

# Linux进程管理

*Linux学习教程*

Linux是一个多用户多工的操作系统。多用户是指多个用户可以在同一时间使用电脑系统；多工是指Linux可以同时执行多个任务，它可以在还未执行完一个任务时又执行另一项任务。

内容基础，语言简短简洁

红联Linux论坛是致力于Linux技术讨论的站点，目前网站收录的文章及教程基本能满足不同水平的朋友学习。

红联Linux门户：[www.linux110.com](http://www.linux110.com)

红联Linux论坛：[www.linuxdiyf.com/bbs](http://www.linuxdiyf.com/bbs)

红联Linux 论坛大全，所有致力点都体现在这

<http://www.linuxdiyf.com/bbs/rf/linux/07.htm>

## 目录

[Linux进程管理](#)

[进程及作业](#)

[创建进程](#)

[手工启动](#)

[调度启动](#)

[调度启动1 . at命令](#)

[2 . cron命令](#)

[3 . crontab命令](#)

[进程的挂起及恢复](#)

[进程管理](#)

[1 . who命令](#)

[2 . w命令](#)

[3 . ps命令](#)

[4 . top命令](#)

[5 . Kill命令](#)

[6 . nohup命令](#)

[基于Webmin的进程管理](#)

[海量Linux技术文章](#)

# Linux 进程管理

发布时间 :2006-11-21 21:45:42

Linux是一个多用户多工的操作系统。多用户是指多个用户可以在同一时间使用电脑系统；多工是指Linux可以同时执行多个任务，它可以在还未执行完一个任务时又执行另一项任务。

操作系统管理多个用户的请求和多个任务。大多数系统都只有一个CPU和一个主存，但一个系统可能有多个二级存储磁盘和多个输入/输出设备。操作系统管理这些资源并在多个用户间共用资源，当你提出一个请求时，给你造成一种假象，好像系统只被你独自占用。而实际上操作系统监控着一个等待执行的任务队列，这些任务包括用户作业、操作系统任务、邮件和显示作业等。操作系统根据每个任务的优先顺序为每个任务分配合适的时间片段，每个时间片段大约都有零点几秒，虽然看起来很短，但实际上已经足够电脑完成成千上万的命令。每个任务都会被系统运行一段时间，然后挂起，系统转而处理其他任务；过一段时间以后再回来处理这个任务，直到某个任务完成，从任务队列中去除。



## 进程及作业

发布时间 :2006-11-21 21:46:24

Linux系统上所有运行的任务都可以称之为一个进程，每个用户任务、每个系统管理守护进程，也都可以称之为进程。Linux用分时管理方法使所有的任务共同分享系统资源。我们所关心的是如何去控制这些进程，让它们能够很好地为用户服务。

进程的一个比较正式的定义是：在自身的虚拟地址空间运行的一个单独的程序。进程与程序是有区别的，进程不是程序，虽然它由程序产生。程序只是一个静态的命令集合，不占系统的运行资源；而进程是一个随时都可能发生变化的、动态的、使用系统运行资源的程序。而且一个程序可以启动多个进程。

Linux系统中所有进程都是相互联系的。除了初始化进程外，所有进程都有一个父进程。新进程不是被创建，而是被复制，或者从以前的进程复制而来。Linux系统中所有的进程都是由一个进程号为1的init进程衍生而来的。而我们在Shell下执行程序启动的进程则是Shell进程的子进程，当然我们启动的进程可以再启动自己的子进程。这样形成了一棵进程树，每个进程都是树中的一个节点，其中树的根是init。

进程和作业的概念也有区别。一个正在执行的进程称为一个作业，而作业可以包含一个或多个进程，尤其是当使用了管道和重定向命令的时候。

作业控制指的是控制正在运行的进程的行为。比如，用户可以挂起一个进程，等一会儿再继续执行该进程。Shell将记录所有启动的进程情况，在每个进程过程中，用户可以任意地挂起进程或重新启动进程。作业控制是许多Shell（包括bash和tcsh）的一个特性，使用户能在多个独立作业间进行切换。

