链滴

Linux 技巧：让进程在后台可靠运行的几种方法

作者：sean
原文链接：https://ld246.com/article/1385291050906
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

<p><p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;backgr
und-color:#FFFFFF;">我们经常会碰到这样的问题，用 telnet/ssh 登录了远程的 Linux 服务器，
行了一些耗时较长的任务， </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;"> 结果却由于网络的不稳定导致任务中途失败。如何让命令提交后不受本
关闭终端窗口/网络断开连接的干扰呢？下面举了一些例子， </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">您可以针对不同的场景选择不同的方式来处理这个问题。 </p
gt;
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">
  </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">场景：</span&gt

  </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">   <span style="font-family:arial, nsimsun, sa
s-serif;line-height:normal;background-color:#FFFFFF;">如果只是临时有一个命令需要长时间
行，什么方法能最简便的保证它在后台稳定运行呢？ </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">解决方法：</spa
>
  </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">   <span style="font-family:arial, nsimsun, sa
s-serif;line-height:normal;background-color:#FFFFFF;">我们知道，当用户注销（logout）或
网络断开时，终端会收到 HUP（hangup）信号从而关闭其所有子进程。因此，我们的解决办法就有
种途径：要么让进程忽略 HUP 信号，要么让进程运行在新的会话里从而成为不属于此终端的子进程
 </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">
  </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">   1. nohup</s
an> </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">   <span style="
ont-family:arial, nsimsun, sans-serif;line-height:normal;background-color:#FFFFFF;">nohup
无疑是我们首先想到的办法。顾名思义，nohup 的用途就是让提交的命令忽略 hangup 信号。</s
an> </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;"><span style="font-family:arial, nsimsun, sans-serif;line-height:normal;ba
kground-color:#FFFFFF;"><span style="font-family:arial, nsimsun, sans-serif;line-height:
ormal;background-color:#FFFFFF;">   &nb
p;nohup 的使用是十分方便的，只需在要处理的命令前加上 nohup 即可，标准输出和标准错误缺省
被重定向到 nohup.out 文件中。一般我们可在结尾加上"&"</s
rong><span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background-
olor:#FFFFFF;">来将命令同时放入后台运行，也可用"><em>filename&lt
/em> 2>&1"<span style="font-family:arial, nsimsun, sans-serif;line-height
normal;background-color:#FFFFFF;">来更改缺省的重定向文件名。
 </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">   nohup
示例：<span style="background-color:#FFFFFF;font-family:arial, ns
msun, sans-serif;line-height:normal;">   <span style="
ackground-color:#FFFFFF;font-family:arial, nsimsun, sans-serif;line-height:normal;">
   </p>

<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;"> </p>
<pre class="prettyprint">[root@pvcent107 ~]# nohup ping <a href="https://ld246.com/
orward?goto=http%3A%2F%2Fwww.ibm.com" target="_blank" rel="nofollow ugc">www.ibm
com &
[1] 3059
nohup: appending output to `nohup.out'
[root@pvcent107 ~]# ps -ef |grep 3059
root 3059 984 0 21:06 pts/3 00:00:00 ping <a href="https://ld246.com/forward?goto=
ttp%3A%2F%2Fwww.ibm.com" target="_blank" rel="nofollow ugc">www.ibm.com
root 3067 984 0 21:06 pts/3 00:00:00 grep 3059
[root@pvcent107 ~]#
 <p> <span style="font-family:arial, nsimsun, sans-serif;line-height:nor
al;background-color:#FFFFFF;">
  &l
;/p>   2.setsid

<p>
 </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">   <span style="
ont-family:arial, nsimsun, sans-serif;line-height:normal;background-color:#FFFFFF;">nohup
无疑能通过忽略 HUP 信号来使我们的进程避免中途被中断，但如果我们换个角度思考，如果我们的
程不属于接受 HUP 信号的终端的子进程，那么自然也就不会受到 HUP 信号的影响了。setsid 就能
助我们做到这一点。 </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">   setsid 示例：</span&gt
<span style="background-color:#FFFFFF;font-family:arial, nsimsun, sans-serif;line-height:no
mal;">   <span style="background-color:#FFFFFF;font-
amily:arial, nsimsun, sans-serif;line-height:normal;">   &l
;/p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;"> </p>
<pre class="prettyprint">[root@pvcent107 ~]# setsid ping <a href="https://ld246.com/
orward?goto=http%3A%2F%2Fwww.ibm.com" target="_blank" rel="nofollow ugc">www.ibm
com
[root@pvcent107 ~]# ps -ef |grep <a href="https://ld246.com/forward?goto=http%3A%2F%
Fwww.ibm.com" target="_blank" rel="nofollow ugc">www.ibm.com
root 31094 1 0 07:28 ? 00:00:00 ping <a href="https://ld246.com/forward?goto=htt
%3A%2F%2Fwww.ibm.com" target="_blank" rel="nofollow ugc">www.ibm.com
root 31102 29217 0 07:29 pts/4 00:00:00 grep <a href="https://ld246.com/forward?goto
http%3A%2F%2Fwww.ibm.com" target="_blank" rel="nofollow ugc">www.ibm.com<br

