 1 = 1
1 * 2 = 2 2 * 2 = 4
1 * 3 = 3 2 * 3 = 6 3 * 3 = 9
1 * 4 = 4 2 * 4 = 8 3 * 4 = 12 4 * 4 = 16
1 * 5 = 5 2 * 5 = 10 3 * 5 = 15 4 * 5 = 20 5 * 5 = 25
1 * 6 = 6 2 * 6 = 12 3 * 6 = 18 4 * 6 = 24 5 * 6 = 30 6 * 6 = 36
1 * 7 = 7 2 * 7 = 14 3 * 7 = 21 4 * 7 = 28 5 * 7 = 35 6 * 7 = 42 7 * 7 = 49
1 * 8 = 8 2 * 8 = 16 3 * 8 = 24 4 * 8 = 32 5 * 8 = 40 6 * 8 = 48 7 * 8 = 56 8 * 8 = 64
1 * 9 = 9 2 * 9 = 18 3 * 9 = 27 4 * 9 = 36 5 * 9 = 45 6 * 9 = 54 7 * 9 = 63 8 * 9 = 72 9 * 9 = 81
wuhan@wuhan-virtual-machine:~$
```

实验 18 批处理操作接口 8：函数

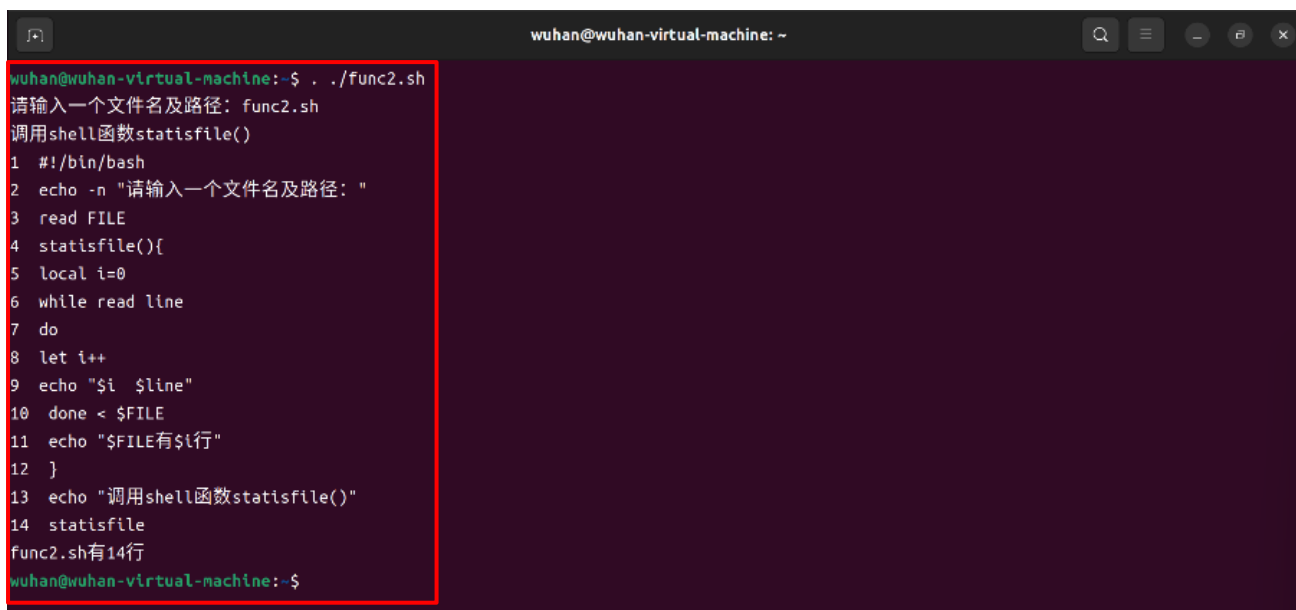
1、建立文件 func2.sh，输出文件内容各行及行数

脚本：

```
#!/bin/bash
echo -n "请输入一个文件名及路径: "
read FILE
statisfile(){
    local i=0
    while read line
    do
        let i++
        echo "$i $line"
    done < $FILE
    echo "$FILE 有$i 行"
}
echo "调用 shell 函数 statisfile()"
statisfile
```

命令：../func2.sh

结果：



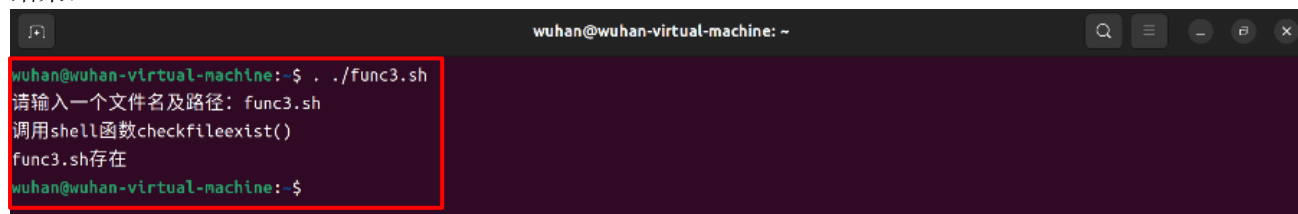
```
wuhan@wuhan-virtual-machine: ~
wuhan@wuhan-virtual-machine:~$ ./func2.sh
请输入一个文件名及路径: func2.sh
调用shell函数statisfile()
1  #!/bin/bash
2  echo -n "请输入一个文件名及路径: "
3  read FILE
4  statisfile(){
5  local i=0
6  while read line
7  do
8  let i++
9  echo "$i $line"
10 done < $FILE
11 echo "$FILE有$i行"
12 }
13 echo "调用shell函数statisfile()"
14 statisfile
func2.sh有14行
wuhan@wuhan-virtual-machine:~$
```

2、建立文件 func3.sh，判断文件是否存在
脚本：

