1 Linux进程

每个用户均可同时运行多个程序。为了区分每一个运行的程序，Linux给每个进程都做了标识，称为进程号，每个进程的进程号是唯一的。

Linux 给每个进程都打上了运行者的标志，用户可以控制自己的进程：给自己的进程分配不同的优先级，也可以随时终止自己的进程。

进程从执行它的用户处继承UID、GID，从而决定对文件系统的存取和访问。

Linux 不可能在一个 CPU 上同时处理多个任务（作业）请求，而是采用 “分时” 技术来处理这些任务请求。

使用PID区分不同的进程：

- 系统启动后的第一个进程是init，它的PID是1。init是由系统内核直接运行的进程（还有另一种情况：coredump 处理进程）。
- 除了init之外，每个进程都有父进程（PPID标识）
- 每个进程还有四个与用户和组相关的识别号
    - 实际用户识别号 （real user ID，RUID）
    - 实际组识别号 （real group ID，RGID）
    - 有效用户识别号 （ effective user ID，EUID）
    - 有效组识别号 （effective group ID，EGID）

1. RUID和RGID的作用：识别正在运行此进程的用户和组。
2. EUID和EGID的作用：确定一个进程对其访问的文件的权限。

进程类型：

- 交互进程：由一个Shell启动的进程；交互进程既可以在前台运行，也可以在后台运行。
- 批处理进程：不与特定的终端相关联，提交到等待队列种顺序执行的进程。
- 守护进程（Daemon）：在Linux在启动时初始化，需要时运行于后台的进程。

进程的启动方式：

- 手工方式：使用操作系统提供的用户接口
    - 前台
    - 后台 （&）
- 调度方式：按照预先指定的时间执行
    - at
    - batch
    - cron

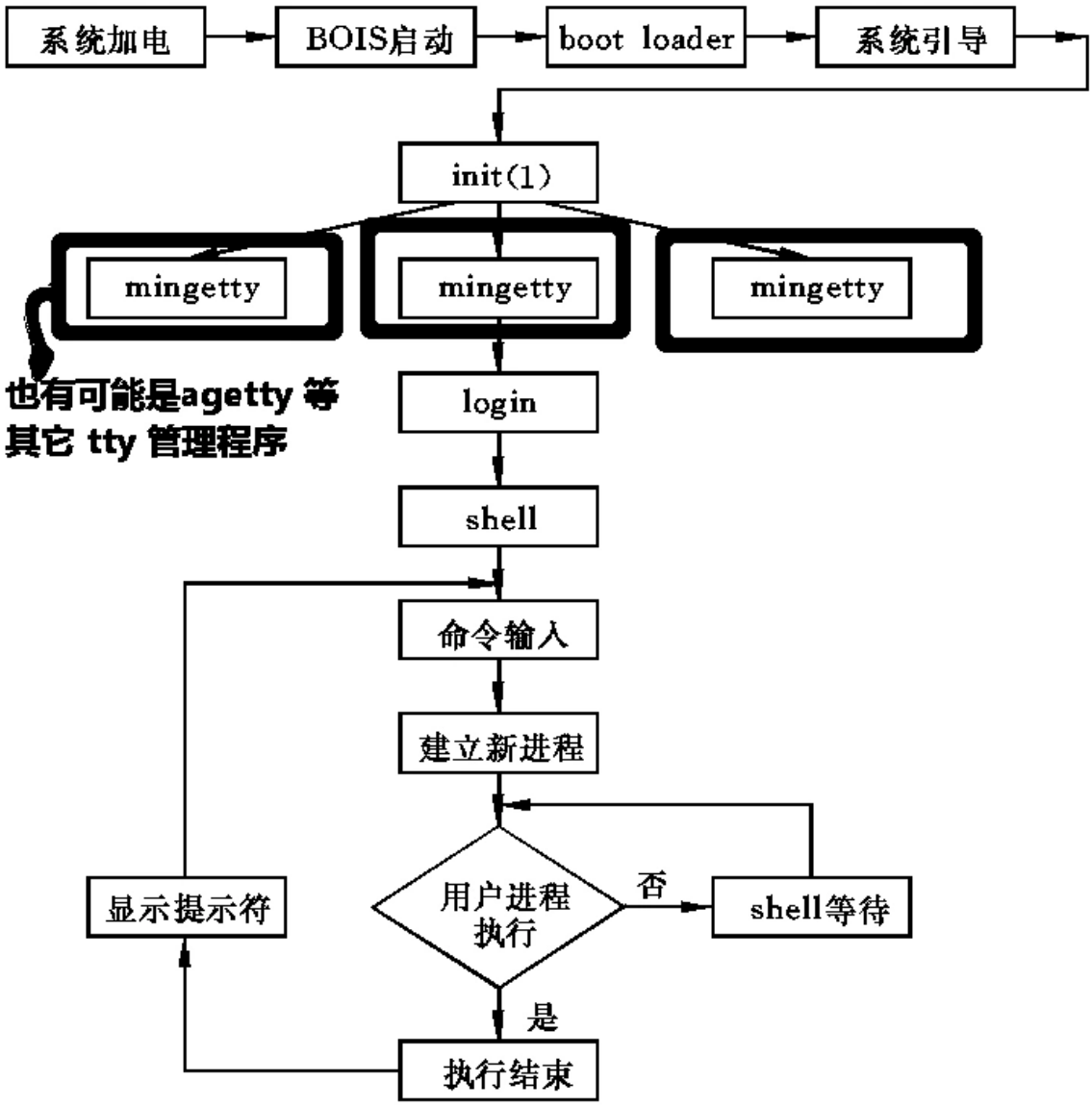
前台进程：指一个程序控制着标准输入/输出，在程序运行时，shell 被暂时挂起，直到该程序运行结束后，才退回到 shell。在这个过程中，用户不能再执行其它程序

