Linux进程管理

# 进程与作业

## Linux中进程的概念

Linxu是一个多用户多任务的系统：多个用户可以同一时间使用计算机系统，多任务是指linux可以同时执行多个任务。每执行一个任务，系统就会启动一个进程（是一个程序在其自身虚拟地址空间中的一次执行活动）。进程可以同时被多个程序调用。

程序只是一个静态的指令集合；进程是一个程序的动态执行过程。它有生命期，是动态的产生和消亡。进程是资源申请、调度和独立运行的单位。程序和进程没有一一对应关系。

Linux系统启动的时候，系统根据进程号PID来区分不同的进程。系统启动的第一个进程是init，它的PID是1。Init是唯一一个有系统内核直接运行的进程。新的进程可以通过fork（）来产生，就是从一个已经存在的进程分出另一个进程 。新产生的进程曾为子进程，之前的进程称为父进程。父进程终止子进程自然终止。系统启动后，init进程会创建login进程等待用户登录。用户登录后，login进程会产生shell进程。此后，用户运行的进程都是shell衍生出来，都是shell的子进程。

每个进程除了有PID，还会有四个识别号。实际用户识别号（Real User ID）,实际用户组识别号（Real Group ID）:识别正在运行此进程的用户和组，也就是用户的UID,GID。有效组识别号和有效用户识别号：确定一个进程对其访问的文件的权限和优先级。如果程序设置了UID和GID位，有效用户/组识别号是程序文件所有者/组。

## 进程的类型

交互进程：shell启动的进程。

批处理进程：不语特定的终端关联

守护进程：linux系统启动时初始化，在后台运行。

## 进程的启动方式

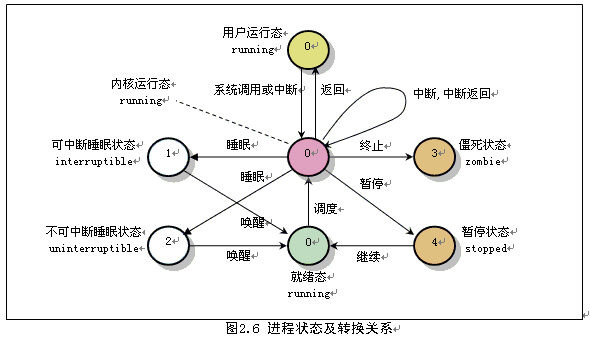
手工启动：前台启动，后台启动(nohup command&)

调度启动：某一时间（at,batch(空闲时)）,周期性（cron）

## 进程的优先级

Linux系统使用的是时间片轮转法来调度进程的执行的。优先级高者优先获得执行权。Red Hat Linux9中优先级为-20~19。默认的优先级是0。

## Linux进程的状态



◆运行状态（TASK\_RUNNING）

当进程正在被CPU执行，或已经准备就绪随时可由调度程序执行，则称该进程为处于运行状态（running）。进程可以在内核态运行，也可以在用户态运行。当系统资源已经可用时，进程就被唤醒而进入准备运行状态，该状态称为就绪态。这些状态（图中中间一列）在内核中表示方法相同，都被成为处于TASK\_RUNNING状态。

◆可中断睡眠状态（TASK\_INTERRUPTIBLE）

当进程处于可中断等待状态时，系统不会调度该进程执行。当系统产生一个中断或者释放了进程正在等待的资源，或者进程收到一个信号，都可以唤醒进程转换到就绪状态（运行状态）。

◆不可中断睡眠状态（TASK\_UNINTERRUPTIBLE）

与可中断睡眠状态类似。但处于该状态的进程只有被使用wake\_up()函数明确唤醒时才能转换到可运行的就绪状态。

◆暂停状态（TASK\_STOPPED）

当进程收到信号SIGSTOP、SIGTSTP、SIGTTIN或SIGTTOU时就会进入暂停状态。可向其发送SIGCONT信号让进程转换到可运行状态。在Linux 0.11中，还未实现对该状态的转换处理。处于该状态的进程将被作为进程终止来处理。

