Screen 管理远程会话

作者：Snowitty
原文链接：https://ld246.com/article/1519734621833
来源网站：链滴

当你使用SSH或者telent远程登录到Linux服务器的时候，是不会经常为一些需要长时间执行的任务感头疼？比如系统备份、编译安装、大数据传输等等。通常这种情况会为这样的任务开一个远程终端窗。因为他们执行的时间太长，在此期间可不能关掉窗口或者断开连接，否则这个任务就被杀掉了，一就白忙了。

SIGHUP 信号

首先让我们来了解为什么关闭窗口或者断开链接会导致正在运行的进程终结

在Linux/Unix中有这样的几个概念：

- 进程组（process group）：一个或多个进程的集合，每一个进程组有唯一一个进程组ID，即进程长进程的ID。
- 会话期（session）：一个或多个进程组的集合，有唯一一个会话期首进程（session leader）。会期ID为首进程的ID。
- 会话期可以有一个单独的控制终端（controlling terminal）。与控制终端连接的会话期首进程叫控制进程（controlling process）。当前与终端交互的进程称为前台进程组。其余进程组称为后台进组。

根据POSIX.1(不了解的请自行Wiki，或者等我再水一篇)定义：

- 挂断信号（SIGHUP）默认的动作是终止程序。
- 当终端接口检测到网络连接断开，将挂断信号发送给控制进程（会话期首进程）。
- 如果会话期首进程终止，则该信号发送到该会话期前台进程组。
- 一个进程退出导致一个孤儿进程组中产生时，如果任意一个孤儿进程组进程处于STOP状态，发送SIHUP和SIGCONT信号到该进程组中所有进程。

因此当网络断开或终端窗口关闭后，控制进程收到SIGHUP信号退出，会导致该会话期内其他进程退。

现在让我们通过一个例子来看。打开两个SSH终端，在其中一个运行top命令

```
[root@hxkvm ~]# top
```

在另一个终端找到top的进程ID为13575其父进程为13514

```
[root@hxkvm ~]# ps -ef|grep top
root     13575 13514  0 19:17 pts/0    00:00:00 top
root     13577 13546  0 19:17 pts/2    00:00:00 grep --color top
```

使用ps-xj命令可以看到：

```
[root@hxkvm ~]# ps -xj|grep 13575
13512 13514 13514 13514 pts/0    13575 Ss       0   0:00 -bash
13514 13575 13575 13514 pts/0    13575 S+       0   0:00 top
13546 13669 13668 13546 pts/2    13668 S+       0   0:00 grep --color 13575
```

关闭第一个SSH窗口可以看到TOP也被杀掉了

```
[root@hxkvm ~]# ps -ef|grep 13575
```

```
root     13814 13546  0 19:50 pts/2    00:00:00 grep --color 13575
```

如果能做到忽略SIGHUP信号，关闭窗口断开链接就不会影响程序的运行了。nohup命令可以达到这目的，如果程序的标准输出/标准错误是终端，nohup默认将其重定向到nohup.out文件。值得注意是nohup命令只是使得程序忽略SIGHUP信号，还需要使用标记**&**把它放在后台运行。

```
nohup <``command``> [argument...] &
```

Screen

虽然nohup很容易使用，但还是比较“简陋”的，对于实行简单的操作行，对于复杂的场景就很麻烦。

我们可以使用一个更为强大的实用程序screen，流行的Linux发行版通常会自带，如果没有请自行安。

```
[root@hxkvm ~]# rpm -qa|grep screen
screen-4.1.0-0.23.20120314git3c2946.el7_2.x86_64
```

简单来说，Screen是一个可以在多个进程之间多路复用一个物理终端的窗口管理器。Screen中有会的概念，用户可以在一个screen会话中创建多个screen窗口，在每一个screen窗口中就像操作一个真的telnet/SSH连接窗口那样。在screen中创建一个新的窗口常用方式：

一、直接在命令行输入screen

Screen将创建一个执行shell的全屏窗口。你可以执行任意shell程序，就像在ssh窗口中那样。在该窗中键入exit退出该窗口，如果这是该screen会话的唯一窗口，该screen会话退出，否则screen自动切到前一个窗口。

二、Screen命令后跟你要执行的程序

Screen创建一个执行该程序的单窗口会话，退出该程序将退出该窗口/会话

我更加习惯Screen目录以现在blog的PIPE程序为例，

```
screen -S pipe
./pipe
```

这样就可以保证pipe的运行不会因为断开putty而中断运行。

想要重新切换回运行的窗口直接：

```
screen -R pipe
```

当然screen也是支持快捷键操作的，这种命令形式在screen中叫做键绑定（key binding），C-a叫命令字符（command character）

可以通过C-a ?来查看所有的键绑定（C即Ctrl+），常用的键绑定有：

C-a ?	**显示所有键绑定信息**
C-a w	显示所有窗口列表
C-a C-a	切换到之前显示的窗口
C-a c	创建一个新的运行shell的窗口并切换到该

口

C-a n	切换到下一个窗口
C-a p	切换到前一个窗口(与C-a n相对)
C-a 0..9	切换到窗口0..9
C-a a	发送 C-a到当前窗口
C-a d	暂时断开screen会话
C-a k	杀掉当前窗口
C-a [	进入拷贝/回滚模式

Screen提供了丰富强大的定制功能。你可以在Screen的默认两级配置文件/etc/screenrc和$HOME/.screenrc中指定更多，例如设定screen选项，定制绑定键，设定screen会话自启动窗口，启用多用户模，定制用户访问权限控制等等。如果你愿意的话，也可以自己指定screen配置文件。

以多用户功能为例，screen默认是以单用户模式运行的，你需要在配置文件中指定multiuser on 来开多用户模式，通过acl*（acladd,acldel,aclchg...）命令，你可以灵活配置其他用户访问你的screen话。更多配置文件内容请参考screen的主页。

参考引用

- GNU Screen的官方网站： http://www.gnu.org/software/screen/
- Screen的man page： http://www.slac.stanford.edu/comp/unix/package/epics/extensions/icConsole/screen.1.html
- IBM developerworks 《使用 screen 管理你的远程会话》