# 2.进程调度模拟程序设计（进程管理）

1．给出进程调度的算法描述（如基于动态优先级和时间片轮转调度算法的描述）。

2．用C语言设计一个对n个并发进程进行调度的程序，每个进程由一个进程控制块（PCB）结构表示，该进程控制块应包括下述信息：进程标识ID、进程优先数PRIORITY（并规定优先数与优先权成正比）、时间片数CHIP、进程已经占用CPU的时间CPUTIME，进程还需要运行的时间ALLTIME（当进程运行完毕时，其值为0）、进程的状态STATE（为简化起见。设每个进程处于运行E（excecuting）、就绪R（ready）和完成F（finish）三种状态之一，并假设起始状态都是就绪状态R。），以及进程队列指针NEXT（用来将PCB排成队列）等，可按照调度算法的不同而增删。

3．调度程序应当包含2种不同的调度算法，运行时可以任选一种，以利于各种方法的分析和比较。

4．程序应能显示或打印各种进程状态和参数变化情况，便于观察。即要显示每个时间片内各进程的情况，并且指出运行进程及就绪和阻塞队列中的内容。

**(1)按优先数调度算法实现处理器调度的程序。**

[提示]：

(1) 假定系统有五个进程，每一个进程用一个进程控制块PCB来代表，进程控制块的格式为：

|  |
| --- |
| 进程名 |
| 指针 |
| 要求运行时间 |
| 优先数 |
| 状态 |

其中，进程名——作为进程的标识，假设五个进程的进程名分别为P1，P2，P3，P4，P5。

指针——按优先数的大小把五个进程连成队列，用指针指出下一个进程的进程控制块的首地址，最后一个进程中的指针为“0”。

要求运行时间——假设进程需要运行的单位时间数。

优先数——赋予进程的优先数，调度时总是选取优先数大的进程先执行。

状态——可假设有两种状态，“就绪”状态和“结束”状态。五个进程的初始状态都为“就绪”，用“R”表示，当一个进程运行结束后，它的状态为“结束”，用“E”表示。

(2) 在每次运行你所设计的处理器调度程序之前，为每个进程任意确定它的“优先数”和“要求运行时间”。

(3) 为了调度方便，把五个进程按给定的优先数从大到小连成队列。用一单元指出队首进程，用指针指出队列的连接情况。例：

队首标志

K2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K1 | P1 | K2 | P2 | K3 | P3 | K4 | P4 | K5 | P5 |
|  | 0 |  | K4 |  | K5 |  | K3 |  | K1 |
|  | 2 |  | 3 |  | 1 |  | 2 |  | 4 |
|  | 1 |  | 5 |  | 3 |  | 4 |  | 2 |
|  | R |  | R |  | R |  | R |  | R |
|  | PCB1 |  | PCB2 |  | PCB3 |  | PCB4 |  | PCB5 |

(4) 处理器调度总是选队首进程运行。采用动态改变优先数的办法，进程每运行一次优先数就减“1”。由于本实验是模拟处理器调度，所以，对被选中的进程并不实际的启动运行，而是执行：

优先数-1

要求运行时间-1

来模拟进程的一次运行。

提醒注意的是：在实际的系统中，当一个进程被选中运行时，必须恢复进程的现场，让它占有处理器运行，直到出现等待事件或运行结束。在这里省去了这些工作。

(5) 进程运行一次后，若要求运行时间≠0，则再将它加入队列（按优先数大小插入，且置队首标志）；若要求运行时间=0，则把它的状态修改成“结束”（E），且退出队列。

(6) 若“就绪”状态的进程队列不为空，则重复上面（4）和（5）的步骤，直到所有进程都成为“结束”状态。

(7) 在所设计的程序中应有显示或打印语句，能显示或打印每次被选中进程的进程名以及运行一次后进程队列的变化。

(8) 为五个进程任意确定一组“优先数”和“要求运行时间”，启动所设计的处理器调度程序，显示或打印逐次被选中进程的进程名以及进程控制块的动态变化过程。

**(2)按时间片轮转法实现处理器调度的程序。**

[提示]：

(1) 假定系统有五个进程，每一个进程用一个进程控制块PCB来代表。进程控制块的格式为：

|  |
| --- |
| 进程名 |
| 指针 |
| 要求运行时间 |
| 已运行时间 |
| 状态 |

其中，进程名——作为进程的标识，假设五个进程的进程名分别为Q1，Q2，Q3，Q4，Q5。

指针——进程按顺序排成循环队列，用指针指出下一个进程的进程控制块的首地址，最后一个进程的指针指出第一个进程的进程控制块首地址。

要求运行时间——假设进程需要运行的单位时间数。

已运行时间——假设进程已经运行的单位时间数，初始值为“0”。

状态——有两种状态，“就绪”和“结束”，初始状态都为“就绪”，用“R”表示。当一个进程运行结束后，它的状态为“结束”，用“E”表示。

(2) 每次运行所设计的处理器调度程序前，为每个进程任意确定它的“要求运行时间”。

(3) 把五个进程按顺序排成循环队列，用指针指出队列连接情况。另用一标志单元记录轮到运行的进程。例如，当前轮到P2执行，则有：

标志单元

K2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K1 | Q1 | K2 | Q2 | K3 | Q3 | K4 | Q4 | K5 | Q5 |
|  | K2 |  | K3 |  | K4 |  | K5 |  | K1 |
|  | 2 |  | 3 |  | 1 |  | 2 |  | 4 |
|  | 1 |  | 0 |  | 0 |  | 0 |  | 0 |
|  | R |  | R |  | R |  | R |  | R |
|  | PCB1 |  | PCB2 |  | PCB3 |  | PCB4 |  | PCB5 |

(4) 处理器调度总是选择标志单元指示的进程运行。由于本实验是模拟处理器调度的功能，所以，对被选中的进程并不实际的启动运行，而是执行：

已运行时间+1

来模拟进程的一次运行，表示进程已经运行过一个单位的时间。

请同学注意：在实际的系统中，当一个进程被选中运行时，必须置上该进程可以运行的时间片值，以及恢复进程的现场，让它占有处理器运行，直到出现等待事件或运行满一个时间片。在这时省去了这些工作，仅用“已运行时间+1”来表示进程已经运行满一个时间片。

(5) 进程运行一次后，应把该进程的进程控制块中的指针值送到标志单元，以指示下一个轮到运行的进程。同时，应判断该进程的要求运行时间与已运行时间，若该进程的要求运行时间≠已运行时间，则表示它尚未执行结束，应待到下一轮时再运行。若该进程的要求运行时间=已运行时间，则表示它已经执行结束，应指导它的状态修改成“结束”（E）且退出队列。此时，应把该进程的进程控制块中的指针值送到前面一个进程的指针位置。

(6) 若“就绪”状态的进程队列不为空，则重复上面的（4）和（5）的步骤，直到所有的进程都成为“结束”状态。

(7) 在所设计的程序中应有显示或打印语句，能显示或打印每次选中进程的进程名以及运行一次后进程队列的变化。

(8) 为五个进程任意确定一组“要求运行时间”，启动所设计的处理器调度程序，显示或打印逐次被选中的进程名以及进程控制块的动态变化过程。