**《计算机操作系统》实验报告**

**实验题目：操作系统的进程调度**

**姓名：汪雨卿 学号：19120191 实验日期：2021.12.07**

**实验环境：**

Visual Studio 2019; C++

**实验目的：**

进程是操作系统最重要的概念之一，进程调度又是操作系统核心的主要内容。本实习要求学生独立地用高级语言编写和调试一个简单的进程调度程序。调度算法可任意选择或自行设计。例如，简单轮转法和优先数法等。本实习可加深对于进程调度和各种调度算法的理解。

**实验内容：**

（1） 编制和调试示例给出的进程调度程序，并使其投入运行。

（2） 自行设计或改写一个进程调度程序，在相应机器上调试和运行该程序，其功能应该不亚于示例。

**提示：**可编写一个反馈排队法（FB 方法）的进程调度程序。该算法的基本思想是设置几个进程就绪队列，如队列 1„„队列 i ，同一队列中的进程优先级相同，可采用先进先出方法调度。各队列的进程，其优先级逐队降低。即队列 1 的进程优先数最高，队列 i 的最低。而时间片，即以此占用 CPU 的时间正好相反，队列 1 的最短，队列 i 则最长。调度方法是开始进入的进程都在队列 1 中参加调度，如果在一个时间片内该进程完不成，应排入队列 2，即优先级要降低，但下一次运行的时间可加长（即时间片加长了）。以此类推，直至排到队列 i。调度时现在队列 1 中找，待队列 1 中已无进程时，再调度队列 2 的进程，一旦队列 1 中有了进程，又应返回来调度队列 1 的进程。这种方法最好设计成运行过程中能创造一定数量的进程，而不是一开始就生成所有进程。

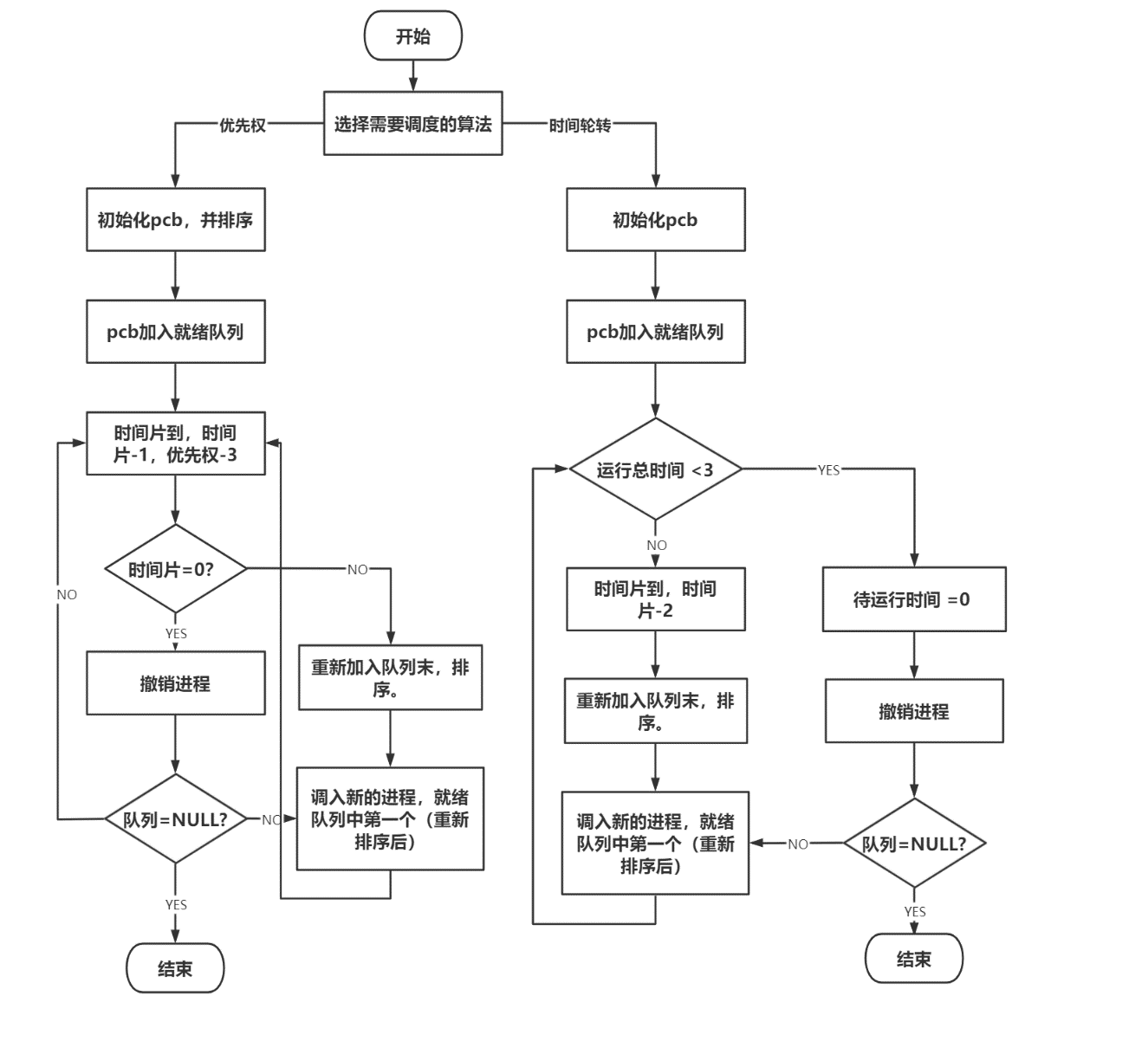
**提示：**可综合各种算法的优先，考虑在各种不同情况下的实施方法，如上述 FB 算法。也可选用有关资料中报导的一些方法，加以分析、简化和实现。

