**&符号**

这两天要在服务器端一直运行一个Python脚本，当然就想到了在命令后面加&符号  
**$ python /data/python/server.py >python.log &**

说明：  
1、 > 表示把标准输出（STDOUT）重定向到 那个文件，这里重定向到了python.log  
2、 & 表示在后台执行脚本  
这样可以到达目的，但是，我们退出shell窗口的时候，必须用exit命令来退出，否则，出之后，该进程也会随着shell的消失而消失（退出、关闭）

## 使用nohup(not hang up)

**$ nohup python /data/python/server.py > python.log3 2>&1 &**

说明：  
1、1是标准输出（STDOUT）的文件描述符，2是标准错误（STDERR）的文件描述符  
1> python.log 简化为 > python.log，表示把标准输出重定向到python.log这个文件  
2、2>&1 表示把标准错误重定向到标准输出，这里&1表示标准输出  
为什么需要将标准错误重定向到标准输出的原因，是因为标准错误没有缓冲区，而STDOUT有。  
这就会导致 commond > python.log 2> python.log 文件python.log被两次打开，而STDOUT和 STDERR将会竞争覆盖，这肯定不是我门想要的  
3、好了，我们现在可以直接关闭shell窗口（我用的是SecureCRT，用的比较多的还有Xshell），而不用再输入exit这个命令来退出shell了

 command &让进程在后台运行

 jobs 查看后台运行的进程

 fg %n 让后台运行的进程n到前台来

 bg %n 让进程n到后台去

 kill %n 杀死job