



Linux 进程管理



李明

E-mail/qq: samlee@lampbrother.net



课程大纲

- 进程的概念
- 进程管理命令
- 计划任务



进程和程序的区别

- 1、程序是静态概念，本身作为一种软件资源长期保存；而进程是程序的执行过程，它是动态概念，有一定的生命期，是动态产生和消亡的。
- 2、程序和进程无一一对应关系。一个程序可以由多个进程共用；另一方面，一个进程在活动中有可顺序地执行若干个程序。



父进程与子进程

- 1、子进程是由一个进程所产生的进程，产生这个子进程的进程称为父进程。
- 2、在Linux系统中，使用系统调用fork创建进程。
fork复制的内容包括父进程的数据和堆栈段以及父进程的进程环境。
- 3、父进程终止子进程自然终止。



前台进程和后台进程

前台进程:

在Shell提示处打入命令后，创建一个子进程，运行命令，Shell等待命令退出，然后返回到对用户给出提示符。这条命令与Shell异步运行，即在前台运行，用户在它完成之前不能执行另一个命令。

后台进程:

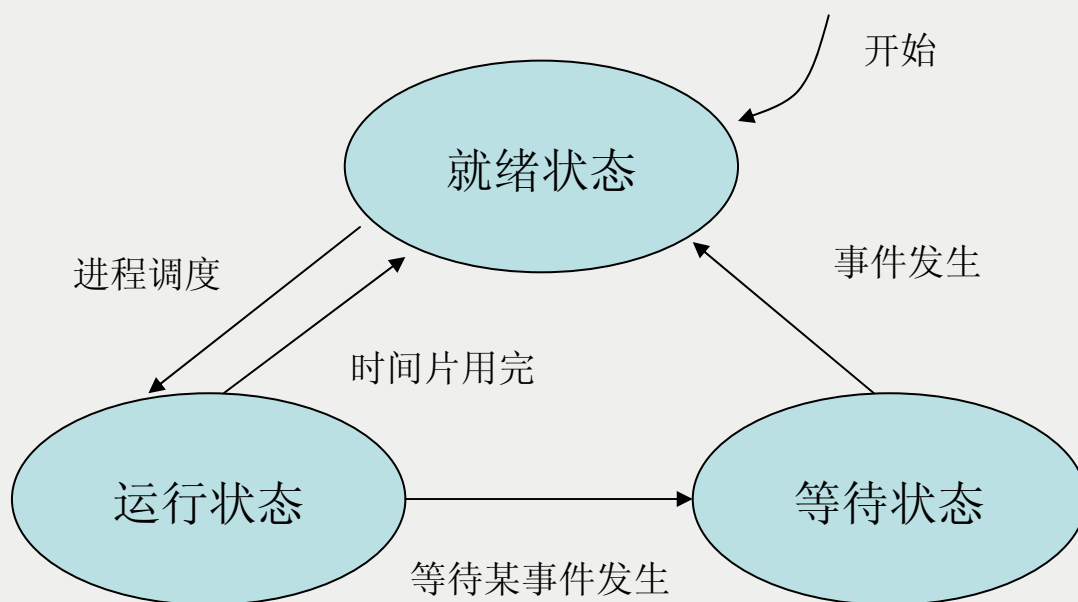
在Shell提示处打入命令，若后随一个&，Shell创建的子进程运行此命令，但不等待命令退出，而直接返回到对用户给出提示。这条命令与Shell同步运行，即在后台运行。**后台进程必须是非交互式的。**



