# test 命令

[Linux test 命令](https://www.linuxmi.com/bash-linux-test-command.html)

## 比较数字

如果要比较解析为数字的元素，可以使用以下比较运算符：

* **-eq**：值 1 等于值 2

* **-ge** : 值 1 大于或等于值 2

* **-gt** : 值 1 大于值 2

* **-le** : 值 1 小于或等于值 2

* **-lt** : 值 1 小于值 2

* **-ne** : 值 1 不等于值 2

### 示例测试

test 1 -eq 2 && echo "yes" || echo "no"

（在屏幕上显示“no”，因为 1 不等于 2）

test 1 -ge 2 && echo "yes" || echo "no"

（在屏幕上显示“no”，因为 1 不大于或等于 2）

test 1 -gt 2 && echo "yes" || echo "no"

（在屏幕上显示“no”，因为 1 不大于 2）

test 1 -le 2 && echo "yes" || echo "no"

（在屏幕上显示“yes”，因为 1 小于或等于 2）

test 1 -lt 2 && echo "yes" || echo "no"

（在屏幕上显示“yes”，因为 1 小于或等于 2）

test 1 -ne 2 && echo "yes" || echo "no"

（在屏幕上显示“yes”，因为 1 不等于 2）

## 比较文本

比较解析为字符串的元素时，请使用以下比较运算符：

* **=**：字符串 1 匹配字符串 2

* **!=** : 字符串 1 与字符串 2 不匹配

* **-n**：字符串长度大于0

* **-z**：字符串长度等于 0

**例子**

test "string1" = "string2" && echo "yes" || echo "no"

（在屏幕上显示“no”，因为“string1”不等于“string2”）

test "string1" != "string2" && echo "yes" || echo "no"

（在屏幕上显示“yes”，因为“string1”不等于“string2”）

test -n "string1" && echo "yes" || echo "no"

（在屏幕上显示“yes”，因为“string1”的字符串长度大于零）

test -z "string1" && echo "yes" || echo "no"

（在屏幕上显示“no”，因为“string1”的字符串长度大于零）

## 比较文件

比较文件时，请使用以下比较运算符：

* **-ef**：文件具有相同的设备和 inode 编号（它们是同一个文件）

* **-nt** : 第一个文件比第二个文件新

* **-ot**：第一个文件比第二个文件旧

* **-b**：文件存在并且是*块特殊的*

* **-c**：文件存在并且是*字符特殊的*

* **-d**：文件存在并且是目录

* **-e** : 文件存在

* **-f** : 文件存在并且是普通文件

* **-g**：文件存在并具有指定的组号

* **-G** : 文件存在且属于用户组

* **-h**或**-L**：文件存在并且是符号链接

* **-k**：文件存在并且设置了粘性位

* **-O** : 文件存在你是所有者

* **-p**：文件存在并且是命名管道

* **-r**：文件存在且可读

* **-s**：文件存在且大小大于零

* **-S** : 文件存在并且是一个socket

* **-t ：**在终端上打开文件描述符

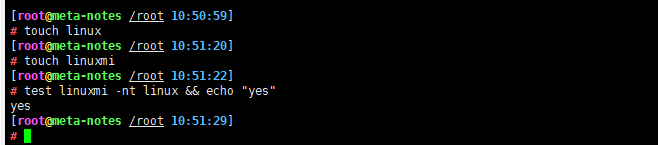
* **-u**：文件存在并且设置了 set-user-id 位

* **-w**：文件存在且可写

* **-x**：文件存在且可执行

**例子**

test linuxmi -nt linux && echo "yes"



（如果 linuxmi 比 linux 新，则显示“yes”字样，如上图）

test -e /root/linuxmi && echo "yes"



（如果 linuxmi 存在，将显示“yes”）

test -O /root/linuxmi && echo "yes"



（如果您拥有 file1，则显示“yes”字样”）

**块特殊**：文件是块设备，这意味着数据以字节块的形式读取。这些通常是设备文件，例如硬盘驱动器。

**特殊字符**：文件在您写入时立即执行，通常是串行端口等设备

## 比较多个条件

到目前为止，一切都在将一件事与另一件事进行比较，但是如果您想比较两个条件怎么办？

例如，如果一只动物有四条腿并且会发出“哞哞”的声音，它可能是一头奶牛。简单地检查四只腿并不能保证你有一头牛，但检查它发出的声音肯定可以。

要同时测试这两个条件，请使用以下语句：

test 4 -eq 4 -a "moo" = "moo" && echo "it is a cow" || echo "it is not a cow"

这里的关键部分是**-a**标志，它代表*and*。

有一种更好和更常用的方法来执行相同的测试，如下所示：

test 4 -eq 4 && test "moo" = "moo" && echo "it is a cow" || echo "it is not a cow"



**test** 命令的分支很重要。如果第一个测试 (4 = 4) 失败，则**test** 命令以非零退出代码终止。因此，我们跳转到双管道符号并且“it is not a cow”打印到标准输出。但是，如果第一个测试成功并因此 **test** 导致退出代码**0**，那么我们跳转到第一个双与号(&&)。下一条语句是另一个测试条件！

如果第二次 test 失败，我们再次跳到双管并从那里继续。然而，如果第二个 test 成功，我们跳转到第二个双& 语句，在这个例子中，它只是将“it is a cow”回显到标准输出，然后终止返回到 shell 提示符。

另一个测试比较两个语句，如果其中一个为真，则输出一个字符串。例如，要检查是否存在名为“linuxmi.txt”的文件或名为“linuxmi.py”的文件，可以使用以下命令：

这里的关键部分是**-o**代表*or*。

test -e linuxmi.txt -o -e linuxmi.py && echo "linuxmi exists" || echo "linuxmi does not exist"

有一种更好和更常用的方法来执行相同的测试，如下所示：

test -e linuxmi.txt || test -e linuxmi.py && echo "linuxmi exists" || echo "linuxmi does not exist"

## 排除 test 关键字

您实际上不需要使用单词**test**来执行比较。您所要做的就是将语句括在方括号中，如下所示：

[ -e linux.py ] && echo "linux.py exists" || echo "file1 does not exist"  
 linux.py exists

[and**]基本上与test**含义 相同**。**

现在您知道这一点，您可以改进比较多个条件，如下所示：

[ 4 -eq 4 ] && [ "moo" = "moo" ] && echo "it is a cow" || echo "it is not a cow"  
  
[ -e linuxmi.py ] || [ -e linuxmi.txt ] && echo "linuxmi exists" || echo "linuxmi does not exist"

## 总结

test 命令在脚本中更有用，因为您可以对照另一个变量测试一个变量的值并控制程序流程。在命令行上，使用它来测试文件是否存在。