

使用screen管理你的远程会话Linux认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022__E4_BD_BF_E7_94_A8scre_c103_645113.htm 最近遇到登录经常断的问题，所以通过一台跳转机，登录到开发环境，使用screen，真是方便啊。可以保留工作场景，断了的话，下次再通过screen进取也没有问题，原来的工作场景还保留着。

----- 你是不是经常需要远程登录到Linux服务器？你是不是经常为一些长时间运行的任务头疼？还在用 nohup 吗？那么来看看 screen 吧，它会给你一个惊喜！你是不是经常需要 SSH 或者 telnet 远程登录到 Linux 服务器？你是不是经常为一些长时间运行的任务而头疼，比如系统备份、ftp 传输等等。通常情况下我们都是为每一个这样的任务开一个远程终端窗口，因为他们执行的时间太长了。必须等待它执行完毕，在此期间可不能关掉窗口或者断开连接，否则这个任务就会被杀掉，一切半途而废了。元凶：SIGHUP 信号 让我们来看看为什么关掉窗口/断开连接会使得正在运行的程序死掉。

在Linux/Unix中，有这样几个概念：进程组（process group）：一个或多个进程的集合，每一个进程组有唯一的一个进程组ID，即进程组长进程的ID。会话期（session）：一个或多个进程组的集合，有唯一的一个会话期首进程（session leader）。会话期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做控制进程（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程组。根

据POSIX.1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SIGHUP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出。我们来看一个例子。打开两个SSH终端窗口，在其中一个运行top命令。

```
[root@tivf09 root]# top
```

在另一个终端窗口，找到top的进程ID为5180，其父进程ID为5128，即登录shell。

```
[root@tivf09 root]# ps -ef|grep top
```

```
root 5180 5128 0 01:03 pts/0 00:00:02 top
```

```
root 5857 3672 0 01:12 pts/2 00:00:00 grep top
```

使用pstree命令可以更清楚地看到这个关系：

```
[root@tivf09 root]# pstree -H 5180|grep top
```

```
top|sshd- -sshd---bash---top
```

使用ps-xj命令可以看到，登录shell（PID 5128）和top在同一个会话期，shell为会话期首进程，所在进程组PGID为5128，top所在进程组PGID为5180，为前台进程组。

```
[root@tivf09 root]# ps -xj|grep 5128
```

```
5128 pts/0 5180 S 0 0:00 -bash
```

```
5128 5180 5180 5128 pts/0 5180 S 0 0:50 top
```

```
3672 18095 18094 3672 pts/2 18094 S 0 0:00 grep 5128
```

关闭第一个SSH窗口，在另一个窗口中可以看到top也被杀掉了。

```
[root@tivf09 root]# ps -ef|grep 5128
```

```
root 18699 3672 0 04:35 pts/2 00:00:00 grep 5128
```

如果我们可以忽略SIGHUP信号，关掉窗口应该就不会影响程序的运行了。nohup命令可以达到这个目的，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。值得注意的是nohup命令只是使得程

序忽略SIGHUP信号，还需要使用标记 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com