条件判断

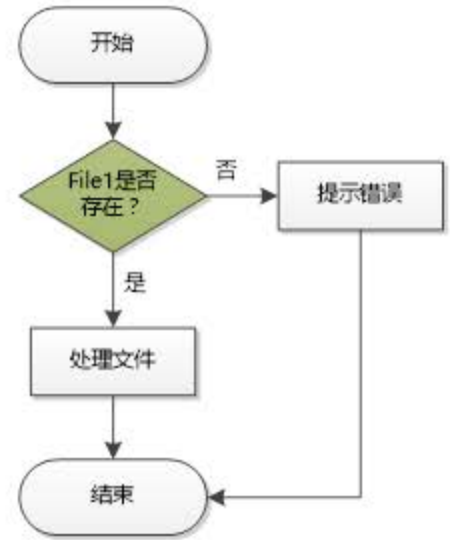
笨办法学

概述

- 条件测试结构 Test Constructs
- 文件测试操作符 File Test Operators
- 整数比较操作符 Integer Comparison
- 字符串比较操作符 String Comparison
- 组合比较 Compound Comparison
- 嵌套的if/then条件测试 Nested if/then Condition Tests

条件测试结构 Test Constructs

- **if/then结构支持以下形式：**
  - **if...fi**
  - **if...else...fi**
  - **if...elif...else...fi**
- **if/then结构判断命令列表的退出状态码是否为0**




```
1 if grep -q Bash file1
2 if echo "Linux" | grep -q "$inu"
3 if test -z "$1"
4 if /usr/bin/test -z "$1"

5 if [ -z "$1" ]
6 if [[ $condition1 || $condition2 ]]
```

```
1 v1=1
2 v2=2
3 [ "$v1" -ne "$v2" ] && echo "Not equal"
```

条件测试结构 Test Constructs

- `if test condition-true` 与`if [ condition-true ]`相同

- [...] 对格式有严格的要求：
  - [ ]中每个组件都需要用空格符分隔
  - [ ]中的变量和常量最好都用双引号引起来
- [[...]]结构比[...]结构更加通用

- 命令`test`支持的类型：
  - 数值比较
  - 字符串比较
  - 文件比较

```
1 if [ condition-true ]
2 then
3    command 1
4    command 2
5    ...
6 else
7    command 3
8    command 4
9    ...
10 fi
```

```
1 if [ -x "$filename" ]; then

2 if :; then echo "OK"; fi
```

示例：if 还可以测试任何命令

root@tomlab1:~

```
 1 #!/bin/bash
 2 # if命令还可以对命令进行测试，并是仅仅是中括号中的条件
 3
 4 # 比较两个文件是否一样
 5 date >testfile1.txt
 6 date >testfile2.txt
 7 if cmp testfile1.txt testfile2.txt &> /dev/null #抑制输出
 8   then echo "Files are identical."
 9   else echo "Files are differ."
10 fi
11
12 # 在grep中，也很有用
13 if grep -q CST testfile1.txt
14   then echo "File contains at lease one occrence of CST."
15 fi
16
17 # 复杂一些的
18 word=Linux
19 letter_sequence=inu
20 if echo "$word" | grep -q "$letter_sequence"
21 then
22   echo "$letter_sequence found in $word"
23 else
24   echo "$letter_sequence not found in $word"
25 fi
26
27 rm testfile[12].txt
28 exit 0
~
"07-001iftest.sh" 28 lines --100%--                    28,6          All
```

示例：什么是真？

- 0 = 真
- 1 = 真
- -1 = 真
- NULL = 假
- 随意字符 = 真
- 未初始化的变量 = 假

root@tomlab1:~

```
12
13 # 1是真！！！
14 echo "Testing \"1\""
15 if [ 1 ]
16 then
17   echo "1 is ture."
18 else
19   echo "1 is false."
20 fi
21 echo
22
23 # -1是真！！！
24 echo "Testing \"-1\""
25 if [ -1 ]
26 then
27   echo "-1 is ture."
28 else
29   echo "-1 is false."
30 fi
31 echo
32
33 # NULL是假
34 echo "Testing \"NULL\""
                34,1          26%
```