后台进程：用户不必等待程序运行结束就可以执行其它程序。运行后台进程的方法是在命令行最后加上 “&”

进程和作业的区别：

- 进程：操作系统的概念，由操作系统负责管理
- 作业：shell程序的概念，由shell程序负责管理

- 一个操作系统可以启动多个shell程序，shell本身也是一个进程
- 一个作业里至少包含一个进程，也可以包含多个进程
- 作业分前台和后台运行之分



管理进程常用命令

使用`ps`命令查看进程状态信息：

- `ps -ef`
- `ps aux`

选项	说明
-e	显示所有进程，等价于 -A。
-f	完全（FULL）显示。增加显示用户名、PPID、进程起始时间。
f/-H	显示进程树，等价于 --forest。
a	显示终端上的所有进程，包括其他用户地进程。
x	显示没有控制终端地进程。
u	面向用户的显示格式。增加显示用户名，进程起始时间，CPU和内存占用百分比等信息。
-u <username>	仅显示指定用户的进程。
l/-l	长格式显示。增加显示进程的UID、PPID和优先权值。
w[w]/-w[w]	加宽显示。通常用于显示完整的命令行。
o/-o <format>	由用户自定义输出列。
--sort <order>	指定按哪/哪些列排序，order格式为：[+|-]key[,[+|-]key[,...]]

`ps`命令输出说明：

❑ ps 的输出依赖于用户所给的选项

UID	用户 ID	START	进程启动时间
USER	用户名	TIME	执行时间
PID	进程 ID	STAT	进程状态
PPID	父进程的 ID	NI	优先权值 / nice 值
TTY	启动进程的终端	CMD	命令名（COMMAND）
RSS	进程所用内存块数	%CPU	进程所用CPU时间百分比 (pcpu)
VSZ	进程所用虚拟内存块数	%MEM	进程所有MEM百分比 (pmem)

还有其他几个常用命令：

- nohup:忽略SIGHUP信号：`nohup 命令 [选项] [参数] [输出文件] &`
  - 用过SpringBoot打jar包部署到Linux下的同学对上面这个命令一定不会陌生啦~
- 杀死进程：`kill pid -9`
- `free`：查看内存使用状况
- `top`：查看实时刷新的系统进程信息
- 进程调度的优先权`nice`命令：
- 进程运行后调整优先级：`renice`命令。

nohup与&启动的程序，在终端还未关闭时，完全不像传统的守护进程，因为其不是会话首进程且持有终端，只是其忽略了SIGHUP信号

从nohup源码就可以看到，其实nohup只做了3件事情

1. dofile函数将输出重定向到nohup.out文件
2. signal函数设置SIGHUP信号处理函数为SIG_IGN宏（指向sigignore函数），以此忽略SIG_HUP信号
3. execvp函数用新的程序替换当前进程的代码段、数据段、堆段和栈段。

