

8. Test

test 命令 是 一个 测试 命令 . 通常 使用 它 来 测试 一些 条件 是否 成立 . test 命令 的 语法 如下 . 其中 方括号 中的 内容 是 要 测试 的 表达式 . 在 Unix 系统 中 , test 命令 通常 是 一个 可执行 文件 (即 , Unix 系统 中 的 命令 通常 都是 可执行 文件) :

```
$ type [  
[ is a shell builtin  
$ which [  
/usr/bin/[  
$ ls -l /usr/bin/[  
lrwxrwxrwx 1 root root 4 Mar 27 2000 /usr/bin/[ -> test
```

在 使用 test 命令 时 , 通常 使用 方括号 来 包裹 要 测试 的 表达式 . 例如 , 测试 变量 foo 是否 等于 字符串 bar , 可以 使用 以下 命令 :

```
if [ $foo = "bar" ]
```

在 上述 命令 中 , 我们 使用 了 单引号 来 包裹 字符串 bar . 这是因为 在 shell 中 , 字符串 通常 是 以 空格 为 分隔符 的 . 如果我们 不使用 单引号 , 那么 字符串 bar 就会被 解释 为 一个 变量 名 . 因此 , 使用 单引号 可以 避免 这种 误解 .

```
if SPACE [ SPACE "$foo" SPACE = SPACE "bar" SPACE ]
```

在 上述 命令 中 , 我们 使用 了 双等号 来 表示 相等 . 这是因为 在 shell 中 , 双等号 是 用于 比较 两个 字符串 是否 相等 的 运算符 . 如果我们 使用 单等号 , 那么 命令 就会被 解释 为 一个 赋值 语句 .

test 命令 通常 是 一个 可执行 文件 . 在 Unix 系统 中 , 通常 使用 以下 命令 来 查看 test 命令 的 帮助 信息 :

test 命令 的 语法 如下 . test 命令 通常 是 一个 可执行 文件 . 在 Unix 系统 中 , 通常 使用 以下 命令 来 查看 test 命令 的 帮助 信息 :

```
if...then...else... 命令 通常 是 一个 可执行 文件 :
```

```
if [ ... ] then  
    # if-code  
else  
    # else-code  
fi
```

```
fi
# do something
esac
.
if [ ... ]
then
# do something
fi
:
```

```
if [ ... ]; then
    # do something
fi
```

```
elif
fi
:
```

```
if [ something ]; then
    echo "Something"
elif [ something_else ]; then
    echo "Something else"
else
    echo "None of the above"
fi
```

```
[something]
echo "Something"
, [ something_else ]
.
[something_else]
echo "Something else"
.
"None of the above"
```

```
X
(-1, 0, 1, hello, bye
).
(
- 1
Dave
):
```

```
$ X=5
$ export X
$ ./test.sh
... output of test.sh ...
$ X=hello
$ ./test.sh
... output of test.sh ...
$ X=test.sh
$ ./test.sh
... output of test.sh ...
```

```
$X
(/etc/hosts)
```

```
#!/bin/sh
if [ "$X" -lt "0" ]
```



```
[ -f $X ] && echo "X is a file" || echo "X is not a file"
[ -n $X ] && echo "X is of non-zero length" || \
    echo "X is of zero length"
```

0. 测试用例 (Test Case) 的编写
 1. 测试用例的编写原则
 2. 测试用例的编写步骤
 3. 测试用例的编写模板
 4. 测试用例的编写示例
 5. 测试用例的编写注意事项
 6. 测试用例的编写工具
 7. 测试用例的编写文档
 8. 测试用例的编写流程
 9. 测试用例的编写方法
 10. 测试用例的编写技巧

X□ □□ □□ □□ □□□ □□ □ □ □□ □□ □□□□ :

```
test.sh: [: integer expression expected before -lt
test.sh: [: integer expression expected before -gt
test.sh: [: integer expression expected before -le
test.sh: [: integer expression expected before -ge
```

```

-lt, -gt, -le  -ge  1000  1000  1000000  1000  1000000  . != 1000
1000  1000  "5"  1000  1000000  "Hello"  1000  1000000  1000  1000000  1000
1000  1000  1000000  .  1000000  1000  1000000  1000  1000000  1000  1000  1000
1000  1000  1000000  1000  :
```

```
echo -en "Please guess the magic number: "
read X
echo $X | grep "[^0-9]" > /dev/null 2>&1
if [ "$?" -eq "0" ]; then
    # If the grep found something other than 0-9
    # then it's not an integer.
    echo "Sorry, wanted a number"
else
    # The grep found only 0-9, so it's an integer.
    # We can safely do a test on it.
    if [ "$X" = "7" ]; then
        echo "You entered the magic number!"
    fi
fi
```

[illegible]

while test [-n "\$X"] :

```
#!/bin/sh
X=0
while [ -n "$X" ]
do
    echo "Enter some text (RETURN to quit)"
    read X
    echo "You said: $X"
done
```

RETURN [-n "\$X"] (X 0).
Justin Heath . [-n "\$X"] \$X
 . \$X [-n "\$X"]
 :
 :

```
$ ./test2.sh
Enter some text (RETURN to quit)
fred
You said: fred
Enter some text (RETURN to quit)
wilma
You said: wilma
Enter some text (RETURN to quit)
```

:
 :

\$

:
 :

```
#!/bin/sh
X=0
while [ -n "$X" ]
do
    echo "Enter some text (RETURN to quit)"
    read X
    if [ -n "$X" ]; then
        echo "You said: $X"
    fi
done
```

if ["\$X" -lt "0"] then echo "X is less than zero" fi

```
if [ "$X" -lt "0" ]
then
    echo "X is less than zero"
fi
```

.....

```
if [ ! -n "$X" ]; then
    echo "You said: $X"
fi
```

if then fi echo . if then fi

```
if [ ! -n "$X" ]
    echo "You said: $X"
```

then fi

Revision #3

Created 16 January 2024 13:25:25 by Enigma

Updated 16 January 2024 13:48:11 by Enigma