Linux - 进程管理（二）

作者：douniwan
原文链接：https://ld246.com/article/1573891709947
来源网站：链滴

上一篇：Linux-进程管理（一）

学习了如何在Linux中查看已经运行的进程的一些信息，这一篇要探索的是如何控制进程及其作业，解进程之间的通信。

<!--more-->

调整优先级

前面从top命令的输出结果中的**NI**值中可以知道进程的优先级，那么如何去设置进程的优先级呢？这有两个命令nice和renice分别用于新创建进程时候给进程设置优先级和修改运行中的进程的优先级。

nice

nice命令以更改过的优先序来执行程序，如果未指定程序，则会印出目前的排程优先序，内定的 adjustment 为 10，范围为 -20（最高优先序）到 19（最低优先序）。

语法：

nice [-n adjustment] [-adjustment] [--adjustment=adjustment] [--help] [--version] [command arg...]]

参数说明

- -n adjustment, -adjustment, --adjustment=adjustment 皆为将该原有优先序的增加 adjustment
- --help 显示求助讯息
- --version 显示版本资讯

renice

如果想要改变正在运行中的进程的优先级，那么就可以用此命该来实现。

语法：

renice priority [[-p] pid ...] [[-g] pgrp ...] [[-u] user ...]

参数说明：

- -p pid 重新指定行程的 id 为 pid 的行程的优先序
- -g pgrp 重新指定行程群组(process group)的 id 为 pgrp 的行程 (一个或多个) 的优先序
- -u user 重新指定行程拥有者为 user 的行程的优先序

详情请看：Linux renice命令

进程的作业控制

作业控制，指的是当前正在运行的进程的行为，也称为进程控制。是shell的一个特性，是用户能够在个独立进程间相互切换。

常用的命名及功能键：

命令或快捷键	功能说明
& 命令	该命令在后台运行
Ctrl + d 快捷键	终止一个正在前运行的进程(含有正常含义)
Ctrl + c 快捷键	终止一个正在前运行的进程(含有强行含义)
Ctrl + z 快捷键	挂起一个正在前运行的进程
jobs 命令	显示后台作业和被挂的进程
bg 命令	重新启动一个挂起的作，并在后台运行
fg 命令	把一个在后台运行的作放到前台运行

关于前后与后台是什么，在上一篇Linux-进程管理（一）的开篇有提过,可以去回顾一下。

&

在命令后加此命令就可以将进程放在后台执行。例如执行一个jar包

```
java -jar test.jar #直接在前台运行，可以被Ctrl + c杀死
java -jar test.jar & #在后台运行，不可以被Ctrl + c杀死
```

除此此外，还有一些进程执行会输出很多到终端，如果我们不想让他输出到终端，出了使用输出重定向方法，牙可以使用&让它后台运行。

Ctrl + z

该命令可以挂起一个正在前台运行的进程。那么这有什么应用场景吗？答案是肯定的，在前面Linux - vm的常用命令有说过在vim中如果不想退出当前界面，又想输入一条命令来查看一些信息时可以使用！来执行一条临时命令。那么如果我想进行其他操作，又不想结束当前的vim程序该怎么做呢？这时实我们就可以使用Ctrl + z来挂起vim，回到终端，从而去做其他的事情。

fg

当我们完成其他的事情后，需要接着做vim中没有完成的事情的时候，就可以使用fg命令调出挂起的程，从而在前台操作继续操作。

bg

同fg一样，不过是到后台运行。

jobs

如果需要查看进程的状态的时候就可以使用jobs。

参数说明

- -l：除了列出作业号之外，同时列出PID
- -r：仅列出正在后台运行的作业
- -s：仅列出正在后台暂停的作业

进程之间的通信

进程之间的通信方式有很多，比如管道，信号，消息队列，共享内存，socket，信号量。

这里我们要说的是信号。

在Linux中我们可以通过不同的信号让进程执行不同的操作。我们可以使用kill命令来将指定的信息发至进程。

kill

```
huny@huny-PC:~$ kill -l  #参看可用的信号参数
 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) SIGRTM
N+13
48) SIGRTMIN+14    49) SIGRTMIN+15    50) SIGRTMAX-14    51) SIGRTMAX-13    52) SIGRT
AX-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
huny@huny-PC:~$
```