```
#!/bin/bash
echo -n "请输入一个文件名及路径: "
read FILE
checkfileexist(){
    if [ -f $FILE ];then
        return 0
    else
        return 1
    fi
}
echo "调用 shell 函数 checkfileexist()"
checkfileexist
if [ $? -eq 0 ];then
    echo "$FILE 存在"
else
    echo "$FILE 不存在"
fi
```

命令: `../func3.sh`

结果：



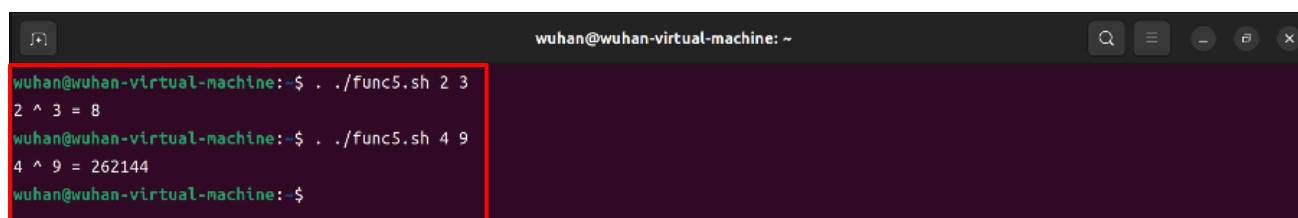
```
wuhan@wuhan-virtual-machine: ~
wuhan@wuhan-virtual-machine:~$ ./func3.sh
请输入一个文件名及路径: func3.sh
调用shell函数checkfileexist()
func3.sh存在
wuhan@wuhan-virtual-machine:~$
```

3、建立文件 func5.sh，计算阶乘
脚本：

```
#!/bin/bash
power(){
    p=1
    i=0
    while [ "$i" -lt $2 ]
    do
        let "p=p*$1"
        let "i=i+1"
    done
    echo "$1 ^ $2 = $p"
}
power $1 $2
```

命令: `../func5.sh 2 3`

结果：



```
wuhan@wuhan-virtual-machine: ~
wuhan@wuhan-virtual-machine:~$ ./func5.sh 2 3
2 ^ 3 = 8
wuhan@wuhan-virtual-machine:~$ ./func5.sh 4 9
4 ^ 9 = 262144
wuhan@wuhan-virtual-machine:~$
```

4、指定位置参数值

脚本:

```
#!/bin/bash
echo "重新设置位置参数值前各位置参数的值: "
cnt=1
for i in $@
do
    echo "\$$cnt=$i"
    let "cnt++"
done
set 1 2 Linux Windows
echo "重新设置位置参数值后各位置参数的值: "
cnt=1
for i in $@
do
    echo "\$$cnt=$i"
    let "cnt++"
done
```

命令: `./set1.sh a b c d 1 2`

结果:



```
wuhan@wuhan-virtual-machine: ~
wuhan@wuhan-virtual-machine:~$ ./set1.sh a b c d 1 2
重新设置位置参数值前各位置参数的值:
$1=a
$2=b
$3=c
$4=d
$5=1
$6=2
重新设置位置参数值后各位置参数的值:
$1=1
$2=2
$3=Linux
$4=Windows
wuhan@wuhan-virtual-machine:~$
```

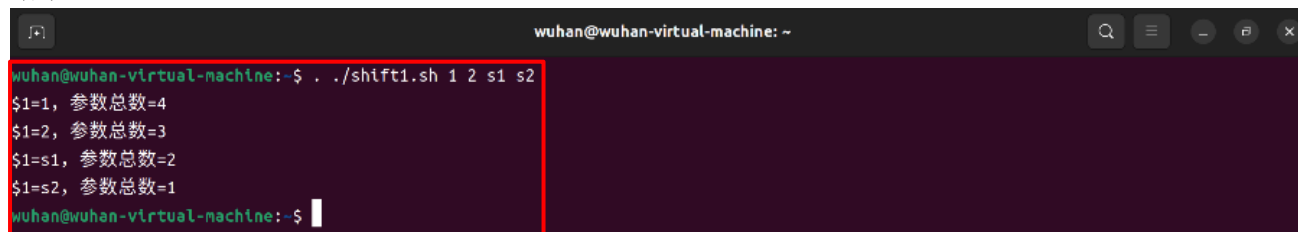
5、给出下列脚本执行结果

脚本:

