实验 6 shell 编程

一、 实验目的

1、 熟悉 shell 编程语法。

2、 掌握 shell 编程基本技巧。

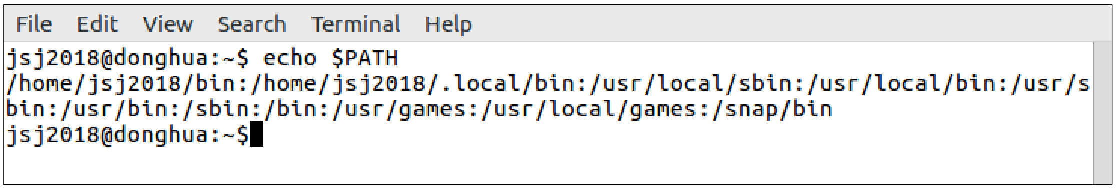
二、 预备知识

查找并学习 bash、sed、grep、awk 脚本编程语法。

三、 实验内容

注意：采用 bash 编程，以下内容中“XXX”为“你的姓名汉语拼音首字母”。

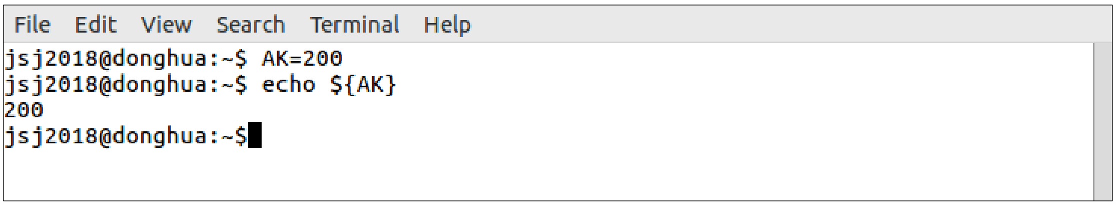
1、 查看当前系统下用户 shell 定义的环境变量的值，如图 E6-1 所示。



```
jsj2018@donghua:~$ echo $PATH
/home/jsj2018/bin:/home/jsj2018/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/s
bin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
jsj2018@donghua:~$
```

图 E6-1 查看 PATH 环境变量

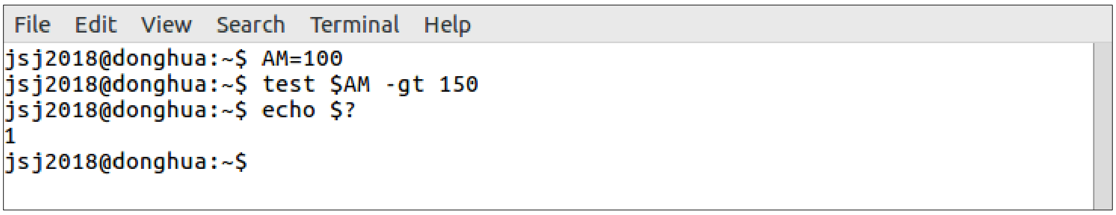
2、 定义变量 AK 的值为 200，并将其显示在屏幕上，如图 E6-2 所示。



```
jsj2018@donghua:~$ AK=200
jsj2018@donghua:~$ echo ${AK}
200
jsj2018@donghua:~$
```

图 E6-2 设置变量

3、 定义变量 AM 的值为 100，并使用 test 命令比较其值是否大于 150，如图 E6-3 所示。



```
jsj2018@donghua:~$ AM=100
jsj2018@donghua:~$ test $AM -gt 150
jsj2018@donghua:~$ echo $?
1
jsj2018@donghua:~$
```

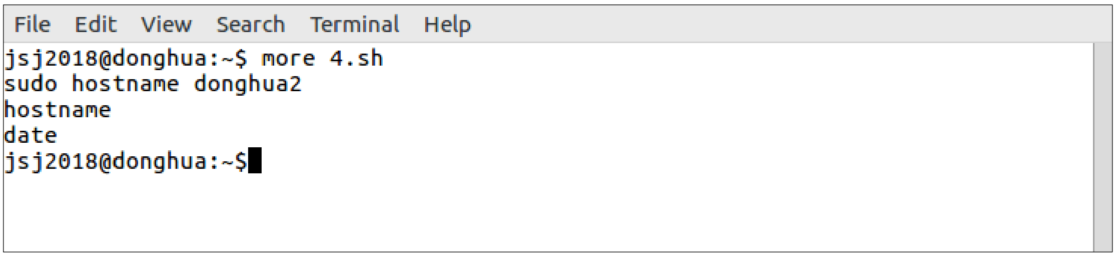
图 E6-3 test 命令操作

4、 创建一个简单的 shell 程序，其功能包括：

（1）将主机名改为你的名字汉语拼音字母；

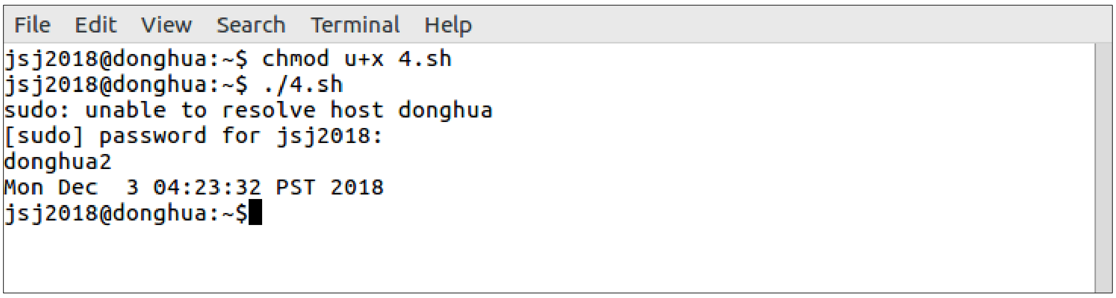
（2）显示计算机主机名以及显示系统日期和时间。

提示：通过“vi 4.sh”编辑shell程序中的内容，编辑完之后退出并保存，可通过more查看。



```
jsj2018@donghua:~$ more 4.sh
sudo hostname donghua2
hostname
date
jsj2018@donghua:~$
```

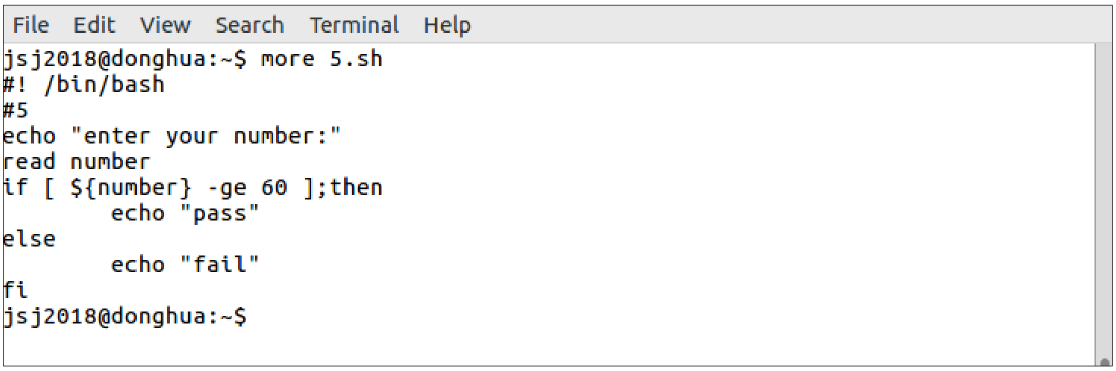
图E6-4 修改主机名的shell程序



```
jsj2018@donghua:~$ chmod u+x 4.sh
jsj2018@donghua:~$ ./4.sh
sudo: unable to resolve host donghua
[sudo] password for jsj2018:
donghua2
Mon Dec  3 04:23:32 PST 2018
jsj2018@donghua:~$
```

