

# 从Socket error丢失网络连接的Linux SSH恢复pts会话

Published on Feb 9, 2022 UTC by Wang Ye <<https://wangye.org>>

日常运维中经常会接触到SSH (Secure Shell, 安全外壳协议), 通过SSH登录到要管理的服务器, 通过各种命令完成整个服务器的维护, 这个过程在内网中或者网络质量较高的时候总是很顺畅, 但是当网络质量比较差的时候就会出现意外断开连接的情况了, 比如比较常见的报错Socket error Event: 32 Error: 10053, 这时候往往需要我们重新发起SSH连接, 问题是新发起的连接无法继续上一个连接执行的命令, 比如在断开连接前我们在虚拟终端执行了一个需要长时间等待的Shell命令, 但是命令没执行完网络中断了, 实际上此时的命令仍然在执行, 如果遇到需要交互的部分还会停下来, 只是我们新的连接将看不到命令的输出, 可以通过`ps aux | grep` 进程名定位到进程。本文将介绍对于此类问题的应对方法供参考

## 1 终端的概念

### 1.1 什么是TTY

TTY 是 Teletype 或 Teletypewriter 的缩写, 原指电传打字机, 后来这种设备逐渐键盘和显示器取代。无论是电传打字机还是键盘显示器, 都是作为计算机的终端设备存在的, 所以 TTY 也泛指计算机的终端 (Terminal) 设备。为了支持终端设备, Linux实现了一个虚拟的设备叫做TTY, 正常通过命令界面登录就会看到类似于下面的界面:

```
Ubuntu 18.04.1 LTS wangye tty1
wangye login:
```

### 1.2 TTY 终端的切换

注意这里终端提示第一行最后的tty1, 这表示你登录的是tty1的虚拟终端, 当然这样的虚拟终端Linux通常会支持多个, 以ttyX表示, 其中X表示数字, 通常情况下tty1~tty6是命令行界面, tty7是图形界面 (GUI, 一般是X桌面会话), 我们可

---

可以通过CTRL+ALT+F1~CTRL+ALT+F6切换命令行终端，CTRL+ALT+F7切换到图形界面。这个小技巧可以让我们快速从假死的图形界面中切换到控制台，然后使用控制台杀死问题进程以恢复系统的运行。

需要知道你当前所登录的终端，只需要键入`tty`命令即可。

### 1.3 什么是PTY

PTY是伪终端 ( Pseudo Terminal ) 的英文缩写，这个概念的提出是为了能够将终端仿真移入用户空间，并完整的兼容TTY 子系统，终端仿真最常见的形式就是在图形界面下启动终端仿真程序 ( 俗称启动命令行界面 )。这时通过`tty`命令就会发现伪终端会在`/dev/pts`路径下创建一个带数字编号的设备文件，同样的我们的SSH连接到Linux系统启动也是伪终端。

要查看当前有多少个活动的伪终端会话 ( Session )，可以直接键入`w`命令，该命令将列出当前所有的伪终端会话，也会显示该会话最后执行的命令行等信息。

### 1.4 PTY伪终端的切换

根据我查阅的资料，没有切换伪终端会话的方法，但也可以变通，具体后面会介绍。

关于终端的概念，这篇文章[《Linux 终端\(TTY\)》](#)已经介绍的很详细，可以参考。

## 2 清理丢失连接的PTY会话

这里以远程SSH会话意外中断为例，一般情况下如果你没有执行任何命令或者程序，又或者命令或者程序已经执行完毕，该会话下无新进程，那么一段时间后Linux系统会自动清理掉这个中断的会话。

但是如果该会话中断前执行了一个需要很长时间才能结束的命令以及执行的命令正在等待交互，那么该会话将会一直存在于系统之中，这个时候通过`w`命令你就可以发现多个会话，其中有当前登录的SSH会话，和之前登录而中断的会话，分别用不同的`pts/X`表示，其中`X`代表数字。

```
root@wangye.org [~]# w
 01:05:41 up 89 days, 21:46,  3 users,  load average: 0.17, 0.09, 0.08
USER      TTY      FROM          LOGIN@      IDLE        JCPU       PCPU WHAT
root      pts/0    10.10.2.11    23:51      53:49      0.04s     0.04s  -
bash
root      pts/2    10.10.2. 11   01:11      0.00s     0.01s     0.01s  w
```

