[**Linux 之 shell 比较运算符**](http://blog.csdn.net/sunboy_2050/article/details/6836382)

[【分享季1】：网友推荐130个经典资源，分享再赠分！](http://topic.csdn.net/u/20120416/10/9123faba-8474-4f2a-a0cf-a1b7d0fe5913.html)

[**Linux 之 shell 比较运算符**](http://blog.csdn.net/sunboy_2050/article/details/6836382)

分类： [Linux](http://blog.csdn.net/sunboy_2050/article/category/655460)2011-09-30 10:16487人阅读[评论](http://blog.csdn.net/sunboy_2050/article/details/6836382#comments)(0)[收藏](javascript:void(0);)[举报](http://blog.csdn.net/sunboy_2050/article/details/6836382#report)

|  |  |  |
| --- | --- | --- |
| 运算符 | 描述 | 示例 |
| 文件比较运算符 | | |
| -e filename | 如果 filename 存在，则为真 | [ -e /var/log/syslog ] |
| -d filename | 如果 filename 为目录，则为真 | [ -d /tmp/mydir ] |
| -f filename | 如果 filename 为常规文件，则为真 | [ -f /usr/bin/grep ] |
| -L filename | 如果 filename 为符号链接，则为真 | [ -L /usr/bin/grep ] |
| -r filename | 如果 filename 可读，则为真 | [ -r /var/log/syslog ] |
| -w filename | 如果 filename 可写，则为真 | [ -w /var/mytmp.txt ] |
| -x filename | 如果 filename 可执行，则为真 | [ -L /usr/bin/grep ] |
| filename1 -nt filename2 | 如果 filename1 比 filename2 新，则为真 | [ /tmp/install/etc/services -nt /etc/services ] |
| filename1 -ot filename2 | 如果 filename1 比 filename2 旧，则为真 | [ /boot/bzImage -ot arch/i386/boot/bzImage ] |
| 字符串比较运算符 （请注意引号的使用，这是防止空格扰乱代码的好方法） | | |
| -z string | 如果 string 长度为零，则为真 | [ -z "$myvar" ] |
| -n string | 如果 string 长度非零，则为真 | [ -n "$myvar" ] |
| string1 = string2 | 如果 string1 与 string2 相同，则为真 | [ "$myvar" = "one two three" ] |
| string1 != string2 | 如果 string1 与 string2 不同，则为真 | [ "$myvar" != "one two three" ] |
| 算术比较运算符 | | |
| num1 -eq num2 | 等于 | [ 3 -eq $mynum ] |
| num1 -ne num2 | 不等于 | [ 3 -ne $mynum ] |
| num1 -lt num2 | 小于 | [ 3 -lt $mynum ] |
| num1 -le num2 | 小于或等于 | [ 3 -le $mynum ] |
| num1 -gt num2 | 大于 | [ 3 -gt $mynum ] |
| num1 -ge num2 | 大于或等于 | [ 3 -ge $mynum ] |

|  |
| --- |
|  |
| 测试命令  test命令用于检查某个条件是否成立，它可以进行数值、字符和文件3个方面的测试，其测试符和相应的功能分别如下。  **（1）数值测试：**  　　-eq 等于则为真。  　　-ne 不等于则为真。  　　-gt 大于则为真。  　　-ge 大于等于则为真。  　　-lt 小于则为真。  　　-le 小于等于则为真。  **（2）字串测试：**  　　= 等于则为真。  　　!= 不相等则为真。  　　-z字串 字串长度伪则为真。  　　-n字串 字串长度不伪则为真。  **（3）文件测试：**  　　-e文件名 如果文件存在则为真。  　　-r文件名 如果文件存在且可读则为真。  　　-w文件名 如果文件存在且可写则为真。  　　-x文件名 如果文件存在且可执行则为真。  　　-s文件名 如果文件存在且至少有一个字符则为真。  　　-d文件名 如果文件存在且为目录则为真。  　　-f文件名 如果文件存在且为普通文件则为真。  　　-c文件名 如果文件存在且为字符型特殊文件则为真。  　　-b文件名 如果文件存在且为块特殊文件则为真 |

**条件变量替换:**   
Bash Shell可以进行变量的条件替换,既只有某种条件发生时才进行替换,替换   
条件放在{}中.   
(1) ${value:-word}

当变量未定义或者值为空时,返回值为word的内容,否则返回变量的值.

(2) ${value:=word}

与前者类似,只是若变量未定义或者值为空时,在返回word的值的同时将 word赋值给value

(3) ${value:?message}

若变量已赋值的话,正常替换.否则将消息message送到标准错误输出(若此替换出现在Shell程序中,那么该程序将终止运行)

(4) ${value:+word}

若变量已赋值的话,其值才用word替换,否则不进行任何替换

(5) ${value:offset}

${value:offset:length} 从变量中提取子串,这里offset和length可以是算术表达式.