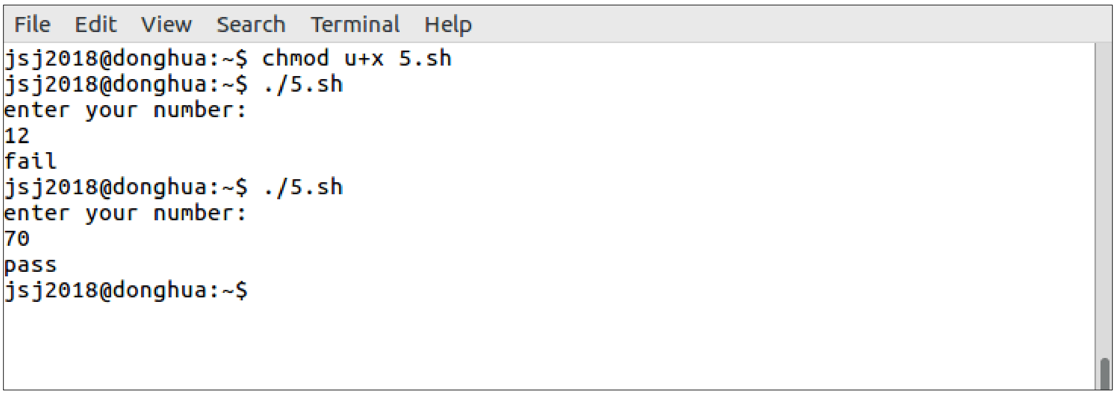
图E6-5 修改主机名的shell程序运行结果

5、 使用 if-then-else 语句创建一个根据输入的分数判断分数是否及格的 shell 程序，如图 E6-6 和 E6-7 所示。



```
jsj2018@donghua:~$ more 5.sh
#! /bin/bash
#5
echo "enter your number:"
read number
if [ ${number} -ge 60 ];then
        echo "pass"
else
        echo "fail"
fi
jsj2018@donghua:~$
```

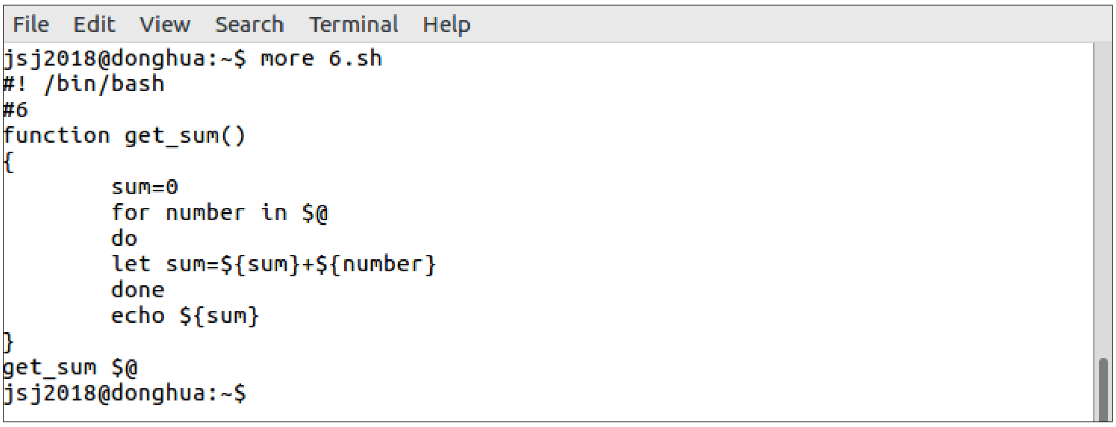
图 E6-6 条件语句使用



```
jsj2018@donghua:~$ chmod u+x 5.sh
jsj2018@donghua:~$ ./5.sh
enter your number:
12
fail
jsj2018@donghua:~$ ./5.sh
enter your number:
70
pass
jsj2018@donghua:~$
```

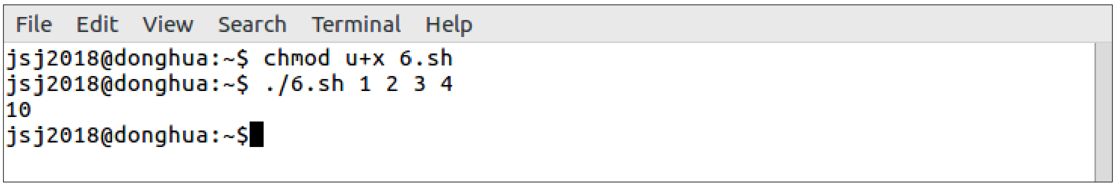
图 E6-7 程序运行测试

6、 使用 for 语句创建求命令行上所有整数之和的 shell 程序，如图 E6-8 和图 E6-9 所示。



```
jsj2018@donghua:~$ more 6.sh
#! /bin/bash
#6
function get_sum()
{
        sum=0
        for number in $@
        do
        let sum=${sum}+${number}
        done
        echo ${sum}
}
get_sum $@
jsj2018@donghua:~$
```

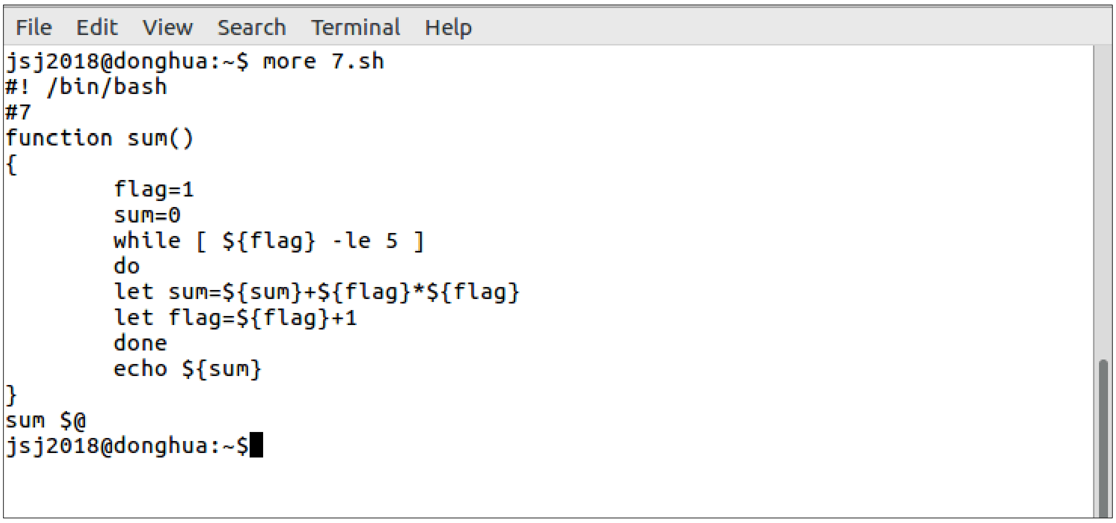
图 E6-8 for 语句使用



```
jsj2018@donghua:~$ chmod u+x 6.sh
jsj2018@donghua:~$ ./6.sh 1 2 3 4
10
jsj2018@donghua:~$
```

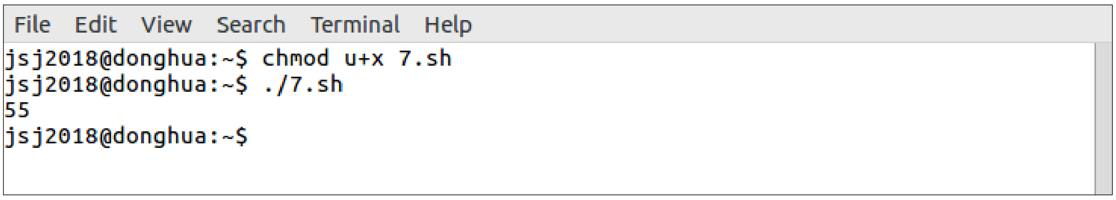
图 E6-9 程序运行测试

7、 使用 while 语句创建一个计算 1 到 5 的平方的 shell 程序，如图 E6-10 和 E6-11 所示。



```
jsj2018@donghua:~$ more 7.sh
#! /bin/bash
#7
function sum()
{
        flag=1
        sum=0
        while [ ${flag} -le 5 ]
        do
        let sum=${sum}+${flag}*${flag}
        let flag=${flag}+1
        done
        echo ${sum}
}
sum $@
jsj2018@donghua:~$
```

