

## 5.22总结：shell脚本命令

### shell变量

#### 系统变量

*HOME*：当前登录用户的家目录//路径  
*USER*：当前用户名  
*\$RANDOM* 可以随机生成 0~32767之间的整数数字

```
echo $HOME #相当于pwd
echo $USER #相当于whoami
echo $[RANDOM%100+1] #生成1-100的随机整数
```

#### 自定义变量

变量名=值；变量名和=之间不能用空格  
用\$符引用变量

```
xm="哥哥"
age=5
echo $xm
echo "$xm是$age岁"

#命令赋值需要用反点``引起
n=`cat /etc/passwd | grep -c "root"`
echo $n
```

#### 特殊变量

*nn*为*number*，\$0代表该脚本名称，\$1 – \$9代表第一到第九个参数（位置参数）# 获取所有输入参数的个数，常用于循环；  
@代表命令行中所有的参数，\$@会把每个参数区分对待；? 返回最后一次命令执行的状态，返回0代表正确执行，返回非0代表执行不正确。

### read 读取终端输入

-p：指定读取值时的提示符；  
-t：指定读取值时等待的时间（秒）。

```
read -p "请输入密码: " s
read -p "请输入密码: " -t 10 pw #10秒后关闭输入
```

## 运算符

### 赋值运算符 =

### 比较运算符

字符串比较: == 等于 != 不等于

数值比较: -eq 等于 -ne 不等于 -lt 小于 -le 小于等于 -gt 大于 -ge 大于等于

类型权限: -f 存在且是文件 -d 存在且是目录 -r 读 (read) -w 写(write) -x 执行 (execute)

### 逻辑运算符

-a 与 -o 或 ! 非

### 算数运算符

\$((运算式)) 或 \${运算式}

+, -, \*, /, % 加, 减, 乘, 除, 取余

## 选择结构

### if [ 条件判断 ]

- if [ 条件判断式 ] 中括号里面两边要加空格!

```
sex="男"
if [ $sex == "男" ]
then echo "去做饭"
elif [ $sex == "女" ]
then echo "去逛街"
else echo "其他"
fi

# 多条件应用
a=3 b=4 c="s"
if [ $a -lt $b -a $c == "s" ]
then echo "true"
else echo "false"
fi
```

```
# 判断文件是否存在
if [ -f /root/a.txt ]
then echo "是文件" cat /root/a.txt
else echo "不是文件"
fi
```

## case 等值判断

```
sex="男"
case $sex in
"男") echo "打游戏" ;;
"女") echo "去逛街" ;;
*) echo "其他" ;;
esac
```

## 循环结构

### for 循环

```
for i in `seq 1 5`
do
    echo $i
done

# 位置参数
for i in $@
do
    echo $i
done

# 字符串
s="a b c dd e"
for i in $s
do
    echo $i
done

# 命令结果
s=`ls /root`
echo $s
for i in $s
do
    echo $i
done
```

# 二重循环：循环里面嵌套循环，外层循环先开始，内层循环结束后，再开始外层的下一个循环

```
for i in `seq 1 5`  
do  
    for j in `seq 1 5`  
    do  
        echo -n "$j" #内层循环  
    done  
    echo "$i" #外层循环  
done
```

## while 循环

```
i=1  
while [ $i -le 10 ]  
do  
    echo $i  
    i=$((i+1))  
done  
  
# 不确定循环次数  
read -p "及格了吗？ (y/n) : " answer  
while [ $answer == "n" ]  
do  
    echo "学习做作业考试"  
    read -p "及格了吗？ (y/n) : " answer  
done  
echo "终于及格了"  
  
#乘法表（双重嵌套）  
for i in `seq 1 9`  
do  
    for j in `seq 1 $i`  
    do  
        echo -n " $j * $i = $((($i * $j)) " "  
    done  
    echo ""  
done
```