(6) ${#value}

变量的字符个数

(7) ${value#pattern}

${value##pattern}   
去掉value中与pattern相匹配的部分,条件是value的开头与pattern相匹配   
#与##的区别在于一个是最短匹配模式,一个是最长匹配模式.

(8) ${value%pattern}

${value%%pattern}   
于(7)类似,只是是从value的尾部于pattern相匹配,%与%%的区别与#与##一样

(9) ${value/pattern/string}

${value//pattern/string}   
进行变量内容的替换,把与pattern匹配的部分替换为string的内容,/与//的区别与上同

注意: 上述条件变量替换中,除(2)外,其余均不影响变量本身的值

#!/bin/bash

var1="1"  
var2="2"

下面是“**与**”运算符**-a**，另外注意，用一个test命令就可以了，还有if条件后面的分号

if test $var1 = "1"-a $var2 = "2" ; then  
echo "equal"  
fi

下面是“**或**”运算符 **-o**，有一个为真就可以

if test $var1 != "1" -o $var2 != "3" ; then  
echo "not equal"  
fi

下面是“**非**”运算符 **！**  
if条件是为真的时候执行，如果使用！运算符，那么原表达式必须为false

if ! test $var1 != "1"; then  
echo "not 1"  
fi

以上三个if都为真，所以三个echo都会打印

------------------------------------------------------------------------------------------------------

**shell字符串比较、判断是否为数字**

|  |
| --- |
| 二元比较操作符,比较变量或者比较数字.注意数字与字符串的区别.  整数比较  -eq 等于,如:if [ "$a" -eq "$b" ]  -ne 不等于,如:if [ "$a" -ne "$b" ]  -gt 大于,如:if [ "$a" -gt "$b" ]  -ge 大于等于,如:if [ "$a" -ge "$b" ]  -lt 小于,如:if [ "$a" -lt "$b" ]  -le 小于等于,如:if [ "$a" -le "$b" ]  < 小于(需要双括号),如:(("$a" < "$b"))  <= 小于等于(需要双括号),如:(("$a" <= "$b"))  > 大于(需要双括号),如:(("$a" > "$b"))  >= 大于等于(需要双括号),如:(("$a" >= "$b"))  字符串比较 = 等于,如:if [ "$a" = "$b" ]  == 等于,如:if [ "$a" == "$b" ],与=等价 注意:==的功能在[[]]和[]中的行为是不同的,如下: 1 [[ $a == z\* ]] # 如果$a以"z"开头(模式匹配)那么将为true 2 [[ $a == "z\*" ]] # 如果$a等于z\*(字符匹配),那么结果为true 3 4 [ $a == z\* ] # File globbing 和word splitting将会发生 5 [ "$a" == "z\*" ] # 如果$a等于z\*(字符匹配),那么结果为true 一点解释,关于File globbing是一种关于文件的速记法,比如"\*.c"就是,再如~也是. 但是file globbing并不是严格的正则表达式,虽然绝大多数情况下结构比较像.  != 不等于,如:if [ "$a" != "$b" ] 这个操作符将在[[]]结构中使用模式匹配.  < 小于,在ASCII字母顺序下.如: if [[ "$a" < "$b" ]] if [ "$a" \< "$b" ] 注意:在[]结构中"<"需要被转义.  > 大于,在ASCII字母顺序下.如: if [[ "$a" > "$b" ]] if [ "$a" \> "$b" ] 注意:在[]结构中">"需要被转义. 具体参考Example 26-11来查看这个操作符应用的例子.  -z 字符串为"null".就是长度为0  -n 字符串不为"null"  注意: 使用-n在[]结构中测试必须要用""把变量引起来.使用一个未被""的字符串来使用! -z或者就是未用""引用的字符串本身,放到[]结构中。虽然一般情况下可以工作,但这是不安全的.习惯于使用""来测试字符串是一种好习惯.  awk '{print $2}' class.txt | grep '^[0-9.]' > res |

**推荐参考：**

[Shell 比较运算](http://blog.sina.com.cn/s/blog_4d2c7ad60100cqxz.html)

[Linux Shell学习简单小结(更新中……)](http://blog.csdn.net/sunboy_2050/article/details/6120376)

**分享到：**

 上一篇：[JNI和NDK的区别](http://blog.csdn.net/sunboy_2050/article/details/6828830)

 下一篇：[Ubuntu下搭建Android NDK开发环境](http://blog.csdn.net/sunboy_2050/article/details/6837469)