链滴

Linux 中的 nohup 与 &

作者：Patrick

原文链接：https://ld246.com/article/1540734519564

来源网站：链滴

之前在解决退出SSH后相关java程序无法后台运行是这样解决的：

Linux将jar注册为service

这样做的好处是可以方便的使用start,stop,status等命令控制，并且可以加入开机自启等等。

但是有些时候对于某些不重要的脚本，或者其他程序，并不想如此设置。只想当前在后台运行就可以。

Linux下使用nohup

命令末尾的“&”

Unix/Linux下一般比如想让某个程序在后台运行，很多都是使用 & 在程序结尾来让程序自动运行。
如我们要运行weblogic在后台：

./startWebLogic.sh &

但是除weblogic这种守护进程外，普通程序如果终端关闭，那么程序也会被关闭。为了能够后台运行那么我们就可以使用nohup这个命令。比如我们有个startWebLogic.sh需要在后台运行，那么就使用ohup：

nohup ./startWebLogic.sh &

提示：

[~]$ appending output to nohup.out

嗯，证明运行成功，同时把程序运行的输出信息放到当前目录的 nohup.out 文件中去，起到一个记录og的作用。

nohup 命令

- 用途：LINUX命令用法，不挂断地运行命令。
- 语法： nohup Command [Arg …] [&]
- 描述：nohup 命令运行由 Command 参数和任何相关的 Arg 参数指定的命令，忽略所有挂断（SIHUP）信号。

在注销后使用 nohup 命令运行后台中的程序。要运行后台中的 nohup 命令，添加 & （ 表示“and的符号）到命令的尾部。

使用重定向解决nohup.out无写权限等问题

场景

有时在使用nohup命令时，会遇到nohup无写权限等问题：

nohup: ignoring input and appending output to nohup.out' **nohup: failed to run command /etc/nginx_check.sh': Permission denied**

Linux重定向

0、1和2分别表示标准输入、标准输出和标准错误信息输出，可以用来指定需要重定向的标准输入或出。

在一般使用时，默认的是标准输出，既1。当我们需要特殊用途时，可以使用其他标号。

例如，将某个程序的错误信息输出到log文件中：./program 2>log。

这样标准输出还是在屏幕上，但是错误信息会输出到log文件中。

另外，也可以实现0，1，2之间的重定向。2>&1：将错误信息重定向到标准输出。

Linux下还有一个特殊的文件/dev/null，它就像一个无底洞，所有重定向到它的信息都会消失得无影踪。

这一点非常有用，当我们不需要回显程序的所有信息时，就可以将输出重定向到/dev/null。

如果想要正常输出和错误信息都不显示，则要把标准输出和标准错误都重定向到/dev/null， 例如：

```
ls 1>/dev/null 2>/dev/null
```

还有一种做法是将错误重定向到标准输出，然后再重定向到 /dev/null，例如：

```
ls >/dev/null 2>&1
```

注意：此处的顺序不能更改，否则达不到想要的效果，此时先将标准输出重定向到 /dev/null，

然后将标准错误重定向到标准输出。

由于标准输出已经重定向到了/dev/null，因此标准错误也会重定向到/dev/null，于是一切静悄悄。

nohup的重定向

将nohup.out重定向至一个有写入权限的路径，或者直接扔到/dev/null中。

```
nohup ./program >/dev/null 2>/dev/null &
```

或者

```
nohup ./program >/dev/null 2>&1 &
```

小结

如果一个命令只使用&来标识，则表示其在当前Session中，运行在后台。如果当前Session关闭或者前的terminal工具关闭，则其附属的进程将会关闭。

正常运行的后台进程都是需要nohup与&，两者并行使用的，方可保证其在后台正常运行。