

## 分支结构

---

### 内容提要

1. 进一步熟悉测试表达式的使用
2. 掌握 `if` 和 `case` 语句的语法

## if 结构

---

### if 语句

`if` 语句的结构如下：

#### 单分支结构

```
if condition
then
    statement1
    statement2
    .....
fi
```

#### 双分支结构

```
if condition
then
    statement1
    statement2
    .....
else
    statement3
    .....
fi
```

#### 多分支结构

```
if condition1
then
    statement1
    statement2
    .....
elif condition2
then
    statement3
    statement4
    .....
elif condition3
then
    statement5
    statement6
    .....
fi
```

#### 多分支结构

```
if condition1
then
    statement1
    statement2
    .....
elif condition2
then
    statement3
    statement4
    .....
elif condition3
then
    statement5
    statement6
    .....
else
    statement7
    .....
fi
```

## if 语句使用举例

下面举几个使用 if 语句的例子：

例1：

```
#!/bin/bash
# filename: areyouok.sh
echo "Are you OK ?"
read answer
# 在 if 的条件判断部分使用扩展的 test 语句 [[...]]
# [[]] 中可以使用正则表达式进行条件匹配
if [[ $answer == [Yy]* || $answer == [Mm]aybe ]]
then
    echo "Glad to hear it."
fi
```

例2：

```
#!/bin/bash
# filename: myssh.sh
# 在条件测试中判断位置参数的个数 $#
if [ "$#" -eq "2" ]; then
    ssh $1@$2
else
    echo "Usage: $0 <username> <FQDN or IPAdress>"
fi
```

例3：

```
#!/bin/bash
# filename: matching_lines.sh

if [ $# -ne 2 ]; then
    echo "Usage: $0 <filename> <regular_expression>"
    exit 1
fi

filename=$1
re=$2

if [ ! -e $filename ]; then
    echo "$1 not exist."
    exit 2
fi
```

```
num_matching=`cat "$filename" | grep --count "$re"`  
  
if [ $num_matching -eq 0 ]; then  
    echo "No matching \"$re\" in $filename."  
else  
    echo "Matching \"$re\" $num_matching lines in $filename."  
fi
```

#### 例4:

```
#!/bin/bash  
# filename: useronline.sh  
# if 语句可以嵌套使用  
if test $# -eq 1  
then  
    # if 的条件部分除了使用 test 语句之外，还可以使用普通的命令进行测试  
    # 当该命令正确执行（ $?=0）返回真，否则（ $?<0）返回假  
    if who|grep $1 >/dev/null  
    then  
        echo $1 is active.  
    else  
        echo $1 is not active.  
    fi  
else  
    echo "Usage: $0 <username>"  
    exit  
fi
```

#### 例5:

```
#!/bin/bash  
# filename: askage.sh  
echo "How old are you?"  
read age  
# 使用 Shell算术运算符 (()) 进行条件测试  
if ((age<0||age>120)); then  
    echo "Out of range !"  
    exit 1  
fi  
# 使用多分支 if 语句  
if ((age>=0&&age<13)); then  
    echo "Child !"  
elif ((age>=13&&age<20)); then  
    echo "Callan !"  
elif ((age>=20&&age<30)); then  
    echo "P III !"  
elif ((age>=30&&age<40)); then  
    echo "P IV !"  
else  
    echo "Sorry I asked."  
fi
```

## case 结构

### case 语句

case 语句的结构如下:

多分支结构

```
case condition in  
pattern1)
```

```
statements1
;;
pattern2)
statements2
;;
.....
patternN)
statementsN
;;
esac
```

## case 语句使用举例

下面举几个使用 **case** 语句的例子：

### 例1：

```
#!/bin/bash
# filename: switch-lang.sh
# usage: . ./switch-lang.sh
cat <<EOF
Please choice a locale:
1) -- en_US.utf8
2) -- en_US.iso88591
3) -- zh_CN.gb18030
4) -- zh_CN.utf8
EOF
read choice
case $choice in
1)
    export LANG="en_US.utf8"
    ;;
2)
    export LANG="en_US.iso88591"
    ;;
3)
    export LANG="zh_CN.gb18030"
    ;;
4)
    export LANG="zh_CN.utf8"
    ;;
*)
    echo Select error.
    ;;
esac
```

### 例2：

```
#!/bin/bash
# filename: switch-lang2.sh
# switch-lang.sh 的命令行参数版
case $1 in
# 在 pattern 部分可以使用 | 分割多个匹配项目
1|eu) export LANG="en_US.utf8" ;;
2|e) export LANG="en_US.iso88591" ;;
3|c) export LANG="zh_CN.gb18030" ;;
4|cu) export LANG="zh_CN.utf8" ;;
*) echo "Usage : . $0 <1|2|3|4|e|eu|c|cu>" ;;
esac
```

- 显示源文件
- 登录