文件测试操作符 File test operators

-e	文件存在
-a	与-e相同，已被"弃用"了
-f	表示是一个普通文件(并不是目录或者设备文件)
-s	文件大小不为零
-d	表示这是一个目录
-b	表示这是一个块设备(软盘、光驱等)
-c	表示这是一个字符设备(键盘、声卡等)
-p	这个文件是一个管道
-h	这是一个符号链接
-L	这是一个符号链接
-S	表示这是一个socket
-t	文件(描述符)被关联到一个终端设备上

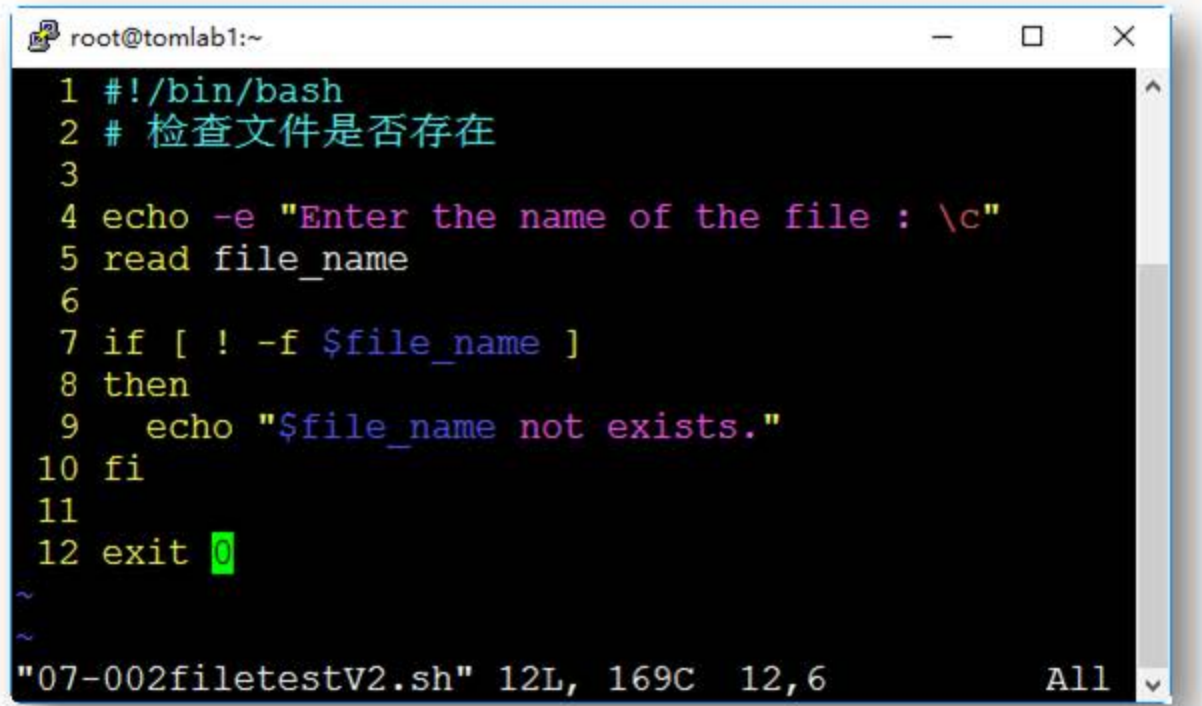
-r	当前用户是否具有可读权限
-w	当前用户是否具有可写权限
-x	当前用户是否具有可执行权限
-g	set-group-id(sgid)标记被设置到文件或目录上
-u	set-user-id (suid)标记被设置到文件上
-k	设置粘贴位
-O	当前用户是否是文件的拥有者
-G	文件的group-id是否与当前用户
-N	从上一次被读取到现在为止，文件是否被修改过
f1 **-nt** f2	文件f1比文件f2新
f1 **-ot** f2	文件f1比文件f2旧
f1 **-ef** f2	文件f1和文件f2是相同文件的硬链接

