

在没有执行`kill -USR1 `cat ${pid_path}``之前，即便已经对文件执行了`mv`命令而改变了文件名，nginx还是会向新命名的文件“`xxx.log_20130909`”照常写入日志数据的。原因在于：linux系统中，内核是根据文件描述符来找文件的。

## 一 对linux文件描述符的理解

文件描述符是linux内核为每个打开的文件命名的一个整数标识。

linux内核为每一个进程生成(或者说维护)一个“文件描述符表”，这个文件描述符表记录的是“此进程所打开的文件(进行标识)”。

在这里的环境中，nginx就是一个运行中的进程，这个进程早就打开了一个日志文件，在文件描述符表是记录了文件的。

即便日志文件的路径改变了，但是还是能够找到(根据文件描述符表可以定位)。

## 二 cat nginx.pid含义

当执行命令“`kill -USR1 `cat ${pid_path}``”的时候，nginx.pid文件中保存的其实就是一个数字(自己可以打开看一下，我这里是894)，nginx 将其主进程的 pid (进程号)写入到了nginx.pid 文件中，所以可以通过cat命令直接拿到其主进程号,直接操作指定的进程号。

`kill -USR1 `cat ${pid_path}`` 就等同于

`kill -USR1 894` #指定发信号(USR1)信号给这个进程编号。

## 三 kill -USR1 `cat \${pid\_path}`

在linux系统中，linux是通过信号与“正在运行的进程”进行通信的。linux系统中，也很多预定义好的信号，像SIGHUP。USR1是用户自定义信号。可以理解为：进程自己定义接到这个信号该干嘛(也就是进程编写者自己确定收到这个信号干嘛还是什么都不做都行，完全交给开发人员自己决定)。而在nginx中，它自己编写了代码处理当我接到USR1信号的时候，让nginx重新打开日志文件。具体原理如下：

1、nginx 的主进程收到USR1信号，会重新打开日志文件(以nginx配置文件中的日志名称命名,就是配置文件中`access_log`项所设置的值，如果文件不存在，会自动创建一个新的文件`xxx.log`)。

2、然后把日志文件的拥有者改为“工作进程(worker进程)”，目的是让worker进程就具备了对日志文件的读写权限（master和worker通常以不同用户运行，所以需要改变拥有者）。

3、nginx主进程会关闭重名的日志文件(也就是刚才使用mv命令重命名成xxx.log\_20130909.log的文件)，并通知工作进程使用新打开的日志文件(刚才主进程打开的文件xxx.log)。具体实现上更细化点就是，主进程把USR1信号发给worker，worker接到这个信号后，会重新打开日志文件(也就是配置文件中约定的xxx.log)