

# 常用Bash技巧总结2

Alex / 20180218 / [free\\_learner@163.com](mailto:free_learner@163.com) / [learning-archive.org](http://learning-archive.org)

更新于2023-07-12，主要是文字排版上的更新，内容基本保持不变。

总结最近学习的一些Bash技巧，包括整数和浮点数运算、间接变量引用、大小字母转换、使用find查找文件并执行命令、使用sed进行文本替换。

## 一、算术运算

### 1. 整数运算（三种方法）

```
x=4
let "y = $x * 3 + 2"
y=`expr $x \* + 2`
y=$(( $x * 3 + 2 ))
```

上面的例子分别使用了 `let`、`expr` 和 `(( ))` 三种不同方法进行整数的运算。注意使用 `expr` 时需要对乘号 `*` 转义。

### 2. 浮点数运算（使用bc）

```
x=4
echo "$x * 0.5 + 1.3" | bc
```

Bash本身不能进行浮点数运算，需要借助bc命令。

## 二、间接变量引用（两种方法）

```
a=3
b=a
eval echo \$$b    ## 输出3
echo ${!b}       ## 输出3
```

变量 `b` 的内容是另一个变量 `a` 的名字，通过变量 `b` 访问变量 `a` 的内容，即为间接变量引用。

## 三、大写字母转换成小写字母（三种方法）

```
var="HELLO WORLD"
echo $var | tr '[:upper:]' '[:lower:]'
echo $var | tr '[A-Z]' '[a-z]'
echo $var | awk '{print tolower($0)}'
```

上面三种方法都是将 HELLO WORLD 转换成 hello world。

## 四、使用find查找文件并执行命令

假设要进行这样一个操作，在当前目录下有若干个子文件夹，每个子文件夹下有一些文本文件，需要提取每个文本文件的第二行，如果用find命令实现呢？

```
find . -name "*.txt" -exec sh -c "cat {} | sed -n 2p" \;
```

. 表示在当前文件夹下搜索； "\*.txt" 表示所有后缀名为 .txt 的文件； sh -c "cat {} | sed -n 2p" 表示对每个搜索到的文件先使用cat显示内容，再传递给sed命令提取第二行内容， {} 表示每个搜索到的文件， \; 表示选项 -exec 的最后一个参数，即标志命令结束；注意 sh -c "cat {} | sed -n 2p" 包含了两个命令，通过这种方式实现顺序执行多个命令；如果需要分别执行多个命令，可以使用连续使用多个 -exec 选项，比如 -exec ls {} \; -exec cat {} \; 表示先显示文件名，再显示文件内容。

## 五、使用sed进行文本替换

sed替换文本的基本格式是 sed 's/old\_string/new\_string/' < file.txt，即将 file.txt 中的 old\_string 替换为 new\_string。下面是一些例子：

```
## 替换第一个满足条件的字符，下面的例子只替换第一个o
```

```
echo hollo | sed 's/o/e/'
```

```
## 替换所有满足条件的字符，下面的例子将所有3换成5
```

```
echo 353 | sed 's/3/5/g'
```

```
## 处理空格，需要使用\转义
```

```
echo "hello world" | sed 's/hello\ world/good night'
```

```
## 与正则表达式结合，将等号后面的内容去掉
```

```
echo "a=b+c" | sed 's/=.*//'
```