```
#!/bin/bash
until [ $# -eq 0 ]
do
    echo "\$1=$1, 参数总数=$#"
    shift
done
```

命令: `./shift1.sh 1 2 s1 s2`

结果:



```
wuhan@wuhan-virtual-machine: ~
wuhan@wuhan-virtual-machine:~$ ./shift1.sh 1 2 s1 s2
$1=1, 参数总数=4
$1=2, 参数总数=3
$1=s1, 参数总数=2
$1=s2, 参数总数=1
wuhan@wuhan-virtual-machine:~$
```

6、给出下列脚本执行结果

脚本:

```
#!/bin/bash
```

```
until [ $# -le 1 ]
```

```
do
```

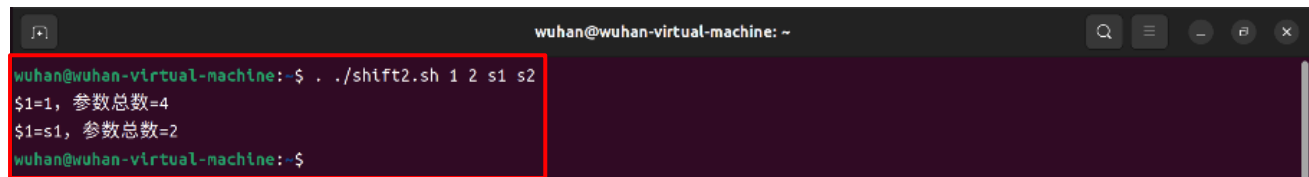
```
    echo "\$1=$1, 参数总数=$#"
```

```
    shift 2
```

```
done
```

命令: ../shift2.sh 1 2 s1 s2

结果:



```
wuhan@wuhan-virtual-machine: ~  
wuhan@wuhan-virtual-machine:~$ ./shift2.sh 1 2 s1 s2  
$1=1, 参数总数=4  
$1=s1, 参数总数=2  
wuhan@wuhan-virtual-machine:~$
```

7、给出下列脚本执行结果

脚本:

```
#!/bin/bash
```

```
Total=0
```

```
expstr=""
```

```
until [ $# -eq 0 ]
```

```
do
```

```
    let "Total=Total+$1"
```

```
    if [ "$expstr" = "" ];then
```

```
        expstr=$1
```

```
    else
```

```
        expstr=$expstr"+"$1"
```

```
    fi
```

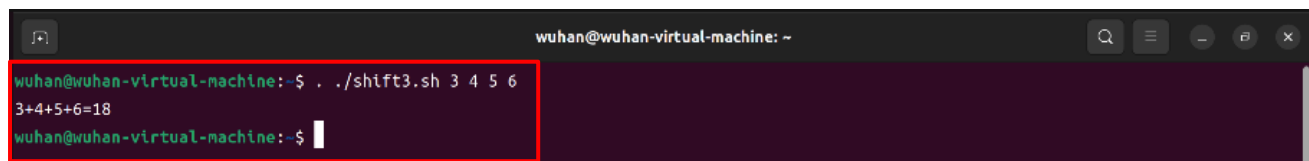
```
    shift
```

```
done
```

```
echo $expstr="$Total"
```

命令: ../shift3.sh 3 4 5 6

结果:



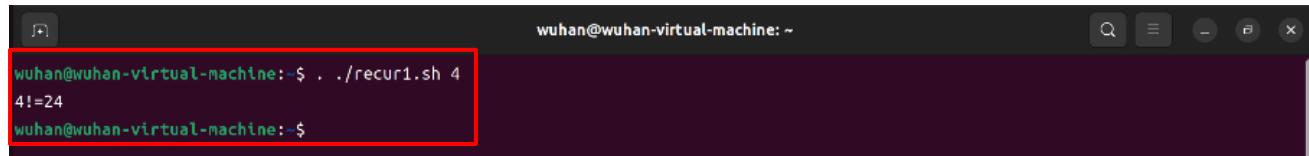
```
wuhan@wuhan-virtual-machine: ~  
wuhan@wuhan-virtual-machine:~$ ./shift3.sh 3 4 5 6  
3+4+5+6=18  
wuhan@wuhan-virtual-machine:~$
```


8、建立递归函数文件 recur1.sh 求阶乘脚本：

```
#!/bin/bash
fac(){
    local n=$1
    if [[ $n -le 0 ]];then
        f=1
    else
        fac $((n-1))
        t=$f
        n=$n
        f=$((n*t))
    fi
}
fac $1
echo "$1!=$f"
```

命令：../recur1.sh 4

结果：

A terminal window titled 'wuhan@wuhan-virtual-machine: ~' with search, menu, and window control icons. The terminal shows the command 'wuhan@wuhan-virtual-machine:~\$./recur1.sh 4' and its output '4!=24'. The command and output lines are highlighted with a red rectangular box.

```
wuhan@wuhan-virtual-machine:~$ ./recur1.sh 4
4!=24
wuhan@wuhan-virtual-machine:~$
```