## [转载]linux查看和关闭后台执行程序

(2014-04-13 09:47:42)

**原文地址：**[linux查看和关闭后台执行程序](http://blog.sina.com.cn/s/blog_90546d6f0101en9y.html" \o "linux查看和关闭后台执行程序" \t "/home/ye/Documents\\x/_blank)**作者：**[小河](http://blog.sina.com.cn/u/2421452143" \o "小河" \t "/home/ye/Documents\\x/_blank)

后台执行的程序会有一个pid,

查看后台程序主要用到jobs,ps

关闭后台程序用到kill

详细说明转自<http://blog.csdn.net/tianlesoftware/article/details/6165753>  
一. 有关进程的几种常用方法

1.1  & 符号

在命令后面加上一个 & 符号，表示该命令放在后台执行，如：

[oracle@singledb ~]$ crontab -l

20 17 \* \* \* /u02/rman\_incremental\_backup.sh &

1.2  Jobs 命令

       该命令可以查看当前有多少在后台运行的命令，jobs 列出当前shell环境中已启动的任务状态，若未指定jobsid，则显示所有活动的任务状态信息。

　　jobs命令执行的结果，＋表示是一个当前的作业，减号表是是一个当前作业之后的一个作业，jobs -l选项可显示所有任务的PID,jobs的状态可以是running, stopped, Terminated,但是如果任务被终止了（kill），shell 从当前的shell环境已知的列表中删除任务的进程标识；也就是说，jobs命令显示的是当前shell环境中所起的后台正在运行或者被挂起的任务信息。如：

[oracle@singledb ~]$ jobs

[1]+  Running   /u02/rman\_incremental\_backup.sh  &

在这里讲一个概念：当前任务

　　如果后台的任务号有2个，[1],[2]；如果当第一个后台任务顺利执行完毕，第二个后台任务还在执行中时，当前任务便会自动变成后台任务号码 “[2]”的后台任务。所以可以得出一点，即当前任务是会变动的。当用户输入“fg”、“bg”和“stop”等命令时，如果不加任何引号，则所变动的均 是当前任务。