例如，当用户编辑一个文字文件，并需要中止做其他事情时，利用作业控制，用户可以让编辑器暂时挂起，返回Shell提示符开始做其他的事情。其他事情做完以后，用户可以重新启动挂起的编辑器，返回到刚才中止的地方，就像用户从来没有离开编辑器一样。这只是一个例子，作业控制还有许多其他实际的用途。

## 创建进程

发布时间 :2006-11-21 21:46:47

在Shell环境下键入需要运行的程序或在图形界面下执行一个程序，其实也就是启动了一个进程。在Linux系统中每个进程都具有一个进程号，用于系统识别和调度。启动一个进程有两个主要途径。手工启动和调度

舳 筋吟鞘李冉 猩柚茫 萦没 G笞孕衅舳

## 手工启动

发布时间 :2006-11-21 21:47:07

由用户输入命令，直接启动一个进程便是手工启动。但手工启动进程又可以分为前台启动和后台启动。

前台启动是手工启动一个进程的最常用的方式。一般用户键入一个命令“ls Cl”，这就已经启动了一个进程，而且是一个前台的进程。这时候系统其实已经处于一个多进程状态。或许有些用户会疑惑：我只启动了一个进程而已。但实际上有许多运行在后台的、系统启动时就已经自动启动的进程正在悄悄运行着。还有的用户在键入“ls Cl”命令以后赶紧使用“ps Cx”查看，却没有看到ls进程，也觉得很奇怪。其实这是因为ls这个进程结束太快，使用 ps查看时该进程已经执行结束了。

直接从后台手工启动一个进程用得比较少一些，除非是该进程甚为耗时，且用户也不急着需要结果的时候。假设用户要启动一个需要长时间运行的格式化文字文件的进程。为了不使整个Shell在格式化过程中都处于“瘫痪”状态，从后台启动这个进程是明智的选择。



## 调度启动

发布时间 :2006-11-21 21:47:25

有时候需要对系统进行一些比较费时而且占用资源的维护工作，这些工作适合在深夜进行，这时候用户就可以事先进行调度安排，指定任务运行的时间或者场合，到时候系统会自动完成这些工作。要使用自动启动进程的功能，就需要掌握以下几个启动命令。

## 调度启动 1 . at 命令

发布时间 :2006-11-21 21:48:11

用户使用at命令在指定时刻执行指定的命令序列。也就是说，该命令至少需要指定一个命令、一个执行时间才可以正常运行。at命令可以只指定时间，也可以时间和日期一起指定。需要注意的是，指定时间有个系统判别问题。比如说，用户现在指定了一个执行时间——凌晨3:20，而发出at命令的时间是头天晚上的20:00，那么究竟是在哪一天执行该命令呢？如果用户在3:20以前仍然在工作，那么该命令将在这个时候完成；如果用户3:20以前就退出了工作状态，那么该命令将在第二天凌晨才得到执行。下面是at命令的语法格式：

```
at [-V] [-q 队列] [-f 文件名] [-mldbv] 时间
```

```
at -c 作业 [作业...]
```

at允许使用一套相当复杂的指定时间的方法，它可以接受在当天的 hh:mm（小时:分钟）式的时间指定。如果该时间已经过去，那么就放在第二天执行。当然也可以使用 midnight（深夜）、noon（中午）、teatime（饮茶时间，一般是下午4点）等比较模糊的词语来指定时间。用户还可以采用12小时计时制，即在时间后面加上AM（上午）或者PM（下午）来说明是上午还是下午。也可以指定命令执行的具体日期，指定格式为month day（月 日）、mm/dd/yy（月/日/年）或者 dd.mm.yy（日.月.年）。指定的日期必须跟在指定时间的后面。

上面介绍的都是绝对计时法，其实还可以使用相对计时法，这对于安排不久就要执行的命令是很有好处的。指定格式为：now + count time-units，now就是当前时间，time-units是时间单位，这里可以是minutes（分钟）、hours（小时）、days（天）、weeks（星期）。count是时间的数量，究竟是几天，还是几小时。