## 作业练习

#1. 输入两个数字判断两个数字的大小

```
read -p "请输入第一个数" num1  
read -p "请输入第二个数" num2
```

```
if [ $num1 -eq $num2 ]
then echo "$num1=$num2"
elif [ $num1 -gt $num2 ]
then echo "$num1>$num2"
else echo "$num1<$num2"
fi
```

#2. 输入一个数字输出这个数字是偶数还是奇数

```
read -p "请输入一个整数: " num
if [ ${num%2} -eq 0 ]
then echo "$num 是偶数"
else echo "$num 是奇数"
fi
```

#3. 判断一个年份是否是闰年, 400整除是世纪闰年, 100不整除并且被4整除是闰年, 其余是平年

```
read -p "请输入一个年份: " y
if [ ${y%400} -eq 0 -o ${y%100} -ne 0 -a ${y%4} -eq 0 ]
then echo "$y是闰年"
else echo "$y是平年"
fi
```

#4. 输入三个数, 不考虑相等, 有小到大打印输出

```
read -p "输入三个数（用空格分隔）: " num1 num2 num3
# num1 <= num2
if [ $num1 -gt $num2 ]
then
    temp=$num1
    num1=$num2
    num2=$temp
fi
# num1 <= num3
if [ $num1 -gt $num3 ]
then
    temp=$num1
    num1=$num3
    num3=$temp
fi
# num2 <= num3
if [ $num2 -gt $num3 ]
then
    temp=$num2
    num2=$num3
    num3=$temp
fi
echo "排序: $num1 $num2 $num3"
```

#5.编写脚本,将系统日志文件（/var/log/messages），拷贝到/root/文件夹中

```
cp /var/log/messages /root/
```

#6.编写脚本，位置参数传入学生的姓名和班级，打印“我的名字叫xx，我的班级是xx班”

```
echo "我的名字叫$1，我的班级是$2班"
```

#7.用if，写一个shell脚本，运行时输入一个数字，如果是666则输出evil，如果是777则输出lucky，如果是其他则输出human。

```
read -p "请输入一个数字: " num
if [ $num -eq 666 ]
then echo "evil"
elif [ $num -eq 777 ]
then echo "lucky"
else echo "human"
fi
```

#8.用case方法改写上面的功能。

```
read -p "请输入一个数字: " num
case $num in
"666")
echo "evil"
;;
"777")
echo "lucky"
;;
*)
echo "human"
;;
esac
```

#9.判断/root/hosts是否是存在并且是普通文件，如果不存在则新建一个，如果已经存在，则追加写入当前时间到此文件中

```
path="/root/hosts"
if [ -f $path ]
then
    date >> $path
    echo "是文件，已追加"
else
    touch $path
    echo "没有文件，已新建"
fi
```

#10.判断当前用户是否为root，如果不是则打印非root账号，是的话就执行ifconfig查看网卡信息（\$USER获取当前用户）

```
if [ $USER == "root" ]
then ifconfig
```

```
else echo "$USER"
fi
```

#11. 位置参数传递一个绝对路径的文件，添加所有人可执行权限且提示添加执行权限成功，如果没有传递参数提示没有传递文件。

```
if [ $# -eq 0 ]
then echo "没有传递文件"
elif [ ! -f "$1" ]
then echo "'$1' 不是文件"
else
chmod a+x "$1"
echo "已为 '$1' 添加所有人可执行权限"
fi
```

#12. 编写脚本判断mysql是否正常启动（端口号3306，查看端口netstat -nltp,然后grep过滤返回行数，判断行数是否大于0）

```
port=`netstat -nltp | grep -c "3306"`
if [ $port -gt 0 ]
then
echo "MySQL服务已启动"
else
echo "MySQL服务未启动"
fi
```

#13. 首先显示当天日期， 然后查找给定的用户是否在系统中工作（cat /etc/passwd）。 如果在系统中，就输出一条欢迎语句（例如 hello, root! ）；如果不在系统中，就 输出一条语句（waiting for root! ）

```
date
read -p "请输入要查找的用户: " usr
n=`cat /etc/passwd | grep -c $usr`
if [ $n -gt 0 ]
then echo "hello,$usr! "
else echo "waiting for $usr! "
fi
```

