**一，shell中的数值运算**

**——使用expr、$[ ] 、let 等整数运算工具；**

**——使用bc 实现小数运算。**

**1）使用expr命令**

**——乘法操作应 采用 \\* 转义，避免\*被作为shell通配符；运算符两边必须要有空格分开；引用变量时必须加$符号；**

**[root@svr7 ~]# x=1234               ###定义变量x**  
**[root@svr7 ~]# expr $x + 45      ###加法**  
**1279**  
**[root@svr7 ~]# expr $x \\* 2       ###乘法，必须添加\对\*转义**  
**2468**  
**[root@svr7 ~]# expr $x / 3        ###除法，只保留整除结果**  
**411**  
**[root@svr7 ~]# expr $x % 26    ###求模（取除法后的余数）**  
**12**  
  
**2）使用$ [ ]  或 $ (( )) 表达式**

**——乘法操作\*无需转义，运算符两侧可以无空格；引用变量可省略 $ 符号；计算结果替换表达式本身，可结合echo命令输出。**

**[root@svr7 ~]# expr $[x+54]         ###加法**  
**1288**  
**[root@svr7 ~]# expr $[x\*2]           ###乘法**  
**2468**

**3) 使用let命令**

**——expr或$[]、$(())方式只进行运算，并不会改变变量的值；而let命令可以直接对变量值做运算再保存新的值。因此变量X=1234，在执行let运算后的值会变更；另外，let运算操作并不显示结果，但是可以结合echo命令来查看：**

**[root@svr7 ~]# x=1**  
**[root@svr7 ~]# y=66**  
**[root@svr7 ~]# let z=x+y**  
**[root@svr7 ~]# echo $z**  
**67**  
**[root@svr7 ~]# let z=x\*y;echo $z**  
**66**  
**[root@svr7 ~]# let z=y-x;echo $z**  
**65**  
**[root@svr7 ~]# let z=y/x;echo $z**  
**66**  
**[root@svr7 ~]# let z=y%x;echo $z**  
**0**  
  
**4）计算的简写**

**i=1**

**let i+=1  ==>  i=i+1**

**let i\*=8  ==>   i=i\*8**

**let i-=2  ==>   i=i-2**

**let i/=3  ==>   i=i/3**

**let i%=2  ==>   i=i%2**

**[root@svr7 ~]# i=1**  
**[root@svr7 ~]# let i+=1 ; echo $i**         **###let i=i+1**  
**2**  
**[root@svr7 ~]# let i\*=8 ; echo $i**     **###let i=i\*8**  
**16**  
**[root@svr7 ~]# let i-=2 ;echo $i**          **###let i=i-2**  
**14**  
**[root@svr7 ~]# let i/=3;echo $i**          **###let** **i=i/3**  
**4**  
**[root@svr7 ~]# let i%=2;echo $i**         **###let  i=i%2**  
**0**  
  
**二 ） 小数运算工具**

**1）bc交互式运算**

**[root@svr7 ~]# bc**

**scale=2            ###设置小数点后几位显示；不设置，则默认显示为整数**  
**2/10**  
**.20**  
**quit                 ###退出**

**2）bc 非交互式运算**

**——将需要运算的表达式通过管道操作交给bc运算。**

**[root@svr7 ~]# echo 'scale=4;1.023+2.548' | bc**  
**3.571**  
**[root@svr7 ~]# echo 'scale=4;1.254\*5.1563' | bc**  
**6.4660**  
=====================================================================================================================  
**二 ）条件测试操作**

**2.1 问题：** **字符串匹配**

**比较整数值的大小**

**识别文件/目录的状态**

**多个条件/操作的逻辑组合**

**2.2 步骤：    语法格式——> 使用“test表达式”  或者 【 表达式  】都可以，表达式两边至少要留一个空格，条件测试操作本身不显示出任何信息。测试的条件是否成立主要体现在命令执行后的返回状态（即 $?），所以可以在测试后查看变量$?的值来做出判断，或者结合&&、||等逻辑操作显示出结果（或作其他操作） 。**

**2.3   字符串匹配**

**1） ==  比较两个字符串是否相同**

**[root@svr7 ~]# [ $USER == root ]**  
**[root@svr7 ~]# echo $?                   ### 查看结果为0则为对，非0为错**  
**0**  
**2） !=   比较两个字符串是否不相同**  
**[root@svr7 ~]# [ $USER != root  ]**  
**[root@svr7 ~]# echo $?**  
**1**  
**3 ） -z   检查变量的值是否未设置（控制）**

 [root@svr5 ~]# var1="Tarena" ; var2=""

 [root@svr5 ~]# [ -z "$var1" ] && echo "空值" || echo "非空值"

 非空值

 [root@svr5 ~]# [ -z $var2 ] && echo "空值" || echo "非空值"

 空值                                     //变量var2已设置，但无任何值，视为空

**&&  逻辑与**

**||     逻辑或**

**例：  A  &&  B   ：执行A命令，仅当A成功，才执行B；**

**A  ||     B   ：执行A命令，仅当A失败，才执行B。**

**A  &&  B   ||   C  ：执行A成功后，才执行B；B执行成功后，就不会再执行C；反之B失败，则执行C**

**A  ；  B   ：执行A，执行B**

**[root@svr7 ~]# [ a == a ] && echo y || echo n**  
**y**  
**[root@svr7 ~]# [ a == b ] && echo y || echo n**  
**n**  
  
**3）整数值比较**

**-eq    等于equal**

**[root@svr7 ~]# x=20**  
**[root@svr7 ~]# [ $x -eq 20 ] && echo "等于" || echo "不等于"**  
**等于**  
**[root@svr7 ~]# [ $x -eq 10 ] && echo "等于" || echo "不等于"**  
**不等于**