1.3 fg 命令（foreground）

       该命令将后台运行的进程调到前台来运行。 具体用法： fg %n。

       这里的N 是jobs 看到的jobnumber。 如：

[oracle@singledb ~]$ fg %1

-bash: fg: job has terminated

[1]+  Exit 1  /u02/rman\_incremental\_backup.sh > /u02/rman.log 2>&1

1.4  ctrl + z 组合键

       该命令将一个正在前台执行的命令放到后台，并且暂停。

1.5  bg命令（background）

       该命令将一个在后台暂停运行的命令，变成继续在后台执行的命令。

       用法：bg %n。 N 是jobs命令查看到的jobnumber。 如：

[oracle@singledb ~]$ /u02/rman\_incremental\_backup.sh

[1]+  Stopped                 /u02/rman\_incremental\_backup.sh

-- 我们运行备份命令，然后按下CTRL+Z

[oracle@singledb ~]$ jobs

[1]+  Stopped                 /u02/rman\_incremental\_backup.sh

--用jobs查看进程状态

[oracle@singledb ~]$ fg %1

/u02/rman\_incremental\_backup.sh

--用fg将进程调到前台运行

[1]+  Stopped                 /u02/rman\_incremental\_backup.sh

--按下ctrl+z

[oracle@singledb ~]$ bg %1

[1]+ /u02/rman\_incremental\_backup.sh &

--用bg将后台暂停的进程继续执行，这里自动加了一个& 符号

[oracle@singledb ~]$ jobs

[1]+  Running                 /u02/rman\_incremental\_backup.sh &

--查看job 状态

1.6  结束正在运行的进程

1.6.1 结束前台进程

       这种只需要按下组合键： ctrl + c 即可。

1.6.2 结束后台运行的进程

       这种情况下也有两种方法：

（1）用jobs查看进程的jobnumber，然后用命令：kill %n 来结束。

（2）用ps 查看进程的pid，然后用命令：kill pid 来结束。

       这里对kill 做点说明，kill除了可以终止进程，还能给进程发送其它信号，使用kill -l 可以察看kill支持的信号。

[oracle@singledb ~]$ kill -l

 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL

 5) SIGTRAP      6) SIGABRT      7) SIGBUS       8) SIGFPE

 9) SIGKILL     10) SIGUSR1     11) SIGSEGV     12) SIGUSR2

13) SIGPIPE     14) SIGALRM     15) SIGTERM     16) SIGSTKFLT

17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP

21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU

25) SIGXFSZ     26) SIGVTALRM   27) SIGPROF     28) SIGWINCH

29) SIGIO       30) SIGPWR      31) SIGSYS      34) SIGRTMIN

35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3  38) SIGRTMIN+4

39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8

43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12

47) SIGRTMIN+13 48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14

51) SIGRTMAX-13 52) SIGRTMAX-12 53) SIGRTMAX-11 54) SIGRTMAX-10

55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7  58) SIGRTMAX-6

59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2

63) SIGRTMAX-1  64) SIGRTMAX

　　SIGTERM是不带参数时kill发送的信号，意思是要进程终止运行，但执行与否还得看进程是否支持。如果进程还没有终止，可以使用kill -SIGKILL pid，这是由内核来终止进程，进程不能监听这个信号。

1.7  ps 命令

       该命令用户列出系统中正在运行的进程。 常用的参数：ps -ef

关于这2个参数含义，参考ps 命令的帮助：

[oracle@singledb ~]$ ps --help

\*\*\*\*\*\*\*\*\* simple selection \*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\* selection by list \*\*\*\*\*\*\*\*\*

-A all processes                      -C by command name

-N negate selection                   -G by real group ID (supports names)

-a all w/ tty except session leaders  -U by real user ID (supports names)

-d all except session leaders         -g by session OR by effective group name

-e all processes                      -p by process ID

T  all processes on this terminal     -s processes in the sessions given

a  all w/ tty, including other users  -t by tty

g  OBSOLETE -- DO NOT USE             -u by effective user ID (supports names)

r  only running processes             U  processes for specified users

x  processes w/o controlling ttys     t  by tty

\*\*\*\*\*\*\*\*\*\*\* output format \*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\* long options \*\*\*\*\*\*\*\*\*\*\*

-o,o user-defined  -f full            --Group --User --pid --cols --ppid

-j,j job control   s  signal          --group --user --sid --rows --info

-O,O preloaded -o  v  virtual memory  --cumulative --format --deselect

-l,l long          u  user-oriented   --sort --tty --forest --version

-F   extra full    X  registers       --heading --no-heading --context

                    \*\*\*\*\*\*\*\*\* misc options \*\*\*\*\*\*\*\*\*

-V,V  show version      L  list format codes  f  ASCII art forest

-m,m,-L,-T,H  threads   S  children in sum    -y change -l format

-M,Z  security data     c  true command name  -c scheduling class

-w,w  wide output       n  numeric WCHAN,UID  -H process hierarchy

二. 前台进程和后台进程的一点讨论

       整理这篇文章的目的就是弄清楚在实际使用中这两种进程的区别。 一般情况下，我们都是通过shell 连接到linux 服务器上，那么，假如我们在这个shell 窗口执行一个备份或者其他命令，即前台命令。 在这个时候，前台命令没有执行完，我们把这个终端关闭了，那么这个命令也就会中断。 即使我们在执行命令时加了& 符号，让进程在后台执行，终端的关闭也会造成命令的结束。

解决这个问题有两种方法：

（1）将命令添加到crontab里，让系统自己调用。

              这种方法将使命令在服务器端执行，这样即使终端关闭，也不影响命令的执行。

（2）使用nohup命令（no hang up）。

       该命令用于保证不挂断地运行命令。如：

[oracle@singledb ~]$ nohup /u02/rman\_incremental\_backup.sh &

[1] 4409

[oracle@singledb ~]$ jobs

[1]+  Running                 nohup /u02/rman\_incremental\_backup.sh &

这里要注意：

       运行nohup命令后，按任意键返回shell 窗口，然后输入exit退出终端，不能直接关闭窗口，直接关闭窗口的话，还是会造成命令进程和终端一起结束。

       在Unix中永久的后台进程称为守护进程（daemon）。守护进程通常从系统启动时自动开始执行，系统关闭时才停止。而nohup 启动的进程虽然也是不中断的执行，但是当进程执行完毕后就会退出。