

退出和退出状态码

笨办法学 

退出和退出状态码

- ▶ Bash命令、脚本和脚本中的函数都会返回退出状态代码
 - ▶ 成功，返回0
 - ▶ 不成功，返回非零值。通常都被解释、翻译成一个错误码



- ▶ 在Bash 脚本中：
 - ▶ 执行`exit nnn`，将退出码nnn传递给shell，`nnn`必须是0 – 255的十进制数
 - ▶ 执行`exit`，由`exit`之前最后执行的命令来决定
 - ▶ 没有`exit`，由最后执行的命令来决定
- ▶ `$?` 保存了最后所执行的命令、脚本的退出状态码

保留的退出状态代码

退出码	含义
1	通用错误
2	shell内建命令使用错误(Bash文档上有说明)
126	命令调用不能执行
127	"command not found"
128	exit的参数错误
128+n	信号"n"的致命错误
130	用Control-C来结束脚本
255*	超出范围的退出状态

示例：退出状态码

```
root@tomlab1:~  
1 #!/bin/bash  
2 # 退出状态码  
3  
4 echo hello  
5 echo $?           # 0  
6  
7 badcmd  
8 echo $?           # 非零的值，比如：127  
9  
10 echo  
11  
12 true # true是Bash的一个内置命令  
13 echo "exit status of \"true\" = $?" # 0  
14  
15 ! true  
16 echo "exit status of \"true\" = $?" # 1  
17  
18 exit 113          # 非零退出码，通常表示错误  
~  
~  
"06-01ExitCode.sh" 18L, 301C written      18,48-65      All
```

作业

- ▶ 先做纸上推演，然后再命令行验证



```
1 bash$ ls | bad_commnad  
  
2 bash$ echo $?  
  
3 bash$ ! ls | bad_commnad  
  
4 bash$ echo $?
```

作业答案

- ▶ 先做纸上推演，然后再命令行验证



```
1 bash$ ls | bad_commnad
  -bash: bad_commnad: command not found

2 bash$ echo $?
  127

3 bash$ ! ls | bad_commnad
  -bash: bad_commnad: command not found

4 bash$ echo $?
  0
```

作业答案

- ▶ 先做纸上推演，然后再命令行验证



```
1 bash$ true | true
2 bash$ echo $?
0

3 bash$ true | false
4 bash$ echo $?
1

5 bash$ false | true
6 bash$ echo $?
0
```

```
7 bash$ ! true | true
8 bash$ echo $?
1

9 bash$ ! true | false
10 bash$ echo $?
0

11 bash$ ! false | false
12 bash$ echo $?
0
```


Bash的set -o pipefail



```
$ man set
```

```
.....
```

```
-o option-name
```

```
The option-name can be one of the following:
```

```
.....
```

pipefail

If set, the return value of a pipeline is the value of the last (**rightmost**) command to exit with a non-zero status, or zero if all commands in the pipeline exit successfully. This option is **disabled by default**.