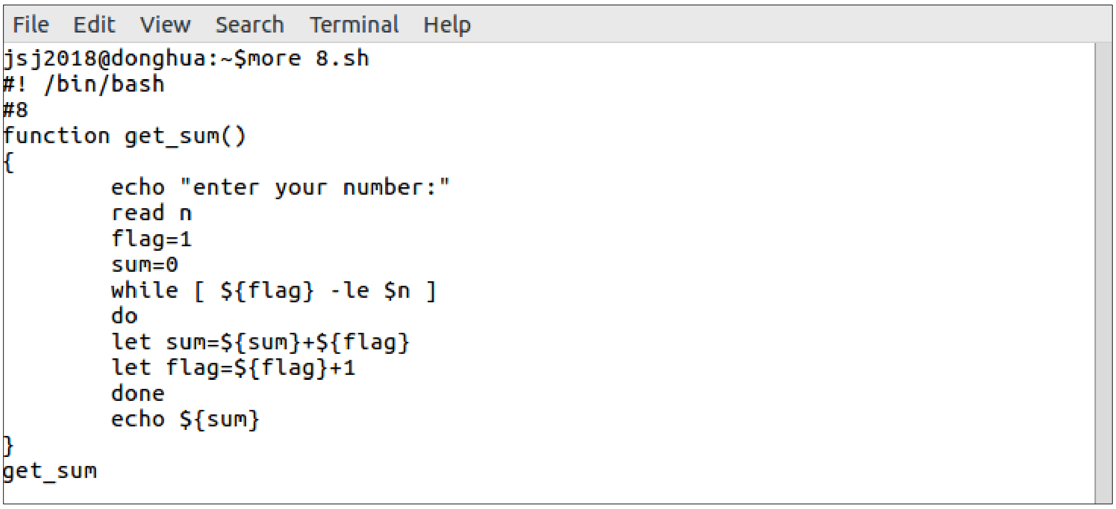
图 E6-10 while 语句使用之一



```
jsj2018@donghua:~$ chmod u+x 7.sh
jsj2018@donghua:~$ ./7.sh
55
jsj2018@donghua:~$
```

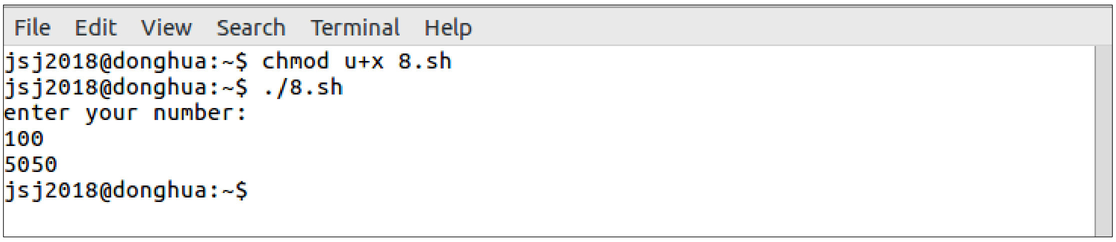
图 E6-11 程序运行测试

8、 使用 while 语句创建一个根据输入的数值 n 求累加和(1+2+3+4+…+n)的 shell 程序，如图 E6-12 和图 E6-13 所示。



```
jsj2018@donghua:~$more 8.sh
#! /bin/bash
#8
function get_sum()
{
        echo "enter your number:"
        read n
        flag=1
        sum=0
        while [ ${flag} -le $n ]
        do
        let sum=${sum}+${flag}
        let flag=${flag}+1
        done
        echo ${sum}
}
get_sum
```

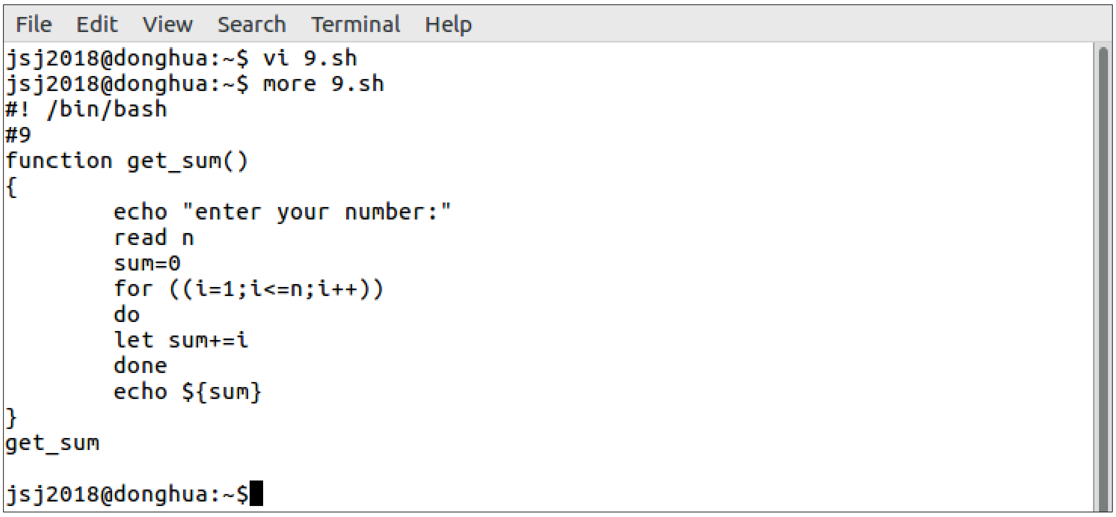
图 E6-12 while 语句使用之二



```
jsj2018@donghua:~$ chmod u+x 8.sh
jsj2018@donghua:~$ ./8.sh
enter your number:
100
5050
jsj2018@donghua:~$
```

图 E6-13 程序运行测试

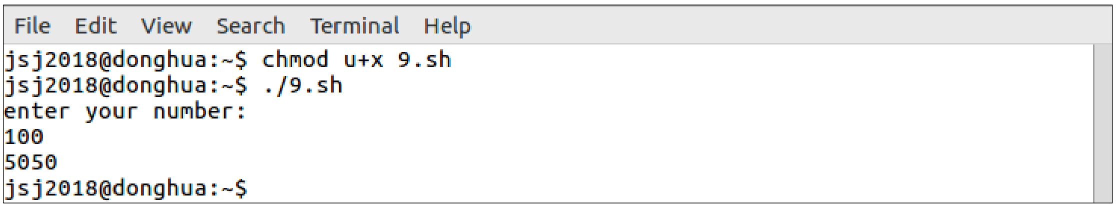
9、 使用 for 语句创建一个 shell 程序，其功能为 1+2+3+4+5+…+n，如图 E6-14 和图 E6-15 所示。



```
jsj2018@donghua:~$ vi 9.sh
jsj2018@donghua:~$ more 9.sh
#! /bin/bash
#9
function get_sum()
{
        echo "enter your number:"
        read n
        sum=0
        for ((i=1;i<=n;i++))
        do
        let sum+=i
        done
        echo ${sum}
}
get_sum

jsj2018@donghua:~$
```

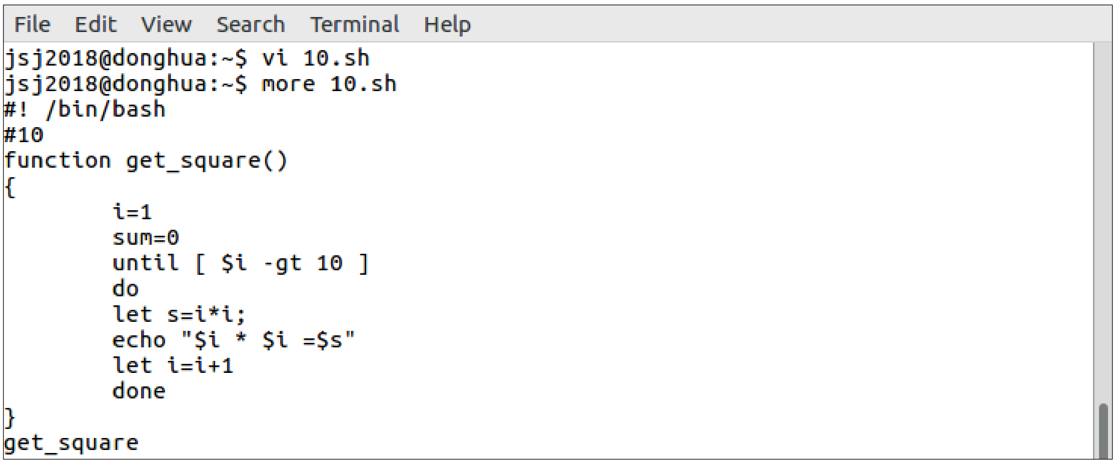
图 E6-14 for 语句使用



```
jsj2018@donghua:~$ chmod u+x 9.sh
jsj2018@donghua:~$ ./9.sh
enter your number:
100
5050
jsj2018@donghua:~$
```

