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。

------------------------------------

进程和线程的区别：

进程： 就是正在执行的程序或命令，每一个进程都是一个运行的实体，都有自己的地址空间，并占用一定的系统资源。

线程： 轻量级的进程；进程有独立的地址空间，线程没有；线程不能独立存在，它由进程创建；相对讲，线程耗费的cpu和内存要小于进程。

-------------------------------------

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

linux 下查端口 netstat -nap| grep "port"

再使用  lsof -i :port    查出使用该端口的进程

再使用kill 杀掉进程

------------------------------------

查看用户信息 w：

1,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

2,查看个别用户信息：w 用户名

3,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:

    ps - report a snapshot of the current processes

    ps命令查找与进程相关的PID号：  **最常用的方法是 ps  aux**  
    ps a 显示现行终端机下的所有程序，包括其他用户的程序。  
    ps -A 显示所有程序。  
    ps c 列出程序时，显示每个程序真正的指令名称，而不包含路径，参数或常驻服务的标示。  
    ps -e 此参数的效果和指定"A"参数相同。  
    ps e 列出程序时，显示每个程序所使用的环境变量。  
    ps f 用ASCII字符显示树状结构，表达程序间的相互关系。  
    ps -H 显示树状结构，表示程序间的相互关系。  
    ps -N 显示所有的程序，除了执行ps指令终端机下的程序之外。  
    ps s 采用程序信号的格式显示程序状况。  
    ps S 列出程序时，包括已中断的子程序资料。  
    ps -t<终端机编号> 指定终端机编号，并列出属于该终端机的程序的状况。  
    ps u 以用户为主的格式来显示程序状况。  
    ps x 显示所有程序，不以终端机来区分。

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

eg:ps -ef|grep \*\*\* 查询进程，如果有多个，可再加通道|grep  \*\*\*

删除进程时，如果有多个进程，不能直接区分启动的目录

则可以使用 cd /proc/PID，再ll一下，就能看到启动的目录了

3，查看系统中所有进程

# ps aux     #查看系统中所有进程，使用BSD操作系统格式

# ps -le        #查看系统中所有进程，使用Linux标准命令格式

ps -aux常用输出信息的含义

    USER：用户名

    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：该进程占用虚拟内存的大小，单位是KB

    RSS：该进程占用实际物理内存的大小，单位是KB

    TTY：进程启动的终端，其中tty1-tty7代表本地控制台终端，tty1-tty6是本地的字符界面终端，tty7是图形终端。pts/0-255代表虚拟终端

    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，s:包含子进程，+：位于后台

    START：该进程的启动时间

    NI：进程优先级

    TIME：进程自从启动以来占用CPU的总时间

    COMMAND/CMD：进程的命令名

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

    # ps -aux | grep sam    查看用户sam执行的进程   
    # ps -uU sam               查看用户sam执行的进程    
    # ps -le | grep init        查看指定进程信息

------------------------------------------------

**进程管理的作用**

1，判断服务器健康状态

2，查看系统中所有进程

3，杀死进程

------------------------------------------------

kill – 杀死进程

为什么要杀死进程

    1,该进程占用了过多的CPU时间  
    2,该进程占用了过多的CPU时间无法运行  
    3,运行时间过长，但没有预期效果  
    4,产生了过多到屏幕或磁盘文件的输出  
    5,无法正常退出

kill

kill [ -s signal | -p ] [ -a ] pid ...   
kill -l [ signal ]   
kill 送出一个特定的信号 (signal) 给行程 id 为 pid 的行程根据该信号而做特定的动作, 若没有指定, 预设是送出终止 (TERM) 的信号   
-s (signal):其中可用的讯号有 HUP (1), KILL (9), TERM (15), 分别代表着重跑, 砍掉, 结束; 详细的信号可以用 kill -l   
-p:印出 pid , 并不送出信号   
-l (signal):列出所有可用的进程信号

常用的进程信号：



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

    killall -i    交互式，询问是否要杀死某个进程

    killall -0   进程名   检测进程是否存在，

    killall -I    忽略进程名的大小写

查找服务进程号：pgrep 服务名称  
关闭进程：pkill 进程名称

    pkill -t  终端号：    按照终端号踢出用户

怎么使用pkill呢？

1，w命令 查询本机已登录的用户，TTY（终端号）

2，pkill -9 -t    pst/1    强制杀死从pst/1虚拟终端登录的进程

------------------------------------------

**把进程放入后台**

1，在命令后加 &

2，在执行命令的过程中，按下ctrl+z快捷键（可以以top命令示范）

**查看后台的工作**

jobs    [-l]

-l    显示工作的PID

"+"号代表最近一个放入后台的工作，也是工作恢复时，默认优先恢复的工作。"-"号代表倒数第二个放入后台的工作。

**将后台暂停的工作恢复到前台执行**

# fg  %工作号

参数：%工作号：    %号可以省略，但是注意工作号和PID的区别

补：如果后台中有多个命令，可以用 fg %jobnumber将选中的命令调出，%jobnumber是通过jobs命令查到的后台正在执行的命令的序号(不是pid)

**把后台暂停的工作恢复到后台继续执行**

# bg    %工作号

注：后台恢复执行的命令，是不能和前台有交互的，否则不能恢复到后台执行

补：如果后台中有多个命令，同样可以用bg %jobnumber将选中的命令调出

比如说，top命令和vi命令，是不能在后台恢复执行的

------------------------------------------

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     **查看系统健康状态**

作用：进程状态显示和进程控制，每3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    默认是3秒  
    c：    显示整个命令行而不仅仅显示命令名

    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

    P：以CPU使用率排名，默认项

    M：以内存的使用率排名

    N：以PID排序

    q：退出top  
--------------------------------------