示例：文件测试操作符

root@tomlab1:~

```
 1 #!/bin/bash
 2 # 检查文件是否存在
 3
 4 echo -e "Enter the name of the file : \c"
 5 read file_name
 6
 7 if [ -f $file_name ]
 8 then
 9   :
10 else
11   echo "$file_name not exists."
12 fi
13
14 exit 0
~
"07-002filetest.sh" 14L, 176C    14,6          All
```

示例：文件测试操作符v2



整数比较操作符 Integer Comparison

- 在中括号中使用 Square Bracket

-eq	equal	等于	if ["$a" -eq "$b"]
-ne	not equal	不等于	if ["$a" -ne "$b"]
-gt	greater than	大于	if ["$a" -gt "$b"]
-ge	greater than or equal	大于等于	if ["$a" -ge "$b"]
-lt	less than	小于	if ["$a" -lt "$b"]
-le	less than or equal	小于等于	if ["$a" -le "$b"]

- 在双圆括号中使用 Double-Parentheses

<	小于	if (("$a" < "$b"))
<=	小于等于	if (("$a" <= "$b"))
>	大于	if (("$a" > "$b"))
>=	大于等于	if (("$a" >= "$b"))

字符串比较操作符 String Comparison

操作符	含义	示例	备注
=	等于	`if [ "$a" = "$b" ]`	
==	等于	`if [ "$a" == "$b" ]`	与=等价.
!=	不等于	`if [ "$a" != "$b" ]`	
<	小于	`if [[ "$a" < "$b" ]]` `if [ "$a" \< "$b" ]`	在[ ]结构中的，需要被转义.
>	大于	`if [[ "$a" > "$b" ]]` `if [ "$a" \> "$b" ]`	在[ ]结构中的，需要被转义.
-z	长度为零	`if [ -z "$a" ]`	字符串长度为零
-n	长度不为零	`if [ -n "$a" ]`	
str	非空	`if [ $a ]`	检查字符串*str*是否是非空字符串

注： 1、<和>，按照ASCII字符进行排序
2、强烈建议把字符串用双引号引用起来

示例：比较操作符

root@tomlab1:~

```
 1 #!/bin/bash
 2
 3 a=4
 4 b=5
 5
 6 if [ "$a" -ne "$b" ]
 7 then
 8   echo "$a is not equal to $b"
 9   echo "(arithmetic comparison)"
10 fi
11 echo
12
13 if [ "$a" != "$b" ]
14 then
15   echo "$a is not equal to $b"
16   echo "(string comparison)"
17 fi
18
19 exit 0
~
~
~
"07-05comparison.sh" 19L, 218C written       19,6          All
```

Tip: 文件名(testtext.txt.gz) 是否是以.gz 结尾的?

- 方法1：
  - 从最右边开始，取3个字符，判断是否是.gz
- 方法2：
  - 参数替换

```
${filename##*.}

语法是：
${var#Pattern}      匹配最短
${var##Pattern}     匹配最长
```