进程状态

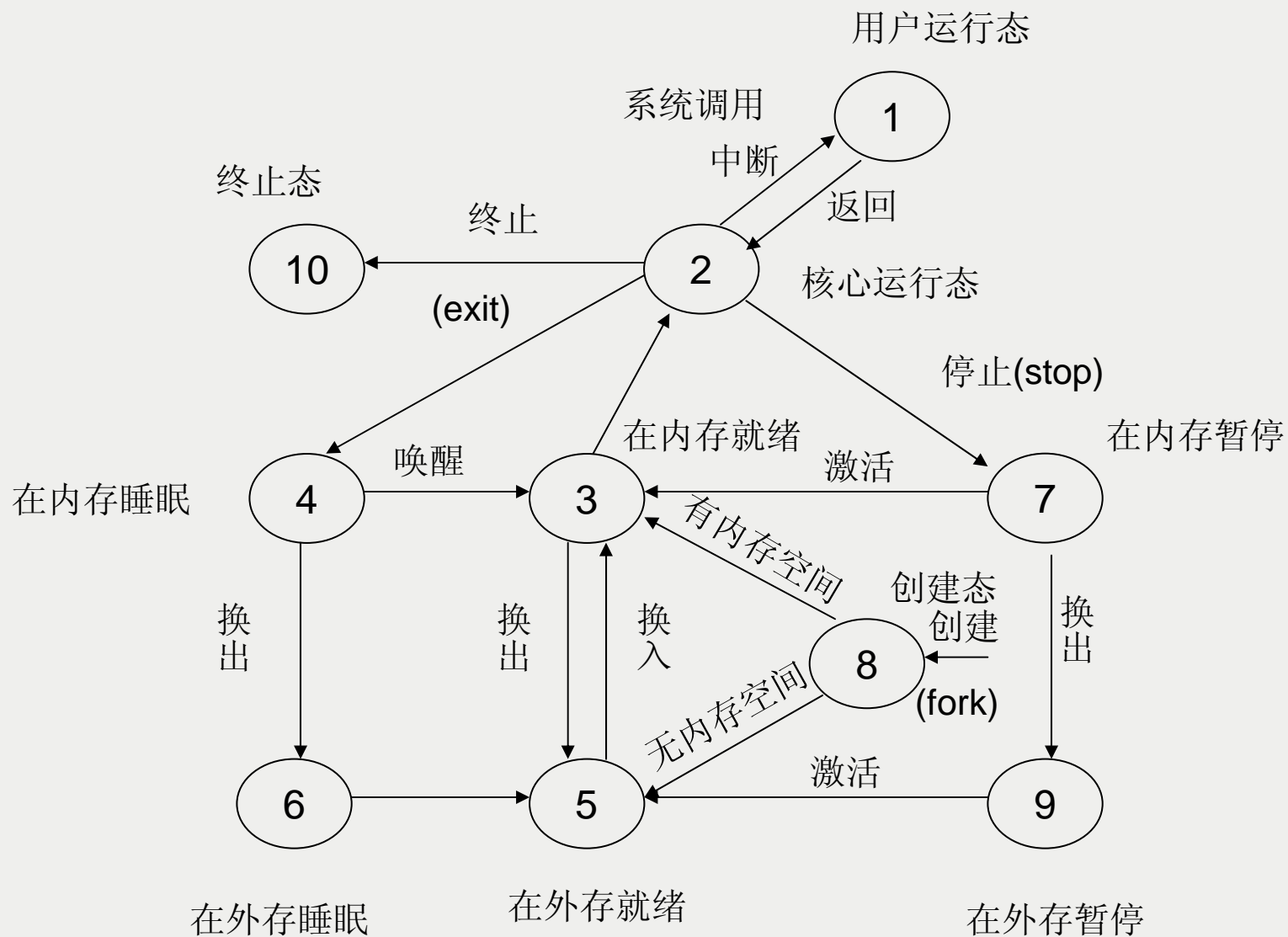
- 就绪：进程已经分配到资源，但因为其它进程正占用CPU，所以暂时不能运行而等待分配CPU的状态。
- 等待：因等待某种事件而暂时不能运行的状态。
- 运行：进程分配到CPU，正在处理器上运行。

进程基本状态及其变化





进程状态细化





进程状态细化

- 用户态运行：在CPU上执行用户代码
- 核心态运行：在CPU上执行核心代码
- 在内存就绪：具备运行条件，只等调度程序为它分配CPU
- 在内存睡眠：因等待某一事件的发生，而在内存中排队等待
- 在外存就绪：就绪进程被交换到外存上继续处于就绪状态
- 在外存睡眠：睡眠进程被交换到外存上继续等待
- 在内存暂停：因调用stop程序而进入跟踪暂停状态，等待其父进程发送命令。
- 在外存暂停：处于跟踪暂停态的进程被交换到外存上
- 创建态：新进程正在被创建、但尚未完毕的中间状态
- 终止态：进程终止自己



