



shell编程-功能语句

主讲：小美老师

创客引领未来

扫微信二维码 获取更多信息





lemon

昵称：小姜老师
华清创客学院，嵌入式讲师

makerU

课程目录

- 说明性语句
- 常用功能性语句
- 测试语句

$R = \text{expr} + -'$

整
字
文
↑
-ne
-eq
!=
==
<
<=

+-
逻辑

输入出



read... class
不分空格)

- shell 程序由零或多条shell语句构成。 shell语句包括三类:说明性语句、功能性语句和结构性语句。

✓ 说明性语句:

以#号开始到该行结束, 不被解释执行

✓ 功能性语句:

任意的shell命令、用户程序或其它shell程序。

✓ 结构性语句:

条件测试语句、多路分支语句、循环语句、循环控制语句等。

说明性语句(注释行)

Q. 怎么设置 shell 类型?

- 注释行可以出现在程序中的任何位置,既可以单独占用一行,也可以接在执行语句的后面. 以#号开始到所在行的行尾部分,都不被解释执行. 例如:

~~#!/bin/sh~~

#

本程序说明

#

command_1

command_2

.....

下面程序段的说明

command_m

.....

command_n

.....

告诉OS用哪种类型的shell来
解释执行该程序

command_2的语句说明

commandn_语句的说明

ubuntu 常用 bash

.....
-n

常用功能性语句(命令)

- read从标准输入读入一行，并赋值给后面的变量，其语法为：

read var

把读入的数据全部赋给var

read var1 var2 var3 (last 不会空格)

把读入行中的第一个单词(word)赋给var1，第二个单词赋给var2，……把其余所有的词赋给最后一个变量。

- 如果执行read语句时标准输入无数据，则程序在此停留等候，直到数据的到来或被终止运行。

应用实例

```
# example1 for read  
echo "Input your name: \c"  
read username  
echo "Your name is $username"
```

```
#example2 for read  
echo "Input date with format yyyy mm dd: \c"  
read year month day  
echo "Today is $year/$month/$day, right?"  
echo "Press enter to confirm and continue\c"  
read answer  
echo "I know the date, bye!"
```

- 算术运算命令expr主要用于进行简单的整数运算，包括加(+)、减(-)、乘(*)、整除(/)和求模(%)等操作。例如：

```
$ expr 12 + 5 \* 3
```

```
27
```

```
$ expr 3 - 8 / 2
```

```
-1
```

```
$ num=9
```

```
$ sum=`expr $num \* 6`
```

```
$ echo $sum
```

```
54
```

反斜线去掉*号的元字符含义

反撇号引用命令的运行结果

test语句

- test语句可测试三种对象：
字符串 整数 文件属性
- 每种测试对象都有若干测试操作符
- 例如：

```
test "$answer" = "yes"
```

变量answer的值是否为字符串yes

```
test $num -eq 18
```

变量num的值是否为整数18

```
test -d tmp
```

测试tmp是否为一个目录名

字符串测试

- `s1 = s2`

测试两个字符串的内容是否完全一样

- `s1 != s2`

测试两个字符串的内容是否有差异

- `-z s1`

测试s1 字符串的长度是否为0

- `-n s1`

测试s1 字符串的长度是否不为0

✓ \$?
0 成功
1 不成
2 有问题

整数测试

- $a \text{ -eq } b$
- $a \text{ -ne } b$
- $a \text{ -gt } b$
- $a \text{ -ge } b$
- $a \text{ -lt } b$
- $a \text{ -le } b$

测试a 与b 是否相等

测试a 与b 是否不相等

测试a 是否大于b

测试a 是否大于等于b

测试a 是否小于b

测试a 是否小于等于b

文件测试

- `-d name` 测试name 是否为一个目录
- `-e name` 测试一个文件是否存在
- `-f name` 测试name 是否为普通文件
- `-L name` 测试name 是否为符号链接
- `-r name` 测试name 文件是否存在且为可读
- `-w name` 测试name 文件是否存在且为可写
- `-x name` 测试name 文件是否存在且为可执行
- `-s name` 测试name 文件是否存在且其长度不为0
- `f1 -nt f2` 测试文件f1 是否比文件f2 更新
- `f1 -ot f2` 测试文件f1 是否比文件f2 更旧



扫微信二维码 获取更多信息