还有一种计时方法就是直接使用today（今天）、tomorrow（明天）来指定完成命令的时间。下面通过一些例子来说明具体用法。

比如，指定在今天下午 5:30 执行某命令。假设现在时间是2001年2月24日中午12:30，其命令格式如下：

```
at 5:30pm
```

```
at 17:30
```

```
at 17:30 today
```

```
at now + 5 hours
```

```
at now + 300 minutes
```

```
at 17:30 24.2.99
```

```
at 17:30 2/24/99
```

```
at 17:30 Feb 24
```

以上这些命令表达的意义是完全一样的，所以在安排时间的时候完全可以根据个人喜好和具体情况自由选择。一般采用绝对时间的24小时计时法可以避免由于用户自己的疏忽造成计时错误的情况发生，例如上例可以写成



at 17:30 2/24/99

这样非常清楚，而且别人也看得懂。

对于at命令来说，需要定时执行的命令是从标准输入或者使用-f选项指定的文件中读取并执行的。如果at命令是从一个使用su命令切换到用户Shell中执行的，那么当前用户被认为是执行用户，所有的错误和输出结果都会送给这个用户。但是如果有邮件送出的话，收到邮件的将是原来的用户，也就是登录时Shell的所有者。

在任何情况下，超级用户都可以使用这个命令。对于其他用户来说，是否可以使用取决于两个文件：`/etc/at.allow`和`/etc/at.deny`。

即时服务器中的Webmin提供了对定时运行命令的图形化配置，登录进入Webmin以后，单击最上边的【系统】进入系统设置模块，选择【命令排程】，即可进入at定时运行命令配置：

使用者：指定运行命令时使用者的身份。

日期：指定运行的日期。

在哪目录执行：指定启动程序的工作目录。

执行命令：指定要运行的命令。

设置完成这些参数以后，单击【建立】即可。

## 2 . cron 命令

发布时间 :2006-11-21 21:48:36

前面介绍的两条命令都会在一定时间内完成一定任务，但是要注意它们都只能执行一次。也就是说，当指定了运行命令后，系统在指定时间完成任务，一切就结束了。但是在很多时候需要不断重复一些命令，比如某公司每周一自动向员工报告头一周公司的活动情况，这时候就需要使用cron命令来完成任务了。实际上，cron命令是不应该手工启动的。cron命令在系统启动时就由一个Shell脚本自动启动，进入后台（所以不需要使用&符号）。一般的用户没有运行该命令的许可权，虽然超级用户可以人工启动cron，不过还是建议将其放到Shell脚本中由系统自行启动。

首先cron命令会搜索/var/spool/cron目录，寻找以/etc/passwd文件中的用户名命名的crontab文件，被找到的这种文件将载入内存。例如一个名为foxy的用户，它所对应的crontab文件就应该是/var/spool/cron/foxy。也就是说，以该用户命名的crontab文件存放在/var/spool/cron目录下面。cron命令还将搜索/etc/crontab文件，这个文件是用不同的格式写成的。cron启动以后，它将首先检查是否有用户设置了crontab文件，如果没有就转入“休眠”状态，释放系统资源。所以该后台进程占用资源极少。它每分钟“醒”过来一次，查看当前是否有需要运行的命令。命令执行结束后，任何输出都将作为邮件发送给crontab的所有者，或者是/etc/crontab文件中MAILTO环境变量中指定的用户。上面简单介绍了一些cron的工作原理，但是cron命令的执行不需要用户干涉；需要用户修改的是crontab中要执行的命令序列，所以下面介绍crontab命令。



### 3 . crontab 命令

发布时间 :2006-11-21 21:49:08

crontab命令用于安装、删除或者列出用于驱动cron后台进程的表格。也就是说，用户把需要执行的命令序列放到crontab文件中执行。每个用户都可以有自己的crontab文件。下面就来看看如何创建一个crontab文件。在/var/spool/cron下的crontab文件不可以直接创建或者直接修改。crontab文件是通过crontab命令得到的。现在假设有个用户名为foxy，需要创建自己的一个crontab文件。首先可以使用任何文本编辑器创建一个新文件，然后向其中写入需要运行的命令和要定期执行的时间。然后保存文件退出。假设该文件为/tmp/test.cron。然后就使用crontab命令来安装这个文件，使之成为该用户的crontab文件。

```
crontab test.cron
```

