操作系统实验 1-进程调度

【实验目的】

（1）理解进程控制块的结构和作用

（2）理解进程运行的并发性

（3）掌握基于动态优先级的时间片轮转调度算法

【实验要求】

（1）设计随机进程产生程序，说明随机性能对算法可能产生的影响

（2）设计一个基于动态优先级的时间片轮转调度算法

（3）以截图方式对程序运行结果进行展示

（4）对测试数据的运行结果进行分析，总结算法特性

（5）若实现可视化效果（图形、动画显示等）有加分

【设计思想】

在多道程序运行环境下，进程数目一般多于处理机数目，使得进程要通过竞争来使用处理机。这就要求系统能按某种算法，动态地把处理机分配给就绪队列中的一个进程，使之运行，分配处理机的任务是由进程调度程序完成的。

一个进程被创建后，系统为了便于对进程进行管理，将系统中的所有进程按其状态，将其组织成不同的进程队列。于是系统有运行进程队列、就绪进程队列和各种事件的进程等待队列。

进程调度的功能就是从就绪队列中挑选一个进程到处理机上运行。进程调度的算法有多种，常用的有优先级调度算法、先来先服务算法、时间片轮转算法等。

进程是程序在处理机上的执行过程。进程存在的标识是进程控制块(PCB)，进程控制块参考结构如下（可根据自己的设计作适当调整）：

```
Typeedef struct node
{   Char    name[10];      /*进程标识符*/
    Int    prio;           /*进程优先数*/
    Int    round;          /*进程时间片轮转时间片*/
    Int    cputime         /*进程占用 CPU 时间*/
    Int    neentime        /*进程到完成还需要的时间*/
    Int    count;          /*计数器*/
    Char    state;         /*进程的状态*/
    Struct node     *next;     /*链指针*/
}PCB;
```

系统创建一个进程，就是由系统为某个程序设置一个 PCB，用于对该进程进行控制和管理。进程任务完成，由系统收回其 PCB，该进程便消亡。每个进程可以有三个状态：运行态、就绪态和完成状态。

使用时间片轮转的动态优先数调度算法完成进程的调度

1) 采用动态优先数法确定进程的优先级别。

2) 设计三个链队列，分别用来表示运行队列、就绪队列和完成队列。

3) 用户输入进程标识符以及进程所需要的时间，申请空间存放进程 PCB 信息。

优先数调度算法为每个进程设一个优先数，它总是把处理机分配给就绪队列中具有最高优先权的进程。初始的进程优先数取决于进程运行所需要的时间，时间长则优先数低。可采取将进程优先数定为一个较大的数（比如 50）减去进程运行所需要的时间。

常用的算法有静态优先数法和动态优先数法。动态优先数法使进程的优先权随时间而改变，进程运行以一个时间片为固定周期 T，每个时间片结束时，动态调整各个进程的优先级，当前正在执行的进程优先级降低（优先数减小），所有就绪进程优先级提升（优先数增加），调整完毕后，重新选择当前优先级最高的进程投入运行。

如果进程所需时间为 0，说明进程运行完毕，将其状态变为完成状态“F”，将此进程 PCB 插入到完成队列中。

重复上述过程，直到就绪队列为空，所有进程都变为完成状态为止。