查看用户信息 w

- w 显示信息的含义
 - JCPU: 以终端代号来区分, 该终端所有相关的进程执行时, 所消耗的CPU时间会显示在这里
 - PCPU: CPU执行程序耗费的时间
 - WHAT: 用户正在执行的操作
- 查看个别用户信息: w 用户名



查看用户信息 w

w显示信息的含义

- **load average:** 分别显示系统在过去1、5、15分钟内的平均负载程度。
- **FROM:** 显示用户从何处登录系统，“:0”的显示代表该用户时从X Window下，打开文本模式窗口登录的
- **IDLE:** 用户闲置的时间。这是一个计时器，一旦用户执行任何操作，该计时器便会被重置



查看系统中的进程 ps

常用选项

- a: 显示所有用户的进程
- u: 显示用户名和启动时间
- x: 显示没有控制终端的进程
- e: 显示所有进程，包括没有控制终端的进程
- l: 长格式显示
- w: 宽行显示，可以使用多个w进行加宽显示



查看系统中的进程 ps

ps常用输出信息的含义

- PID: 进程号
- PPID: 父进程的进程号
- TTY: 进程启动的终端
- STAT: 进程当前状态

S休眠状态, D不可中断的休眠状态, R运行状态, Z僵死状态, T停止

- NI: 进程优先级



查看系统中的进程 ps

ps常用输出信息的含义

- TIME: 进程自从启动以来启用CPU的总时间
- COMMAND/CMD: 进程的命令名
- USER: 用户名
- %CPU: 占用CPU时间和总时间的百分比
- %MEM: 占用内存与系统内存总量的百分比



查看系统中的进程 ps

ps应用实例

ps 查看隶属于自己的进程

ps -u or -l 查看隶属于自己进程详细信息

ps -le or -aux 查看所有用户执行的进程的
详细信息

ps -aux --sort pid 可按进程执行的时间、
PID、UID等对进程进行排序



查看系统中的进程 ps

ps应用实例

```
# ps -aux | grep sam
```

```
# ps -uU sam
```

查看系统中指定用户执行的进程

```
# ps -le | grep init
```

查看指定进程信息



kill – 杀死进程

为什么要杀死进程

- 该进程占用了过多的CPU时间
- 该进程缩住了一个终端，使其他前台进程无法运行
- 运行时间过长，但没有预期效果
- 产生了过多到屏幕或磁盘文件的输出
- 无法正常退出



kill – 关闭进程

- ❑ 关闭进程: **kill** 进程号
- ❑ **kill -9** 进程号 (强行关闭)
- ❑ **kill -1** 进程号 (重启进程)
- ❑ 关闭图形程序: **xkill**
- ❑ 结束所有进程: **killall**
- ❑ 查找服务进程号: **pgrep** 服务名称
- ❑ 关闭进程: **pkill** 进程名称



nice和renice

□ nice

- 指定程序的运行优先级
- 格式: `nice -n command`
- 例如: `nice -5 myprogram`

□ renice

- 改变一个正在运行的进程的优先级
 - 格式: `renice n pid`
 - 例如: `renice -5 777`
- * 优先级取值范围为 $(-20, 19)$ *



nohup

使进程在用户退出登陆后仍旧继续执行，nohup命令将执行后的数据信息和错误信息默认储存到文件nohup.out中
格式：

nohup program &



进程的挂起和恢复

□ 进程的中止（挂起）和终止

- 挂起（Ctrl+Z）
- 终止（Ctrl+C）

□ 进程的恢复

- 恢复到前台继续运行（fg）
- 恢复到后台继续运行（bg）

□ 查看被挂起的进程（jobs）



top

作用：进程状态显示和进程控制，每5秒钟自动刷新一次（动态显示）

常用选项：

- d: 指定刷新的时间间隔
- c: 显示整个命令行而不仅仅显示命令名



top

常用命令：

- u: 查看指定用户的进程
- k: 终止执行中的进程
- h or ? : 获得帮助
- r: 重新设置进程优先级
- s: 改变刷新的时间间隔
- W: 将当前设置写入 ~/.toprc 文件中



