

实验二 处理器调度

1、实验内容

设计一个按时间片轮转法实现处理器调度的程序。

2、基本思想

这个实验主要是针对对于进程的时间轮转调度。

“时间片轮转 (Round Robin, RR) 调度算法是按一定的时间片 (time slice) q 轮流地运行就绪队列中的各个进程。如果其为一个定值, 即各个进程运行同样长的时间片, 则轮转法是一种机会均等的调度算法。”

我的做法是将各个所有进程的 PCB 连接成循环队列, 用一个指针指向队列中当前运行的进程。进程运行时间由时间片大小决定, 并且时间到用时钟中断中断运行的进程, 并切向下一个进程, 使其继续运行。如果有进程运行完毕, 则将该进程的 PCB 从循环队列中移除, 直至所有的进程运行完毕。具体的流程见流程图。

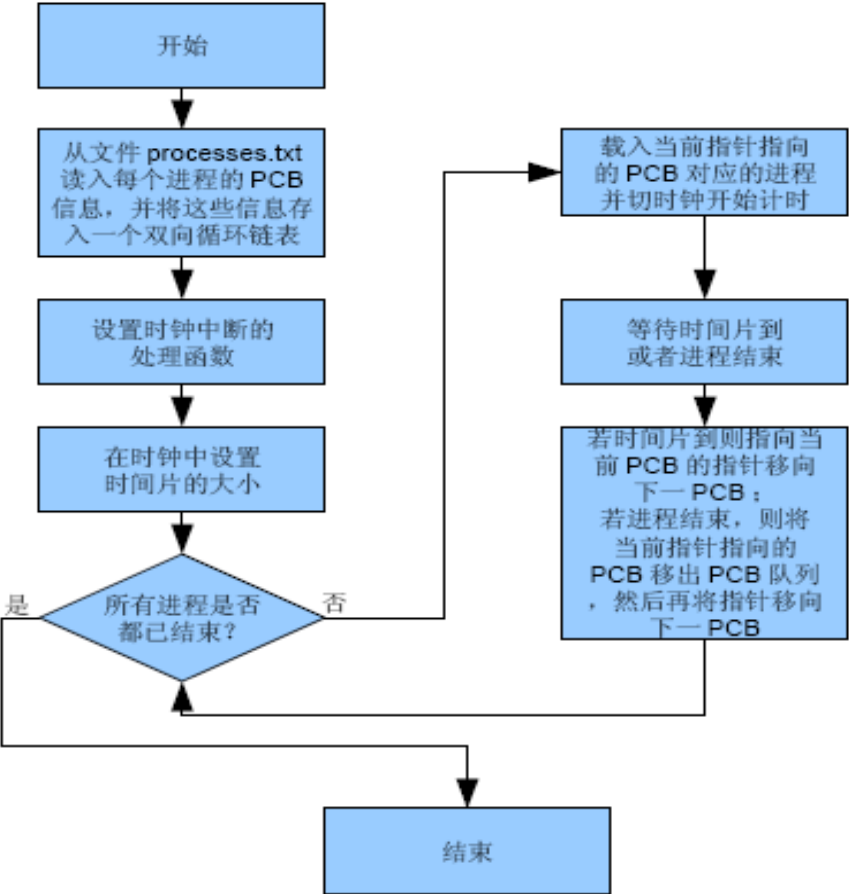


Illustration 1: 实验一流程图

3、实验平台及软件

Debian Linux 4.1.1-21 with Linux Kernel 2.6.18-6-686, GCC version 4.1.2 and GDB version 6.4.90

4、调试过程

(1) 测试数据

10

Q1 20

Q2 30

Q3 44

Q4 100

Q5 90

Drawing 1: 输入数据

(2) 输出结果

Process Q1 begins to run!

Process Q1 has used up its time slice! So begins to wait!

Process Q2 begins to run!

Process Q2 has used up its time slice! So begins to wait!

Process Q3 begins to run!

Process Q3 has used up its time slice! So begins to wait!

Process Q4 begins to run!

Process Q4 has used up its time slice! So begins to wait!

Process Q5 begins to run!

Process Q5 has used up its time slice! So begins to wait!

Process Q1 begins to run!

Process Q1 has been finished!

Process Q2 begins to run!

Process Q2 has used up its time slice! So begins to wait!

Process Q3 begins to run!

Process Q3 has used up its time slice! So begins to wait!

Process Q4 begins to run!

Process Q4 has used up its time slice! So begins to wait!

Process Q5 begins to run!

Process Q5 has used up its time slice! So begins to wait!

Process Q2 begins to run!

Process Q2 has been finished!

Process Q3 begins to run!

Process Q3 has used up its time slice! So begins to wait!

Drawing 2: 输出数据第一部分

Process Q4 begins to run!

Process Q4 has used up its time slice! So begins to wait!

Process Q5 begins to run!

Process Q5 has used up its time slice! So begins to wait!

Process Q3 begins to run!

Process Q3 has used up its time slice! So begins to wait!

Process Q4 begins to run!

Process Q4 has used up its time slice! So begins to wait!

Process Q5 begins to run!

Process Q5 has used up its time slice! So begins to wait!

Process Q3 begins to run!

Process Q3 has been finished!

Process Q4 begins to run!

Process Q4 has used up its time slice! So begins to wait!

Process Q5 begins to run!

Process Q5 has used up its time slice! So begins to wait!

Process Q4 begins to run!

Process Q4 has used up its time slice! So begins to wait!

Process Q5 begins to run!

Process Q5 has used up its time slice! So begins to wait!

Process Q4 begins to run!

Process Q4 has used up its time slice! So begins to wait!

Process Q5 begins to run!

Process Q5 has used up its time slice! So begins to wait!

Drawing 3: 输出数据第二部分

Process Q4 begins to run!
Process Q4 has used up its time slice! So begins to wait!

Process Q5 begins to run!
Process Q5 has used up its time slice! So begins to wait!

Process Q4 begins to run!
Process Q4 has used up its time slice! So begins to wait!

Process Q5 begins to run!
Process Q5 has been finished!

Process Q4 begins to run!
Process Q4 has been finished!

All processes have finished working!

Drawing 4: 输出数据第三部分

5、总结

通过自此实验，我加深了对操作系统中采用的进程调度算法的理解，并且实现了时间片轮转算法的模拟程序；最后，此次实验是我对中断机制有了更好的理解。

6、参考文献

黄水松，黄干平，曾平，李蓉蓉。（2003）。《计算机操作系统》。武汉：武汉大学。