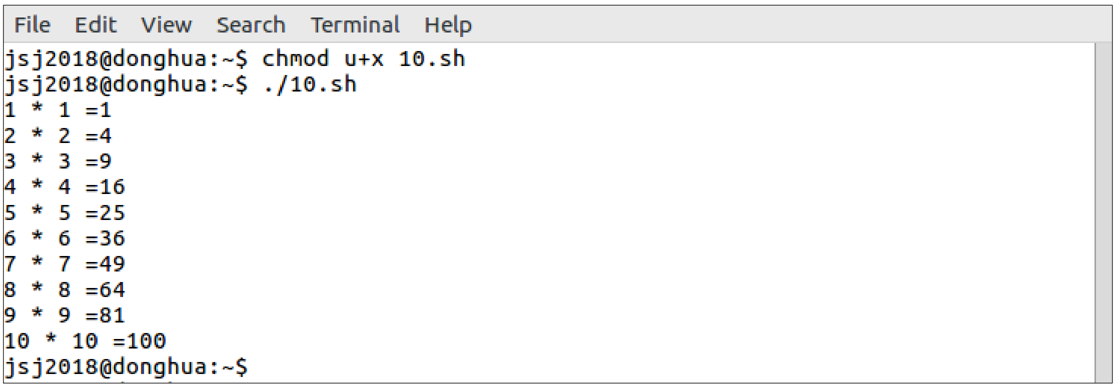
图 E6-15 程序运行测试

10、使用 until 语句创建一个 shell 程序，其功能为计算 1~10 的平方，如图 E6-16 和图 E6-17 所示。



```
jsj2018@donghua:~$ vi 10.sh
jsj2018@donghua:~$ more 10.sh
#! /bin/bash
#10
function get_square()
{
        i=1
        sum=0
        until [ $i -gt 10 ]
        do
        let s=i*i;
        echo "$i * $i =$s"
        let i=i+1
        done
}
get_square
```

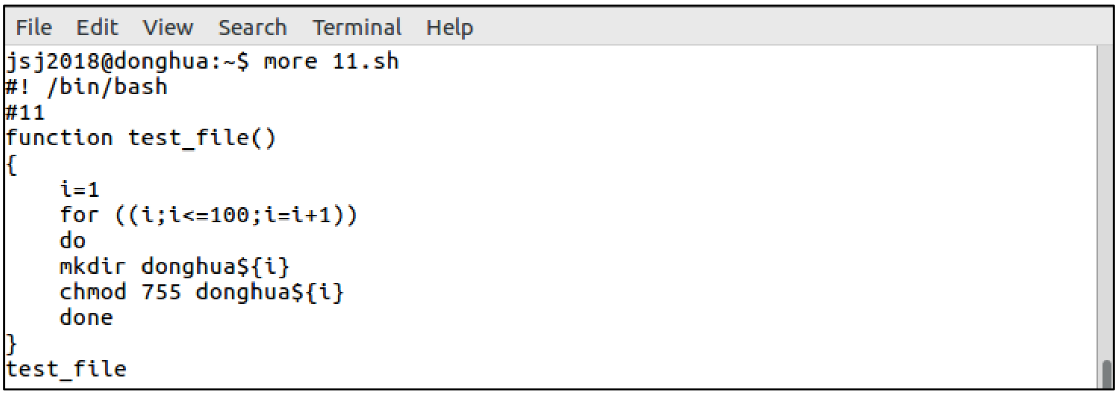
图 E6-16 until 语句使用



```
jsj2018@donghua:~$ chmod u+x 10.sh
jsj2018@donghua:~$ ./10.sh
1 * 1 =1
2 * 2 =4
3 * 3 =9
4 * 4 =16
5 * 5 =25
6 * 6 =36
7 * 7 =49
8 * 8 =64
9 * 9 =81
10 * 10 =100
jsj2018@donghua:~$
```

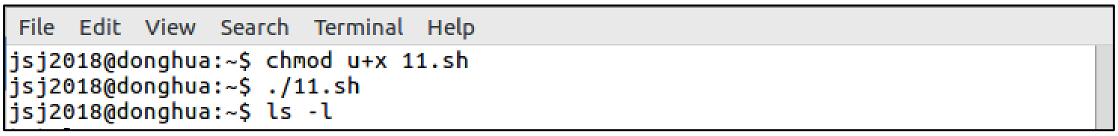
图 E6-17 程序运行测试

11、 设计一个 shell 程序，在/home 目录下建立 100 个目录，即 XXX1～XXX100，并设置每个目录的权限，其中文件所有者的权限为：读、写、执行；文件所有者所在组的权限为：读、执行；其他用户的权限为：读、执行；可通过 ls -l 的命令查看。如图 E6-18 和 E6-19 所示。

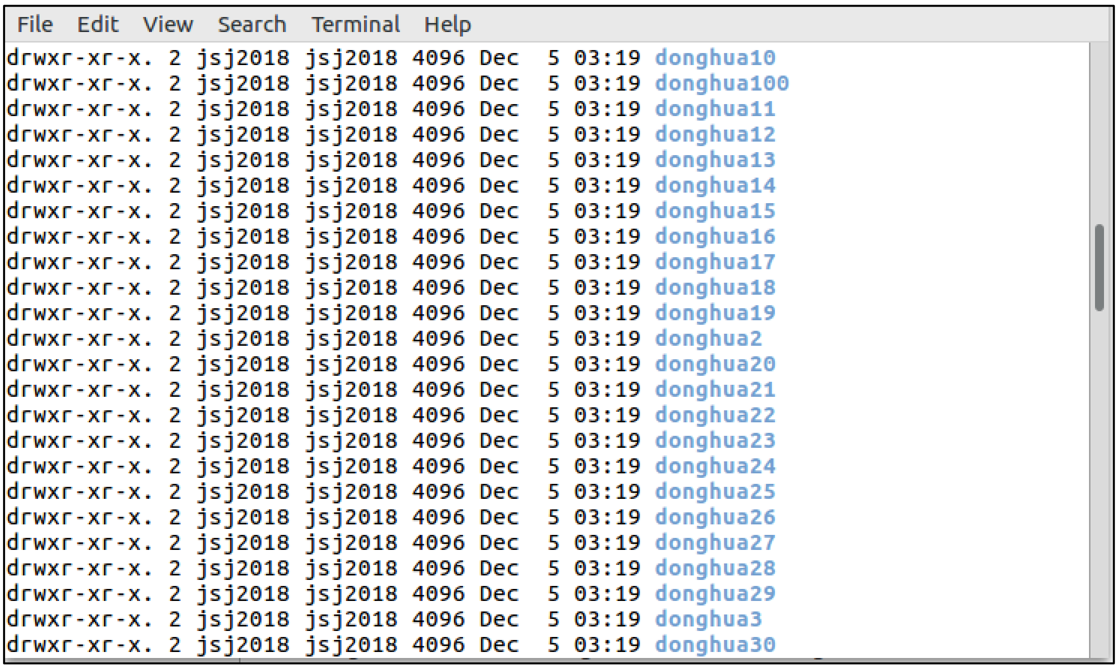


```
jsj2018@donghua:~$ more 11.sh
#! /bin/bash
#11
function test_file()
{
    i=1
    for ((i;i<=100;i=i+1))
    do
    mkdir donghua${i}
    chmod 755 donghua${i}
    done
}
test_file
```

图 E6-18 shell 程序示例



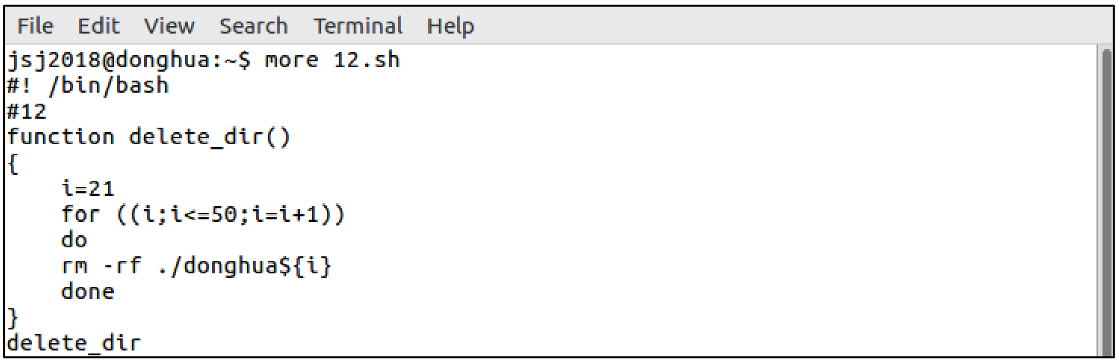
```
jsj2018@donghua:~$ chmod u+x 11.sh
jsj2018@donghua:~$ ./11.sh
jsj2018@donghua:~$ ls -l
```



```
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua10
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua100
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua11
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua12
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua13
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua14
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua15
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua16
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua17
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua18
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua19
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua2
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua20
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua21
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua22
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua23
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua24
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua25
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua26
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua27
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua28
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua29
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua3
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua30
```