这样一个crontab 文件就创建好了。可以转到/var/spool/cron目录下面查看，发现多了一个foxy文件。这个文件就是所需的crontab文件。

在crontab文件中如何输入需要执行的命令和时间呢？该文件中每行都包括6个域，其中前5个域是指定命令被执行的时间，最后一个域是要被执行的命令。每个域之间使用空格或者跳位字符分隔。格式如下：

```
minute hour day-of-month month-of-year day-of-week commands
```

第1项是分钟，第2项是小时，第3项是一个月的第几天，第4项是一年的第几个月，第5项是一周的星期几，第6项是要执行的命令。这些项都不能为空，必须填入。如果用户不需要指定其中的几项，那么可以使用\*代替。因为\*是万用字符，可以代替任何字符，所以就可以认为是任何时间，也就是该项被忽略了。

即时服务器中的Webmin提供了对定时运行命令的图形化配置。登录进入Webmin以后，单击最上边的【系统】进入系统配置模块，选择【Cron任务调度】，即可进入Cron配置。

可以看到，这里列出了当前Cron定时运行的所有任务，单击命令上的链接，可以对该定时任务进行编辑，界面和添加新Cron任务是一样的。单击【建立新的定时执行工作】，进入添加界面。

可以看到，需要为Cron任务指定：执行任务的用户身份、执行的命令，以及发送给用户的命令输入和执行程序的时间，设置以后单击【新建】即可。



## 进程的挂起及恢复

发布时间 :2006-11-21 21:49:29

作业控制允许将进程挂起并可以在需要时恢复进程的运行，被挂起的作业恢复后将从中止处开始继续运行。只要在键盘上按【Ctrl+Z】，即可挂起当前的前台作业。在键盘上按【Ctrl+Z】后，将挂起当前执行的命令cat。

使用jobs命令可以显示 Shell的作业清单，包括具体的作业、作业号以及作业当前所处的状态。恢复进程执行时，有两种选择：用fg命令将挂起的作业放回到前台执行；用bg命令将挂起的作业放到后台执行。灵活使用上述命令，将给自己带来很大的方便。

## 进程管理

发布时间 :2006-11-21 21:50:01

由于Linux是个多用户系统，有时候也要了解其他用户现在在干什么；同时Linux是一个多进程系统，经常需要对这些进程进行一些调配和管理；而要进行管理，首先就要知道现在的进程情况：究竟有哪些进程？进程情况如何？所以需要查看进程。

## 1 . who 命令

发布时间 :2006-11-21 21:50:28

该命令主要用于查看当前线上的用户情况。这个命令非常有用。如果用户想和其他用户创建即时通信，比如使用talk命令，那么首先要确定的就是该用户确实上线，不然talk进程就无法创建起来。又如，系统管理员希望监视每个登录的用户此时此刻的行为，也要使用who命令。who命令应用起来非常简单，可以比较准确地掌握用户的情况，所以使用非常广泛。



## 2 . w 命令

发布时间 :2006-11-21 21:50:57

该命令也用于显示登录到系统的用户情况，但是与who不同的是，w命令功能更加强大，它不但可以显示有谁登录到系统，还可以显示出这些用户当前正在进行的工作，并且统计资料相对who命令来说更加详细和科学。可以认为w命令就是who命令的一个增强版。w命令的显示项目按以下顺序排列：当前时间；系统启动到现在的时间；登录用户的数目；系统在最近1秒、5秒和15秒的平均负载。然后是每个用户的各项资料，项目显示顺序如下：登录账号、终端名称、远程主机名、登录时间、空闲时间、JCPU、PCPU、当前正在运行进程的命令行。其中JCPU时间指的是和该终端（tty）连接的所有进程占用的时间。这个时间里并不包括过去的后台作业时间，但却包括当前正在运行的后台作业所占用的时间。而PCPU时间则是指当前进程（即在WHAT项中显示的进程）所占用的时间。