execvp 函数执行后，新程序（并没有fork进程）会继承一些调用进程属性，比如：进程id、会话id，控制终端等

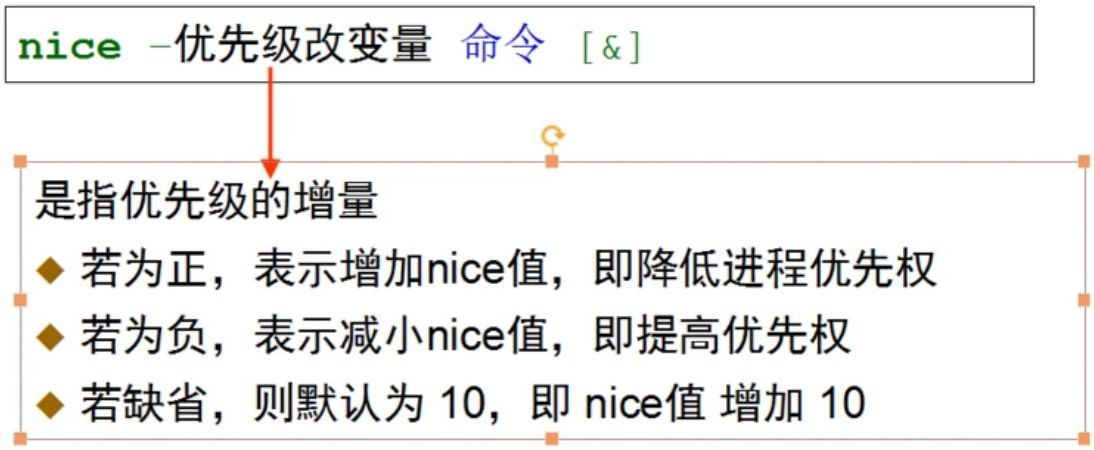
登录连接断开之后

```
    1  36308  36308  34401 ?           -1 SN       0   0:00 bash -c tail -f /var/log/messages | grep sys
36308  36311  36308  34401 ?           -1 SN       0   0:00  \_ tail -f /var/log/messages
36308  36312  36308  34401 ?           -1 SN       0   0:00  \_ grep sys
```

在终端关闭后，**nohup起到类似守护进程的效果，但是跟传统的守护进程还是有区别的**

1、nohup创建的进程工作目录是你执行命令时所在的目录
2、0 1 2 标准输入 标准输出 标准错误 指向nohup.out文件
3、nohup创建的进程组中，除首长进程的父进程id变为1之外，其余进程依然保留原来的会话id、进程组id、父进程id，都保持不变

- 在启动进程时就指定优先级： **nice**



作业控制是指控制当前正在运行的进程的行为。

- 暂时停止某个运行程序 使用`Ctrl+z`
- 列举作业号码和名称：`jobs`
- 在后台恢复运行：`bg [%作业号码]`
- 在前台恢复运行：`fg [%作业号码]`
- 发送信号：`kill -[信号] pid`

2 守护进程

始终在后台运行并响应合法请求的程序称为守护进程。守护进程不是由用户启动运行的，也不与终端关联。

- 一个实际运行中的系统一般会有多个守护进程在运行，且各个系统中运行的守护进程都不尽相同。
- 除非程序异常中止或者人为终止，否则它们将一直运行下去直至系统关闭。
- UNIX/Linux的守护进程在Windows系统中被称作“服务”。

守护进程的分类：

- 系统守护进程
  - 计划性任务 daemon：如 atd、crond

    - 系统日志 daemon：如 rsyslogd
    - 打印假脱机 daemon：如 cupsd、lpd
    - 网络参数设置 daemon：如 network
- 网络守护进程：
    - 各种网络协议侦听 daemon
    - 如：sshd、httpd、postfix、vsftpd
- 网络超级服务器（Supper Server）
    - 如：xinetd 或 inetd

超级服务器的引入`xinetd`：

- 对于系统所要提供的每一种网络服务，都必须运行一个监听某个端口连接发生的守护程序，这通常意味着系统资源的浪费。
- 为了避免系统资源浪费引入了“超级服务器”。超级服务器启动后同时监听它所管理的服务的所有端口
- 当有客户提出服务请求时
    - 超级服务器会判断这是对哪一个服务的请求，然后再开启与此服务相应的守护进程
    - 由超级服务器产生的某服务的进程处理客户的请求，当处理结束便终止此服务进程
    - 超级服务器本身继续监听其他服务请求

