


# Linux如何查看和控制进程



一米八是我呀

发布时间：18-11-18 13:29

## Linux如何查看和控制进程

前言：程序是保存在外部存储介质（如硬盘）中的可执行机器代码和数据的静态集合，而进程是在CPU及内存中处于动态执行状态的计算机程序。在Linux系统中，每个程序启动后可以创建一个或多个进程。例如，提供Web服务的httpd程序，当有大量用户同时访问Web页面时，httpd程序可能会创建多个进程来提供服务。

- 程序
- 保存在硬盘、光盘等介质中的可执行代码和数据

静态保存的代码

- 进程
- 在 CPU 及内存中运行的程序代码

动态执行的代码

父、子进程

每个进程可以创建一个或多个进程

下面我们将一起来学习查看进程信息及控制进程相关的操作命令。

### 一、查看进程

使用不同的命令工具可以从不同的角度查看进程状态。

常用的进程查看命令

#### 1.ps命令——查看静态的进程统计信息（Processes Statistic）

常见的选项：

- a：显示当前终端下的所有进程信息，包括其他用户的进程。
- u：使用以用户为主的格式输出进程信息。
- x：显示当前用户在所有终端下的进程。
- e：显示系统内的所有进程信息。
- l：使用长（long）格式显示进程信息。

## 作者最新文章

- 实验报告——MySQL的安装
- Shell脚本应用——条件测试操作
- shell脚本与计划任务

## 相关文章

Linux编程基础，码农们需要知道的一些Bash常识



在EXCEL 2013中注册日期控件的方法



从C语言代码分析Linux系统是如何创建进程的



性能测试资源监控操作指南之Linux类操作系统

需要注意的是，有一部分选项是不带“-”前缀的（添加“-”前缀后含义可能会有出入）。习惯上将上述选项组合在一起使用，如，“ps aux”或“ps -elf”

