

(77条消息) 关闭linux终端还让程序继续执行的方法_昙花逐月的博客-CSDN博客_关闭终端后程序继续运行



昙花逐月 于 2022-05-27 12:25:05 发布 1374 已收藏 7

版权声明：本文为博主原创文章，遵循CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

一、前言

我们在使用linux时，会经常要把一个程序放到后台运行，这时候就需要在程序的最后使用【&】符号，但是当直接关闭这个终端时，程序就会退出，下面我介绍5种方法，在关闭终端后程序还继续运行。

二、思考

首先要考虑一个问题，直接关闭终端为什么后台程序就退出了？那是因为直接关闭，终端会向当前环境启动的后台任务发送SIGHUP信号，让运行的程序挂起，那么忽略SIGHUP信号就行啦。

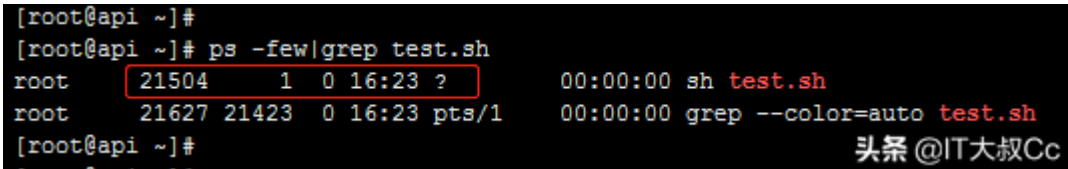
三、nohup

最常用的命令就是【nohup】，从名字就看出是否挂起运行命令，关闭终端或者退出某个账号，进程也继续保持运行状态，同样要配合【&】符号一起使用。

用法：nohup Command [Arg ...]&

```
$ nohup sh test.sh &
# 直接关闭当前终端，再打开一个查看
$ ps -few|grep test.sh
123
```

- 1
- 2
- 3
- 4



从上面的输出可以看出，我们运行的test.sh的父进程变成了【PID 1】，由linux主进程接管。

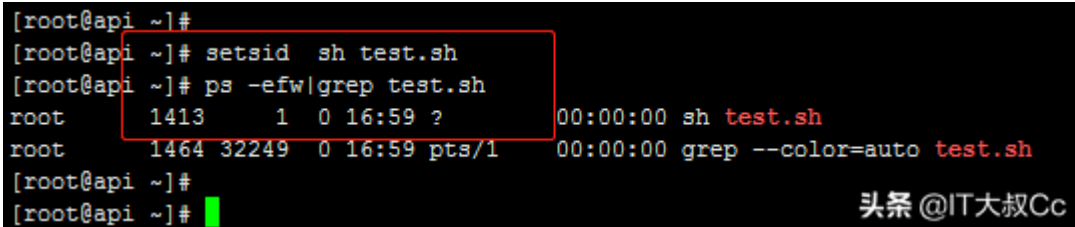
四、setsid

nohup是忽略SIGHUP信号，那有没有其他的思路？

使用setsid命令可以使程序打开新的进程，该进程继承了父进程组的ID。从下图可以看到，test.sh脚本的父进程已经是init 1，此时不管关闭还是退出终端，已经和后台的程序没有关系了。

```
$ setsid sh test.sh
1
```