[root@pvcent107 ~]#
<span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background-color:#FF
FFF;">值得注意的是，上例中我们的进程 ID(PID)为 31094，而它的父 ID（PPID）为 1（即为 init
进程 ID），并不是当前终端的进程 ID。请将此例与 nohup
<span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background-color:#FF
FFF;">中的父 ID 做比较。

<p>
 </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">
  </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">3.&： </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background

color:#FFFFFF;"> </p>
<p style="font-family:arial, nsimsun, sans-serif;font-size:0.76em;background-color:#FFFFFF;
>   这里还有一个关于 subshell 的小技巧。我们知道，将一个或多命名包含在“()”中就能让这些命令在子 shell 中运行中，从而扩展出很多有趣的功能，我们现在要论的就是其中之一。 </p>
<p style="font-family:arial, nsimsun, sans-serif;font-size:0.76em;background-color:#FFFFFF;
>   当我们将"&"也放入“()”内之后，我们就会发现所提交的业并不在作业列表中，也就是说，是无法通过 jobs 来查看的。让我们来看看为什么这样就能躲过 HUP 信号的影响吧。 </p>
   subshell 示例:
br>
<span style="background-color:#FFFFFF;font-family:arial, nsimsun, sans-serif;line-height:no
mal;">
<span style="background-color:#FFFFFF;font-family:arial, nsimsun, sans-serif;line-height:no
mal;">
<p>
 </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;"> </p>
<pre class="prettyprint">[root@pvcent107 ~]# (ping <a href="https://ld246.com/forwa
d?goto=http%3A%2F%2Fwww.ibm.com" target="_blank" rel="nofollow ugc">www.ibm.com
/a> &)
[root@pvcent107 ~]# ps -ef |grep <a href="https://ld246.com/forward?goto=http%3A%2F%
Fwww.ibm.com" target="_blank" rel="nofollow ugc">www.ibm.com
root 16270 1 0 14:13 pts/4 00:00:00 ping <a href="https://ld246.com/forward?goto=h
tp%3A%2F%2Fwww.ibm.com" target="_blank" rel="nofollow ugc">www.ibm.com
root 16278 15362 0 14:13 pts/4 00:00:00 grep <a href="https://ld246.com/forward?goto
http%3A%2F%2Fwww.ibm.com" target="_blank" rel="nofollow ugc">www.ibm.com<br

[root@pvcent107 ~]#
<span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background-color:#FF
FFF;">从上例中可以看出，新提交的进程的父 ID（PPID）为 1（init 进程的 PID），并不是当前端的进程 ID。因此并不属于当前终端的子进程，从而也就不会受到当前终端的 HUP 信号的影响了。&
t;/span>
<p>
 </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">
  </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">
  </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">场景：</span&gt

  </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">   <span style="font-family:arial, nsimsun, sa
s-serif;line-height:normal;background-color:#FFFFFF;">我们已经知道，如果事先在命令前加上 nohup 或者 setsid 就可以避免 HUP 信号的影响。但是如果我们未加任何处理就已经提交了命令，如何补救才能让它避免 HUP 信号的影响呢？
  </p><
r>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">解决方法：</spa
> </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background
color:#FFFFFF;">   <span style="font-family:arial, nsimsun, sa
s-serif;line-height:normal;background-color:#FFFFFF;">这时想加 nohup 或者 setsid 已经为已晚，只能通过作业调度和 disown 来解决这个问题了。<span style="font-family:arial, nsimsun,

sans-serif;line-height:normal;background-color:#FFFFFF;">让我们来看一下 disown 的帮助信：   </p>
<p>   </p>
<pre class="prettyprint">disown [-ar] [-h] [jobspec ...]
Without options, each jobspec is removed from the table of
active jobs. If the -h option is given, each jobspec is not
removed from the table, but is marked so that SIGHUP is not
sent to the job if the shell receives a SIGHUP. If no jobspec
is present, and neither the -a nor the -r option is supplied,
the current job is used. If no jobspec is supplied, the -a
option means to remove or mark all jobs; the -r option without
a jobspec argument restricts operation to running jobs. The
return value is 0 unless a jobspec does not specify a valid
job.
<p>
 </p>
<p> 可以看出，我们可以用如下方式来达成我们的目的。 </p>
<p> </p>
<ul style="font-size:0.76em;font-family:Simsun;background-color:#FFFFFF;">
<li style="font-family:arial, nsimsun, sans-serif;"> 用 disown -h jobspe 来使某个作业忽略 HUP 信号。
<li style="font-family:arial, nsimsun, sans-serif;"> <span style="font-size:0.76em;"&gt用 disown -ah 来使所有的作业都忽略 HUP 信号。<span style="font-size:0.76em;"&gt
<li style="font-family:arial, nsimsun, sans-serif;"> <span style="font-size:0.76em;"&gt用 disown -rh 来使正在运行的作业&t;span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background-color:#FFFFF;">忽略 HUP 信号。