## 3 . ps 命令

发布时间 :2006-11-21 21:51:20

ps命令就是最基本的同时也是非常强大的进程查看命令。使用该命令可以确定有哪些进程正在运行以及运行的状态、进程是否结束、进程有没有僵死、哪些进程占用了过多的资源等等。总之大部分信息都可以通过执行该命令得到。

ps命令最常用的还是用于监控后台进程的工作情况，因为后台进程是不和屏幕键盘这些标准输入/输出设备进行通信的，所以如果需要检测其情况，便可以使用ps命令了。

ps [选项]

下面对命令选项进行说明

-e 显示所有进程。

-f 全格式。

-h 不显示标题。

-l 长格式。

-w 宽输出。

a 显示终端上的所有进程，包括其他用户的进程。

r 只显示正在运行的进程。

x 显示没有控制终端的进程。

O[+|-] k1 [, [+|-] k2 [, ...]] 根据SHORT KEYS, k1, k2中快捷键指定的多级排序顺序显示进程列表。对于ps的不同格式都存在着默认的顺序指定。这些默认顺序可以被用户的指定所覆盖。其中 + 字符是可选的，- 字符是倒转指定键的方向。

最常用的3个参数是u, a, x。

下面是命令

```
#ps axu
```

的输出格式：

```
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
```

USER：进程所有者。

PID：进程号。

%CPU：占用的 CPU 使用率。

%MEM：占用的内存使用率。

VSZ：占用的虚拟内存大小。

RSS：占用的内存大小。

TTY：终端的次设备号码。

STAT：该进程的状态。

D：不可中断的休眠（通常表示该进程正在进行I/O动作）。

R：正在执行中。

S：休眠状态。

T：暂停执行。

Z：不存在但暂时无法消除。

W：没有足够的内存分页可分配。

<：高优先顺序的进程。

N：低优先顺序的进程。

L：有内存分页分配并锁在内存内（即时系统或定制I/O）。

START：进程开始时间。

TIME：执行的时间。

COMMAND：所执行的命令

例：

```
#ps
```

```
PID TTY TIME CMD
```

```
2791 tty0 00:00:00 tcsh
```

```
3092 tty0 00:00:00 ps
```

```
#ps -A
```

```
PID TTY TIME CMD
```

```
1 ? 00:00:03 init
```



2 ? 00:00:00 kflushd

3 ? 00:00:00 kpiod

4 ? 00:00:00 kswapd

5 ? 00:00:00 mdrecoveryd

.....

# ps -aux

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

root 1 0.0 0.7 1096 472 ? S Sep10 0:03 init [3]

root 2 0.0 0.0 0 0 ? SW Sep10 0:00 [kflushd]

root 3 0.0 0.0 0 0 ? SW Sep10 0:00 [kpiod]

root 4 0.0 0.0 0 0 ? SW Sep10 0:00 [kswapd]

## 4 . top 命令

发布时间 :2006-11-21 21:51:54

top命令和ps命令的基本作用是相同的，用于显示系统当前的进程和其他状况。但是top是一个动态显示过程，即可以通过用户按键来不断刷新当前状态。如果在前台执行该命令，它将独占前台，直到用户终止该程序为止。准确地说，top命令提供了即时的对系统处理器的状态监视。它将显示系统中CPU最“敏感”的任务列表。该命令可以按CPU使用、内存使用和执行时间对任务进行排序；而且该命令的很多特性都可以通过交互式命令或者在个人定制文件中进行设置。

