

1、查看当前系统下用户shell 定义的环境变量的值

```
[151020216@localhost ~]$ echo $PATH
/usr/lib64/qt-3.3/bin:/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/151020216/bin
[151020216@localhost ~]$
```

2、定义变量 AK 的值为 200，并将其显示在屏幕上。

```
[151020216@localhost ~]$ AK=200
[151020216@localhost ~]$ echo $AK
200
[151020216@localhost ~]$
```

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。

```
File Edit View Search Terminal Help
[151020216@localhost ~]$ AM=100
[151020216@localhost ~]$ test $AM -gt 150
[151020216@localhost ~]$ echo $?
1
[151020216@localhost ~]$
```

4、创建一个简单的shell 程序，其功能为：

- 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。

```
[151020216@localhost ~]$ cat -n my.sh
 1 #!/bin/bash
 2 sudo hostname zhanghaijie
 3 hostname
 4 date
 5
 6
[151020216@localhost ~]$ sudo ./my.sh
zhanghaijie
Mon Dec 11 16:54:43 EST 2017
[151020216@localhost ~]$
```

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。

```
[151020216@localhost ~]$ ./my.sh
Enter your score:20
fail
[151020216@localhost ~]$ ./my.sh
Enter your score:80
pass
[151020216@localhost ~]$
```

```
[151020216@localhost ~]$ cat my.sh
#!/bin/bash
echo -n "Enter your score:"
read score
if [ $score -ge 60 ]
then
    echo "pass"
else
    echo "fail"
fi

[151020216@localhost ~]$
```

6、使用for语句创建求命令行上所有整数之和的shell程序。

```
[151020216@localhost ~]$ ./my.sh 1 2 3
sum=6
[151020216@localhost ~]$ cat my.sh
#!/bin/bash
sum=0
for n in $@
do
    let sum=$sum+$n
done
echo "sum=$sum"

[151020216@localhost ~]$
```

7、使用while语句创建一个计算1到5的平方的shell程序。

```
[151020216@localhost ~]$ ./my.sh
1
4
9
16
25
[151020216@localhost ~]$ cat my.sh
#!/bin/bash
n=1
while [ $n -le 5 ]
do
    let s=${n}*${n}
    echo $s
    let n=$n+1
done

[151020216@localhost ~]$
```

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。

```
[151020216@localhost ~]$ ./my.sh 5
Enter your number:
100
5050
[151020216@localhost ~]$ cat my.sh
#!/bin/bash

function get_sum()
{
    echo "Enter your number:"
    read max
    flag=1
    sum=0
    while [ $flag -le $max ]
    do
        let sum=${sum}+${flag}
        let flag=${flag}+1
    done
    echo $sum
}
get_sum
[151020216@localhost ~]$
```

9、使用for 语句创建一个shell 程序，其功能为1+2+3+4+5+...+n。

```
[151020216@localhost ~]$ ./my.sh
Enter your number:
100
5050
[151020216@localhost ~]$ cat my.sh
#!/bin/bash

function get_sum()
{
    echo "Enter your number:"
    read n
    sum=0
    for((i=1;i<=n;i++))
    do
        let sum+=i
    done
    echo ${sum}
}
get_sum
[151020216@localhost ~]$
```

10、使用until 语句创建一个shell 程序，其功能为计算1~10的平方。

```
[151020216@localhost ~]$ ./my.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

[151020216@localhost ~]$ cat my.sh
#!/bin/bash

function get_square()
{
    i=1
    until [ $i -gt 10 ]
    do
        let s=i*i
        echo "$i *$i = $s"
        let i+=1
    done
}

get_square
[151020216@localhost ~]$
```

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；

```

drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx67
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx68
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx69
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx7
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx70
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx71
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx72
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx73
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx74
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx75
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx76
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx77
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx78
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx79
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx8
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx80
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx81
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx82
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx83
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx84
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx85
drwxr-xr-x. 2 151020216 root 4096 Dec 11 17:54 xxx86

```

```

#!/bin/bash

function test_file()
{
    i=1
    for((i;i<=100;i++))
    do
        mkdir xxx${i}
        chmod 755 xxx${i}
    done
}
test_file
151020216@localhost ~]$

```

12、编写shell程序，实现自动删除前1题中50个目录的功能。账号名为XXX1至XXX50；

```

[151020216@localhost ~]$ ./my.sh
[151020216@localhost ~]$ ls
a.out      ourcall    xxx100    xxx56    xxx62    xxx68    xxx74    xxx80    xxx86    xxx92    xxx98
Desktop    ourcall.c xxx51     xxx57    xxx63    xxx69    xxx75    xxx81    xxx87    xxx93    xxx99
Documents  Pictures   xxx52     xxx58    xxx64    xxx70    xxx76    xxx82    xxx88    xxx94
Downloads  Public     xxx53     xxx59    xxx65    xxx71    xxx77    xxx83    xxx89    xxx95
Music      Templates xxx54     xxx60    xxx66    xxx72    xxx78    xxx84    xxx90    xxx96
my.sh      Videos    xxx55     xxx61    xxx67    xxx73    xxx79    xxx85    xxx91    xxx97
[151020216@localhost ~]$

```

```
[151020216@localhost ~]$ cat -n my.sh
 1  #!/bin/bash
 2
 3  function delete_dir()
 4  {
 5      i=1
 6      for((i;i<=50;i++))
 7      do
 8          rm -rf ./xxx${i}
 9      done
10  }
11  delete_dir
12
[151020216@localhost ~]$
```

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 !”，60-70 分为“Passed !”，70-80 分为“Medium !”，80-90 分为“Good !”，90-100 为“Excellent !”。

```
File Edit View Search Terminal Help
[151020216@localhost ~]$ vim my.sh
[151020216@localhost ~]$ ./my.sh
Enter your score:
90
Excellent!
[151020216@localhost ~]$ ./my.sh
Enter your score:
60
Pass!
[151020216@localhost ~]$ ./my.sh
Enter your score:
50
Failed!
[151020216@localhost ~]$
```

```
[151020216@localhost ~]$ cat my.sh
#!/bin/bash

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
[151020216@localhost ~]$
```

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。

```
[root@localhost 151020216]# cat prgx
50 16 * * * rm -r /abc/ *
50 17 * * 1 tar zcvf backup.tar.gz /data
55 17 * * * umount /dev/cdrom
```

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；

```
[151020216@localhost ~]$ cat my.sh
#!/bin/bash

function delete_dir()
{
    i=80
    for((i=1;i<=90;i++))
    do
        if [ -d ./xxx${i} ]
        then
            rm -rf ./xxx${i}
        fi
    done
}
delete_dir
[151020216@localhost ~]$
```

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到30。

```
151020216 xxx11 xxx14 xxx17 xxx2 xxx22 xxx25 xxx28 xxx30 xxx6 xxx9
xxx1 xxx12 xxx15 xxx18 xxx20 xxx23 xxx26 xxx29 xxx4 xxx7 zhj
xxx10 xxx13 xxx16 xxx19 xxx21 xxx24 xxx27 xxx3 xxx5 xxx8
[root@localhost home]# cd 151020216
[root@localhost 151020216]# cat my.sh
#!/bin/bash

function add_user()
{
    sudo groupadd testgroup
    i=1
    for((i=1;i<=30;i++))
    do
        user=${i}
        if ((i<10))
        then
            use=0${i}
        fi
        sudo useradd xxx${user}
        sudo usermod -g testgroup xxx${user}
    done
}
add_user
```