示例1：ps aux 将以简单列表的形式显示出进程信息。如下图

```
[root@localhost ~]# ps aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	19356	1492	?	Ss	10:27	0:00	/sbin/init
root	2	0.0	0.0	0	0	?	S	10:27	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S	10:27	0:00	[migration/0]
root	4	0.0	0.0	0	0	?	S	10:27	0:00	[ksoftirqd/0]
root	5	0.0	0.0	0	0	?	S	10:27	0:00	[migration/0]
root	6	0.0	0.0	0	0	?	S	10:27	0:00	[watchdog/0]
root	7	0.0	0.0	0	0	?	S	10:27	0:00	[migration/1]
root	8	0.0	0.0	0	0	?	S	10:27	0:00	[migration/1]
root	9	0.0	0.0	0	0	?	S	10:27	0:00	[ksoftirqd/1]
root	10	0.0	0.0	0	0	?	S	10:27	0:00	[watchdog/1]
root	11	0.0	0.0	0	0	?	S	10:27	0:00	[events/0]
root	12	0.0	0.0	0	0	?	R	10:27	0:00	[events/1]
root	13	0.0	0.0	0	0	?	S	10:27	0:00	[cgroup]
root	14	0.0	0.0	0	0	?	S	10:27	0:00	[khelper]
root	15	0.0	0.0	0	0	?	S	10:27	0:00	[netns]
root	16	0.0	0.0	0	0	?	S	10:27	0:00	[async/mgr]
root	17	0.0	0.0	0	0	?	S	10:27	0:00	[pm]
root	18	0.0	0.0	0	0	?	S	10:27	0:00	[sync_supers]
root	19	0.0	0.0	0	0	?	S	10:27	0:00	[bdi-default]
root	20	0.0	0.0	0	0	?	S	10:27	0:00	[kintegrityd/0]
root	21	0.0	0.0	0	0	?	S	10:27	0:00	[kintegrityd/1]

上图中的输出信息中，第1行为列表标题，其中各字段的含义描述如下：

USER：启动该进程的用户账号名称

PID：该进程的ID号，在当前系统中是唯一的

TTY：该进程在哪个终端上运行。“？”表未知或不需要终端

STAT：显示了进程当前的状态，如S（休眠）、R（运行）、Z（僵死）、<（高优先级）、N（低优先级）、s（父进程）、+（前台进程）。对处于僵死状态的进程应予以手动终止。

START：启动该进程的时间

TIME：该进程占用CPU时间

COMMAND：启动该进程的命令的名称

%CPU：CPU占用的百分比

%MEM：内存占用的百分比

VSZ：占用虚拟内存（swap空间）的大小

RSS：占用常驻内存（物理内存）的大小

示例2：ps -elf 以长格式显示系统中的进程信息，包含更丰富的内容。大概意思都一样，PPID为父进程的PID。



Win10家庭版系统也能使用组策略啦！终于关闭系统自动更新了



F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TIME	TTY	TIME	CMD
4	S	root	1	0	0	80	0	-	4838	poll_w	10:27	?	00:00:00	/sbin/init
1	S	root	2	0	0	80	0	-	0	kthrea	10:27	?	00:00:00	[kthreadd]
1	S	root	3	2	0	-40	-	-	0	migrat	10:27	?	00:00:00	[migration/0]
1	S	root	4	2	0	80	0	-	0	ksofti	10:27	?	00:00:00	[ksoftirqd/0]
1	S	root	5	2	0	-40	-	-	0	cpu_st	10:27	?	00:00:00	[migration/0]
5	S	root	6	2	0	-40	-	-	0	watchd	10:27	?	00:00:00	[watchdog/0]
1	S	root	7	2	0	-40	-	-	0	migrat	10:27	?	00:00:00	[migration/1]
1	S	root	8	2	0	-40	-	-	0	cpu_st	10:27	?	00:00:00	[migration/1]
1	S	root	9	2	0	80	0	-	0	ksofti	10:27	?	00:00:00	[ksoftirqd/1]
5	S	root	10	2	0	-40	-	-	0	watchd	10:27	?	00:00:00	[watchdog/1]
1	S	root	11	2	0	80	0	-	0	worker	10:27	?	00:00:00	[events/0]
1	R	root	12	2	0	80	0	-	0	-	10:27	?	00:00:00	[events/1]
1	S	root	13	2	0	80	0	-	0	worker	10:27	?	00:00:00	[cgroup]
1	S	root	14	2	0	80	0	-	0	worker	10:27	?	00:00:00	[khelper]
1	S	root	15	2	0	80	0	-	0	worker	10:27	?	00:00:00	[netns]
1	S	root	16	2	0	80	0	-	0	async_	10:27	?	00:00:00	[async/mgr]
1	S	root	17	2	0	80	0	-	0	worker	10:27	?	00:00:00	[pm]
1	S	root	18	2	0	80	0	-	0	bdi_sy	10:27	?	00:00:00	[sync_supers]
1	S	root	19	2	0	80	0	-	0	bdi_fo	10:27	?	00:00:00	[bdi-default]
1	S	root	20	2	0	80	0	-	0	worker	10:27	?	00:00:00	[kintegrityd/0]
1	S	root	21	2	0	80	0	-	0	worker	10:27	?	00:00:00	[kintegrityd/1]
1	S	root	22	2	0	80	0	-	0	worker	10:27	?	00:00:00	[kblockd/0]
1	S	root	23	2	0	80	0	-	0	worker	10:27	?	00:00:00	[kblockd/1]

示例3：ps 直接执行不带任何选项，只显示当前用户会话中打开的进程。

```
[root@localhost ~]# ps
```

```
[root@localhost 桌面]# ps
  PID TTY          TIME CMD
 2224 pts/0      00:00:00 bash
 2234 pts/0      00:00:00 _ps
```

示例4：结合管道操作和grep命令进行过滤，用于查询某一个进程的信息。