守护进程的启动方式：

- 独立启动
    - 独立运行的守护进程由init脚本负责管理，脚本存放在`/etc/rc.d/init.d/`目录下
    - 所有的系统服务都是独立运行的。如：crond、syslogd等
    - 一些常用的网络守护进程是独立运行的。如：httpd等
- 瞬态启动
    - 由网络超级服务器（xinetd）运行的守护进程，由xinetd管理的守护进程的配置文件存在`/etc/xinetd.d/`目录下
    - 默认的xinetd的主配置文件是`/etc/xinetd.conf`

- 一些不常用的网络守护进程是由xinetd启动的，如：telnet、 tftp等
- xinetd本身是独立运行的守护进程

管理守护进程常用命令

`chkconfig`命令的功能

- 添加指定的新服务
- 清除指定的服务
- 显示由chkconfig管理的服务
- 改变服务的运行级别
- 检查服务的启动状态

`chkconfig --list`会显示出对应的运行级别：

- 0： 关机
- 1： 单用户
- 2： 无网络的多用户
- 3： 命令行模式
- 4： 未用
- 5： GUI（图形桌面 模式）
- 6： 重启

用`ntsysv`管理守护进程

使用`service`管理守护进程

- `service --status-all`
- `service server-name status`
- `service server-name start|stop|restart`

> telnet服务端口可以改变吗？如果可以改变，连接telnet服务应注意什么问题？

答：telnet服务端口可以改变。连接telnet服务的时候，应该注意端口号修改为正在提供telnet服务的端口号。

> 修改telnet配置文件，需要xinetd服务重启吗？为什么？

答：需要重新启动xinetd服务，因为xinetd作为超级服务器，它负责管理telnet服务的启动，也要同时查看telnet的服务配置文件。当telnet服务配置文件修改的时候，xinetd服务需要知道配置文件的变化，重新启动会重新读取配置文件的内容，使之生效。

> Telnet为什么可以看到脚本程序的执行结果？

答：telnet是一个远程的字符界面的网络工具，它实现了远程字符界面的标准输入和输出功能；脚本执行的结果是输出到标准输出设备，也就是字符界面的屏幕，telnet将标准输出通过网络传递到telnet客户端的屏幕上显示，因此它可以看到脚本程序执行的结果。

> 通过ntsysv命令和chkconfig命令打开守护进程启动运行的结果是否完全一样？有什么不同？

答：ntntsysv命令关闭守护进程，只是改变了当前用户运行级别的守护进程开关，具体来说，改变了3号运行级别命令行模式的守护进程开关。chkconfig命令默认改变3、4、5级别的全部开关，chkconfig还可以指定某个运行级别的守护进程开关。

3 安排自动化任务

调度任务的守护进程：

- atd
- crond

安排调度任务的几个命令：

- 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

3.1 atd

atd守护进程负责监控一次性任务的执行，atd守护进程的执行参数`/etc/sysconfig/atd`

控制普通用户的使用：

- 若`/etc/at.allow`存在，仅列在其中的用户允许使用
- 若`/etc/at.allow`不存在，检查`/etc/at.deny`，没有列于其中的所有用户允许使用
- 若两个文件均不存在，仅允许root用户使用
- 空的`/etc/at.deny`文件，表示允许所有用户使用（默认值）

如何使用：

- 安装命令`yum install at`
- atd的启动`service atd start`
- atd服务的查看`chkconfig --list | grep atd`或者`ps -aef | grep atd`
- `at`命令格式及参数`at [-q 队列] [-f 文件名] 时间`

3.2 cron

- crond守护进程负责监控周期性任务的执行
- crond守护进程的执行参数配置文件`/etc/sysconfig/crond`

控制普通用户的使用：

- 若`/etc/cron.allow`存在，仅列在其中的用户允许使用
- 若`/etc/cron.allow`不存在，检查`/etc/cron.deny`，没有列于其中的所有用户允许使用
- 若两个文件均不存在，仅允许root用户使用
- 空的`/etc/cron.deny`文件，表示允许所有用户使用（默认值）

crond启动以后，每分钟唤醒一次，检测如下文件的变化并将其加载到内存

- `/etc/crontab`：是crontab格式（man 5 crontab）的文件
- `/etc/cron.d/*`：是crontab格式（man 5 crontab）的文件
- `/var/spool/cron/*`：是crontab格式（man 5 crontab）的文件
- `/etc/anacrontab`：是anacrontab格式（man 5 anacrontab）的文件

crontab文件的格式

CentOS
The Community ENTerprise Operating System

- 注释行以 # 开头
- 详情参见 man 5 crontab
- 每一行由5个时间字段及命令组成

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

- 五个时间字段
  - 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]**