组合比较 Compound Comparison

- 通过**布尔操作符**来组合比较
  - `-a` 逻辑与

```
[ exp1 -a exp2 ]
```

  - `-o` 逻辑或

```
[ exp1 -o exp2 ]
```

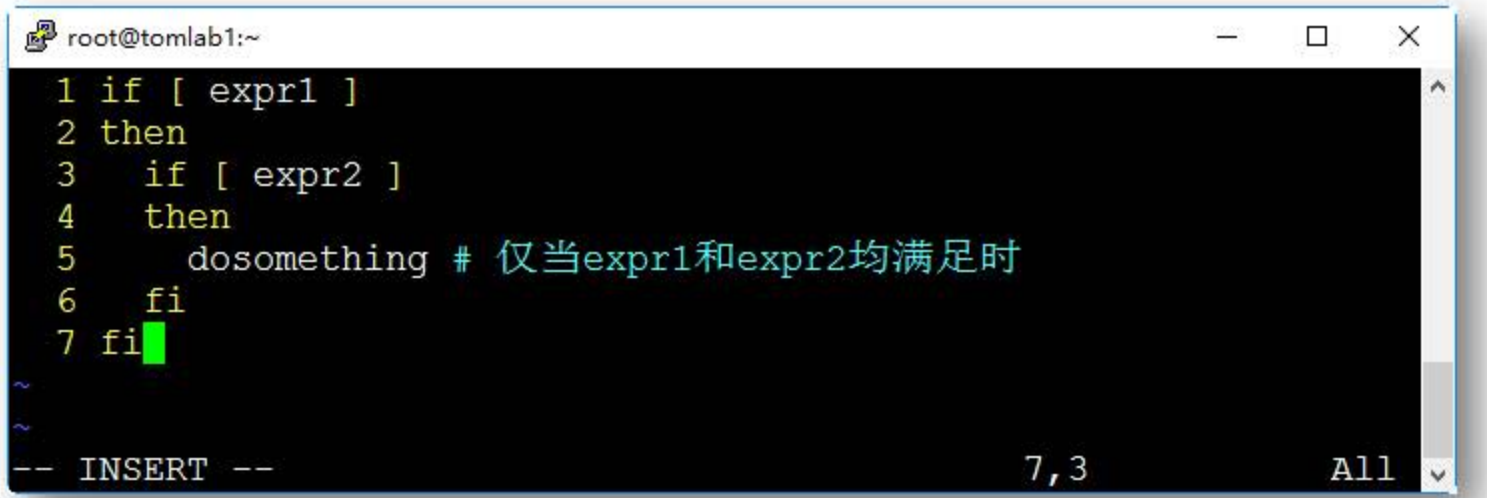
- 通过**比较操作符**来组合比较
  - `&&` 逻辑与

```
[[ exp1 && exp2 ]]
```

  - `||` 逻辑或

```
[[ exp1 || exp2 ]]
```

嵌套的if条件测试　　Nested if Condition Tests



示例脚本 /etc/X11/xinit/xinitrc

root@tomlab1:~

```
12 #       Mike A. Harris <mharris@redhat.com>
13
14 # Mandatorily source xinitrc-common, which is common code shared between the
15 # Xsession and xinitrc scripts which has been factored out to avoid duplication
16 . /etc/X11/xinit/xinitrc-common
17
18 # The user may have their own clients they want to run.  If they don't,
19 # fall back to system defaults.
20 if [ -f $HOME/.Xclients ]; then
21     exec $CK_XINIT_SESSION $SSH_AGENT $HOME/.Xclients || \
22     exec $CK_XINIT_SESSION $SSH_AGENT $HOME/.Xclients
23 elif [ -f /etc/X11/xinit/Xclients ]; then
24     exec $CK_XINIT_SESSION $SSH_AGENT /etc/X11/xinit/Xclients || \
25     exec $CK_XINIT_SESSION $SSH_AGENT /etc/X11/xinit/Xclients
26 else
27     # Failsafe settings.  Although we should never get here
28     # (we provide fallbacks in Xclients as well) it can't hurt.
29     [ -x /usr/bin/xsetroot ] && /usr/bin/xsetroot -solid '#222E45'
30     [ -x /usr/bin/xclock ] && /usr/bin/xclock -geometry 100x100-5+5 &
31     [ -x /usr/bin/xterm ] && xterm -geometry 80x50-50+150 &
32     [ -x /usr/bin/twm ] && /usr/bin/twm
33 fi
                                                              33,1          Bot
```

总结

- 条件测试结构 Test Constructs
- 文件测试操作符 File Test Operators
- 整数比较操作符 Integer Comparison
- 字符串比较操作符 String Comparison
- 组合比较 Compound Comparison
- 嵌套的if/then条件测试 Nested if/then Condition Tests