# 进程运行轨迹的跟踪与统计

## 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？

单进程编程时程序有着严格的执行顺序以及逻辑，而多进程编写的程序是同时进行的。

单进程的程序在无其他外界因素干扰下数据是同步的，二多进程编写的程序虽然会共享一些资源，但由于执行顺序的未知，需要考虑进程间的关系，数据是异步的。要考虑进程间的通信，互斥等各种情况。

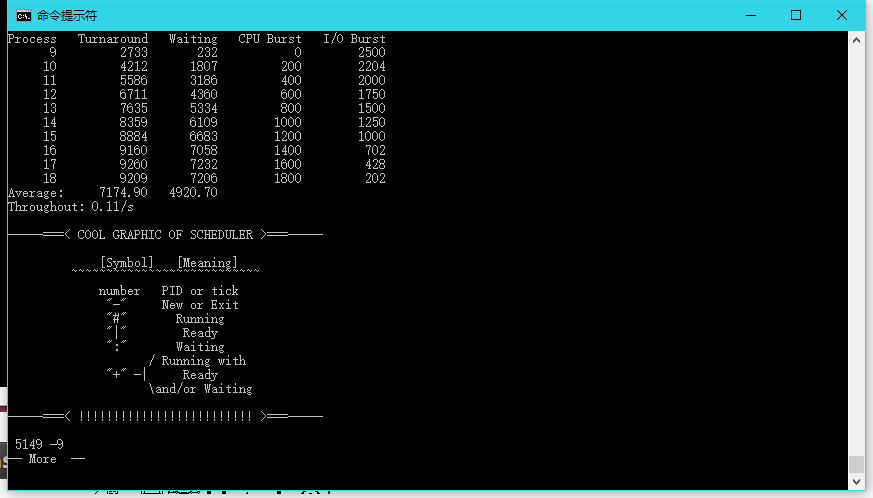
但多进程编程虽然比单进程复杂，但有更广泛的用途。

## 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？

在include/linux/sched.h中修改#define INIT\_TASK宏定义中counter和priority数值

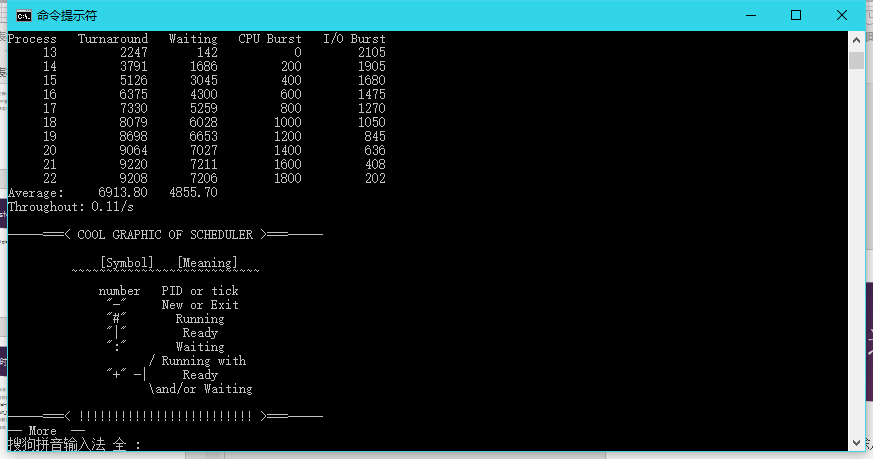
0,15,15, 分别对应state、counter和priority， 将priority值修改，即可实现对时间片大小的调整。时间片修改为15与25

25

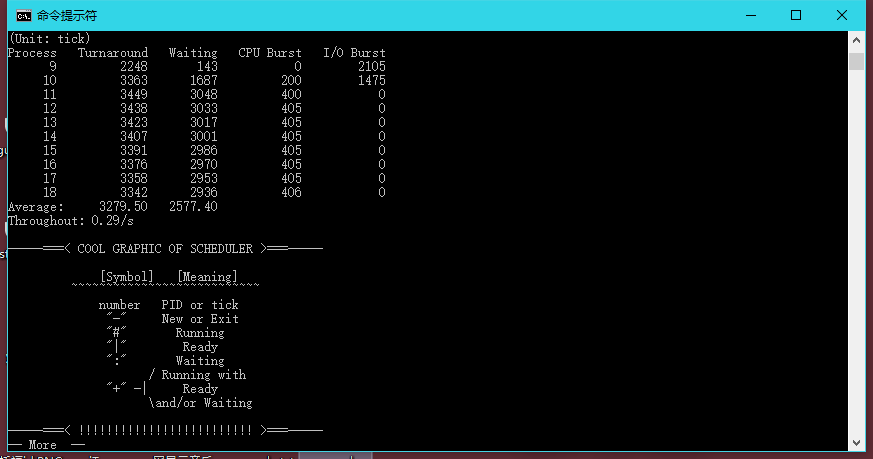


这种情况下，io brust和默认的15情况下差距并不是太大。可能是程序中I/O操作占用的时间对于总时间影响的权重过大，导致处理时间体现的并不明显。平均时间的变化也不大，原因是，子进程连续占用cpu的时间要比时间片大很多。

15



5



* 在一定的范围内，平均等待时间，平均完成时间的变化随着时间片的增大而减小。这是因为在时间片小的情况下，cpu将时间耗费在调度切换上，所以平均等待时间增加。时间片过大，中断和睡眠造成的切换次数变多，平均等待时间变长。
* 时间片过小，切换次数因为时间片到时变多，大多数进程等待时间也会变长,导致平均等待时间还是会变长。
* 所以时间片一定要长短合适。
* 通过在实验结束后对相关资料的查询和学习中，我们得知若时间片增大到一定范围后， RR轮转调度就变成了FCFS先来先服务，参数将不再有明显的变化。
* 随着时间片修改，吞吐量始终变化不明显，我们研究后应该是因为系统在一定时间内所能完成的进程数量是一定的。