```
[root@localhost ~]# ps aux | grep bash
```

```
[root@localhost 桌面]# ps aux | grep bash
root      2224  0.0  0.1 108336 1776 pts/0    Ss   21:43   0:00 /bin/bash
root      2247  0.0  0.0 103256   844 pts/0    S+   21:47   0:00 grep bash
```

## 2.top命令——查看进程动态信息

以全屏交互式的界面显示进程排名，及时跟踪包括CPU、内存等系统资源占用情况，默认情况下每三秒刷新一次，其作用基本类似于Windows系统中的任务管理器。

示例：

```
[root@localhost ~]# top
```

Tasks: 121 total, 1 running, 120 sleeping, 0 stopped, 0 zombie  
Cpu(s): 0.0%us, 0.2%sy, 0.0%ni, 99.8%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st  
Mem: 1923392k total, 1223652k used, 699740k free, 113892k buffers  
Swap: 4194296k total, 0k used, 4194296k free, 889964k cached

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1843	root	20	0	98.0m	4148	3124	S	0.3	0.2	0:00.18	sshd
5113	root	20	0	15036	1184	904	R	0.3	0.1	0:00.11	top
1	root	20	0	19356	1496	1188	S	0.0	0.1	0:00.58	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:00.75	migration/0
4	root	20	0	0	0	0	S	0.0	0.0	0:00.02	ksoftirqd/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
6	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
7	root	RT	0	0	0	0	S	0.0	0.0	0:00.18	migration/1
8	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/1
9	root	20	0	0	0	0	S	0.0	0.0	0:00.07	ksoftirqd/1
10	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/1

上图中输出信息开头部分相关信息的含义如下：

无响应的进程数。

- CPU信息：us，用户占用；sy，内核占用；ni，优先级调度占用；id，空闲CPU；wa，I/O等待占用；hi，硬件中断占用；si，软件中断占用；st，虚拟化占用。了解空闲的CPU百分比，主要看%id部分。
- Mem（内存）信息：total，总内存空间；used，已用内存；free，空闲内存；buffers，缓存区域。
- Swap（交换空间）信息：total，总交换空间；used，已用交换空间；free，空闲交换空间；cached，缓存空间。

在top命令的全屏操作界面中，按P键根据CPU占用情况对进程列表进行排序，或按M键根据内存占用情况排序，按N键根据启动时间进行排序，按h键可以获得top程序的在线帮助信息，按q键可以正常地退出top程序。

若通过top排名工具发现某个进程CPU占用率非常高，需要终止该进程的运行时，可以在top操作界面按k键，然后在列表上方将会出现“PID to kill”的提示信息，根据提示输入指定进程的PID号并按enter键确认即可终止对应的进程。

（个人感觉没必要都记得清清楚楚，了解一下吧，知道大概意思就行，用的时候拉出来看看）

### 3.pgrep命令——根据特定条件查询进程PID信息

示例：

```
[root@localhost ~]# pgrep -l "log"
2538 rsyslogd
2113 mcelog
[root@localhost ~]# pgrep -l -U teacher -t tty1
27483 bash
27584 vim
```

-l: 显示进程名

-U: 指定特定用户  
-t: 指定终端

### 4.pstree命令——查看进程树，以树形结构列出进程信息

示例：



## 二、控制进程

### 1. 启动进程

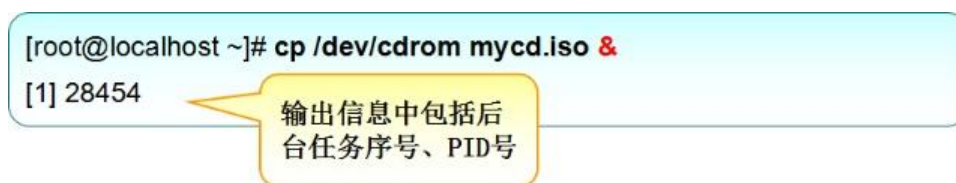
进程的启动方式：

- **手工启动：**由用户手工输入命令或执行程序的路径，可以至少启动一个进程。手工启动包括：前台启动和后台启动。

前台启动：用户输入命令，直接执行程序

后台启动：在命令行尾加入“&”符号

示例：



后台启动后直接放入后台运行，而不占用前台的命令操作界面，方便用户进行其他操作。

- **调度启动：**用于服务器维护工作中，例如当需要执行一些比较费时而且占用资源的任务（如数据备份），这些任务更适合在相对空闲的时候（如夜间）执行。这时就需要用户事先进行调度安排，指定任务运行的时间，当系统到达指定设定时间时会自动启动并完成指定的任务。调度启动的计划任务进程均在后台运行，不会占用用户的命令终端。调度启动可以通过at、crontab命令进行设置。

使用 at 命令，设置一次性计划任务

使用 crontab 命令，设置周期性计划任务

（关于at和crontab的配置，准备下篇文章再讲解）

### 2. 改变进程的运行方式

#### 1) 挂起当前的进程



机搜索东西时，发现他搜索的老慢了甚至电脑都有点卡，我们突然不想让它搜了，就想马上让它停止搜索，就是这种感觉。

## 2) 查看后台的进程

使用**jobs**命令，可以查看当前终端在后台的进程任务，结合“-l”选项可以同时显示出该进程对应的PID号

示例：