<p>
 </p>
<p> </p>
<p style="font-family:arial, nsimsun, sans-serif;font-size:0.76em;background-color:#FFFFFF;>   需要注意的是，当使用过 disown 之后，会将把目标作业从作业表中移除，我们将不能再使用 jobs 来查看它，但是依然能够用 ps -ef 查找到它。 </p>
<p style="font-family:arial, nsimsun, sans-serif;font-size:0.76em;background-color:#FFFFFF;>   但是还有一个问题，这种方法的操作对象是作业，如果我们在运命令时在结尾加了"&"来使它成为一个作业并在后台运行，那就万事大吉了，我们可以通过 jobs 命令来得到所有作业的列表。但是如果并没有把当前命令作为作来运行，如何才能得到它的作业号呢？答案就是用 CTRL-z（按住 Ctrl 键的同时按住 z 键）了！ </>
<p style="font-family:arial, nsimsun, sans-serif;font-size:0.76em;background-color:#FFFFFF;>   CTRL-z 的用途就是将当前进程挂起（Suspend），然后我们就以用 jobs 命令来查询它的作业号，再用 bg jobspec 来将它入后台并继续运行。需要注意的是，如果挂起会影响当前进程的运行结果，请慎用此方法。 </p&gt

disown 示例 1（如果提交命令时已经用“&”将命令放入后台运行，则可以直接使用 disown”）

<p>
 </p>
<p>   </p>
<pre class="prettyprint">[root@pvcent107 build]# cp -r testLargeFile largeFile &
[1] 4825
[root@pvcent107 build]# jobs
[1]+ Running   cp -i -r testLargeFile largeFile &
[root@pvcent107 build]# disown -h %1
[root@pvcent107 build]# ps -ef |grep largeFile
root   4825   968 1 09:46 pts/4   00:00:00 cp -i -r testLargeFile largeFile
root   4853   968 0 09:46 pts/4   00:00:00 grep largeFile
[root@pvcent107 build]# logout
disown 示例 2（如果提交命令时未使用“&”将命令放入后台运行，可使用 CTRL-z “bg”将其放入后台，再使用“disown”）

<p>
 </p>
<p>   </p>
<pre class="prettyprint">[root@pvcent107 build]# cp -r testLargeFile largeFile2
[1]+ Stopped   cp -i -r testLargeFile largeFile2
[root@pvcent107 build]# bg %1
[1]+ cp -i -r testLargeFile largeFile2 &
[root@pvcent107 build]# jobs
[1]+ Running   cp -i -r testLargeFile largeFile2 &
[root@pvcent107 build]# disown -h %1
[root@pvcent107 build]# ps -ef |grep largeFile2
root   5790 5577 1 10:04 pts/3   00:00:00 cp -i -r testLargeFile largeFile2
root   5824 5577 0 10:05 pts/3   00:00:00 grep largeFile2
[root@pvcent107 build]#

<p>
 </p>
<p> 场景：</span&gt

  </p>
<p>   我们已经知道了如何让进程免受 HUP 号的影响，但是如果有大量这种命令需要在稳定的后台里运行，如何避免对每条命令都做这样的操作？ </p>
<p> 解决方法：</spa
>
  </p>
<p>   <span style="font-family:arial, nsimsun, sa
s-serif;line-height:normal;background-color:#FFFFFF;">此时最方便的方法就是 screen 了。单的说，screen 提供了 ANSI/VT100 的终端模拟器，使它能够在一个真实终端下运行多个全屏的伪端。screen 的参数很多，具有很强大的功能，我们在此仅介绍其常用功能以及简要分析一下为什么用 screen 能够避免 HUP 信号的影响。
  </p>
<p> <span style="font-family:arial, nsimsun, sans-serif;line-height:normal;background

color:#FFFFFF;">   使用 screen 很方便，有以几个常用选项： </p>
<p> </p>
<ul style="font-size:0.76em;font-family:Simsun;background-color:#FFFFFF;">
<li style="font-family:arial, nsimsun, sans-serif;"> 用 screen -dmS session name 来建立一个处于断开模式下的会话（并指定其会话名）。
<li style="font-family:arial, nsimsun, sans-serif;"> 用 screen -list 来列出所有会话。
<li style="font-family:arial, nsimsun, sans-serif;"> 用 screen -r sessionname 来重新连接指定会话。
<li style="font-family:arial, nsimsun, sans-serif;"> 用快捷键 CTRL-a d 来暂时断开当会话。

<p> screen 示例 </p>
<p> </p>
<pre class="prettyprint">[root@pvcent107 ~]# screen -dmS Urumchi
[root@pvcent107 ~]# screen -list
There is a screen on:
12842.Urumchi (Detached)
1 Socket in /tmp/screens/S-root.
[root@pvcent107 ~]# screen -r Urumchi
<p></p> 转自：
http://www.ibm.com/developerworks/cn/linux/l-cn-nohup/

<p></p>
<p>
 </p><p class='fn-clear'>该文章同步自 <i style='margin-right:5px;'>田尘殇 SeanSnow 自己决定一切 做自己做的人</i></p>

原文链接：Linux 技巧：让进程在后台可靠运行的几种方法