

By scpman

BLOG:[www.scpman.com](http://www.scpman.com)

QQ 群: 158601242

## 第二天 读懂你的输入

在第一天过程中，我们初步接触 `shell`，学习了下一个关于驴的故事。

回想下: `$0 $1 $2..`

今天的课程中我们将学习对用户输出的参数进行判断。

例子: 如果你说别人坏话? 那就要挨打了。

故事 2:

当你吃零食时，猫儿在你身边叫的时候，你听到了，然后你将手中的零食，分了一块给小猫，小猫得到零食后，就离开你了。很显然这只猫是吃货。

下面我们拆分这个故事，并实现它。放开那只猫让你来。

这个故事里，听，分，离开是关键字

主人公听到后，做出的反应是分一点给猫，然后猫得到后就走了

### 2、1 read

`shell:`

`read` 从命令行读取一行用户输入

根据编程来看用的最多的是:

`read -p "你的描述" ppp`

意思:

在命令行输出一个提示用户输入的等待，然后定义了变量叫 `ppp` (名字随便取，好的习惯是，变量名要用意义，好比，西瓜为什么叫西瓜不叫土豆? 因为这是常识)

然后用户输入后，将用户的输入读进变量 `ppp`

我们接到了变量，然后就应该干嘛? 对这变量使用之。

`2.1.sh`

`#!/bin/sh`

`#code by scpman`

`#read`

`read -p "What's your Name: " name`

`echo "Hi," $name`

`echo "now Time : `date`"`

```
scpman# cat 2.1.sh
#!/bin/sh
#code by scpman
#read
read -p "what's your Name: " name

echo "Hi," $name
echo "now Time : `date`"
scpman# ./2.1.sh
what's your Name: scpman
Hi, scpman
now Time : Sun Jan 20 06:35:43 CST 2013
```

这个符号在键盘的按键1  
左边，可以直接执  
行shell命令  
`date`直接在脚本，执  
行date命令

在这个脚本中，我们提示用户输入名字，然后输出 hi,用户输出的名字，并在脚本中输出了当前系统时间。

## 2、2 if

在 2.1 中，我们已经接到用户输入了，那你给猫一只狗，猫会吃吗？任何时候，我们都不相信用户的输入！猫有原则，给只狗就是不吃，只吃老鼠（主人公在吃的零食）

如果你给的是老鼠，猫就吃，否则，就不吃。

那就用 if 来了解这个事吧

语法：

if ....then...fi

如果给的是老鼠，就吃

if ...else....fi

如果给的不是老鼠，而是别的呢

If....elif...elif....else...fi

如果给的是老鼠，或者给饼干，给别的等

if 也在一些固定的表达式用来判断一些问题：

列出一些：

条件表达式

文件表达式

if [ -f file ] 如果文件存在

if [ -d ... ] 如果目录存在

if [ -s file ] 如果文件存在且非空

if [ -r file ] 如果文件存在且可读

if [ -w file ] 如果文件存在且可写

if [ -x file ] 如果文件存在且可执行

整数变量表达式

if [ int1 -eq int2 ] 如果 int1 等于 int2

if [ int1 -ne int2 ] 如果不等于

if [ int1 -ge int2 ] 如果 >=

if [ int1 -gt int2 ] 如果 >

if [ int1 -le int2 ] 如果 <=

if [ int1 -lt int2 ] 如果 <

字符串变量表达式

|                               |                              |
|-------------------------------|------------------------------|
| if [ \$a = \$b ]              | 如果 string1 等于 string2        |
|                               | 字符串允许使用赋值号做等号                |
| if [ \$string1 != \$string2 ] | 如果 string1 不等于 string2       |
| if [ -n \$string ]            | 如果 string 非空(非 0)，返回 0(true) |
| if [ -z \$string ]            | 如果 string 为空                 |
| if [ \$sting ]                | 如果 string 非空，返回 0 (和-n 类似)   |

条件表达式引用变量要带\$，并最好加"\$变量名"（通过失败的经验得来的）

## 2.3 实例

### 1、如果给的是老鼠，猫就吃

思考：可能的情况：

- A 如果你只是挥了下手，什么都没给
- B 如果你给的不是老鼠
- C 如果你给了老鼠，却扔给了别人

```
#!/bin/sh
#code by scpman
#cat's food
read -p "Please input the cat's food:" food
if [ -n "$food" ]
then
    echo $food
fi
```

```
scpman# cat 2.3.1.sh
#!/bin/sh
#code by scpman
#cat's food
read -p "Please input the cat's food: " food
if [ -n "$food" ]
then
    echo the food:$food
fi
scpman# sh 2.3.1.sh
Please input the cat's food: pig
the food:pig
scpman#
```

