
一、实验内容

设计实现处理机调度算法。

二、实验目的

多道系统中，当就绪进程数大于处理机数时，须按照某种策略决定哪些进程优先占用处理机。本实验模拟实现处理机调度，以加深了解处理机调度的工作。

三、实验要求

- 1、设计一个有几个进程并发执行的进程调度程序，每个进程由一个进程控制块(PCB)表示，进程控制块通常应包括下述信息：进程名，进程 PID，进程优先级，进程状态、PCB 指针等，且可按照调度算法的不同而增删。
- 2、调度算法包括：优先权调度算法和时间片轮转调度算法。
- 3、程序应能显示或打印各进程状态和参数的变化情况，便于观察。
- 4、可随时增加新进程。

四、实验分析与设计

优先权调度算法

1. 优先级调度的含义

(1) 当该算法用于作业调度时，系统从后备作业队列中选择若干个优先级最高的，且系统能满足资源要求的作业装入内存运行。

(2) 当该算法用于进程调度时，将把处理机分配给就绪进程队列中优先级最高的进程。

2. 优先级调度算法细分成如下两种方式：

a. 非抢占式优先级算法

在这种调度方式下，系统一旦把处理机分配给就绪队列中优先级最高的进程后，该进程就能一直执行下去，直至完成；或因等待某事件的发生使该进程不得不放弃处理机时，系统才能将处理机分配给另一个优先级高的就绪进程。

b. 抢占式优先级调度算法

在这种调度方式下，进程调度程序把处理机分配给当时优先级最高的就绪进程，使之执行。一旦出现了另一个优先级更高的就绪进程时，进程调度程序就停止正在执行的进程，将处理机分配给新出现的优先级最高的就绪进程。

3. 优先级的类型

进程的优先级可采用静态优先级和动态优先级两种，优先级可由用户自定或由系统确定。

静态优先级

(1) 含义

静态优先级是在创建进程时确定进程的优先级，并且规定它在进程的整个运行期间保持不变。

(2) 确定优先级的依据

确定优先级的依据通常有下面几个方面：

①进程的类型。通常系统进程优先级高于一般用户进程的优先级；交互型的用户进程的优先级高于批处理作业所对应的进程的优先级。

②进程对资源的需求。例如，进程的估计执行时间及内存需求量少的进程，应赋予较高的优先级，这有利缩小作业的平均周转时间。

③根据用户的要求。用户可以根据自己作业的紧迫程度来指定一个合适的优先级。

(3) 优缺点

静态优先级法的优点是：简单易行、系统开销小。

缺点是：不太灵活，很可能出现低优先级的作业(进程)，长期得不到调度而等待的情况。

静态优先级法仅适用于实时要求不太高的系统。

动态优先级

(1) 含义

动态优先级是在创建进程时赋予该进程一个初始优先级，然后其优先级随着进程的执行情况的变化而改变，以便获得更好的调度性能。

(2) 优缺点

动态优先级优点是使相应的优先级调度算法比较灵活、科学，可防止有些进程一直得不到调度，也可防止有些进程长期垄断处理机。动态优先级缺点是需要花费相当多的执行程序时间，因而花费的系统开销比较大。

4. 设计思路

本实验采用抢占式动态优先权，设置三个队列：运行队列（CPU）、就绪队列、等待队列。

运行队列中只有一个进程，就是正在运行的进程；

就绪队列为准备好的进程队列，就绪队列满后新来的进程或被替换掉的进程

均放入等待队列。

每次调度前检查就绪队列是否有位置，如果有位置则把等待队列中的进程按顺序放入就绪队列。

正在运行的进程优先权数值会升高（优先权 0~255，0 代表最低优先权），一旦就绪队列中有进程优先权比当前正在运行的进程优先权高时，自动替换，被替换的进程放到队尾。

运行完毕的进程会被删除。

设置 witing 变量记录在就绪队列中的进程等待时间。

开辟线程，允许在处理机调度的同时添加新进程。

设计图形界面，更加直观。

时间片轮转调度算法

1.基本原理

在早期的时间片轮转法中,系统将所有的就绪进程按先来先服务的原则,排成一个队列,每次调度时,把 CPU 分配给队首进程,并令其执行一个时间片.时间片的大小从几 ms 到几百 ms.当执行的时间片用完时,由一个计时器发出时钟中断请求,调度程序便据此信号来停止该进程的执行,并将它送往就绪队列的末尾;然后,再把处理机分配给就绪队列中新的队首进程,同时也让它执行一个时间片.这样就可以保证就绪队列中的所有进程,在一给定的时间内,均能获得一时间片的处理机执行时间。

2. 时间片大小的确定

- 1.系统对响应时间的要求
- 2.就绪队列中进程的数目
- 3.系统的处理能力

3.设计思路

本实验采用抢占式动态优先权，设置三个队列：运行队列（CPU）、就绪队列、等待队列。

运行队列中只有一个进程，就是正在运行的进程；

就绪队列为准备好的进程队列，就绪队列满后新来的进程或被替换掉的进程均放入等待队列。

每次调度前检查就绪队列是否有位置，如果有位置则把等待队列中的进程按顺序放入就绪队列。

预先为每个进程分配时间片，时间片提前设定（如 10），设计链表数据结构模拟队列，将到来的进程依次插入队列，每次从队头取出一个进程放入 CPU，时间片随时间递减，如果时间片减少到 0 而进程仍未运行完毕，则将其放到队尾，并将其时间片重置，此后从当前就绪队列队头取出一个进程放入 CPU，以此类推。

运行完毕的进程会被删除。

设置 `witing` 变量记录在就绪队列中的进程等待时间。

开辟线程，允许在处理机调度的同时添加新进程。

设计图形界面，更加直观。

五、实验运行结果和相关源代码

1. 实验结果

优先权调度算法

通过选择界面选择优先权调度算法，为方便调试，提前放入了一些进程，程序运行时自动调到就绪队列中并执行优先权调度算法，允许插入新进程，点击 **Start** 自动开始模拟调度过程。点击 **New** 按钮可添加新进程



处理器调度

Running:

Ready:

New

Start

Suspend

Reversing

Running PCB

Ready PCB





[illegible]

通过选择界面选择时间片调度算法，为方便调试，提前放入了一些进程，程序运行时自动调到就绪队列中并执行优先权调度算法，允许插入新进程，点击 **Start** 自动开始模拟调度过程。点击 **New** 按钮可添加新进程



