

操 作 系 统

实 验 报 告

实验3 进程运行轨迹的跟踪与统计

班级： 1303104

组员： 1130310427/刘志铭

组员： 1133710314/郑榕鹏

教师： 曲 明 成

实验3 进程运行轨迹的跟踪与统计

1. 回答问题
2. 结合自己的体会，谈谈从程序设计者的角度看，单进程编程和多进程编程最大的区别是什么？

答：多进程能更加充分的利用CPU和IO设备，但是进程间的切换不仅给程序编写，也给操作系统的调用带来挑战。对于规模较小的程序，比较适合单进程编程，编程逻辑清晰，不易出错，而且执行效率也高。而对于比较复杂、规模较大的程序任务，则多进程的优势非常明显，多进程的独立性强，有各自独立的代码段和数据段，相互影响较小，适合于独立工作。

1. 你是如何修改时间片的？仅针对样本程序建立的进程，在修改时间片前后，log文件的统计结果（不包括Graphic）都是什么样？结合你的修改分析一下为什么会这样变化，或者为什么没变化？

答: 在没有调用过nice系统调用的情况下，时间片的初值就是进程0的priority。

即在linux-0.11/include/linux/sched.h中，宏INIT\_TASK中定义的：

#define INIT\_TASK { 0,15,15, //分别对应state;counter;和priority;

在linux 0.11中，所有的进程都是进程1的子进程，而进程1是进程0的子进程。修改了进程的初始时间片的大小，就改变了随手的所有进程的时间片的大小。

我将宏的定义做了如下的修改：

#define INIT\_TASK { 0,15,100, //分别对应state;counter;和priority;

在修改时间片前，由于样本程序中创建的子进程一次连续占用cpu的时间比时间片要大，因此样本程序中子进程还未执行完，就被调度为就绪状态了。表现为占用cpu的子进程在R状态后15个滴答后就转为J状态了。

而把时间片修改到100个滴答后，进程13的只用了个位数的轮转就执行完毕了，使得进程执行有了很大的改变。而对于样本程序中不占用cpu只占用I/O的进程则没变化，因为他们一旦执行，就进入睡眠等待状态，等待I/O处理。