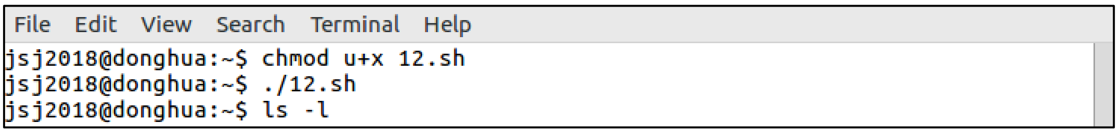
图 E6-19 程序运行测试

12、编写 shell 程序，实现自动删除第 11 题中 30 个目录的功能。目录名为 XXX21 至 XXX50；如图 E6-20 和图 E6-21 所示。

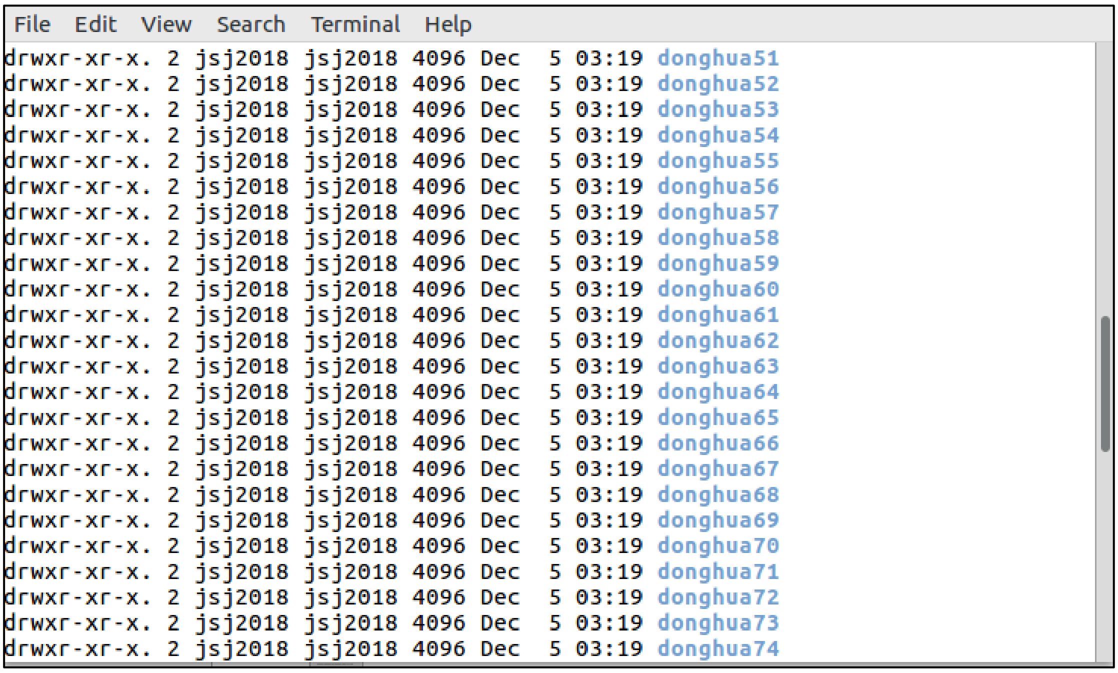


```
jsj2018@donghua:~$ more 12.sh
#! /bin/bash
#12
function delete_dir()
{
    i=21
    for ((i;i<=50;i=i+1))
    do
    rm -rf ./donghua${i}
    done
}
delete_dir
```

图 E6-20 shell 程序示例



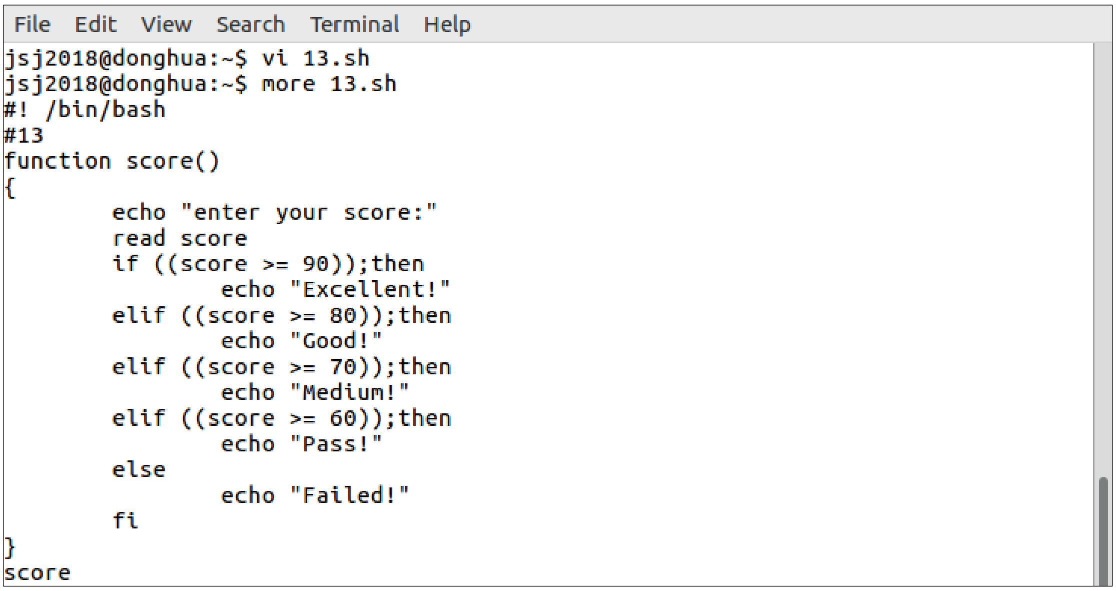
```
jsj2018@donghua:~$ chmod u+x 12.sh
jsj2018@donghua:~$ ./12.sh
jsj2018@donghua:~$ ls -l
```



```
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua51
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua52
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua53
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua54
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua55
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua56
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua57
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua58
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua59
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua60
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua61
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua62
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua63
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua64
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua65
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua66
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua67
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua68
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua69
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua70
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua71
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua72
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua73
drwxr-xr-x. 2 jsj2018 jsj2018 4096 Dec  5 03:19 donghua74
```

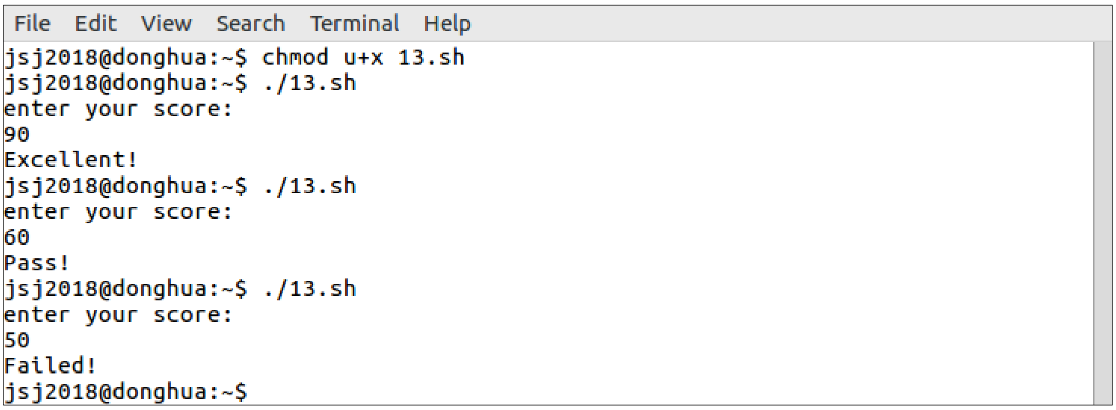
图 E6-21 程序运行测试

13、编写一个 shell 程序，要求：根据从键盘输入的学生成绩，显示相应的成绩等级，其中 60 分以下为“Failed！”，60-69 分为“Passed！”，70-79 分为“Medium！”，80-89 分为“Good！”，90-100 为“Excellent！”，如图 E6-22 和图 E6-23 所示。



```
jsj2018@donghua:~$ vi 13.sh
jsj2018@donghua:~$ more 13.sh
#! /bin/bash
#13
function score()
{
        echo "enter your score:"
        read score
        if ((score >= 90));then
                echo "Excellent!"
        elif ((score >= 80));then
                echo "Good!"
        elif ((score >= 70));then
                echo "Medium!"
        elif ((score >= 60));then
                echo "Pass!"
        else
                echo "Failed!"
        fi
}
score
```

