**进程基础**

1. 进程的概念

进程是一个程序的一次动态执行过程，是系统进行资源分配和调度的基本单元。

程序和进程的区别：

1. 程序是一段静态的代码，是存放在磁盘上的有序指令的集合，没有任何执行的概念。(程序指的是可执行文件，而并非源码)
2. 进程是一个动态的概念，它是程序的执行过程，包括动态创建、调度和消亡的整个过程，它是程序执行和资源管理的最小单位。

Linux内核通过唯一的进程标示符PID来标示每一个进程。新创建的进程PID通常是前一个进程的PID加1，不过PID的值有上限(最大值 = PID\_MAX\_DEFAULT – 1，通常为32767)，可以查看/proc/sys/kernel/pid\_max来确定该系统的进程上限数。

在Linux中获取当前进程的进程号PID和父进程号PPID的系统调用函数分别为getpid()和getppid()。

1. 进程的结构

进程不但包括程序的指令和数据，而且包括程序计数器(PC)和处理器的所有寄存器及存储临时数据的进程堆栈，因此正在执行的进程包括处理器当前的一切活动。

简单来说，进程由正文段(text)、用户数据段(data)和系统数据段(PC、register、stack)组成。

1. 进程的分类
2. 交互进程：该类进程由shell控制和运行，在前台和后台都可以运行；
3. 批处理进程：该类进程不属于某个终端，它被提交到一个队列中以便顺序执行；
4. 守护进程：该类进程在后台运行。它一般在linux启动时开始执行，系统关闭时才结束。

让一个进程运行于后台的方法：

1. ./a.out &；
2. 首先让一个进程处于暂停状态kill -19 pid/ctrl + z；然后，恢复其运行kill -18 pid。
3. 进程的状态(R S D T Z)
4. R------正在运行或准备运行(在运行队列中等待)；
5. S------睡眠状态(等待可中断状态)：等待一个事件的发生或某种系统资源。处在这种状态下的进程可以被信号中断；
6. D------等待不可中断状态：将信号传递至该类进程，不能改变其状态；
7. T------停止态、暂停态、挂起态(kill -19 pid或ctrl+z)：进程被中止；
8. Z------僵尸态：进程已终止，但还在进程向量数组中占有一个task\_struct结构，等待父进程调用wait()来收尸。
9. 进程的内存结构

Linux操作系统采用虚拟内存管理技术，使得每个进程都有各自互不干涉的进程地址空间。该地址空间是达纤维4GB的线性虚拟空间，用户所看到和接触到的都是该虚拟地址，无法看到实际的物理内存地址。利用这种虚拟地址不仅能够起到保护操作系统的效果(用户不能直接访问物理内存)，而且更重要的是，用户程序可以使用比实际物理内存更大的空间。

4GB的进程地址空间会被分成两个部分：用户空间和内核空间。用户地址空间是从0到3GB，内核地址空间占据3GB到4GB。用户进程在通常情况下只能访问用户空间的虚拟地址，不能访问内核空间的虚拟地址。只有用户进程使用系统调用时可以访问到内核空间。

六、相关的命令ps pstree top renice nice kill

ps：

1. ps -ef显示系统当前所有的进程，可获取pid和ppid信息；
2. ps –efL显示系统中的线程信息(LWP)；
3. ps aux可获取进程所处的状态信息；(前两个命令常与grep命令结合使用)
4. ps axj可获取系统中的守护进程的信息；(PID、PPID、PGID、SID、TTY等)
5. pstree以树形结构显示系统的所有进程。

top：

top命令可动态显示系统的进程信息，可以获取到NI---[-20,19]和PR---[0,39]值，即获取进程的优先级信息。NI和PR的关系：PR=20+NI。NI值越大，进程越越谦让，优先级就越低。

nice：

按用户指定的优先级运行进程。例如：nice -6 ./a.out

renice：

改变正在运行进程的优先级。例如：renice prinumber pid

kill：

1. kill –l------列出系统所支持的所有信号的信息；
2. kill –num pid------用于向某个进程发送信息。



