

《Linux基础》



第九讲 进程管理

进程基本介绍

- 运行中的程序就是进程。系统会给每一个进程分配一个数字进行标记，此数字就是进程ID（PID）。
- 父进程：如果进程B由进程A来创建，则进程A就是进程B的父进程，进程B就是进程A的子进程。
- PPID：父进程的进程ID。
- 在shell中输入命令运行，则shell就是父进程，运行的程序就是shell的子进程。（守护进程除外，守护进程的父进程是init[进程ID为1]）。

UID, EUID, GID, EGID

- UID (user id) : 每一个进程都有一个所属用户ID, 就是运行程序的用户的ID。每一个进程都有一个父进程, 通常情况下去进程的UID继承自父进程。但是系统通过系统调用传递参数更改进程的UID以标识进程的创建者。
- EUID (effective user id) : 有效用户ID, 表示进程对于文件和资源的访问权限。大多数情况下EUID和UID是相同的。对于设置了相关标志位的程序, 运行时会改变EUID, 系统根据EUID赋予进程相应的权限。
- GID (group id) : 进程的组ID, 程序所属组的ID, 参考UID解释。
- EGID (effective group id) : 有效组ID, 参考EUID解释。

进程管理相关的命令

ps 查看当前进程

kill 向进程发送信号，通常是终止进程

bg shell内建命令，后台任务继续执行，就像在命令后面加入&

fg shell内建命令，后台任务转至前台

jobs shell内建命令，显示后台运行的任务

pgrep 搜索进程

top/htop 动态监控进程情况，系统资源使用情况

nice/renice 调整进程优先级

查看进程

- 使用示例：

- `ps -e` //查看所有进程

- `ps -u root` //查看所有root用户的进程

- `ps -aux` //查看详细信息，如有效用户，pid，CPU占用，内存占用等

- `ps -ejH` //显示进程树

- 运行 `man ps` 查看ps手册。

- 分页查看进程信息：`ps -aux | less`

- 按列显示：`ps -o user,pid,ppid,tty,comm,args -e`

根据名称关键字搜索进程

- Linux提供了pgrep命令查询进程信息。
- 使用方式：`pgrep [options] KEYWORD`
 - `pgrep -a sh` 搜索名称含有sh的进程，并列出详细信息。
 - `pgrep -l sh` 仅仅列出名称和PID。
- 另一种方式：`ps -e | grep KEYWORD`

终止进程：kill

- kill：向指定的进程发送信号。 `kill -l` 查看所有的信号。
- 示例：
 - `kill 1234` //向PID为1234的进程发送终止信号。
- 默认的信号是SIGTERM，此信号通常会终止程序的运行，但程序可能仍然继续运行。这个时候可以向进程传递SIGKILL信号，会强制进程结束。
 - `kill -9 1234` //向PID为1234的进程发送kill信号。
- SIGINT，使用ctrl+c终止程序运行时发送的就是SIGINT信号，SIGINT、SIGTERM可以被程序捕获并进行自定义处理，而SIGKILL不能被捕获。

后台任务

- 一个任务如果运行时间太长，或者是需要长期运行的情况，此时想要获取终端控制权。可以把任务转至后台。
- 在命令后面加入&便可以直接使任务直接在后台运行。
- 而在运行中输入Ctrl+Z会把当前任务转至后台**暂停**。此时使用jobs可以显示后台的任务，每个任务都有一个编号。使用bg [后台任务编号]可以使任务在后台继续执行。
- fg [后台任务编号]把后台的任务转至前台执行。

动态监控进程以及资源占用

- Ubuntu自带top命令动态查看进程。htop是top的升级版。
- htop需要下载安装，安装命令：`sudo apt-get install htop`。
- saidar可以动态监控资源占用，但不显示进程的信息