图 E6-22 shell 程序示例



```
jsj2018@donghua:~$ chmod u+x 13.sh
jsj2018@donghua:~$ ./13.sh
enter your score:
90
Excellent!
jsj2018@donghua:~$ ./13.sh
enter your score:
60
Pass!
jsj2018@donghua:~$ ./13.sh
enter your score:
50
Failed!
jsj2018@donghua:~$
```

图 E6-23 程序运行测试

14、某系统管理员每天需做一定的重复工作，请按照下列要求，编制一个解决方案：

（1）在下午4 :50 删除/abc 目录下的全部子目录和全部文件；

（2）每逢周一下午5:50 将/data 目录下的所有目录和文件归档并压缩为文件：backup.tar.gz；

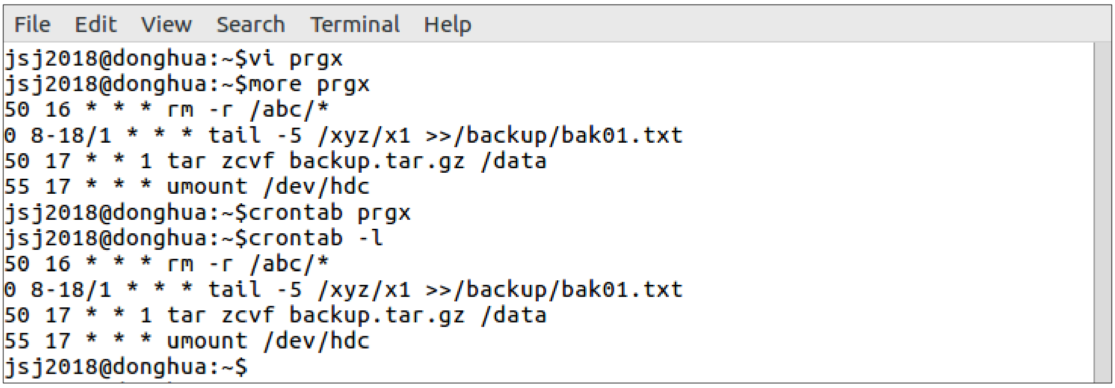
（3）在下午5:55 将IDE 接口的CD-ROM 卸载（假设：CD-ROM 的设备名为hdc）；

（4）在早晨开机后启动。

用 vi 创建编辑一个名为 prgx 的 crontab 文件；prgx 文件内容如下：

```
50  16  *  *  *  rm  -r  /abc/*
0  8—18/1  *  *  *  tail  -5  /xyz/x1  >>  /backup/bak01.txt
50  17  *  *  1  tar  zcvf  backup.tar.gz  /data
55  17  *  *  *  umount  /dev/hdc
```

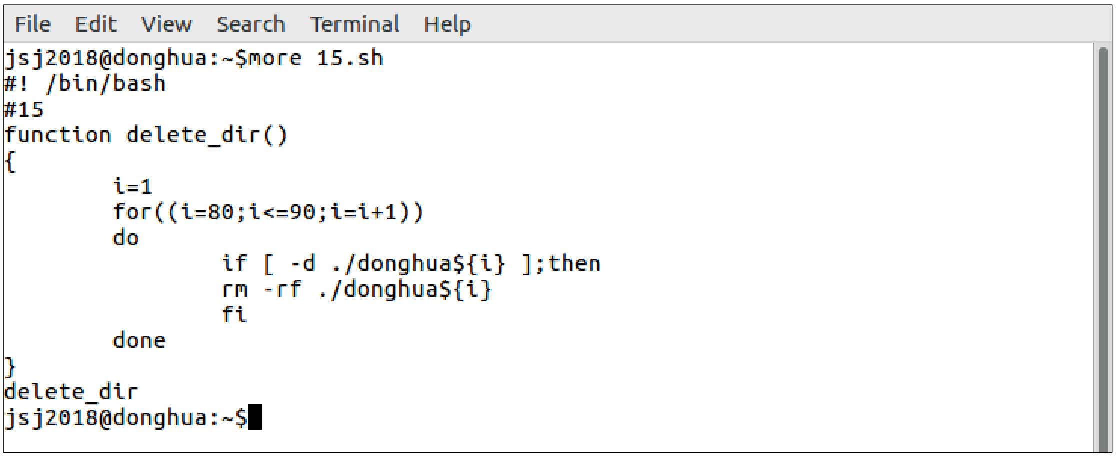
然后执行命令 crontab prgx 将文件提交给 cron 进程，最后使用 crontab -l 命令列出该 crontab 文件。



```
jsj2018@donghua:~$vi prgx
jsj2018@donghua:~$more prgx
50 16 * * * rm -r /abc/*
0 8-18/1 * * * tail -5 /xyz/x1 >>/backup/bak01.txt
50 17 * * 1 tar zcvf backup.tar.gz /data
55 17 * * * umount /dev/hdc
jsj2018@donghua:~$crontab prgx
jsj2018@donghua:~$crontab -l
50 16 * * * rm -r /abc/*
0 8-18/1 * * * tail -5 /xyz/x1 >>/backup/bak01.txt
50 17 * * 1 tar zcvf backup.tar.gz /data
55 17 * * * umount /dev/hdc
jsj2018@donghua:~$
```

图E6-24 程序运行测试

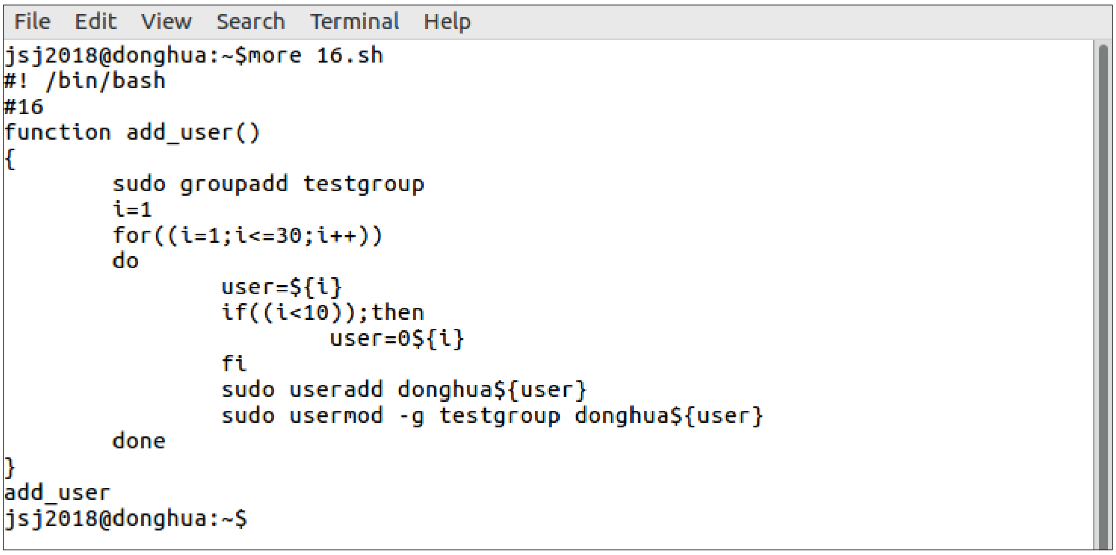
15、设计一个 Shell 程序，查看/home 目录下是否有名为 XXX80～XXX90 的目录，如果有，把它们删除掉。注意：不要删除其它的目录；如图 E6-25 所示。



```
jsj2018@donghua:~$more 15.sh
#! /bin/bash
#15
function delete_dir()
{
        i=1
        for((i=80;i<=90;i=i+1))
        do
                if [ -d ./donghua${i} ];then
                rm -rf ./donghua${i}
                fi
        done
}
delete_dir
jsj2018@donghua:~$
```

