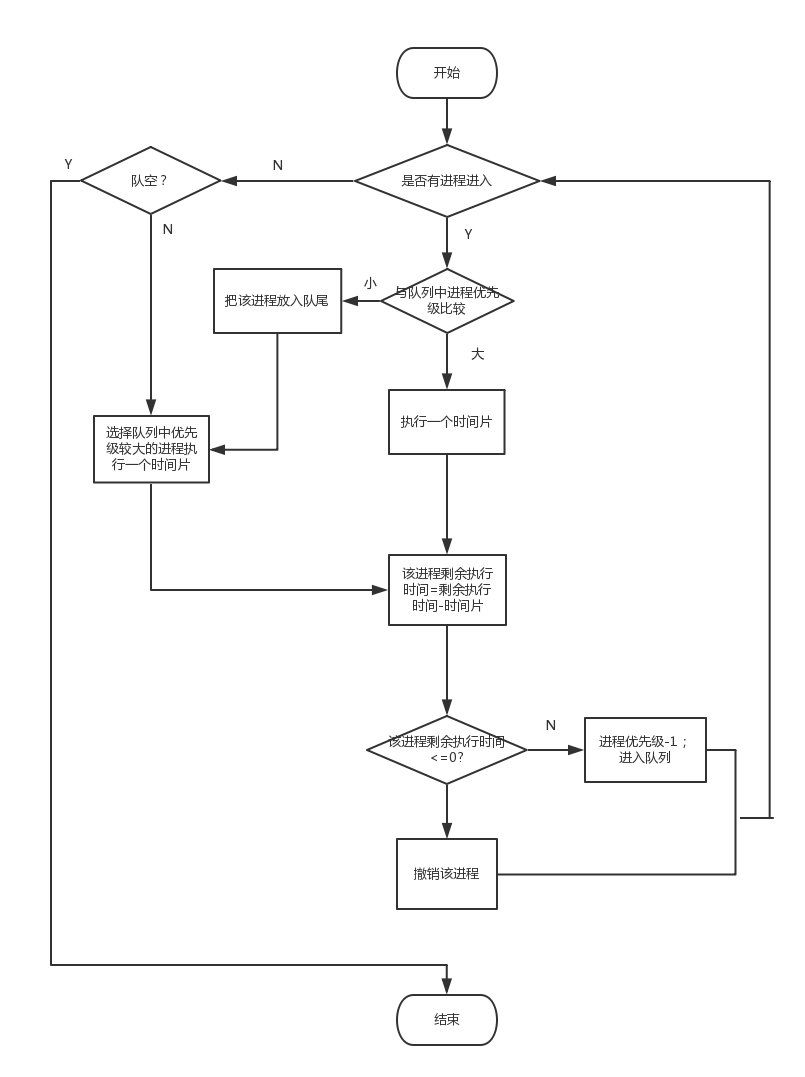
进程调度

实验目的:

理解操作系统中进程调度的概念和调度算法

学习并掌握Windows进程控制以及进程间通信的基本知识.

基本要求:



1. 实现进程调度程序（scheduleprocess）,负责各系统的运行
2. 实现创建进程命令

格式 creat<name><time> name：进程名 time：计划运行时间，用户通过本命令发送创建进程请求，将进程信息提交各系统。系统创建进程为其分配一个唯一的进程标识PID，并将状态置为READY，然后放入就绪队列中。

1. 实现撤销进程命令

格式：remove<name>输入撤销命令后，系统会删除待撤销的进程在缓冲区中的内容，如果输入有误，程序会有出错提示。

1. 实现察看进程状态命令

格式：current 打印出当前运行进程和就绪队列中各进程的信息。状态信息应该包括：进程PID、进程名、进程状态（ready、run、wait）

1. 实现时间片轮转调度算法

处理机总是优先调度运行就绪队列中的第一个进程，当时间片结束后就把该进程放在就绪队列的尾部。在系统的实现以及运行中，不必考虑Windows操作系统本身的进程调度。假设所有的作业均由scheduleprocess调度执行，而且进程在分配给它的时间片内总是不间断地运行。

程序的改进：

程序中采用的是轮转调度策略，可以对策略进行改进，实现多级反馈的轮转调度策略。每个进程有其动态的优先级，在用完分配的时间片后，可以被优先级更高的进程抢占运行。就绪队列中的进程等待时间越长，其优先级越高。每个进程都具有以下两种优先级。

初始优先级（initial priority）:在进程创建时指定，保持不变，直到进程结束。

当前优先级（current priority）:由scheduleprocess调度更新，用以调度进程运行。Scheduleproccess总是选择当前优先级最高的那个进程来运行。进程当前运行优先级的更新主要取决于以下两种情况：

1. 一个进程在就绪对列中等待了若干个时间片（如5个），则将它的当前优先级加1。
2. 若当前运行的进程时间片到，则中止其运行（抢占式多任务），将其放入就绪队列中，它的当前优先级也恢复为初始优先级

进程调度算法：

1. 先来先服务
2. 轮转调度
3. 分级轮转法优先级法
   1. 非抢占方式（优先占有法）
   2. 抢占方式（优先剥夺法）

使用函数介绍：

1. CreatThread()
2. SuspendThread

格式：DWORD SuspendThread（HANDLE hThread）hThread线程句柄

1. DWORD ResumeThread(HANDLE hThread)恢复（唤醒）线程，对于暂停计数为n的线程，必须恢复n次才能为其分配CPU。该函数正确执行则返回线程的前一个暂停计数，否则返回0XFFFFFFFF
2. Bool Terminate Thread(HANDLE hThread,DWORD dwExitCode) hThread线程句柄, dwExitCode线程中止时的推出代码
3. Sleep()
4. CloseHandle（）
5. VOID EnterCriticalSection(LPCRTICAL\_SECTION lpCriticalSection) lpCriticalSection:指向临界区对象的指针。功能说明：等待指定临界区对象的所有权。当线程被赋予所有权时，该函数返回
6. VOID LeaveCriticalSection(LPCRTICAL\_SECTION lpCriticalSection) lpCriticalSection:指向临界区对象的指针。功能说明：释放指定临界区对象的所有权。