比如调试断点，没执行到一个断点，就转入暂停态。

◆僵死状态（TASK\_ZOMBIE）

当进程已停止运行，但其父进程还没有询问其状态时，则称该进程处于僵死状态。

当一个进程的运行时间片用完，系统就会使用调度程序强制切换到其它的进程去执行。另外，如果进程在内核态执行时需要等待系统的某个资源，此时该进程就会调用sleep\_on()或sleep\_on\_interruptible()自愿地放弃CPU的使用权，而让调度程序去执行其它进程。进程则进入睡眠状态（TASK\_UNINTERRUPTIBLE或TASK\_INTERRUPTIBLE）。

只有当进程从“内核运行态”转移到“睡眠状态”时，内核才会进行进程切换操作。在内核态下运行的进程不能被其它进程抢占，而且一个进程不能改变另一个进程的状态。为了避免进程切换时造成内核数据错误，内核在执行临界区代码时会禁止一切中断。

## 作业

正在执行的一个或者多个相关的进程被称为作业。

# 进程管理命令

## 查看系统进程

### ps命令

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

常用命令：

查看系统中指定用户执行的进程：ps -aux | grep sam

查看指定进程信息：ps –ef | grep java

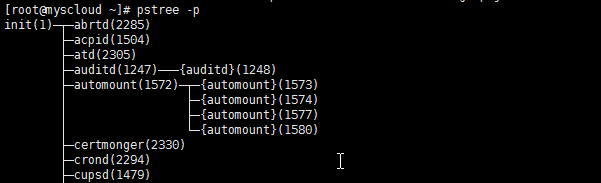
常见显示信息:

|  |
| --- |
| USER：进程拥有者  PID：进程号  PPID：父进程的进程号  %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  VSZ：占用的虚拟内存大小  RSS：占用的内存大小  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 ,W没有足够内存可分配  NI：进程优先级  TIME：进程自从启动以来启用CPU的总时间  COMMAND/CMD：进程的命令名 |

### pstree命令

###显示进程树

pstree -p



### top命令

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）

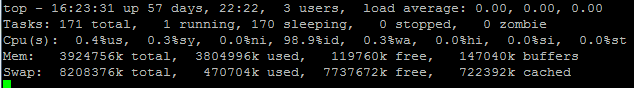
常用选项：

* 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 –d 20 //每20s刷新

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



top命令的前五行：正常运行时间行 、进程统计数行、CPU统计行、内存统计行、交换区和缓存统计行

|  |
| --- |
| USER：进程拥有者  PID：进程号  PR：进程优先级  NI：进程优先级数值  VIRT：进程占用的虚拟内存  RES：进程占用实际物理内存  SHR：进程使用的共享内存  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 ,W没有足够内存可分配  %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  TIME：进程自从启动以来启用CPU的总时间  COMMAND/CMD：进程的命令名 |

### pgrep命令

pgrep命令以名称为依据从运行进程队列中查找进程，并显示查找到的进程id。

使用方法：pgrep -l 进程名称



### lsof命令

作用：列出进程调用或者打开文件的信息

lsof [选项]

* -c 字符串：只列出以字符串开头的进程打开的文件
* -u 用户名：只列出某个进程打开的文件
* -p pid：列出某个pid进程打开的文件
* lsof 文件名 #显示开启文件的进程名

输出字段信息

|  |
| --- |
| COMMAND：进程的名称  PID：进程的标识符  USER：进程的所有者  FD：文件描述符，应用程序通过文件描述符识别该文件  TYPE：文件类型，如DIR，REG等  DEVICE：磁盘的名称  NODE：索引节点  NAME：打开文件的全路径 |

### pidof命令

pidof命令用于查找指定名称的进程的进程号id号

语法： pidof(选项)(参数)

选项：

* -s：仅返回一个进程号；
* -c：仅显示具有相同“root”目录的进程；
* -x：显示由脚本开启的进程；
* -o：指定不显示的进程ID。

参数 进程名称：指定要查找的进程名称。

## 查看系统作业

###数字代表工作号 +代表最后一个放入后台的工作，-代表表示倒数第二个###放入后台的任务

jobs -l(显示工作pid)