sh是直接调用这个脚本，用sh的方式执行，这样执行时，可以在开关不写#!/bin/sh，写是没错的。后面会讲调试脚本

在这里我们用了-n检测变量是否在

你很大方，给了头pig

要是输入就输出，没有输入就退出，并提示用户什么都没输入。

如果得到了输入就进行判断，如果给的是老鼠就说谢谢，如果不是老鼠就不要，并提示用户

# vi 2.3.2.sh

```
#!/bin/sh
#code by scpman
#cat's food
read -p "Please input the cat's food: " food
if [ -n "$food" ]
then
    echo the food:$food
```

```
else
    echo You give cat a null
fi
```

```
if [ "$food" = 'laoshu' ]
then
    echo 3Q,cat love laoshu
elif [ "$food" = 'pig' ]
then
    echo sorry,cat not love pig
else
    echo cat not like others!
```

Fi

# sh 2.3.2.sh

Please input the cat's food:

You give cat a null #这里为空应该提示完直接退出？

cat not like others! #这一句怎么也输出了？

Shell:

exit 脚本退出

# vi 2.3.3.sh

```
#!/bin/sh
```

```
#code by scpman
```

```
#cat's food
```

```
read -p "Please input the cat's food: " food
```

```
if [ -n "$food" ]
```

```
then
```

```
    echo the food:$food
```

```
else
```

```
    echo You give cat a null
```

```
    exit #####看这里，#号代表注释，这样当条件走这时，就直接提示并退出 了
```

```
fi
```

```
if [ "$food" = 'laoshu' ]
```

```
then
```

```
    echo 3Q,cat love laoshu
```

```
elif [ "$food" = 'pig' ]
```

```
then
```

```
    echo sorry,cat not love pig
```

```
else
```

```
    echo cat not like others!
```

Fi

# sh 2.3.3.sh

Please input the cat's food:

You give cat a null

这次就对了

```
# sh 2.3.3.sh
```

Please input the cat's food: pig

the food:pig

sorry,cat not love pig

```
scpman# sh 2.3.3.sh
```

Please input the cat's food: lkjdsfldsf

the food:lkjdsfldsf

cat not like others!

```
scpman# sh 2.3.3.sh
```

Please input the cat's food: laoshu

the food:laoshu

3Q,cat love laoshu

当然用不了，写这么多行，就能搞定，本书的所有例子，不要求有多少行，只要求思路清晰希望大家少写 NB 的代码（只有自己想几次才看懂的代码！）

看到着，免得被大家骂，咱们赶紧来一个有用的例子吧！

## 2.4

### 小实例：

判断/usr/test 目录在不在，如果不在就输出，然后建立这个目录

建立完这个目录后，读/etc/passwd 文件，找出文件中所有用户并在/usr/test/下建立出各用户名字对应的文件

编程：先看懂你要做什么，然后喝口水，想好了思路再开始，一气呵成。

外行：好，这个需求简单，几分钟写完了，然后再看下要求，再重复来，键盘也是有尊严的

```
# vi 2.4.sh
```

```
#!/bin/sh
```

```
#code by scpman
```

```
#mkdir ,touch file
```

```
mulu="/usr/test"
```

```
if [ -d "$mulu" ]
```

```
then
```

```
    echo $mulu is have
```

```
else
```

```
    mkdir $mulu
```

```
    echo $mulu create ok
```

```
fi
```

```
for username in `cat /etc/passwd |grep -v "^#" | awk -F: '{print $1}`
```

```
do
```

```
cd $mulu
```

```
touch $mulu/$username
```

```
scpman# cat 2.4.sh
#!/bin/sh
#code by scpman
#mkdir ,touch file
mulu="/usr/test"
if [ -d "$mulu" ]
then
    echo $mulu is have
else
    mkdir $mulu
    echo $mulu create ok
fi
for username in `cat /etc/passwd |grep -v "^#" | awk -F: '{print $1}`
do
    cd $mulu
    touch $mulu/$username
done
ls -l $mulu
```

我们定义了变量mulu

我们用if -d表达式来检查目录是否存在

cd到这个目录，cd命令很重要，很多时候，出错都是因为路径的问题，cd的意义这时就显得尤为重要

用for循环读文件，明天讲

ls 最后列出下，文件是否已经创建

```
done
```

```
ls -l $mulu
```

脚本中的 for 循环部分明天讲

```
# sh 2.4.sh
```

```
/usr/test create ok
```

```
total 0
```

```
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 _dhcp
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 _pflogd
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 bin
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 bind
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 daemon
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 games
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 kmem
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 mailnull
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 man
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 news
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 nobody
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 operator
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 pop
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 proxy
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 root
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 scpman
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 smmsp
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 sshd
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 toor
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 tty
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 uucp
-rw-r--r-- 1 root wheel 0 Jan 20 07:42 www
```

第二天结束了，你是否有所收获？最起码知道了，猫是吃什么的吧？