

# 常用bash shell整理

gaccob

2014 年 1 月 26 日

## 1. bash脚本的参数

手工的处理方式:

- \$0: 指脚本自己, 例如test.sh.
- \$1: 第一个参数, 依次类推\$2, \$3 ...
- \$#: 参数的个数.
- \$@: 所有的参数, 是一个数组.
- \$\*: 所有的参数, 与\$@不一样的是它是一个字符串.

一个简单的使用sample:

```
1  for arg in "$@"
2  do
3      echo $arg
4  done
```

getopts的处理方式:

(这里只讨论短选项, 长选项需要getopt, 会复杂一些).

```
1  while getopts "a:bc" arg
2  do
3      case $arg in
4          a) echo -e "argument a: $OPTARG\n" ;;
5          b) echo -e "argument b\n" ;;
6          c) echo -e "argument c\n" ;;
7          ?) echo -e "usage: ./test <-a *> <-b> <-c>" ;;
8      esac
9  done
```

上面这个sample中:

- "a:bc"代表了-a需要参数, b和c是无参数的选项.
- case ? 代表了任何不识别的选项.

## 2. 获取当前脚本目录

```
1 function get_pwd()  
2 {  
3     echo "$( cd "$( dirname "$0" )" && pwd )";  
4 }
```

## 3. 遍历嵌套文件夹

```
1 function loop_dir()  
2 {  
3     dir="$1";  
4     for sub in $dir  
5     do  
6         if [ -d "$sub" ]; then  
7             loop_dir $sub;  
8         else  
9             # do something  
10        fi  
11    done  
12 }
```

## 4. expect的典型用法

通过expect完成自动scp的过程参考:

```
1 function scp_target()  
2 {  
3     scp_src="$1";  
4     scp_dst="$2";  
5     scp_dst_passwd="$3";  
6     expect -c "set timeout -1;  
7         spawn scp -p -o StrictHostKeyChecking=no -r $scp_src  
8             $scp_dst;  
9         expect "*assword:*" { send $scp_dst_passwd\r\n; };  
10        expect eof {exit;}; "  
11 }
```

## 5. 进程锁

通过flock实现进程锁，尤其在执行shell脚本时比较实用，具体可以参考man page.

一个典型用法如下：

```
1  # non-block (fail exit with code 1), exclusive lock
2  flock -xn /tmp/test.lock -c test.sh
3
4  # non-block, wait 5 seconds
5  flock -n -w 5 /tmp/test.lock -c test.sh
```

## 6. 绑定cpu

在某些情况下，可以通过绑定server进程到固定的cpu，减少context切换，一定程度上提高性能。这个时候需要用到taskset.

taskset的具体用法可以参照man page，一个常见的sample如下：

```
1  taskset -c 0,2,4-6 -p 14285
```