**-ne    不等于not equal**

**[root@svr7 ~]# [ $x -ne 10 ] && echo "不等于" || echo "等于"  
不等于  
[root@svr7 ~]# [ $x -ne 20 ] && echo "不等于" || echo "等于"  
等于**

**-gt    大于**

**[root@svr7 ~]# [ $x -gt 10 ] && echo "大于" || echo "否"**  
**大于**  
**[root@svr7 ~]# [ $x -gt 20 ] && echo "大于" || echo "否"**  
**否**  
**[root@svr7 ~]# [ $x -gt 30 ] && echo "大于" || echo "否"**  
**否**

**-ge   大于等于**

**[root@svr7 ~]# [ $x -ge 30 ] && echo "大于或等于" || echo "否"**  
**否**  
**[root@svr7 ~]# [ $x -ge 20 ] && echo "大于或等于" || echo "否"**  
**大于或等于**  
**[root@svr7 ~]# [ $x -ge 10 ] && echo "大于或等于" || echo "否"**  
**大于或等于**

**-lt    小于**

**[root@svr7 ~]# [ $x -lt 10 ] && echo "小于" || echo "否"**  
**否**  
**[root@svr7 ~]# [ $x -lt 20 ] && echo "小于" || echo "否"**  
**否**  
**[root@svr7 ~]# [ $x -lt 30 ] && echo "小于" || echo "否"**  
**小于**

**-le   小于等于**

**[root@svr7 ~]# [ $x -le 30 ] && echo "小于或等于" || echo "否"**  
**小于或等于**  
**[root@svr7 ~]# [ $x -le 20 ] && echo "小于或等于" || echo "否"**  
**小于或等于**  
**[root@svr7 ~]# [ $x -le 10 ] && echo "小于或等于" || echo "否"**  
**否**  
  
**案例： 每分钟判断当前登录用户，超过3个则发邮件报警**

**[root@svr7 ~]# vim baojing.sh  
#! /bin/bash  
#先写邮件的内容：**

**echo "用户数超过了3个**

**." > mail.txt**  
**【 $(who | wc -l ) -gt 3】 && mail  -s "警报"  root < /root/mail.txt**  
**[root@svr7 ~]# crontab -e**  
**\*/1 \* \* \* \*  sh /root/baojing.sh**  
  
**4) 识别文件/目录的状态**

**1） -e 判断对象是否存在（不管三目录还是文件）**

**[root@svr7 ~]# [ -e "/etc/passwd" ] && echo '存在' || echo '不存在'**  
**存在**

**[root@svr7 ~]# [ -e "/root/abc " ] && echo '存在' || echo '不存在'**  
**不存在**

**2) -d  判断对象是否为目录（存在且是目录）**

**[root@svr7 ~]# [ -d "/usr/src" ] && echo '是' || echo '不是'**  
**是**  
**[root@svr7 ~]# [ -d "/etc/fstab" ] && echo '是' || echo '不是'**  
**不是**

**3) -f  判断对象是否为文件（存在且为文件）**

**[root@svr7 ~]# [ -f "/etc/fstab" ] && echo '是' || echo '不是'**  
**是**  
**[root@svr7 ~]# [ -f "/mnt" ] && echo '是' || echo '不是'**  
**不是**

**4） -r  判断对象是否可读 （此测试对root用户无效，无论文件是否设置r权限，root都可读）**

**[root@svr7 ~]# [ -r "/etc/shadow" ] && echo '可读' || echo '不可读'**  
**可读**  
**[root@svr7 ~]# su hahe**  
**[hahe@svr7 root]$ [ -r "/etc/shadow" ] && echo '可读' || echo '不可读'**  
**不可读**  
**5） -w 判断对象是否可写**

**6） -x 判断对象是否具有可执行权限**

**案例： 如果/test目录存在，就无操作；**

**如果/test目录没有，就创建/test；**

**如果/test/目录下，有nb.txt 就无操作；**

**如果/tset目录下，没有nb.txt就创建。**

**[root@svr7 ~]#vim test.sh**

**#! /bin/bash**

**[ -d "/test"  ]  || mkdir /test**

**[ -f "/test/nb.txt" ] || touch /test/nb.txt**

================================================================================================================

**三） 使用if选择结构**

**1） if 单分支语法组成：**

**if   条件测试 ； then**

**命令序列**

**fi**

**2） if 双分支语法组成：**

**if   条件测试 ；  then**

**命令序列1**

**else   命令序列2**

**fi**

**案例1：只能ping 一个固定的IP**

**#! /bin/bash**  
**ping -c2 192.168.4.254 &> /dev/null**  
**if [ $? -eq 0 ] ;then**  
**echo "通了"**  
**else**  
**echo "通不了"**  
**fi**  
**案例2：可以自定义ping IP**

**#! /bin/bash**  
**read -p "请输入一个正确的IP地址：" IP**  
**ping -c2 $IP &> /dev/null**  
**if [ $? -eq 0 ] ;then**  
**echo "通了"**  
**else**  
**echo "通不了"**  
**fi**  
**扩展内容：ping -c2  -i0.1  -W1   里， -c2代表ping 2次； -i0.1 代表ping通时的间隔时间为0.1s; -W1 代表ping不通时的间隔时间改为1s**

**3） if 多分支语法组成：**

**if   条件测试1 ;  then**

**命令序列1**

**elif 条件测试2；**

**then 命令序列2**

**else 命令序列3**

**fi**