**[linux 如何让程序后台执行](https://www.cnblogs.com/maseng/p/3739030.html)**

$ (./test.sh &)

$ setsid ./test.sh &

$ nohup ./test.sh &

具体的转自：http://digdeeply.org/archives/06281192.html

在Linux中，如果要让进程在后台运行，一般情况下，我们在命令后面加上&即可，实际上，这样是将命令放入到一个作业队列中了：

|  |  |
| --- | --- |
| Example | |
| 1  2  3  4  5 | $ ./test.sh &  [1] 17208    $ jobs -l  [1]+ 17208 Running                 ./test.sh & |

对于已经在前台执行的命令，也可以重新放到后台执行，首先按ctrl+z暂停已经运行的进程，然后使用bg命令将停止的作业放到后台运行：

|  |  |
| --- | --- |
| Example | |
| 1  2  3  4  5  6  7  8 | $ ./test.sh  [1]+  Stopped                 ./test.sh    $ bg %1  [1]+ ./test.sh &    $ jobs -l  [1]+ 22794 Running                 ./test.sh & |

但是如上方到后台执行的进程，其父进程还是当前终端shell的进程，而一旦父进程退出，则会发送hangup信号给所有子进程，子进程收到 hangup以后也会退出。如果我们要在退出shell的时候继续运行进程，则需要使用nohup忽略hangup信号，或者setsid将将父进程设为 init进程(进程号为1)

|  |  |  |
| --- | --- | --- |
| Example | | |
| 1  2  3  4  5  6  7  8  9 | | $ echo $$  21734    $ nohup ./test.sh &  [1] 29016    $ ps -ef | grep test  515      29710 21734  0 11:47 pts/12   00:00:00 /bin/sh ./test.sh  515      29713 21734  0 11:47 pts/12   00:00:00 grep test |
| Example | | |
| 1  2  3  4  5  6 | $ setsid ./test.sh &  [1] 409    $ ps -ef | grep test  515        410     1  0 11:49 ?        00:00:00 /bin/sh ./test.sh  515        413 21734  0 11:49 pts/12   00:00:00 grep test | |

上面的试验演示了使用nohup/setsid加上&使进程在后台运行，同时不受当前shell退出的影响。那么对于已经在后台运行的进程，该怎么办呢？可以使用disown命令：

|  |  |
| --- | --- |
| Example | |
| 1  2  3  4  5  6  7  8  9  10  11 | $ ./test.sh &  [1] 2539    $ jobs -l  [1]+  2539 Running                 ./test.sh &    $ disown -h %1    $ ps -ef | grep test  515        410     1  0 11:49 ?        00:00:00 /bin/sh ./test.sh  515       2542 21734  0 11:52 pts/12   00:00:00 grep test |

另外还有一种方法，即使将进程在一个subshell中执行，其实这和setsid异曲同工。方法很简单，将命令用括号() 括起来即可：

|  |  |
| --- | --- |
| Example | |
| 1  2  3  4  5 | $ (./test.sh &)    $ ps -ef | grep test  515        410     1  0 11:49 ?        00:00:00 /bin/sh ./test.sh  515      12483 21734  0 11:59 pts/12   00:00:00 grep test |

注：本文试验环境为Red Hat Enterprise Linux AS release 4 (Nahant Update 5),shell为/bin/bash，不同的OS和shell可能命令有些不一样。例如AIX的ksh，没有disown，但是可以使用nohup -p PID来获得disown同样的效果。

还有一种更加强大的方式是使用screen，首先创建一个断开模式的虚拟终端，然后用-r选项重新连接这个虚拟终端，在其中执行的任何命令，都能达到nohup的效果，这在有多个命令需要在后台连续执行的时候比较方便：

|  |  |
| --- | --- |
| Example | |
| 1  2  3  4  5  6  7  8 | $ screen -dmS screen\_test    $ screen -list  There is a screen on:          27963.screen\_test       (Detached)  1 Socket in /tmp/uscreens/S-jiangfeng.    $ screen -r screen\_test |