图 E6-25 shell 程序示例

16、设计一个 shell 程序，添加一个新组 testgroup，然后添加属于这个组的 30 个用户，用户名的形式为 XXX??，其中??从 01～30 ，如图 E6-26 所示。可通过 cat /etc/passwd 看到所添加的新组。



```
jsj2018@donghua:~$more 16.sh
#! /bin/bash
#16
function add_user()
{
        sudo groupadd testgroup
        i=1
        for((i=1;i<=30;i++))
        do
                user=${i}
                if((i<10));then
                        user=0${i}
                fi
                sudo useradd donghua${user}
                sudo usermod -g testgroup donghua${user}
        done
}
add_user
jsj2018@donghua:~$
```

图 E6-26 shell 程序示例

四、 拓展

1、 查看本机是否有 sed、grep、awk，如果没有则安装。

2、 文件名 datafile，其内容如下：

Steve Blenheim:238-923-7366:95 Latham Lane, Easton, PA 83755:11/12/56:20300

Mary Boop:245-836-8357:635 Cutesy Lane, Hollywood, CA 91464:6/23/23:14500

Igor Chevsky:385-375-8395:3567 Populus Place, Caldwell, NJ 23875:6/18/68:23400

Norma Corder:397-857-2735:74 Pine Street, Dearborn, MI 23874:3/28/45:245700

Jennifer Cowan:548-834-2348:583 Laurel Ave., Kingsville, TX 83745:10/1/35:58900

Mary Evich:284-758-2857:23 Edgecliff Place, Lincoln, NB 92086:7/25/53:85100

Karen Evich:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200

Mary Evich:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200

Fred Fardbarkle:674-843-1385:20 Parak Lane, DeLuth, MN 23850:4/12/23:780900

Fred Fardbarkle:674-843-1385:20 Parak Lane, DeLuth, MN 23850:4/12/23:780900

Lori Gortz:327-832-5728:3465 Mirlo Street, Peabody, MA 34756:10/2/65:35200

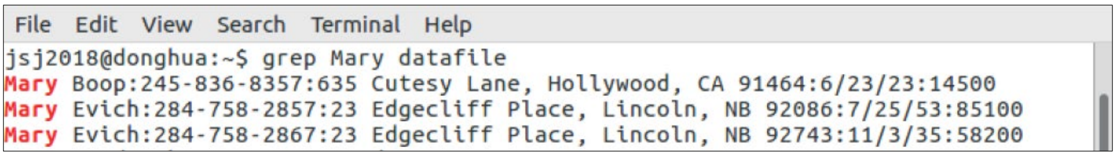
Paco Gutierrez:835-365-1284:454 Easy Street, Decatur, IL 75732:2/28/53:123500

Ephram Hardy:293-259-5395:235 CarltonLane, Joliet, IL 73858:8/12/20:56700

Jam Ikeda:834-938-8376:23445 Aster Ave., Allentown, NJ 83745:12/1/38:45000

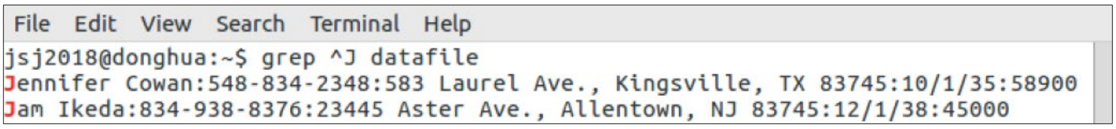
采用 grep、sed 和 awk 命令做如下操作：

（1）显示所有包含 Mary 的行。（提示：$ grep Mary datafile）



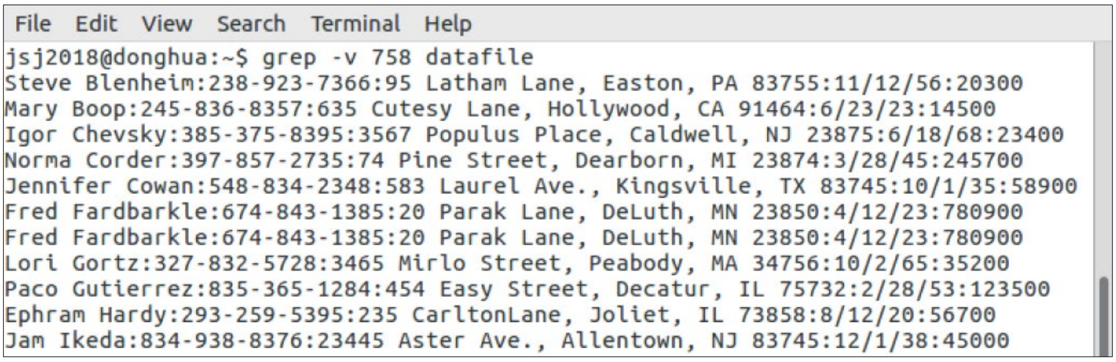
```
jsj2018@donghua:~$ grep Mary datafile
Mary Boop:245-836-8357:635 Cutesy Lane, Hollywood, CA 91464:6/23/23:14500
Mary Evich:284-758-2857:23 Edgecliff Place, Lincoln, NB 92086:7/25/53:85100
Mary Evich:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
```

（2）显示所有以 J 开始的人名所在的行。（提示：$ grep ^J datafile）



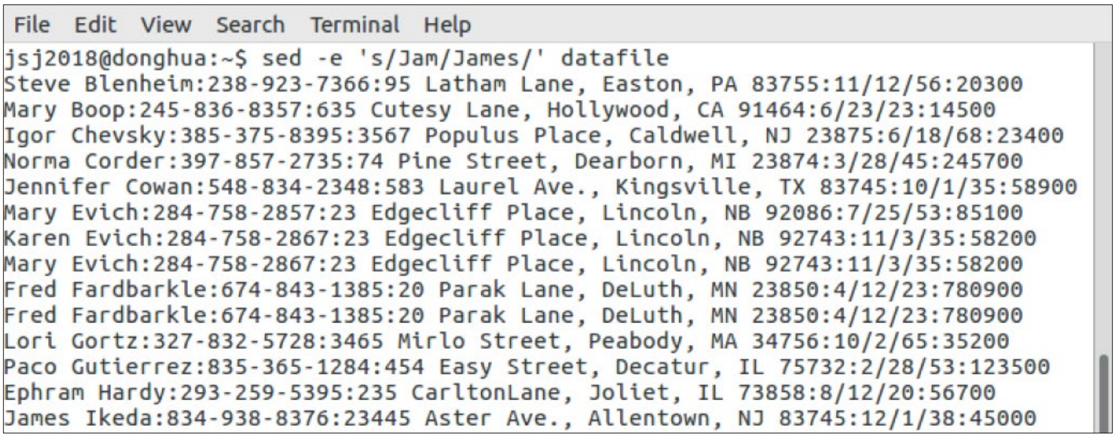
```
jsj2018@donghua:~$ grep ^J datafile
Jennifer Cowan:548-834-2348:583 Laurel Ave., Kingsville, TX 83745:10/1/35:58900
Jam Ikeda:834-938-8376:23445 Aster Ave., Allentown, NJ 83745:12/1/38:45000
```

（3）显示所有不包括 758 的行。（提示：$ grep -v 758 datafile）



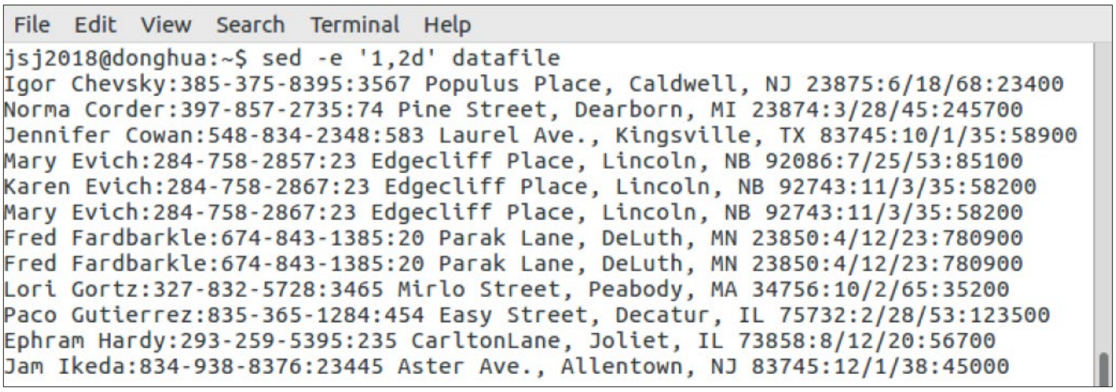
```
jsj2018@donghua:~$ grep -v 758 datafile
Steve Blenheim:238-923-7366:95 Latham Lane, Easton, PA 83755:11/12/56:20300
Mary Boop:245-836-8357:635 Cutesy Lane, Hollywood, CA 91464:6/23/23:14500
Igor Chevsky:385-375-8395:3567 Populus Place, Caldwell, NJ 23875:6/18/68:23400
Norma Corder:397-857-2735:74 Pine Street, Dearborn, MI 23874:3/28/45:245700
Jennifer Cowan:548-834-2348:583 Laurel Ave., Kingsville, TX 83745:10/1/35:58900
Fred Fardbarkle:674-843-1385:20 Parak Lane, DeLuth, MN 23850:4/12/23:780900
Fred Fardbarkle:674-843-1385:20 Parak Lane, DeLuth, MN 23850:4/12/23:780900
Lori Gortz:327-832-5728:3465 Mirlo Street, Peabody, MA 34756:10/2/65:35200
Paco Gutierrez:835-365-1284:454 Easy Street, Decatur, IL 75732:2/28/53:123500
Ephram Hardy:293-259-5395:235 CarltonLane, Joliet, IL 73858:8/12/20:56700
Jam Ikeda:834-938-8376:23445 Aster Ave., Allentown, NJ 83745:12/1/38:45000
```

