

Linux 基础



第 8 讲 进程管理

Linux 进程基础

- 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，一般以 PID 表示。

Linux 进程基础

- 父进程和子进程：如果进程 B 由进程 A 创建，则 A 是 B 的父进程，B 是 A 的子进程。
- PPID：父进程 ID。
- 在 shell 中运行命令，shell 是其父进程。（如果程序运行后，创建了守护进程，此时会脱离 shell 控制。）

UID

- `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, 通常情况父子进程的 UID 继承自父进程。

EUID

- **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。
- **EUID** 可通过系统调用设置, 更改后, 进程的权限就会发生变化。

GID, EGID

- 组 ID 和有效组 ID。
- 参考 UID 和 EUID 的解释。

进程管理相关命令

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

- `ps` 是用于查看进程的命令。
- 直接输入 `ps` 显示的是在当前 `shell` 中运行的进程。
- `ps -e` 显示所有进程。

查看进程详细信息

- 运行 `ps aux` 可以按照 BSD 风格查看信息

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.1	225540	9264	?	Ss	08:27	0:02	/sbin/init splash
root	2	0.0	0.0	0	0	?	S	08:27	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	08:27	0:00	[rcu_gp]
root	4	0.0	0.0	0	0	?	I<	08:27	0:00	[rcu_par_gp]
root	6	0.0	0.0	0	0	?	I<	08:27	0:00	[kworker/0:0H-kb]
root	8	0.0	0.0	0	0	?	I<	08:27	0:00	[mm_percpu_wq]
root	9	0.0	0.0	0	0	?	S	08:27	0:00	[ksoftirqd/0]
root	10	0.1	0.0	0	0	?	I	08:27	0:35	[rcu_sched]
root	11	0.0	0.0	0	0	?	I	08:27	0:00	[rcu_bh]
root	12	0.0	0.0	0	0	?	S	08:27	0:00	[migration/0]
root	13	0.0	0.0	0	0	?	S	08:27	0:00	[watchdog/0]

按列显示

- 很多时候需要按照指定的列显示，尤其在 `shell` 脚本中，要进行严格的匹配查找。
- 按照‘用户， PID ， PPID ， 命令， 参数’的顺序显示：

```
ps -e -o user,pid,ppid,comm,args
```

其他使用示例

- 显示进程树: `ps -ejH`
- 显示某一进程详细信息: `ps -l [PID]`
- 分页显示: `ps aux | less`

搜索进程

- `pgrep` 用于搜索进程：
 - `pgrep -a sh` 搜索名称含有 `sh` 的进程并显示详细信息。
 - `pgrep -l ab` 搜索名称含有 `ab` 的进程，显示名称和 PID。
- 使用管道（正则匹配）：
 - `ps -e -o pid,comm,args | grep 'node.*serv'`

终止进程

- 终止进程的命令： `kill` 。
- 从名称上看， `kill` 就是终止进程，事实上， `kill` 是向进程发送信号。
- 默认发送的信号是 `SIGTERM` ，这个信号表示中断进程，通常情况，进程就会退出。

kill 与信号

- Linux/Unix 通过信号机制控制进程， `kill` 命令可以用来发送信号。
- `kill -l` 可以查看所有的信号，显示信号名称和信号的数字编号。

在 shell 中 Ctrl+C 快捷键产生的是 SIGINT 信号，这个信号默认也是终止进程的。

kill 与信号

- 除了终止进程， `kill` 还可以向指定进程发送特定的信号。
- 比如，通过 `kill -l` 查到时钟信号（ `SIGALRM` ）的数字是 14，可以向指定进程发送时钟信号：

`kill -14 [PID]` # 如果 pid 是 1234, 则运行 `kill -14 1234`

或者使用名称的形式

`kill -s SIGALRM [PID]`

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运行。
- 如果需要暂停任务，也可以把任务转至后台暂停。

暂停任务

- 在 `shell` 中运行命令，`Ctrl+Z` 快捷键可以把任务转到后台暂停。
- 这种情况其实是对进程发送了 `SIGSTOP` 信号：
`kill -19 [PID]`

管理后台任务

- 命令 `jobs` 可以显示后台的任务，每个任务都有一个编号。
- 命令 `fg [任务编号]` 可以把任务转到前台继续执行。
- 命令 `bg [任务编号]` 让任务在后台继续执行。

直接在后台运行命令

- 在 `shell` 输入命令，如果最后加上 `&`，则会在后台执行命令。
- 注意，如果输出没有重定向，则仍然会在 `shell` 中显示，在后台运行的任务一般把输出进行重定向。

* 进程优先级 *

- 运行 `nice` 会显示当前 `shell` 的优先级。一般情况下，子进程会继承父进程的优先级。系统内核在进程运行时也会动态调整。
- `nice ps` 会在当前优先级加上一个数值（默认是 10）运行 `ps`。`nice` 调整数字的范围是 -20 ~ 19。数字越小优先级越高。

练习

- 安装 `htop` , `saidar` , `pstree` 。
- `htop` 和 `saidar` 用于动态监控。
- `pstree` 显示进程树。
- 熟悉 `ps` 常用方式, 以及如何搜索进程。
- 熟悉如何终止进程。