

shell中组合命令的方法



在shell中，有两种组合命令的方法，在介绍这两种方法之前，我们先来看一个小示例。

目前有两个目录，这两个目录分别为/test1和/test2，目录中分别存在如下文件。

```
[www.zsythink.net]# ls /test1 /test2
/test1:
t1  t2  t3

/test2:
t1
[www.zsythink.net]#
[www.zsythink.net]#
```

此刻，如果我们想要统计这两个目录中一共有多少个文件，该怎样编写shell呢，编写如下的shell命令，能够达到目的吗？

```
[www.zsythink.net]# ls /test1; ls /test2 | wc -l
```

上述命令是无法统计出两个目录中的文件数量的，因为执行结果如下。

```
[www.zsythink.net]# ls /test1; ls /test2 | wc -l
t1  t2  t3
1
[www.zsythink.net]#
[www.zsythink.net]#
```

从上述执行结果可以看出，前两条命令并没有一起执行，而是第一条命令执行完毕后，再把后面的命令当做一个整体执行了，那么如果我们想要让前两条命令当做一个整体执行该怎样做呢？

没错，这时候就要用到我们之前说的两种方法，两种方法的示例如下。

```
[www.zsythink.net]# ls /test1; ls /test2 | wc -l
4
[www.zsythink.net]# { ls /test1; ls /test2; } | wc -l;
4
[www.zsythink.net]#
[www.zsythink.net]#
```

上图中，我们分别使用小括号"()"与 大括号"{}"将前两个命令组合在了一起。两种方法都能达到我们的目的，但是，这两种方法是有所不同的，先说说第一种，当把两条命令括起以后，这两条命令相当于在当前shell的子shell中执行，也就是说，这两条命令并不是在当前进程中运行的，而是在当前进程的子进程中运行的，而使不同，被大括号括起的命令是在当前进程中执行的。那么，我们怎样证明上述理论呢，我们可以做一个小实验，如下图。

```
[www.zsythink.net]# cd;exit
[www.zsythink.net]#
[www.zsythink.net]# { cd; exit; }
logout
Connection closed by foreign host.


Disconnected from remote host(192.168.1.108:22) at 21:45:02.

Type 'help' to learn how to use Xshell prompt.
[c=V~]$
```

当我们使用小括号将命令聚合以后，因为是在子进程中执行，所以即使exit命令生效，也是退出子进程，而不是退出当前shell进程，但是如果使用大括号将命令组合在前shell进程中运行命令，所以，当exit命令生效时，则会退出当前shell进程。

好了，这就是使用"大括号"与使用"小括号"的不同之处，其实，在使用大括号时，还需要注意一些额外的注意点，细心如你一定已经发现了，上述所有使用"大括号"括号内的所有命令都用空格隔开了，而且大括号内的每个命令都必须以分号";"结尾，即使是括号内的最后一个命令，也需要以分号结尾，而且需要用空格与大括号隔开。使用这种方法的注意点，其实，这种用大括号聚合命令的方法是使用了函数中"代码块"的语法，在函数中，一个函数的代码块就会用大括号括起，表示是一个函数的代码块，我们也可以这样理解，我们将多个命令用大括号括起，表示他们是一个代码块，是一个整体，我们可以把这种没有函数名只有函数体的代码块理解为匿名函数，当然，解shell的函数，没有关系，只需要记住这种用法即可。





我的微信公众号

关注"实用运维笔记"微信公众号，当博客
中有新文章时，可第一时间得知哦~

Shell (bash)