#14. 接受一个参数，若给定参数不是目录， 则给出提示信息； 否则使用 ls -l 命令列出该目录下的内容，并输出有多少个子目录（d 开头），多少 个普通文件(-开头)。显示示例如下： 目录的数量：16 文件的数量：25

```
read -p "请输入一个路径:" path
if [ -d $path ]
then
ls -l $path
d=`ls -l $path | grep -c "^d"`
f=`ls -l $path | grep -c "^-"`
echo "目录数量: $d"
echo "文件数量: $f"
```

```
else
    echo "'$path'不是目录"
fi
```

#上机练习11

#1. 写一个循环，输出20-40

```
for i in `seq 20 40`
do
    echo $i
done
```

#2. 从1-50的单数，倒序

```
for i in `seq 50 -1 1`
do
    echo $i
done
```

#3. 计算并显示 1 到 100 之间的奇数之和

```
for i in `seq 1 100`
do
    if [ ${i%2} -ne 0 ]
    then
        sum=$((sum+$i))
    fi
done
echo $sum
```

#5. 有若干只鸡兔同在一个笼子里，从上面数有35个头，从下面数有94只脚。问笼中各有多少只鸡和兔？

```
ck=35
rb=0
leg=70
while [ $leg -ne 94 ]
do
    ck=$((ck-1))
    rb=$((rb+1))
    leg=$((ck*2+rb*4))
done
echo "$ck只鸡和$rb只兔"
```

#6. 猴子吃桃问题：猴子第一天摘下若干个桃子，当即吃了一半，还不瘾，又多吃了一个，第二天早上又将剩下的桃子吃掉一半，又多吃了一个。以后每天早上都吃了前一天剩下的一半零一个。到第10天早上想再吃时，见只剩下一个桃子了。求第一天共摘了多少。

```
n=1
for i in `seq 1 9`
do
```



```
    n=$((n+1)*2)
done
echo "第一天共摘了$n个桃"
```

#7. 创建用户组wanmei, 批量建立用户wm1到wm9, 用户属于wanmei组, 创建完成查看/etc/group和/etc/passwd文件

```
g="wanmei"
u="wm"
groupadd $g
for i in `seq 1 9`
do
    u=wm$i
    useradd -g $g $u
done
cat /etc/group /etc/passwd
```

#8. 删除上面的用户和用户组

```
g="wanmei"
u="wm"
for i in `seq 1 9`
do
    u=wm$i
    userdel $u
done
groupdel $g
```

#9. 实现传入一个目录作为参数, 如果传入的不是目录则提示不是目录, 如果是目录则将该目录下所有的普通文件的文件名都罗列出来

```
read -p "请输入一个目录:" dir
if [ ! -d $dir ]
then
    echo "不是目录"
else
    names=`ls $dir`
    echo "$names"
    for i in $names
    do
        if [ -d $dir/$i ] #必须要带绝对路径
        then
            echo "$i"
        fi
    done
fi
```

#10. 猜数字游戏: 电脑随机产生一个100以内的数字(1-99), 通过每次键盘输入来猜数字, 输入的小了就提示往大猜, 输入大了就提示往小里猜, 直到输入正确数字(通过\$RANDOM获取随机数)

```
n=$((RANDOM%99+1))
```

```
read -p "现在生成了一个1-99的随机数，请输入你猜的数字：" num
while true
do
    if [ $num -eq $n ]
    then
        echo "恭喜你猜对了！"
        break
    elif [ $num -lt $n ]
    then
        read -p "猜错了，往大猜：" num
    elif [ $num -gt $n ]
    then
        read -p "猜错了，往小猜：" num
    else
        read -p "输入不合法，请重新输入：" num
    fi
done
```