下面是该命令的语法格式：

```
top [-] [d delay] [q] [c] [s] [S] [i]
```

其中：

d 指定每两次屏幕信息刷新之间的时间间隔。当然用户可以使用s交互命令来改变它。

q 该选项将使top没有任何延迟地进行刷新。如果调用程序有超级用户许可权，那么top将以尽可能高的优先顺序运行。

S 指定累计模式。

s 使top命令在安全模式中运行。这将去除交互命令所带来的潜在危险。

i 使top不显示任何闲置或者僵死进程。

c 显示整个命令行而不只是显示命令名。

top命令显示的项目很多，默认值是每5秒刷新一次，当然这是可以设置的。

显示的各项目为：uptime项显示的是系统启动时间、已经运行的时间和3个平均负载值（最近1秒、5秒、15秒的负载值）。processes表示自最近一次刷新以来的运行进程总数。当然这些进程被分为正在运行的、休眠的、停止的等很多种类。进程和状态显示可以通过交互命令t来实现。

CPU states用于显示用户模式、系统模式、优先顺序进程（只有优先顺序为负的列入考虑）和闲置等各种情况所占用CPU时间的百分比。优先顺序进程所消耗的时间也被列入到用户和系统的时间中，所以总的百分比将大于100%。

Mem为内存使用情况统计，其中包括总的可用内存、空闲内存、已用内存、共用内存和缓存所占内存的情况。

Swap为交换空间统计，其中包括总的交换空间、可用交换空间、已用交换空间。

PID 每个进程的ID。

PPID 每个进程的父进程ID。



UID 每个进程所有者的UID。

USER 每个进程所有者的用户名。

PRI 每个进程的优先顺序别。

NI 该进程的优先顺序值。

SIZE 该进程的代码大小加上资料大小再加上堆叠空间大小的总数，单位是KB。

TSIZE 该进程的代码大小。对于内核进程这是一个很奇怪的值。

DSIZE 资料 and 堆叠的大小。

TRS 文本驻留大小。

D 被标记为“不干净”的页项目。

LIB 使用的库页的大小。对于ELF进程没有作用。

RSS 该进程占用的实体内存的总数量，单位是KB。

SHARE 该进程使用共用内存的数量。

STAT 该进程的状态。其中，S代表休眠状态；D代表不可中断的休眠状态；R代表运行状态；Z代表僵死状态；T代表停止或跟踪状态。

TIME 该进程自启动以来所占用的总CPU时间。如果进入的是累计模式，那么该时间还包括这个进程子进程所占用的时间。且标题会变成CTIME。

%CPU 该进程自最近一次刷新以来所占用的CPU时间和总时间的百分比。

%MEM 该进程占用的实体内存占总内存的百分比。

COMMAND 该进程的命令名称，如果一行显示不下，则会进行截取。内存中的进程会有一个完整的命令行。

下面介绍在top命令执行过程中可以使用的一些交互命令。从使用角度来看，熟练的掌握这些命令比掌握选项还重要一些。这些命令都是单字母的，如果在命令行选项中使用了s选项，则可能其中一些命令会被遮掉。

<空格> 立即刷新显示。

Ctrl+L 擦除并且重写屏幕。

h或者? 显示帮助画面，给出一些简短的命令总结说明。

k 终止一个进程。系统将提示用户输入需要终止的进程PID，以及需要发送给该进程的信号。一般的终止进程可以使用信号15；如果不能正常结束那就使用信号9强制结束该进程。默认值是信号15。在安全模式中此命令被遮罩。

i 忽略闲置和僵死进程。这是一个开关式命令。



q 退出程序。

r 重新安排一个进程的优先顺序。系统提示用户输入需要改变的进程PID以及需要设置的进程优先顺序值。输入一个正值将使优先顺序降低，反之则可以使该进程拥有更高的优先权。默认值是10。

S 切换到累计模式。

s 改变两次刷新之间的延迟时间。系统将提示用户输入新的时间，单位为s。如果有小数，就换算成ms。输入0则系统将不断刷新，默认值是5s。需要注意的是，如果设置太小的时间，很可能会引起不断刷新，从而根本来不及看清显示的情况，而且系统负载也会大大增加。

f或者F 从当前显示中添加或者删除项目。

o或者O 改变显示项目的顺序。

l 切换显示平均负载和启动时间信息。

m 切换显示内存信息。

t 切换显示进程和CPU状态信息。

c 切换显示命令名称和完整命令行。

M 根据驻留内存大小进行排序。

P 根据CPU使用百分比大小进行排序。

T 根据时间/累计时间进行排序。

W 将当前设置写入 ~/.toprc 文件中。这是写top配置文件的推荐方法。

从上面的介绍中可以看到，top命令是一个功能十分强大的监控系统工具，尤其对于系统管理员而言更是如此。一般的用户可能会觉得ps命令其实就够用了，但是top命令的强劲功能确实提供了不少方便。下面来看看实际使用的情况。

