

# Shell运算符

2022年10月21日 17:29

- Shell 和其他编程语言一样，支持多种运算符，包括：
  - 算数运算符
  - 关系运算符
  - 布尔运算符
  - 字符串运算符
  - 文件测试运算符
- 原生bash不支持简单的数学运算，但是可以通过其他命令来实现，例如 `awk` 和 `expr`，`expr` 最常用。

• 例如，两个数相加(注意使用的是反引号 ``` 而不是单引号 `'`)：

```
#!/bin/bashval=`expr 2 + 2`echo"两数之和为：$val"#两数之和为：4
```

- 表达式和运算符之间要有空格，例如 `2+2` 是不对的，必须写成 `2 + 2`，这与我们熟悉的大多数编程语言不一样。
- 完整的表达式要被 ``` 包含，注意这个字符不是常用的单引号，在 `Esc` 键下边。
- `centos`中可用 `"$((运算式))"` 或 `"${运算式}"`

## 1. 算术运算符

下表列出了常用的算术运算符，假定变量 `a` 为 10，变量 `b` 为 20：

运算符	说明	举例
+	加法	<code>`expr \$a + \$b`</code> 结果为 30。
-	减法	<code>`expr \$a - \$b`</code> 结果为 -10。
*	乘法	<code>`expr \$a * \$b`</code> 结果为 200。
/	除法	<code>`expr \$b / \$a`</code> 结果为 2。
%	取余	<code>`expr \$b % \$a`</code> 结果为 0。
=	赋值	<code>a=\$b</code> 将把变量 <code>b</code> 的值赋给 <code>a</code> 。
==	相等。用于比较两个数字，相同则返回 <code>true</code> 。	<code>[ \$a == \$b ]</code> 返回 <code>false</code> 。
!=	不相等。用于比较两个数字，不相同则返回 <code>true</code> 。	<code>[ \$a != \$b ]</code> 返回 <code>true</code> 。

注意：

- 乘号 `(*)` 前边必须加反斜杠 `()` 才能实现乘法运算；
- `if...then...fi` 是条件语句，后续将会讲解。
- 在 `MAC` 中 `shell` 的 `expr` 语法是：`$( (表达式) )`，此处表达式中的 `"*"` 不需要转义符号 `"\"` 。

## 2. 关系运算符

- 关系运算符只支持数字，不支持字符串，除非字符串的值是数字。
- 下表列出了常用的关系运算符，假定变量 `a` 为 10，变量 `b` 为 20：

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

- 关系运算符实例如下：

```
#!/bin/basha=10b=20if[ $a -eq $b ]thenecho"$a -eq $b: a 等于 b"elseecho"$a -eq $b: a 不等于 b"fiif[ $a -ne $b ]thenecho"$a -ne $b: a 不等于 b"elseecho"$a -ne $b: a 等于 b"fiif[ $a -gt $b ]thenecho"$a -gt $b: a 大于 b"elseecho"$a -gt $b: a 不大于 b"fiif[ $a -lt $b ]thenecho"$a -lt $b: a 小于 b"elseecho"$a -lt $b: a 不小于 b"fiif[ $a -ge $b ]thenecho"$a -ge $b: a 大于或等于 b"elseecho"$a -ge $b: a 小于 b"fiif[ $a -le $b ]thenecho"$a -le $b: a 小于或等于 b"elseecho"$a -le $b: a 大于 b"fi#10 -eq 20: a 不等于 b#10 -ne 20: a 不等于 b#10 -gt 20: a 不大于 b#10 -lt 20: a 小于 b#10 -ge 20: a 小于 b#10 -le 20: a 小于或等于 b
```

