**Shell测试语句**

关于条件测试命令可以使用man test命令查看

测试语句语法

1.test 测试语句

2.[ 测试语句 ]

3.[[ 测试语句 ]]

4.((测试语句))

其中1和2除了语法不一样，功能基本是一样的，而第3种语法他可以使用通配符进行匹配，第四种语法除了用于整数的比较，更多的还是进行一些整数的运算，不过平时经常用到的还是第2、3种语法结构。

条件测试类型

1. 文件测试
2. 数值测试
3. 字符串测试
4. 逻辑测试

**文件测试**

格式：

1.test 文件测试符 file

2.[ 文件测试符 file ] #中括号的两边都需要有空格

3.[[ 文件测试符 file ]] #中括号两边需要有空格

常用文件测试操作符：

-b: 文件是否存在，并且为块文件

-c：文件是否存在，并且为字符文件

-d: 文件是否存在，并且为一个目录（在linux下,万物皆为文件）

-e: 文件是否存在

-f: 文件是否存在，并且为普通文件

-s: 文件是否存在，并且文件大小大于零

-L: 文件是否存在，并且为链接文件

-r: 文件是否存在，并且文件可读

-w: 文件是否存在，并且文件可写

-x: 文件是否存在，并且文件可执行

file1 -nt file2 文件1是否比文件2新(文件新旧是按照文件修改时间来区分的)

file1 -ot file2 文件1是否比文件2旧

实例：

[root@docker ~]# touch wanger #新建一个普通文件

[root@docker ~]# test -f wanger && echo 0 || echo 1

0 #测试文件是否是普通文件，如果是输出0，否则输出1

[root@docker ~]# [ -d wanger ] && echo 0 || echo 1

1 #测试文件是否是一个目录，如果是输出0，否则输出1

[root@docker ~]# touch wangerxiao 创建一个普通文件wangerxiao

[root@docker ~]# test wanger -nt wangerxiao && echo 0 || echo 1

1 #比较wanger是否比wangerxiao新，如果为真输出0，否则输出1

[root@docker ~]# [ wanger -ot wangerxiao ] && echo 0 || echo 1

0 #比较wanger是否比wangerxiao旧，如果为真输出0，否则输出1

**字符串测试：**

格式：

1. test 字符串测试符 string
2. [ 字符串测试符 string ]
3. [[ 字符串测试符 string ]]

常用字符串测试操作符：

-z string :字符串长度是否为0

-n string :字符串长度是否不为0

string1 = string2 :string1和string2是否相等

string1 != string2 :string1和string2是否不相等

实例：

[root@docker ~]# name=wanger #定义一个变量

[root@docker ~]# [ -z $name ] && echo 0 || echo 1

1 #判断字符串是否为空，为空则返回0，否则返回1

[root@docker ~]# test -n $name && echo 0 || echo 1

1. #判断字符串是否不为空，如果是则返回0，否则返回1

[root@docker ~]# [ $name != "wanger" ] && echo 0 || echo 1

1 #判断变量是否不等于”wanger”，如果是返回0,否则返回1

[root@docker ~]# test $name = "wanger" && echo 0 || echo 1

1. #判断变量是否等于”wanger”,如果是返回0，否则返回1

**数值测试：**

格式：

test num1 数值测试操作符 num2

[ num1 数值测试操作符 num2 ]

[[ num 数值测试操作符 num2 ]]

((num1数值测试操作符num2))

常用数值测试操作符：

num1 -eq num2 : 数字1是否等于数字2

num1 -ge num2 : 数字1是否大于等于数字2

num1 -gt num2 : 数字1是否大于数字2

num1 -le num2 : 数字1是否小于等于数字2

num1 -lt num2 : 数字1是否小于数字2

num1 -ne num2 : 数字1是否不等于数字2

注意：以上几种操作符不能用在(())格式中，在(())中应使用”< > = != >= <=”比较符号来进行数值的比较

实例：

[root@docker ~]# test 6 -le 5 && echo 0 ||echo 1

0 #判断6是否小于等于5，如果是返回0，否则返回1

[2]+ Done test 64 -gt 63

[root@docker ~]# [ 6 -gt 5 ]&& echo 0 || echo 1

0 #判断6是否大于5，如果大于返回0，否则返回1

[root@docker ~]# [[ 6 = 5 ]]&& echo 0 || echo 1

1. #判断6是否等于5，如果等于返回0，否则返回1

[root@docker ~]# ((5!=6))&&echo 0 ||echo 1

1. #判断5是否等于6，如果不等于返回0，等于则返回1

**逻辑运算符：**

-a : 相当于&&，表示逻辑与，两端条件都为真，结果为真

-o : 相当于||,表示逻辑或，两端条件只要有一个为真，结果就为真

！: 逻辑非，两端相反结果为真

实例：

test=”123” #定义test变量

[root@docker ~]# [ ! -z $test -o $test = "123" ] &&echo 0 || echo 1

0 #判断test变量是否不为空或者是否等于123，如果是返回0，在[]格式中只能使用-o和-a，而不能用&&和||

[root@docker ~]# test -z $test && test $test = "123" &&echo 0 ||echo 1

1 #判断test变量是否为空并且test变量是否等于123，如果是返回0

[root@docker ~]# [[ -z $test || $test = "123" ]] &&echo 0 ||echo 1

0 #判断test变量是否为空或者test变量是否等于123，如果是返回0