**Linux基础入门之实验十五进程的概念**

实验十三的学习内容包括进程与程序、进程的衍生、工作管理。

1. 进程与程序

程序就是执行一系列有逻辑、有顺序结构的指令，帮我们达成某个结果。

进程是程序在一个数据集合上的一次执行过程。进程具有动态性、并发性、独立性、异步性、结构性等特性。从进程的功能和服务对象角度进程可以分为用户进程和系统进程，从应用程序的服务类型角度进程可以分为交互进程、批处理进程和守护进程。

线程是操作系统能够进行运算调度的最小单位。它被包含在进程之中，是进程中的实际运作单位。一条线程指的是进程中一个单一顺序的控制流，一个进程中可以并发多个线程，每条线程并行执行不同的任务。

1. 进程的衍生

子进程就是父进程通过系统调用fork()而产生的复制品，fork()就是把父进程的 PCB 等进程的数据结构信息直接复制过来，只是修改了 PID，所以一模一样，只有在执行exec()之后才会不同，而早先的fork()比较消耗资源后来进化成vfork(),效率高了不少。

fork() 是一个系统调用（system call），它的主要作用就是为当前的进程创建一个新的进程，这个新的进程就是它的子进程，这个子进程除了父进程的返回值和 PID 以外其他的都一模一样，如进程的执行代码段，内存信息，文件描述，寄存器状态等。exec()也是系统调用，作用是切换子进程中的执行程序也就是替换其从父进程复制过来的代码段与数据段。

1. 进程组与sessions

每一个进程都会是一个进程组的成员，而且这个进程组是唯一存在的，他们是依靠 PGID（process group ID）来区别的，而每当一个进程被创建的时候，它便会成为其父进程所在组中的一员。

进程一般通过使用 getpgrp() 系统调用来寻找其所在组的 PGID，领导进程可以先终结，此时进程组依然存在，并持有相同的PGID，直到进程组中最后一个进程终结。

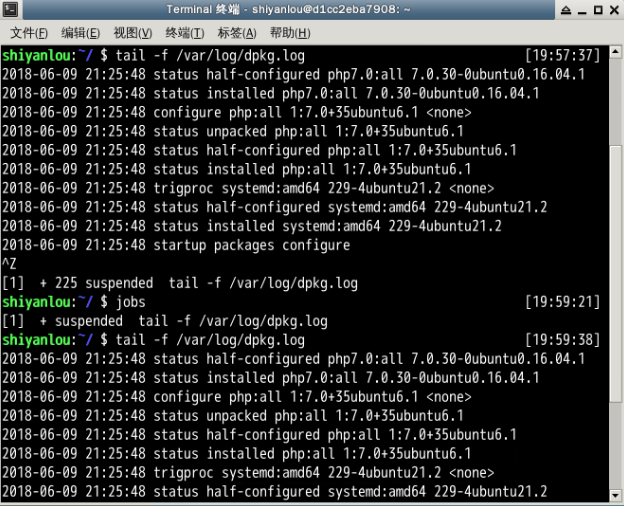
Session 主要是针对一个 tty 建立，Session 中的每个进程都称为一个工作(job)。每个会话可以连接一个终端(control terminal)。当控制终端有输入输出时，都传递给该会话的前台进程组。Session 意义在于将多个 jobs 囊括在一个终端，并取其中的一个 job 作为前台，来直接接收该终端的输入输出以及终端信号。 其他 jobs 在后台运行。

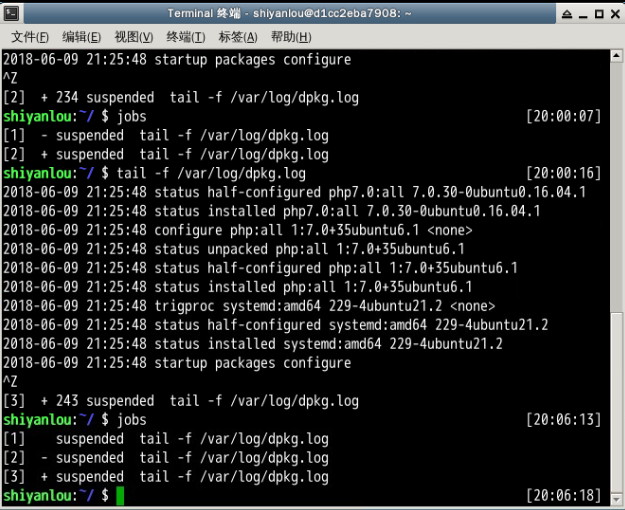
1. 工作管理

命令在后台运行：

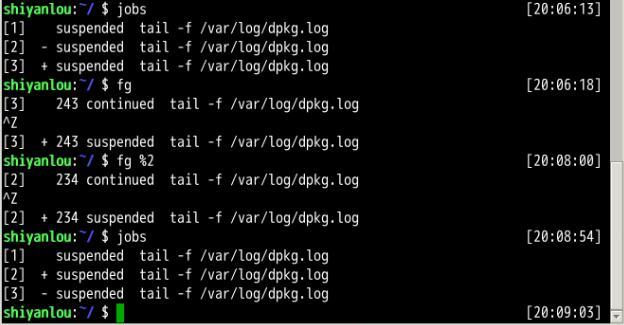


命令并查看停止与丢到后台的工作：





将后台工作拿到前台：



后台工作运作：