- 1
- 2



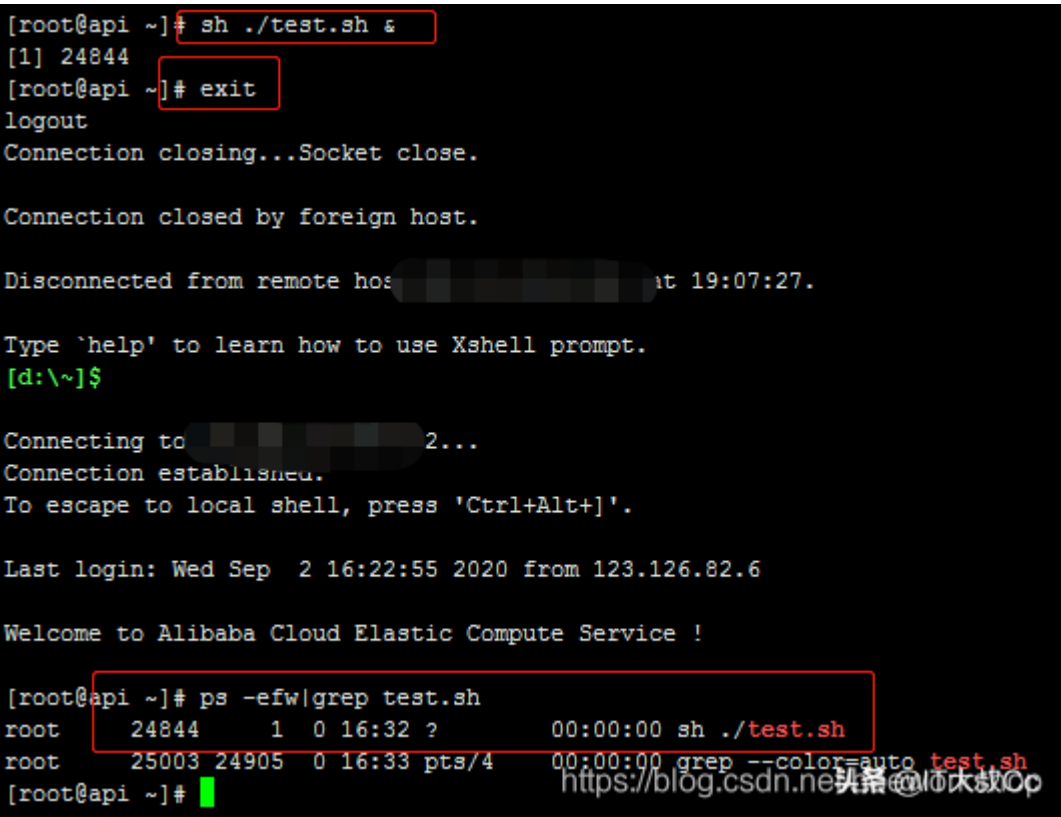
setsid与nohup的一个重要区别是：不需要使用&压入后台。此法使运行的进程以root身份运行，多少有些安全隐患。

五、exit

使用exit命令退出终端，我们的程序还是会继续运行，这是为什么呢？这是因为使用exit命令退出终端时不会向终端所属任务发SIGHUP信号，这是huponexit配置项控制的，默认是off，可以使用shopt命令查看。

```
[root@api ~]# shopt |grep huponexit
huponexit off
12
```

- 1
- 2
- 3



将huponexit 配置成on，再次使用exit命令退出，所属的任务就会跟随退出。

```
[root@api ~]# shopt -s huponexit
[root@api ~]# shopt |grep huponexit
huponexit on
123
```

- 1
- 2
- 3
- 4

六、 trap

trap命令是linux内置的命令

```
trap [接收到指定信号时将要采取的行动] signal_spec
1
```

- 1
- 2

常见的信号如下：

信号	值	描述
1	SIGHUP	挂起进程
2	SIGINT	终止进程
3	SIGQUIT	停止进程
9	SIGKILL	无条件终止进程
15	SIGTERM	尽可能终止进程
17	SIGSTOP	无条件停止进程，但不是终止进程
18	SIGTSTP	停止或暂停进程，但不终止进程
19	SIGCONT	继续运行终止的进程

当一个脚本要在后台运行，不希望终端退出后关闭脚本程序，我们可以在的脚本中增加trap命令来忽略SIGHUP信号(挂起进程)。

```
#!/bin/bash
trap "" HUP
while true;do
    date >> /root/test.txt
    sleep 1
done
123456
```

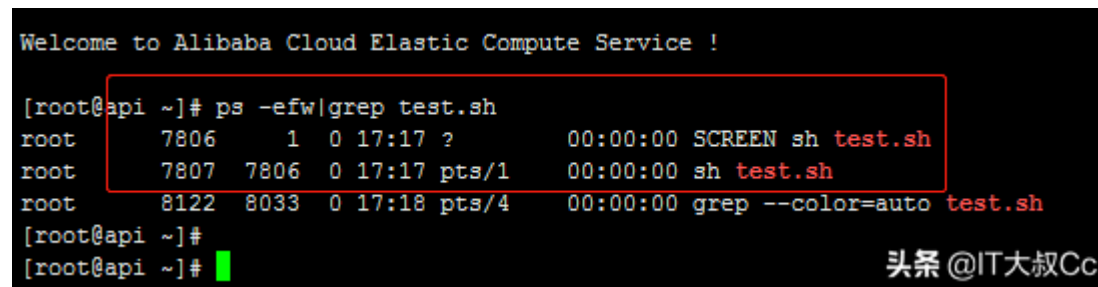
- 1
- 2
- 3
- 4
- 5
- 6
- 7

七、screen

screen命令用于管理多个终端，它可以创建终端，让程序在里面运行。

```
screen sh test.sh &
1
```

- 1
- 2



从上图可以看出，test.sh脚本的父进程是screen终端进程【PID 7806】，而screen终端的父进程【PID 1】，同样可以实现后台的运行。