#将后台暂停的任务恢复到前台运行

fg %工作号

#将后台暂停的任务恢复到后台运行（一些交互的任务不能实现如top)

bg %工作号

注意：&后台运行 ctrl+z代表后台暂停

## 设置进程优先级

### nice命令

nice:指定程序的运行优先级

格式：nice -n 数字command

例如：nice -n -5 myprogram

### renice命令

作用：改变一个正在运行的进程的优先级

格式：renice -n 数字 -p pid

例如：renice -n -5 -p 777 // 优先级取值范围为（-20，19）

## 终止进程

杀死进程的目的：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。

常用信号：

* SIGHUP(1)：从终端发出结束信号。
* SIGINT(2)：从键盘发出中断信号（Ctrl+c）
* SIGQUT(3)：从键盘发出退出信号（Ctrl+\）
* SIGFPE(8)：浮点异常
* SIGKILL(9)：结束接受信号的进程（强行杀死进程）
* SIGTERM(15)：kill命令默认的终止信号
* SIGCHLD(17)：子进程终止或结束的信号
* SIGSTOP(19)：从键盘来的执行信号（Ctrl+d）

### kill命令

ps –aux | grep java

kill -9 pid

### killall命令（作用同pkill）

killall -9 进程名

# 任务调度

## at调度

功能：安排一个或多个命令在指定的时间运行一次

使用：at 时间，回车就可以进入编辑了，完成后按ctrl+d提交。



at的命令格式及参数

|  |
| --- |
| at服务管理：service atd start/stop/restart/status  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 |

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：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）。

## cron调度

作用：用于生成cron进程所需要的crontab文件

### crontab的命令

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

|  |
| --- |
| 格式：minute hour day-of-month month-of-year day-of-week commands  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  例如：\*/1 \* \* \* \* /home/smb/shell/checkMongoStatus.sh |

用户配置文件：每个用户都有自己的cron配置文件,通过crontab -e 就可以编辑,一般情况下我们编辑好用户的cron配置文件保存退出后,系统会自动就存放于/var/spool/cron/目录中,文件以用户名命名。

注意事项：

|  |
| --- |
| 所有字段都不能为空，必须填入，不知道的值使用统配符\*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。  使用\*/数字表示增量。\*/1:每分钟，每小时。。。  命令应该给出绝对路径，为了防止相同命令名的攻击型文件。  用户必须具有运行所对应的命令或程序的权限  在写脚本的时候只要在加上：. /etc/profile |

### cron**的启动与关闭**

由于Cron是Linux的内置服务，可以用以下的方法启动.关闭这个服务:

service crond start/stop/restart/status

### 系统定时任务

第一种：系统自带，把脚本拷贝到目录 /etc/cron.(daily,hourly,monthly,

weekly/ )  中。系统会每天或者每小时，每月，每星期执行。

第二种：在/etc/crontab中配置。

### anacron配置

anacron是用来保证在系统关机的时候错过的定时任务，可以在系统开机之后

再执行。目录 /etc/cron.(daily,hourly,monthly, weekly/)中的脚本

anacron检测周期：anacron会使用一天，7天，1个月的检测周期。在系统

的/var/spool/anacron/目录中存在cron.{daily,weekly,monthly}文件，用于

记录上次cron执行的时间。和当前的时间作比较，如果两个时间的差值大于

anacron指定的时间差值，说明有任务被揉执行。

anacron配置文件：vim /etc/anacrontab

|  |
| --- |
| SHELL=/bin/sh  PATH=/sbin:/bin:/usr/sbin:/usr/bin  MAILTO=root  # 最大随机延迟  RANDOM\_DELAY=45  # anacron执行执行范围3点到22点  START\_HOURS\_RANGE=3-22  #period in days delay in minutes job-identifier command  1 5 cron.daily nice run-parts /etc/cron.daily  7 25 cron.weekly nice run-parts /etc/cron.weekly  @monthly 45 cron.monthly nice run-parts /etc/cron.monthly  注意：/usr/bin/run-parts是一个shell命令，用于执行指定目录下的所有文件。 |