（4）把 Jam 的名字改成 James。（提示：$ sed -e 's/Jam/James/' datafile）



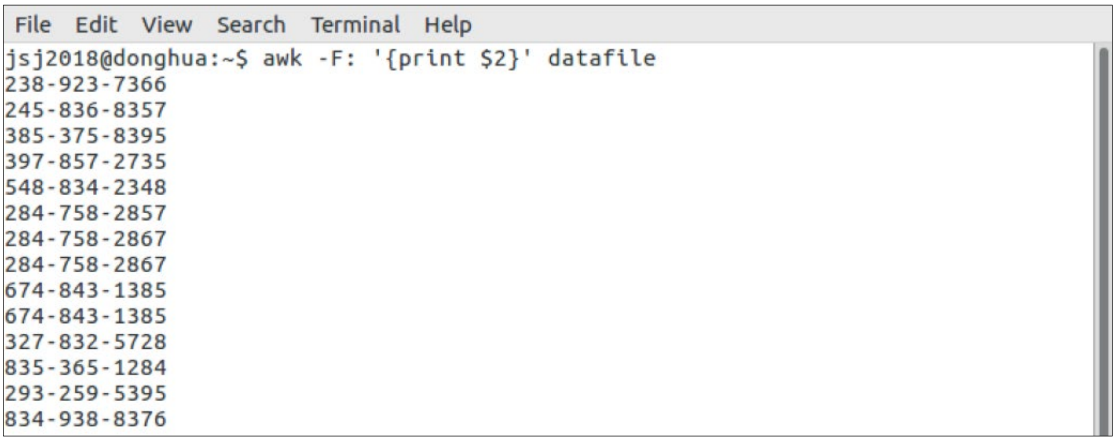
```
jsj2018@donghua:~$ sed -e 's/Jam/James/' datafile
Steve Blenheim:238-923-7366:95 Latham Lane, Easton, PA 83755:11/12/56:20300
Mary Boop:245-836-8357:635 Cutesy Lane, Hollywood, CA 91464:6/23/23:14500
Igor Chevsky:385-375-8395:3567 Populus Place, Caldwell, NJ 23875:6/18/68:23400
Norma Corder:397-857-2735:74 Pine Street, Dearborn, MI 23874:3/28/45:245700
Jennifer Cowan:548-834-2348:583 Laurel Ave., Kingsville, TX 83745:10/1/35:58900
Mary Evich:284-758-2857:23 Edgecliff Place, Lincoln, NB 92086:7/25/53:85100
Karen Evich:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
Mary Evich:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
Fred Fardbarkle:674-843-1385:20 Parak Lane, DeLuth, MN 23850:4/12/23:780900
Fred Fardbarkle:674-843-1385:20 Parak Lane, DeLuth, MN 23850:4/12/23:780900
Lori Gortz:327-832-5728:3465 Mirlo Street, Peabody, MA 34756:10/2/65:35200
Paco Gutierrez:835-365-1284:454 Easy Street, Decatur, IL 75732:2/28/53:123500
Ephram Hardy:293-259-5395:235 CarltonLane, Joliet, IL 73858:8/12/20:56700
James Ikeda:834-938-8376:23445 Aster Ave., Allentown, NJ 83745:12/1/38:45000
```

（5）删除头 2 行。（提示：$ sed -e '1,2d' datafile）

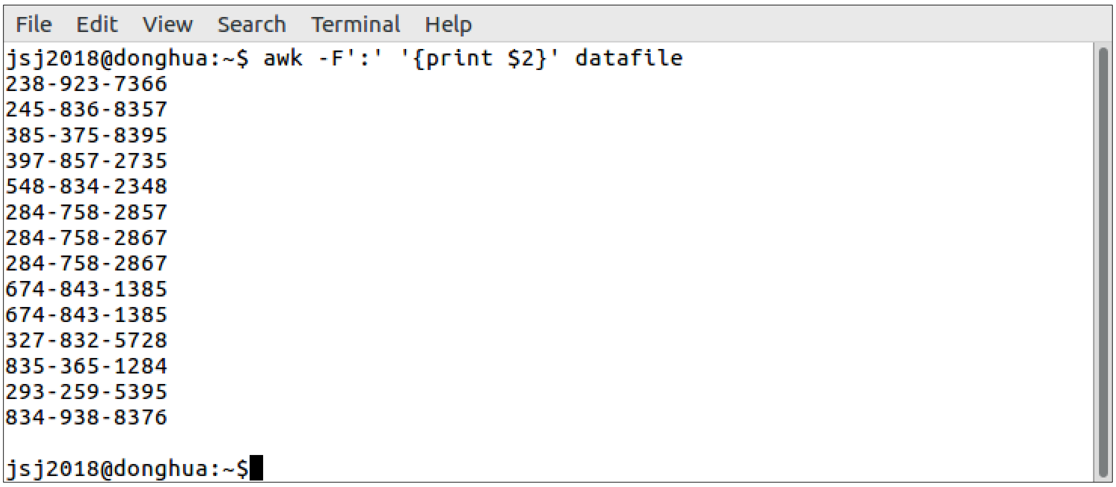


```
jsj2018@donghua:~$ sed -e '1,2d' datafile
Igor Chevsky:385-375-8395:3567 Populus Place, Caldwell, NJ 23875:6/18/68:23400
Norma Corder:397-857-2735:74 Pine Street, Dearborn, MI 23874:3/28/45:245700
Jennifer Cowan:548-834-2348:583 Laurel Ave., Kingsville, TX 83745:10/1/35:58900
Mary Evich:284-758-2857:23 Edgecliff Place, Lincoln, NB 92086:7/25/53:85100
Karen Evich:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
Mary Evich:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
Fred Fardbarkle:674-843-1385:20 Parak Lane, DeLuth, MN 23850:4/12/23:780900
Fred Fardbarkle:674-843-1385:20 Parak Lane, DeLuth, MN 23850:4/12/23:780900
Lori Gortz:327-832-5728:3465 Mirlo Street, Peabody, MA 34756:10/2/65:35200
Paco Gutierrez:835-365-1284:454 Easy Street, Decatur, IL 75732:2/28/53:123500
Ephram Hardy:293-259-5395:235 CarltonLane, Joliet, IL 73858:8/12/20:56700
Jam Ikeda:834-938-8376:23445 Aster Ave., Allentown, NJ 83745:12/1/38:45000
```

（6）显示所有的电话号码。（提示：$ awk -F: '{print $2}' datafile 或$ awk -F':' '{print $2}' datafile）



```
jsj2018@donghua:~$ awk -F: '{print $2}' datafile
238-923-7366
245-836-8357
385-375-8395
397-857-2735
548-834-2348
284-758-2857
284-758-2867
284-758-2867
674-843-1385
674-843-1385
327-832-5728
835-365-1284
293-259-5395
834-938-8376
```



```
jsj2018@donghua:~$ awk -F':' '{print $2}' datafile
238-923-7366
245-836-8357
385-375-8395
397-857-2735
548-834-2348
284-758-2857
284-758-2867
284-758-2867
674-843-1385
674-843-1385
327-832-5728
835-365-1284
293-259-5395
834-938-8376

jsj2018@donghua:~$
```