



Linux自动化运维特训班

条件测试

讲师：孔丹

大纲

- 条件测试： 主要介绍Shell程序中的文件、变量、字符串数值以及逻辑等条件测试

条件测试

- 为了能够正确处理Shell程序运行过程中遇到的各种情况，Linux Shell提供了一组测试运算符。通过这些运算符，Shell程序能够判断某种或者几个条件是否成立。条件测试在各种流程控制语句，例如判断语句和循环语句中发挥了重要的作用，所以，了解和掌握这些条件测试是非常重要的。

条件测试的基本语法

□ 在Shell程序中，用户可以使用测试语句来测试指定的条件表达式的条件的真或者假。当指定的条件为真时，整个条件测试的返回值为0；反之，如果指定的条件为假，则条件测试语句的返回值为非0值。

格式1: test 条件表达式

格式2: [条件表达式]

格式3: [[条件表达式]]

注意: test 和条件表达式至少一个空格

[]中条件表达式左右至少一个空格

[[]]中条件表达式左右至少一个空格

文件测试表达式

□ [操作符 文件或目录]

操作符	说明	举例
-b file	检测文件是否是块设备文件，如果是，则返回 true。	[-b \$file] 返回 false。
-c file	检测文件是否是字符设备文件，如果是，则返回 true。	[-c \$file] 返回 false。
-d file	检测文件是否是目录，如果是，则返回 true。	[-d \$file] 返回 false。
-f file	检测文件是否是普通文件（既不是目录，也不是设备文件），如果是，则返回 true。	[-f \$file] 返回 true。
-g file	检测文件是否设置了 SGID 位，如果是，则返回 true。	[-g \$file] 返回 false。
-k file	检测文件是否设置了粘着位(Sticky Bit)，如果是，则返回 true。	[-k \$file] 返回 false。
-p file	检测文件是否是具名管道，如果是，则返回 true。	[-p \$file] 返回 false。
-u file	检测文件是否设置了 SUID 位，如果是，则返回 true。	[-u \$file] 返回 false。
-r file	检测文件是否可读，如果是，则返回 true。	[-r \$file] 返回 true。

文件测试表达式

□ [操作符 文件或目录]

//普通文件测试

```
[root@localhost ~]# touch file1.txt
```

```
[root@localhost ~]# [ -f file1.txt ] && echo yes || echo no  
yes
```

//目录测试

```
[root@localhost ~]# [ -d /tmp ] && echo yes || echo no  
yes
```

//测试文件属性

```
[root@localhost ~]# ll file1.txt
```

```
-rw-r--r-- 1 root root 0 8月 28 12:30 file1.txt
```

```
[root@localhost ~]# [ -r file1.txt ] && echo yes || echo no  
yes
```

```
[root@localhost ~]# [ -x file1.txt ] && echo yes || echo no  
no
```


关系运算操作符

□ 关系运算符列表

运算符	说明	举例
-eq	检测两个数是否相等，相等返回 true。	[\$a -eq \$b] 返回 true。
-ne	检测两个数是否相等，不相等返回 true。	[\$a -ne \$b] 返回 true。
-gt	检测左边的数是否大于右边的，如果是，则返回 true。	[\$a -gt \$b] 返回 false。
-lt	检测左边的数是否小于右边的，如果是，则返回 true。	[\$a -lt \$b] 返回 true。
-ge	检测左边的数是否大等于右边的，如果是，则返回 true。	[\$a -ge \$b] 返回 false。
-le	检测左边的数是否小于等于右边的，如果是，则返回 true。	[\$a -le \$b] 返回 true。

判断变量是不是数字：

```
[root@localhost ~]# num10=123
[root@localhost ~]# num20=ssss1114ss
[root@localhost ~]# [[ "$num10" =~ ^[0-9]+$ ]];echo $?
0
[root@localhost ~]# [[ "$num20" =~ ^[0-9]+$ ]];echo $?
1
```

布尔运算符

□ 布尔运算符列表

运算符	说明	举例
!	非运算，表达式为 true 则返回 false，否则返回 true。	[! false] 返回 true。
-o	或运算，有一个表达式为 true 则返回 true。	[\$a -lt 20 -o \$b -gt 100] 返回 true。
-a	与运算，两个表达式都为 true 才返回 true。	[\$a -lt 20 -a \$b -gt 100] 返回 false。

布尔运算符

□ 示例

```
[root@localhost ~]# [ -f /etc/hosts -a -f /etc/services ] &&  
echo yes || echo no  
yes
```

```
[root@localhost ~]# [[ -f /etc/hosts && -f /etc/services ]] &&  
echo yes || echo no  
yes
```

字符串运算符

□ 字符串运算符列表

运算符	说明	举例
=	检测两个字符串是否相等，相等返回 true。	[\$a = \$b] 返回 false。
!=	检测两个字符串是否相等，不相等返回 true。	[\$a != \$b] 返回 true。

字符串运算符

□ 示例

// -n 如果字符串长度不为零输出yes, 否则输出no

```
[root@localhost ~]# [ -n "hello" ] && echo yes || echo no  
yes
```

```
[root@localhost ~]# [ -n "" ] && echo yes || echo no  
no
```

// -z 如果字符串长度为零输出yes, 否则输出no

```
[root@localhost ~]# [ -z "hello" ] && echo yes || echo no  
no
```

```
[root@localhost ~]# [ -z "" ] && echo yes || echo n  
yes
```

// 字符串相等比较

注意: = 的左右有空格, 没有空格将会导致逻辑错误。

```
[root@localhost ~]# [ "HELO" = "hello" ] && echo yes || echo no  
no
```

```
[root@localhost ~]# [ "HELO" != "hello" ] && echo yes || echo no  
yes
```

[]和[][]的区别

[]中使用	[[]]中使用
--------	----------

-a	
-o	
!	

&&	
!	

说明

与，两端都为真，则真
或，两端有一个为真则为真
非，相反则为真

总结

- 条件测试的基本语法
- 文件测试表达式
- 字符串测试表达式
- 整数测试表达式

作业

- 1、取出/etc/inittab文件的第6行;
- 2、取出当前系统上所有用户的shell, 要求, 每种shell只显示一次, 并且按顺序进行显示;
 - 使用cut、sort结合管道实现
- 3、如果/var/log/messages文件的行数大于100, 就显示好的文件
- 4、显示/etc目录下所有以pa开头的文件, 并统计其个数;
- 5、如果用户hadoop不存在就添加, 否则显示用户已存在
- 6、编写一个 Shell 程序 test1, 程序执行时从键盘读入一个目录名, 然后 显示这个目录下所有文件的信息
- 7、编写一个 Shell 程序 test2, 从键盘读入 x、y 的值, 然后做加法运算, 最后输出结果



谢谢观看

更多好课，请关注[万门大学APP](#)