第1行的项目依次为当前时间、系统启动时间、当前系统登录用户数目、平均负载。第2行为进程情况，依次为进程总数、休眠进程数、运行进程数、僵死进程数、终止进程数。第3行为CPU状态，依次为用户占用、系统占用、优先进程占用、闲置进程占用。第4行为内存状态，依次为平均可用内存、已用内存、空闲内存、共用内存、缓存使用内存。第5行为交换状态，依次为平均可用交换容量、已用容量、闲置容量、快取内存容量。接下来就是和ps相仿的各进程情况列表了。

## 5 . Kill 命令

发布时间 :2006-11-21 21:52:41

当需要中断一个前台进程的时候，通常是使用组合键【Ctrl+C】；但是对于一个后台进程恐怕就不是一个组合键所能解决的了，这时就必须求助于kill命令。该命令可以终止后台进程。终止后台进程的原因很多，或许是该进程占用的CPU时间过，或许是该进程已经挂死。总之这种情况是经常发生的。

kill命令是通过向进程发送指定的信号来结束进程的。如果没有指定发送信号，那么默认值为TERM信号，它将终止所有不能捕获该信号的进程。至于那些可以捕获该信号的进程可能就需要使用kill（9）信号了，该信号是不能被捕捉的。

kill命令的语法格式很简单，大致有以下两种方式：

```
kill [-s 信号 | -p ] [ -a ] 进程号 ...
```

```
kill -l [信号]
```

其中，

-s 指定需要送出的信号，既可以是信号名也可以是对应数位。

-p 指定kill命令只是显示进程的pid，并不真正送出结束信号。

-l 显示信号名称列表，这也可以在/usr/include/Linux/signal.h文件找到。

下面看看该命令的使用。

例在执行一条find命令时由于时间过长，决定终止该进程。

首先应该使用ps命令来查看该进程对应的PID。键入ps，显示如下：

```
PID TTY TIME COMMAND
```

```
285 1 00:00:00 -bash
```

```
287 3 00:00:00 -bash
```

```
289 5 00:00:00 /sbin/mingetty tty5
```

```
290 6 00:00:00 /sbin/mingetty tty6
```

```
312 3 00:00:00 telnet bbs3
```

```
341 4 00:00:00 /sbin/mingetty tty4
```

```
345 1 00:00:00 find / -name foxy.jpg
```



```
348 1 00:00:00 ps
```

可以看到，该进程对应的PID是345，现在使用kill命令来终止该进程。键入：

```
# kill 345
```

再用ps命令查看，就可以看到，find进程已经被终止了。

例：杀掉进程11721。

```
# ps
```

```
PID TTY TIME COMMAND
```

```
11668 p1 00:00:00 -tcsh
```

```
11721 p1 00:00:00 cat
```

```
11737 p1 00:00:00 ps
```

```
# kill 11721
```

```
[1] Terminated cat
```

有时候可能会遇到这样的情况，某个进程已经挂起或闲置，使用kill命令却终止不掉。这时候就必须发送信号9，强行关闭此进程。当然这种“野蛮”的方法很可能会导致打开的文件出现错误或者资料丢失。所以不到万不得已的时候不要使用强制结束的办法。如果连信号9都不回应，那恐怕就只有重新启动电脑了。

