

## 循环结构

---

### 内容提要

1. 掌握 **while** 语句和 **until** 语句的语法
2. 掌握 **for** 语句的两种语法及适用范围
3. 学会构造 **foreach** 型循环中的列表
4. 进一步巩固条件测试表达式的使用

## 当型循环和直到型循环

---

### while 语句和 until 语句

**while** 和 **until** 语句的结构如下:

#### 当型循环结构

```
while condition
do
    statement1
    statement2
    .....
done
```

当 **condition** 条件满足时执行循环体，否则退出循环

#### 直到型循环结构

```
until condition
do
    statement1
    statement2
    .....
done
```

当 **condition** 条件满足时结束循环，否则继续执行循环体

### while 语句和 until 语句使用举例

下面举几个使用 **while** 和 **until** 语句的例子:

#### 例1:

```
#!/bin/bash
# filename: sum1to100_while.sh
# 在进行数值计算时使用((...))实现C语言风格的循环很方便
((i=0,s=0))
# 使用当型循环
while ((i<100)); do ((i++,s+=i)); done
```

```
echo sum\$(1..100)\=$s
```

上面的代码与下面的传统代码等价：

```
#!/bin/sh
i=0
s=0
while [ $i -lt 100 ]
do
    i=`expr $i + 1` # 或 let "i = $i + 1"
    s=`expr $s + $i` # 或 let "s = $s + $i"
done
echo sum\$(1..100)\=$s
```

例2：

```
#!/bin/bash
# filename: sum1to100_until.sh
# 在进行数值计算时使用((...))实现C语言风格的循环很方便
((i=0,s=0))
# 使用直到型循环
until ((i==100)); do ((i++,s+=i)); done
echo sum\$(1..100)\=$s
```

例3：

```
#!/bin/bash
# filename: scp_files_from_remote.sh
# 显示存有文件名列表的文件 files.txt 通过管道传递给 while 循环
# read x 语句每次读取一个文件名进行处理（执行循环体）
cat files.txt |while read x
do
    scp user@remote:"\"$x\"" .
done
```

您可以参考上面的例子 `myssh.sh` 改造脚本 `scp_files_from_remote.sh`，从而接受命令行参数。

例4：

```
#!/bin/bash
# filename: guess_number.sh
# 对被猜的数进行赋值
num=8
# 使用永真循环、条件退出的方式接收用户的猜测并进行判断
while [ 1 ]; do
    echo "Please input a number: "
    read user_num

    if [ $user_num -lt $num ]; then
        echo "The number you inputed is less then my NUMBER."
    elif [ $user_num -gt $num ]; then
        echo "The number you inputed is greater then my NUMBER."
    elif [ $user_num -eq $num ]; then
        echo "Congratulate: my NUMBER is $num."
        break
    fi
done
```

`bash` 提供了两个循环控制语句：

- **break**：用来跳出循环
- **continue**：只会跳过本次循环，忽略本次循环剩余的代码，进入循环的下一迭代

## foreach型循环和计数型循环

## for 语句

for 语句的结构如下:

### foreach型循环结构

```
for variable in WordList
do
    statement1
    statement2
    .....
done
```

in WordList 可省略, 省略时相当于 in "\$@"

### C语言型的for循环结构

```
for (( exp1; exp2; exp3 ))
do
    statement1
    statement2
    .....
done
```

## for 语句使用举例

下面举几个使用 for 语句的例子:

例1:

```
#!/bin/bash
# filename: constant_as_list.sh
# 使用字面列表作为 WordList
for x in centos ubuntu gentoo opensuse
do
    echo "$x"
done
```

例2:

```
#!/bin/bash
# filename: seq_as_list.sh
# 使用 seq 命令生成 WordList
for x in `seq 100`
do
    echo "$x"
done
```

seq 命令的语法如下:

```
# 生成序列 [1..LAST], 步长为 1
seq LAST
# 生成序列 [FIRST..LAST], 步长为 1
seq FIRST LAST
# 生成序列 [FIRST..LAST], 步长为 INCREMENT
seq FIRST INCREMENT LAST
```

例3:

```
#!/bin/bash
```

```
# filename: star_as_list.sh
# 使用 * 作为 WordList, 表示当前目录下的所有文件
# 将当前目录下的所有的大写文件名改为小写文件名
for fname in *; do
    # 使用命令替换生成小写的文件名赋予新的变量 fn
    fn=$(echo $fname | tr A-Z a-z)
    # 若新生成的小写文件名与原文件名不同, 将文件名改为小写的名字
    [[ $fname == $fn ]] || mv $fname $fn
    # 上面的语句与下面的两条语句均等效
    # [[ $fname != $fn ]] && mv $fname $fn
    # if [[ $fname != $fn ]] ; then mv $fname $fn ; fi
done
```

#### 例4:

```
#!/bin/bash
# filename: ls_as_list.sh
# 使用 ls 的结果作为 WordList
if [ "$#" -ne "1" ]; then
    echo "Usage: $0 <directory>"
    exit 1
fi
# 将指定目录 ($1) 下的所有文件名改为小写字母
for i in `ls $1`; do mv $i $(echo $i | tr [A-Z] [a-z]); done
```

#### 例5:

```
#!/bin/bash
# filename: file-content_as_list.sh
# 使用文本文件内容作为 WordList
for x in `cat files.txt`
do
    echo "$x"
done
```

#### 例6:

```
#!/bin/bash
# filename: change_file_SUFFIX.sh
# 将当前目录下所有以 ($1) 为后缀的文件改为以 ($2) 为后缀的文件
if [ "$#" -eq "2" ]; then
    for f in *.$1; do mv $f `basename $f . $1`.$2; done
else
    echo "Usage: $0 <SUFFIX1> <SUFFIX2>"
    echo "Exmaple: $0 txt doc"
fi
```

#### 例7:

```
#!/bin/bash
# filename: addusers.sh
# 成批添加50个用户, 使用C语言风格的 for 语句
for (( num=1; num<=50; num++ ))
do
    if ((num<10)); then
        st="st0$num"
    else
        st="st$num"
    fi
    useradd $st
    # 删除用户的口令, 以便用户可以无口令登录, 首次登录后要用 passwd 命令设置口令
    passwd -d $st
done
```

#### 例8:

```
#!/bin/bash
# filename: vbird_to_utf8.sh
# 将下载的 vbird 离线文件转换为 UTF-8。
# 先做如下准备工作
cd
mkdir vbird
cd vbird
wget http://ccw2.ocit.edu.tw/c095571/basic.zip
wget http://ccw2.ocit.edu.tw/c095571/server.zip
unzip basic.zip
unzip server.zip
mkdir basic-utf8 server-utf8
# 使用 for 嵌套循环对 basic 和 server 目录下的文件进行转换
for dir in basic server
do
    dest=${dir}-utf8
    for i in $dir/*.htm; do
        F=`basename $i`
        iconv -f BIG-5 -t UTF-8 -o $dest/$F $i
        # if [[ "$?" != 0 ]] ; then echo $F; fi
        perl -pi -e "s|charset=big5|charset=UTF-8|sgi;" $dest/$F
    done
done
# 再做些后续处理
cp -f basic-utf8/* basic
cp -f server-utf8/* server
rm -rf basic-utf8/ server-utf8/
```

- 显示源文件
- 登录