



## shell编程-分支语句

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

- 条件语句
- 多路分支语句

先回忆 C 语言的, 然后再变化.

# 结构性语句

- 结构性语句主要根据程序的运行状态、输入数据、变量的取值、控制信号以及运行时间等因素来控制程序的运行流程。
- 主要包括：条件测试语句（两路分支）、多路分支语句、循环语句、循环控制语句和后台执行语句等。

## 条件语句

- if...then...fi

- 语法结构:

if 表达式

then 命令表

fi

- 如果表达式为真，则执行命令表中的命令；否则退出if语句，即执行fi后面的语句。
- if和fi是条件语句的语句括号，必须成对使用；
- 命令表中的命令可以是一条，也可以是若干条。

## 复习-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

## 实例

shell程序prog2.sh（测试命令行参数是否为已存在的文件或目录）。用法为：

./prog2.sh file

代码如下：

```
#The statement of if...then...fi          (注释语句)
if [ -f $1 ]                             (测试参数是否为文件)
then
    echo "File $1 exists"                (引用变量值)
fi
if [ -d $HOME/$1 ]                       (测试参数是否为目录)
then
    echo "File $1 is a directory"        (引用变量值)
fi
```

创客, 引领未来 | You Make the Future

## 实例

执行prog2程序:

```
$ ./prog2.sh prog1.sh
```

```
File prog1.sh exists
```

*\$0为prog2.sh; \$1为prog1.sh, 是一个已存在的文件.*

```
$ ./prog2.sh backup
```

```
File backup is a directory
```

*\$0为prog2.sh; \$1为backup, 是一个已存在的目录.*

如果不带参数,或大于一个参数运行prog2, 例如:

```
$ ./prog2.sh (或 $ ./prog2.sh file1 file2)
```

会出现什么结果?

创客, 引领未来 | *You Make the Future*



## 条件语句

### if...then...else...fi

- 语法结构为:

```
if      表达式
then 命令表1
else 命令表2
fi
```

- 如果表达式为真, 则执行命令表1中的命令, 再退出if语句; 否则执行命令表2中的语句, 再退出if语句.

注意: 无论表达式是否为真, 都有语句要执行.

# test命令的使用

test命令测试的条件成立时，命令返回值为真(0)，否则返回值为假(非0)。

## 方式1

```
test $name -eq $1  
echo $?
```

## 方式2

```
if test -f $filename  
then  
.....  
fi
```

## 方式3

```
if [ -f $filename ]  
then  
...  
fi
```

用方括号替代test语句

至少一个空格

至少一个空格

## 实例

例子: shell程序prog3.sh, 用法为:  
./prog3.sh file

内容如下:

```
#The statement of if...then...else...fi
if [ -d $1 ]
then
    echo "$1 is a directory"
    exit (退出当前的shell程序)
else
    if [ -f $1 ]
    then
        echo "$1 is a common file"
    else
        echo "unknown"
    fi
fi
```

## 实例

运行prog3.sh程序:

```
$ ./prog3.sh backup  
backup is a directory
```

```
$ ./prog3.sh prog1  
prog1 is a common file
```

```
$ ./prog3.sh abc  
unknown
```

prog3.sh是对prog2.sh的优化, 逻辑结构更加清晰合理!

# 多路分支语句

## case...esac

多路分支语句case用于多重条件测试，语法结构清晰自然。其语法为：

✓ case 字符串变量 in ← case语句只能检测字符串变量

✓ 模式1)

命令表1

各模式中可用文件名元字符,以右括号结束

模式2

✓ 模式3)

命令表2

一次可以匹配多个模式用“|”分开

;;

命令表以单独的双分号行结束,退出case语句

.....

✓ 模式n)

命令表n

模式 n常写为字符\* 表示所有其它模式

..

最后一个双分号行可以省略

✓ esac

# 实例

实例. 程序prog4.sh检查用户输入的文件名,用法为:

```
./prog4.sh    string_name
# The statement of  case...esac
if  [  $#  -eq  0  ]
then
    echo  "No argument is declared"
    exit
fi
case  $1  in
    file1)
        echo  "User selects file1"
        ;;
    file2)
        echo  "User selects file2"
        ;;
    *)
        echo  "You must select either file1 or file2!"
        ;;
esac
```

创客, 引领未来 | *You Make the Future*



扫微信二维码 获取更多信息