## Linux所有的信号

讯 号

描 述

SIGABRT

处理退出信号



SIGALRM

报警时钟

SIGFPE

错误的算术操作

SIGHUP

挂起

SIGILL

非法命令

SIGINT

中断信号

SIGKILL

Kill（不能忽略）

SIGPIPE

向管道写，没有程序读

SIGQUIT

退出信号

SIGSEGV

非法内存引用

SIGTERM

终止信号

SIGUSR1

用户定义信号1

SIGUSR2

用户定义信号2

SIGCHLD

子进程终止或停止

SIGCONT

如果停止，继续执行

SIGSTOP

停止执行（不能忽略）

SIGTSTP

停止信号

SIGTTIN

后台进程尝试读

SIGTTOU

后台进程尝试写

SIGBUS

汇流排错误

SIGPOLL

Pollable event

SIGPROF

剖析计时器到时

SIGSYS

非法系统调用

SIGTRAP



跟踪/中断点陷阱

SIGURG

在 socket 中高频宽资料可用

SIGVTALRM

虚拟计时器到时

SIGXCPU

超过CPU时间限制

SIGXFSZ

超过文件大小限制

## 6 . nohup 命令

发布时间 :2006-11-21 21:53:27

理论上，我们一般退出Linux系统时会把所有的程序全部结束，包括那些后台程序。有些时候，例如你正在编辑一个很长的程序，但是你下班或是有事需要先退出系统，你又不希望系统把你编辑那么久的程序结束，希望退出系统时，程序还能继续执行。这时，我们就可以使用nohup命令使进程在用户退出后仍继续执行。

一般我们都是让这些进程在后台执行，结果会写到用户自己目录下的nohup.out文件里（也可以使用输出重定向，让它输出到一个特定的文件）。

例如：

```
# nohup sort sales.dat &
```

这条命令告诉sort命令忽略用户已退出系统，它应该一直运行，直到进程完成。利用这种方法，可以启动一个要运行几天甚至几周的进程，而且在它运行时，用户不需要去登录。

nohup命令把一条命令的所有输出和错误信息送到nohup.out文件中。若将输出重定向，则只有错误信息放在nohup.out文件中。

## 基于 Webmin 的进程管理

发布时间 :2006-11-21 21:53:52

即时服务器中的Webmin提供了对定时进程管理的图形化配置，登录进入Webmin以后，单击最上边的【系统】进入系统配置模块，选择【进程管理器】即可进入进程管理。

可以看到，视窗中列出所有的进程，目前是按照进程号PID的大小顺序来排列显示的。在最上端的有几个链接：用户、内存和CPU分别用来使用不同的属性顺序来显示进程。“搜寻”链接用来搜索某个特定的进程。单击该链接。

用户可以在此指定进程属主、进程的工作目录、进程打开的文件、使用CPU的百分比、打开的端口等属性来搜索进程。



## 海量 Linux 技术文章

发布时间 :2006-11-15 11:32:55

下面是linux技术文章快速入口。需要联网：

Linux 技术交流

<http://www.linuxdiyf.com/bbs/forum-3-1.html>

Linux 应用

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=1>

Linux 安装及学习指导

<http://www.linuxdiyf.com/bbs/forum-45-1.html>

Linux 系统安装

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=45&filter=type&typeid=11>

Linux 学习指导

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=45&filter=type&typeid=12>

Linux 软件安装

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=45&filter=type&typeid=13>

shell

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=3>

Linux 壁纸

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=4>

红旗

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=5>

Redhat

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=6>

SuSE

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=7>

## Linux 认证

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=9>

## Linux下载分享{酷件、书籍、视频分享 }

<http://www.linuxdiyf.com/bbs/forum-6-1.html>

## 服务器应用

<http://www.linuxdiyf.com/bbs/forum-7-1.html>

## 数据库应用

<http://www.linuxdiyf.com/bbs/forum-8-1.html>

## Linux 编程与内核

<http://www.linuxdiyf.com/bbs/forum-9-1.html>

## UniX 技术文章

<http://www.linuxdiyf.com/bbs/forum-32-1.html>

## Linux 业界声音、新闻

<http://www.linuxdiyf.com/bbs/forum-11-1.html>

## Linux 人才招聘信息

<http://www.linuxdiyf.com/bbs/forum-46-1.html>

制作：红联Linux论坛

祝您阅读愉快！

内容来源于网络，汇集成一本电子书籍，让大家都得到成长，感谢原创作者，衷心感谢！

