

作业二：shell 编程的返回值实现

邵柯欣

专业：信息与计算科学学号：3200103310

2022 年 6 月 28 日

1 代码 (分为两个部分)

来源：Beginning+Linux+Programming 的第 48 页

1.1 第一部分：子函数

创建yes_or_no函数，用来根据用户的反馈确定返回值。

```
#!/bin/bash

yes_or_no(){
    echo "Is your name $* ?"      %打印带有用户输入行参的语句
    while true
    do
        echo -n "Enter yes or no:" %第一次提示用户选择
        read x                    %存储用户的输入
        case "$x" in
            y | yes ) return 0;;
            n | no )  return 0;;
            * )          echo "Answer yes or no"
        esac
    done
}
```

1.2 第二部分: main 函数

```
echo "Original parameter are $*"

if yes_or_no "$1"
then
    echo "Hi $1, nice name"
else
    echo "Never mind"
fi
exit 0
```

2 对文件权限的修改

创建一个My_name文件，将上面两段代码写到该文件中。

但是，如果你直接用

```
./My_name
```

命令去尝试运行这个文件，你会得到

```
bash: ./My_name: 权限不足
```

的错误反馈。

这段程序并没有像我们所期待的那样运行起来。

为了解决这个问题，我们输入

```
ls -l
```

来查看我们对这个文件的权限，我们会发现，My_name的权限状态为

```
rw-rw-r--
```

表明我们没有运行的权限。

于是，我们输入

```
chmod 777 ~/My_name
```

来更改权限状态。

再次输入

```
ls -l
```

我们可以看到My_name的权限状态为

```
rw-rw-rw-
```

接下来，我们就可以通过

```
./My_name
```

命令去尝试运行这个文件了。

3 尝试不同的输入

通过观察程序我们可以看到，程序中出现了\$* \$1两个变量。
根据课本前面的介绍,我们知道\$*表示./My_name的所有形参,\$1表示./My_name的第一个行参。

同理可知，\$i表示./My_name的第i个行参(i>0)。
特别注意的是，当i=0时，\$i表示./My_name自身的名字，即./My_name。

3.1 输入输出样例一

如果我们运行

```
./My_name Rick Neil
```

此时我们输入了两个行参，所以\$*上的值为，运行结果如样例一所示。

```
$ ./My_name
Original parameter are
Is your name ?
```

3.2 输入输出样例二

如果我们直接运行

```
./My_name Rick Neil
```

此时我们输入了两个行参，所以\$*的值是Rick Neil，\$1的值是Rick运行结果如样例二所示。

```
$ ./My_name Rick Neil
Original parameter are Rick Neil
Is your name Rick ?
```

3.3 输入输出样例三

如果我们直接运行

```
./My_name Rick
```

此时我们只输入一个行参，所以\$*和\$1的值相同都为Rick，运行结果如样例三所示。

```
$ ./My_name Rick
Original parameter are Rick
Is your name Rick ?
```

4 总结

通过该shell程序样例，我掌握了以下知识点：

- 掌握了如何用shell编程实现输入和输出返回值；
- 了解了`$*`, `$i(i>=0)`变量的含义；
- 能熟练运用`chown`, `chmod`函数更改shell程序文件的所有者和权限。
- ...