在上面的输出中

HUP 1 终端断线

INT 2 中断（同 Ctrl + C）

QUIT 3 退出（同 Ctrl + \）

TERM 15 终止

KILL 9 强制终止

CONT 18 继续（与STOP相反， fg/bg命令）

STOP 19 暂停（同 Ctrl + Z）

示例：

首先打开一个进程

```
huny@huny-PC:~/Desktop$ tail -f test.txt # 打开一个tail进程
iiiiiiii
iiiiiiiii
iiiiiiiiii
iiiiiiiiiii
iiiiiiiiiiii
iiiiiiiiiiiii
iiiiiiiiiiiii
iiiiiiiiiiiiii
iiiiiiiiii
```

另外打开一个终端，查找tail进程的PID，然后杀掉这个进程

```
huny@huny-PC:~/Desktop$ ps -ef | grep tail
huny     15897 15791  0 15:49 pts/3    00:00:00 grep tail
huny@huny-PC:~/Desktop$ ps -ef | grep tail
huny     15922 15758  0 15:50 pts/2    00:00:00 tail -f test.txt
huny     15987 15791  0 15:51 pts/3    00:00:00 grep tail
huny@huny-PC:~/Desktop$ kill -9 15922 #根据查找的结果，杀掉tail进程
huny@huny-PC:~/Desktop$ ps -ef | grep tail #再次查找tail进程
huny     16007 15791  0 15:52 pts/3    00:00:00 grep tail
huny@huny-PC:~/Desktop$
```

从上面的例子中，我们了解到只要我们知道进程的PID，只要我们有相应的权限我们就可以跟那个进进行相应的通信。

守护进程

什么是守护进程?

- 守护进程：在后台运行，且不受终端控制的一种进程。
- 通过ps axj | more指令可以查看到Linux下的守护进程

创建守护进程的流程

1.创建子进程，终止父进程

因为守护进程脱离终端的控制，一方面因为让父进程退出，就会使得当前终端可以执行其它的指令，之后其它的操作都让子进程去执行，让子进程已一种僵尸状态来执行，就可以达到脱离终端的目的。一方面，需要创建新的会话，而创建会话的进程不能是组长进程，让父进程创建一个子进程，那么子程一定不是组长进程。

2.子进程创建会话

因为子进程继承了父进程的一些信息，包括会话相关的消息，子进程必须要脱离父进程所在的会话，须创建新的会话，让子进程变为新会话的话首进程

3.屏蔽SIGCHLD信号

为了防止僵尸进程的出现

4.改变当前工作目录

因为子进程以父进程为模板，那么子进程就会和父进程处在同一个工作目录下，所以需要改变子进程工作目录

5.重定向文件描述符

因为子进程是以父进程为模板，那么子进程就有和父进程一样的文件描述符表，父进程打开的文件，进程也能操作，所以需要将文件描述符重定向，重定向至”/dev/null/”，这个目录相当于一个垃圾桶

6.重建文件创建掩码umask

因为创建文件时，文件创建掩码会屏蔽掉一些权限对应的位，所以需要重建文件创建掩码

创建守护进程的方法

nohup 后面加上进程名 再加上&

nobup命令使进程忽略hangup信号，&使进程在后台运行。

proc

/proc：这个目录本身是一个虚拟文件系统。他放置的数据都是在内存当中，例如系统内核、进程、部设备的状态及网络状态等。因为这个目录下的数据都是在内存当中，所以本身不占任何硬盘空间，较重要的文件有：/proc/cpuinfo /proc/dma,/proc/interrupts,/proc/oports,/proc/net，等

详细信息请转至linux系统/proc目录内容简介

参考文章：

- 作业控制与进程管理
- Linux--守护进程
- linux系统/proc目录内容简介
- Linux 命令大全