## 3. 布尔运算符

- 下表列出了常用的布尔运算符，假定变量 `a` 为 10，变量 `b` 为 20：

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

- 布尔运算符实例如下：

```
#!/bin/basha=10b=20if[ $a != $b ]thenecho"$a != $b: a 不等于 b"elseecho"$a == $b: a 等于 b"fiif[ $a -lt 100 -a $b -gt 15 ]thenecho"$a 小于 100 且 $b 大于 15：返回 true"elseecho"$a 小于 100 且 $b 大于 15：返回 false"fiif[ $a -lt 100 -o $b -gt 100 ]thenecho"$a 小于 100 或 $b 大于 100：返回 true"elseecho"$a 小于 100 或 $b 大于 100：返回 false"fiif[ $a -lt 5 -o $b -gt 100 ]thenecho"$a 小于 5 或 $b 大于 100：返回 true"elseecho"$a 小于 5 或 $b 大于 100：返回 false"fi#10 != 20：a 不等于 b#10 小于 100 且 20 大于 15：返回 true#10 小于 100 或 20 大于 100：返回 true#10 小于 5 或 20 大于 100：返回 false
```

## 4. 逻辑运算符

- 以下介绍 Shell 的逻辑运算符，假定变量 a 为 10，变量 b 为 20：

运算符	说明	举例
&&	逻辑的 AND	[[ \$a -lt 100 && \$b -gt 100 ]] 返回 false
	逻辑的 OR	[[ \$a -lt 100    \$b -gt 100 ]] 返回 true

- 逻辑运算符实例如下：

```
#!/bin/bash
a=10
b=20
if [[ $a -lt 100 && $b -gt 100 ]]
then echo "返回 true"
else echo "返回 false"
fi
if [[ $a -lt 100 || $b -gt 100 ]]
then echo "返回 true"
else echo "返回 false"
fi
```

## 5. 字符串运算符

- 下表列出了常用的字符串运算符，假定变量 a 为 "abc"，变量 b 为 "efg"：

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。
-z	检测字符串长度是否为0，为0返回 true。	[ -z \$a ] 返回 false。
-n	检测字符串长度是否为0，不为0返回 true。	[ -n "\$a" ] 返回 true。
\$	检测字符串是否为空，不为空返回 true。	[ \$a ] 返回 true。

- 字符串运算符实例如下：

```
#!/bin/bash
a="abc"
b="efg"
if [ $a = $b ]
then echo "a 等于 b"
else echo "a 不等于 b"
fi
if [ $a != $b ]
then echo "a 不等于 b"
else echo "a 等于 b"
fi
if [ -z $a ]
then echo "a: 字符串长度为 0"
else echo "a: 字符串长度不为 0"
fi
if [ -n $a ]
then echo "a: 字符串长度不为 0"
else echo "a: 字符串长度为 0"
fi
if [ $a ]
then echo "a: 字符串不为空"
else echo "a: 字符串为空"
fi
```

## 6. 文件测试运算符

- 文件测试运算符用于检测 Unix 文件的各种属性。
- 属性检测描述如下：

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named pipe，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。
-w file	检测文件是否可写，如果是，则返回 true。	[ -w \$file ] 返回 true。
-x file	检测文件是否可执行，如果是，则返回 true。	[ -x \$file ] 返回 true。
-s file	检测文件是否不为空（文件大小是否大于0），不为空返回 true。	[ -s \$file ] 返回 true。
-e file	检测文件（包括目录）是否存在，如果是，则返回 true。	[ -e \$file ] 返回 true。

其他检查符：

-S：判断某文件是否 socket。 -L：检测文件是否存在并且是一个符号链接。 实例

- 变量 file 表示文件 /var/www/runoob/test.sh，它的大小为 100 字节，具有 rwx 权限。下面的代码，将检测该文件的各种属性：

```
#!/bin/bash
file="/var/www/runoob/test.sh"
if [ -r $file ]
then echo "文件可读"
else echo "文件不可读"
fi
if [ -w $file ]
then echo "文件可写"
else echo "文件不可写"
fi
if [ -x $file ]
then echo "文件可执行"
else echo "文件不可执行"
fi
if [ -d $file ]
then echo "文件是目录"
else echo "文件不是目录"
fi
if [ -s $file ]
then echo "文件不为空"
else echo "文件为空"
fi
if [ -e $file ]
then echo "文件存在"
else echo "文件不存在"
fi
```