Expl let bc

整数运算 + - \* / 加减乘除

· expr $X + 78                     //加法

· 1312

· [root@svr5 ~]# expr $X - 78                     //减法

· 1156

· [root@svr5 ~]# expr $X \\* 78                     //乘法，操作符应添加\转义

· 96252

· [root@svr5 ~]# expr $X / 78                     //除法，仅保留整除结果

· 15

· [root@svr5 ~]# expr $X % 78                     //求模

· 64

1. [root@svr5 ~]# X=1234
2. [root@svr5 ~]# echo $[X+78]
3. 1312
4. [root@svr5 ~]# echo $[X-78]
5. 1156
6. [root@svr5 ~]# echo $[X\*78]
7. 96252
8. [root@svr5 ~]# echo $[X/78]
9. 15
10. [root@svr5 ~]# echo $[X%78]
11. 64

· [root@svr5 ~]# X=1234

· [root@svr5 ~]# let y=X+22

· [root@svr5 ~]# let X++; echo $X                    # X++(X=X+1)

· [root@svr5 ~]# let X--; echo $X                    # X--(X=X-1)

· [root@svr5 ~]# let X+=78 ; echo $X                    # X+=78(X=X+78)

· [root@svr5 ~]# let X-=78 ; echo $X                 # X-=78(X=X-78)

· [root@svr5 ~]# let X\*=78 ; echo $X                 # X\*=78(X=X\*78)

· [root@svr5 ~]# let X/=78 ; echo $X                 # X/=78(X=X/78)

· [root@svr5 ~]# let X%=78 ; echo $X                 # X%=78(X=X%79)

bc交互式运算

bc非交互式运算

1. [root@svr5 ~]# echo 'scale=4;12.34+5.678' | bc
2. 18.018
3. [root@svr5 ~]# echo 'scale=4;12.34-5.678' | bc
4. 6.662
5. [root@svr5 ~]# echo 'scale=4;12.34\*5.678' | bc
6. 70.0665
7. [root@svr5 ~]# echo 'scale=4;12.34/5.678' | bc
8. 2.1733

## 条件测试操作

== 比较两个字符串是否相同

!= 比较两个字符串是否不相同

· [root@svr5 ~]# echo $?                        //查看结果0为对，非0为错

一行执行多条命令的情况

1. # A && B                     //仅当A命令执行成功，才执行B命令
2. # A || B                        //仅当A命令执行失败，才执行B命令
3. # A ; B                        //执行A命令后执行B命令，两者没有逻辑关系

整数值比较

1. eq 等于 [root@svr5 ~]# X=20                         //定义一个测试变量
2. [root@svr5 ~]# [ $X -eq 20 ] && echo "相等" || echo "不相等"
3. 相等
4. [root@svr5 ~]# [ $X -eq 30 ] && echo "相等" || echo "不相等"
5. 不相等

Gt 大于

Ge 大于等于

Lt 小于

Le 小于等于

Ne 不等于

提取当前登录的用户数

who | wc -l

1. [root@svr5 ~]# who | wc -l                                 //确认已登录的用户数
2. 4
3. [root@svr5 ~]# N=$(who | wc -l)                             //赋值给变量N
4. [root@svr5 ~]# [ $N -gt 5 ] && echo "超过了" || echo "没超过"
5. 没超过

识别文件/目录的状态

-e 判断对象是否存在（不管是目录还是文件）

-d 判断对象是否为目录（存在且是目录）

-f 判断对象是否为文件（存在且是文件）

-r 判断对象是否可读

-w 判断对象是否可写

## 使用if选择结构

if单分支的语法组成

if 条件测试

then 命令序列

fi

if双分支的语法组成：

if 条件测试

then 命令序列1

else 命令序列2

fi

if多分支的语法组成：

1. if 条件测试1
2. then 命令序列1
3. elif 条件测试2
4. then 命令序列2
5. else 命令序列n
6. fi

if多分支结构实际上相当于多层if嵌套：

1. if 条件测试1 ; then
2. 命令序列1
3. else
4. if 条件测试2 ; then
5. 命令序列2
6. else
7. .. ..
8. 命令序列n
9. fi
10. Fi

检测并判断指定的主机是否可ping通

Ping 命令 -c 次数 -i 秒数 -w 回应时间段

冽 ping 192.168.4.1 -c 2 -i 0.3 -w 3

拼同 两次 零点三秒 回应

· #!/bin/bash

· ping -c 3 -i 0.2 -W 3 $1 &> /dev/null

· if [ $? -eq 0 ] ; then

· echo "Host $1 is up."

· else

· echo "Host $1 is down."

· fi

·