计划任务

- 为什么要设置计划任务
- 计划任务的命令
 - at 安排作业在某一时刻执行一次
 - batch 安排作业在系统负载不重时执行一次
 - cron 安排周期性运行的作业



at命令的功能和格式

功能： 安排一个或多个命令在指定的时间运行一次

at的命令格式及参数

- `at [-f 文件名] 时间`
- `at -d` or `atrm` 删除队列中的任务
- `at -l` or `atq` 查看队列中的任务



at命令指定时间的方式

□ 绝对计时方法

- midnight noon teatime
- hh:mm [today]
- hh:mm tomorrow
- hh:mm 星期
- hh:mm MM/DD/YY

□ 相对计时方法

- now + n minutes
- now + n hours
- now + n days



at命令指定时间的方式

- ❑ 指定在今天下午 17:30 执行某命令（假设现在时间是下午14:30，2011年1月11日）。
- ❑ 命令格式如下：
 - at 5:30pm
 - at 17:30
 - at 17:30 today
 - at now + 3 hours
 - at now + 180 minutes
 - at 17:30 11.1.11
 - at 17:30 1/11/11



at命令使用范例

□ 交互方式

at 9:00

□ 使用命令文件方式

1. 生成文件at.script:

2. 使用at命令

at -f at.script 9:00 2/2/11

or

at < at.script 9:00 2/2/11



at配置文件

作用： 限制哪些用户可以使用at命令

- /etc/at.allow

- /etc/at.deny



at的执行结果和配置文件

如果/etc/at.allow文件存在，那么只有列在此文件中的用户才可以使用at命令；若/etc/at.allow文件不存在，则检查/etc/at.deny文件是否存在。若/etc/at.deny存在，则在此文件中列出的用户都不能使用at命令。如果两个文件都不存在，则只有超级用户可以使用at命令。如果两个文件都存在而且均为空，则所有用户都可以使用at命令。



batch命令

- 作用：安排一个或多个命令在系统负载较轻时运行一次（一般情况下负载较轻指平均负载降到0.8以下）。
- 使用方法同at



crontab命令格式

- ❑ 作用：用于生成cron进程所需要的crontab文件
- ❑ crontab的命令格式

crontab {-l|-r|-e}

- -l 显示当前的crontab
- -r 删除当前的crontab
- -e 使用编辑器编辑当前的crontab文件。



crontab文件格式

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

□ 其中

- | | | |
|-----------------|-----------|--------|
| ■ Minute | 一小时中的哪一分钟 | [0~59] |
| ■ hour | 一天中的哪个小时 | [0~23] |
| ■ day-of-month | 一月中的哪一天 | [1~31] |
| ■ month-of-year | 一年中的哪一月 | [1~12] |
| ■ day-of-week | 一周中的哪一天 | [0~6] |
| ■ commands | 执行的命令 | |



书写注意事项

- ❑ 选都不能为空，必须填入，不知道的值使用通配符*表示任何时间
- ❑ 每个时间字段都可以指定多个值，不连续的值用, 间隔，连续的值用-间隔。
- ❑ 命令应该给出绝对路径
- ❑ 用户必须具有运行所对应的命令或程序的权限



crontab应用示例

应用范例：

```
# crontab -e
```

```
55 17 * * 1-5 /usr/bin/wall < /etc/motd
```

```
0 18 * * 1-5 /sbin/init 5
```

```
0 0 1,15 * 1 /bin/cp -R /etc /backup/etc.0111
```

* crontab文件建立好以后可以到/var/spool/cron
目录确认



cron配置文件

作用： 限制哪些用户可以使用crontab命令

- /etc/cron.allow
- /etc/cron.deny



进程处理方式

- ❑ standalone 独立运行
- ❑ xinetd 进程托管
- ❑ atd、crond 计划任务



Thanks



技术交流 <http://www.lampbrother.net/linux.php>
视频下载 <http://www.lampbrother.net/video.html>