## 2.1 结束终端会话所运行的进程

可能读者会想到我们可以直接`kill`掉中断会话启动的所有进程，然后让系统清理掉这个会话，确实，这不失为一个好的办法，如果所启动的进程无关紧要，那么可以一试。

通过`ps -ft pts/0`命令获取依赖于`pts/0`会话下的进程，这里的`pts/0`是你通过`w`命令获取的，根据实际情况修改。

```
root@wangye.org [~]# ps -ft pts/0
UID          PID     PPID  C  STIME TTY          TIME CMD
root         221857 221761  0  Oct09 pts/0        00:00:00 -bash
```

找到PID所对应的进程号，执行`kill 进程号`的命令，优雅请进程退出，当然对于顽固的进程，在确保安全的前提下使用`kill -9 进程号`强制杀死进程。

## 2.2 直接结束终端会话

如果依次执行结束进程的命令较为麻烦，还有一种简便的方式，那就是使用`pkill`命令结束终端会话，还是先通过`w`命令查到不需要的终端会话号，假设我们这里是`pts/0`，那么可以这么执行`pkill`完成清理，`pkill -9 -t pts/0`。

---

更多细节这里可以参考这篇文章 [《How to kill or terminate unwanted tty/pts sessions in Linux?》](#) 的介绍。

### 3 从丢失连接的PTY会话恢复

如果在丢失会话中操作的进程很重要呢，我们能否恢复到丢失的会话继续先前的操作？非常遗憾的是无法切换到先前的会话，但是我们有个变通的办法——那就是接管进程。

#### 3.1 使用reptyr命令

使用reptyr命令，这个命令需要安装，在Debian或者Ubuntu下可以直接运行 `sudo apt-get install reptyr` 安装，安装完成后执行 `reptyr -h` 获得帮助如下：

```
root@wangye.org [~]# reptyr -h
Usage: reptyr [-s] PID
       reptyr -l|-L [COMMAND [ARGS]]
-l     Create a new pty pair and print the name of the slave.
       if there are command-line arguments after -l
       they are executed with REPTYR_PTY set to path of pty.
-L     Like '-l', but also redirect the child's stdio to the slave.
-s     Attach fds 0-2 on the target, even if it is not attached to a
tty.
-T     Steal the entire terminal session of the target.
       [experimental] May be more reliable, and will attach all
       processes running on the terminal.
-h     Print this help message and exit.
-v     Print the version number and exit.
-V     Print verbose debug output.
```

我们通过2.1节介绍的 `ps -ft` 命令得到会话的进程ID号，然后在root身份下使用 `reptyr -T 进程ID号`，不出意外这个进程会被转移到当前的会话下，原来丢失连接的会话如果没有执行的进程将会被系统清理（或者也可以安全的杀死该会话）。

---

关于此命令的一些讨论可以参考[《Moving an already-running process to Screen》](#)。

## 4 总结

对于一次丢失连接的SSH会话，按照下面的步骤操作：

- ①再次连接SSH，使用`w`命令检查会话是否存在；
- ②通过`w`或者`ps -ft` 会话号检查丢失连接前是否执行了长时间或者需要交互的命令；
- ③判断命令是否能够自行结束，如果不能，那么该命令是否重要，不重要的命令直接杀死命令进程或者杀死所依赖的会话；
- ④重要的命令进程使用`reptyr -T` 进程ID号在当前会话下获得进程并安全的杀死旧的会话。

## External References (Links)

《Linux 终端(TTY)》

<https://www.cnblogs.com/sparkdev/p/11460821.html>

《How to kill or terminate unwanted tty/pts sessions in Linux?》

<https://www.crybit.com/kill-unwanted-tty-unix/>

《Moving an already-running process to Screen》

<https://serverfault.com/questions/55880/moving-an-already-running-process-to-screen/284795#284795>