```
[root@localhost ~]# jobs
[1]-  Stopped          cp /dev/cdrom mycd.iso
[2]+  Stopped          top
```

## 3) 将后台的进程恢复运行

**bg** (BackGround) 命令，可以将后台中暂停执行（如，按Ctrl+Z组合键挂起）的任务恢复运行，继续在后台执行

**fg** (ForeGround) 命令，可以将后台任务重新恢复到前台运行

示例：

```
[root@localhost ~]# jobs
[1]-  Stopped          cp /dev/cdrom mycd.iso
[2]+  Stopped          top
[root@localhost ~]# fg 1
```

除非后台中的任务只有一个，否则bg和fg命令都需要指定后台进程的任务编号作为参数

## 3. 终止进程执行

### 1) Ctrl+C组合键

强制中断正在执行的命令，如，命令长时间没有响应的情况下。

### 2) kill命令

用于终止指定PID号的进程，**需要使用进程的PID号作为参数**。无特定选项时，kill命令将给该进程发送终止信号并正常退出运行，有时可能该进程已经无法响应终止信号，这时可以结合“-9”（这是数字9不是字母g，有时候容易看走眼）选项强制终止进程。强制终止进程可能会导致程序运行的部分数据丢失，因此不到不得已时慎用。

示例

```
[root@localhost ~]# kill -9 2869 //强制终止目标进程
[root@localhost ~]# pgrep -l "portmap" //确认进程已终止（查询时无结果）
```

### 3) killall命令

用于终止指定名称的所有进程，当需要结束系统中多个相同名称的进程时，使用killall命令将更加方便，效率更高。Killall命令同样也有“-9”选项。

示例：

```
[root@localhost ~]# vim testfile1 //挂起第一个vim测试进程
[1]+  Stopped          vim testfile1
[root@localhost ~]# vim testfile2 //挂起第二个vim测试进程
[1]+  Stopped          vim testfile2
[root@localhost ~]# job -l //确认待终止的进程信息
[1] -  3029  停止      vim testfile1
[2] +  3030  停止      vim testfile2
[root@localhost ~]# killall -9 vim //通过进程名终止多个进程
[1]- 已杀死          vim testfile1
[2]+ 已杀死          vim testfile2
```

### 4) pkill命令

根据特定条件终止相应的进程

常用选项：（大部分选项与pgrep命令基本类似）

-U：根据进程所属的用户名终止相应进程

-t：根据进程所在的终端终止相应进程

示例：

```
[root@localhost ~]# pgrep -l -U "hackli" //确认目标进程的相关信息
3045 bash
[root@localhost ~]# pkill -9 -U "hackli" //强行终止用户hackli的进程
[root@localhost ~]# pgrep -l -U "hackli" //确认目标进程已被终止
```

下一篇文章将总结如何创建计划任务（at、crontab计划任务）