（3）直观地评测各种调度算法的性能。

**操作过程：**

**1. 设计算法的流程图**

图1.1调度算法逻辑框图中展示了本程序的算法逻辑过程。



**图 1 调度算法逻辑框图**

**2. 设计进程控制块PCB的数据结构**

**将进程控制块的PCB用类来表达，设置PID，Priority，CPUtime，ALLtime，State，Finish。并设置对应的getter 和 setter 方法。**

* **PID：标志进程块的序号**
* **Priority；标志PCB的优先级**
* **CPUtime：记录CPU已经运行的时间片数**
* **ALLtime：记录该进程还需要运行的时间片数**
* **State：表示进程当前的运行状态 （1.运行 2.就绪 3. 等待 -1.未初始化）**
* **Finish：标志进程是否完成（TRUE 完成，False 未完成）**

**图2 PCB类实现中展示了具体代码实现。**



**图 2 PCB类实现**

**3. 设计两种调度算法的具体实现**

**a. 优先调度序算法的实现**

**图3-4 优先级调度算法代码实现 显示了该调度算法的具体实现**

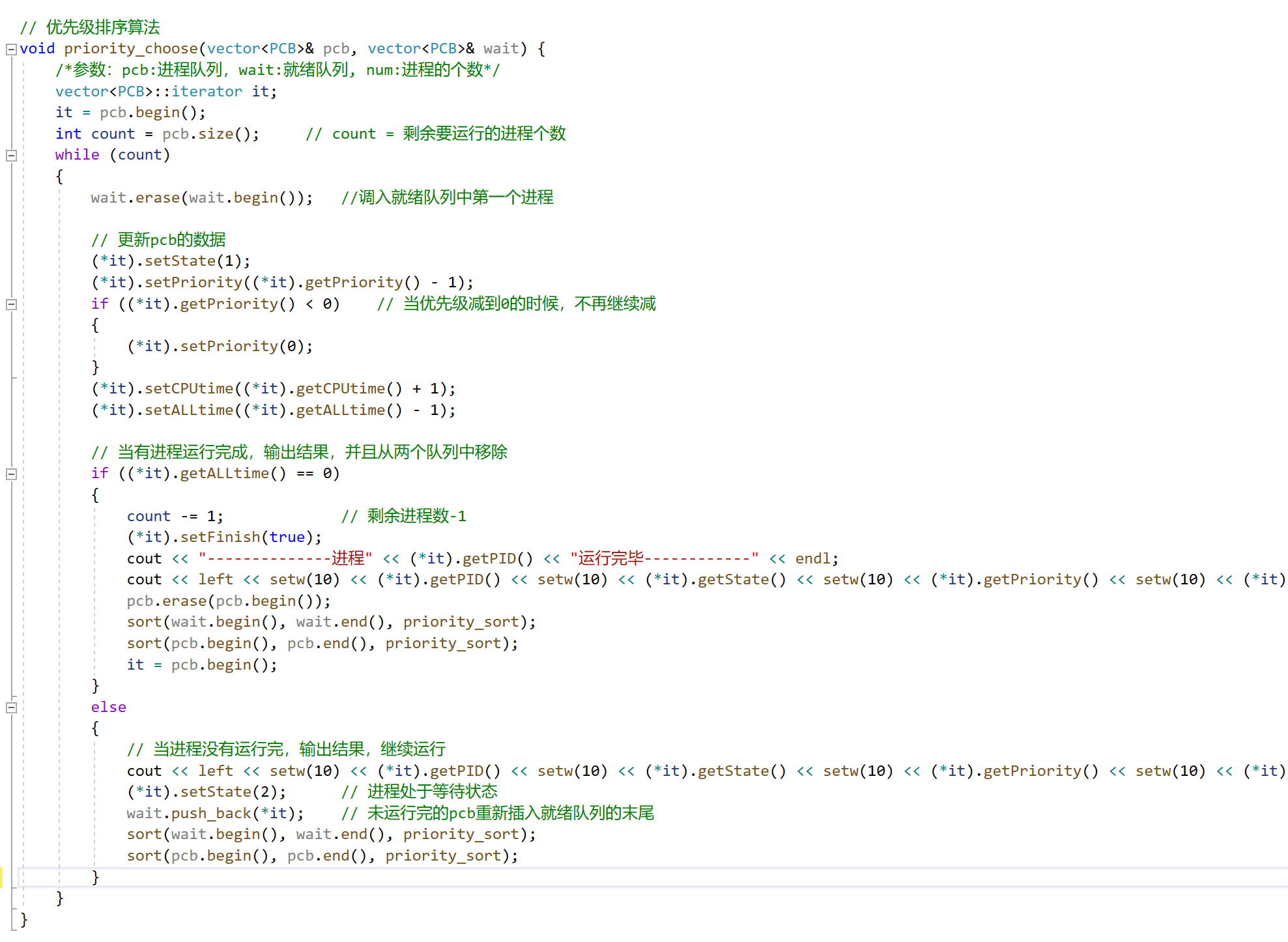


图 3优先级调度算法代码实现

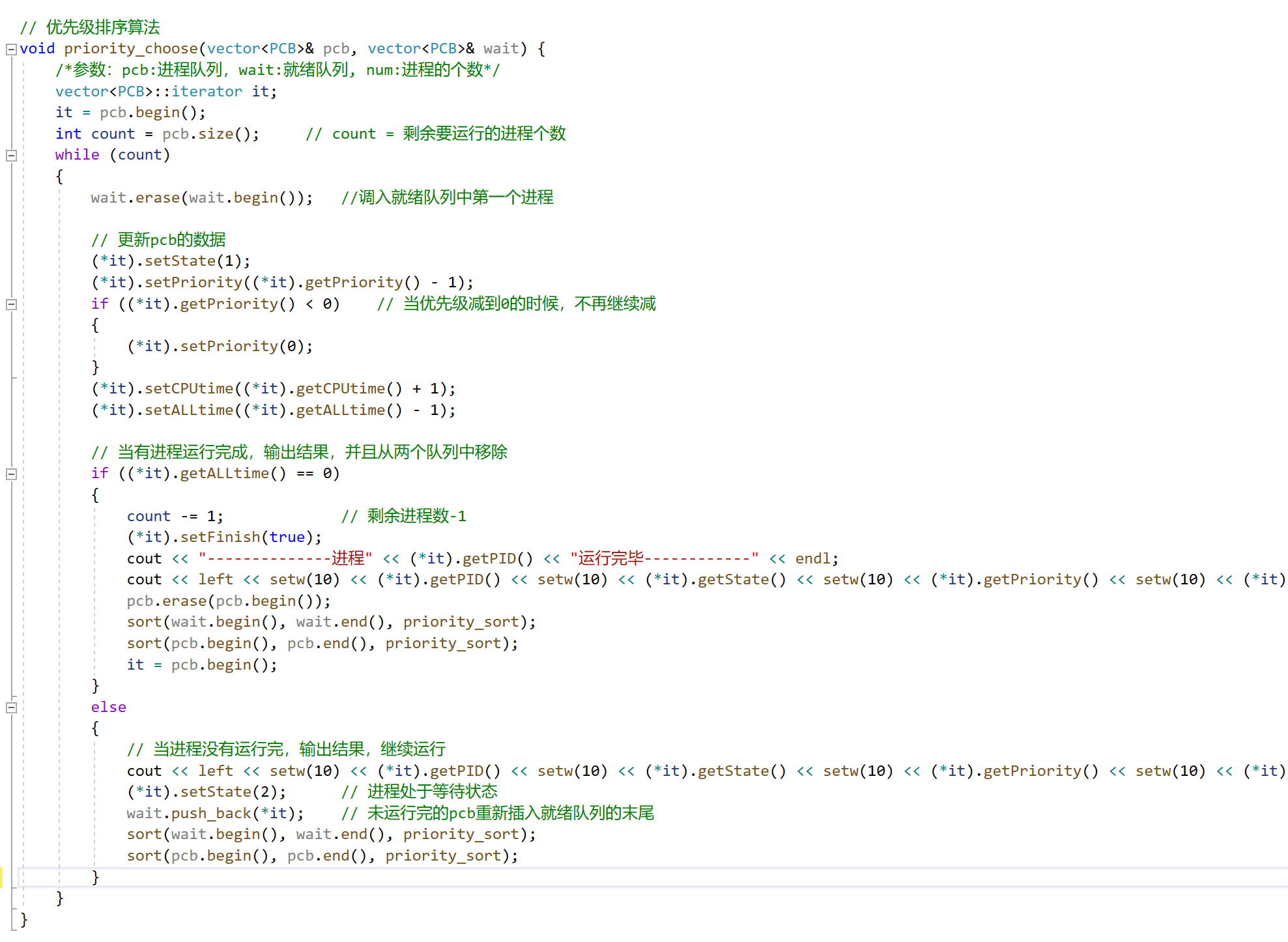


图 4优先级调度算法代码实现

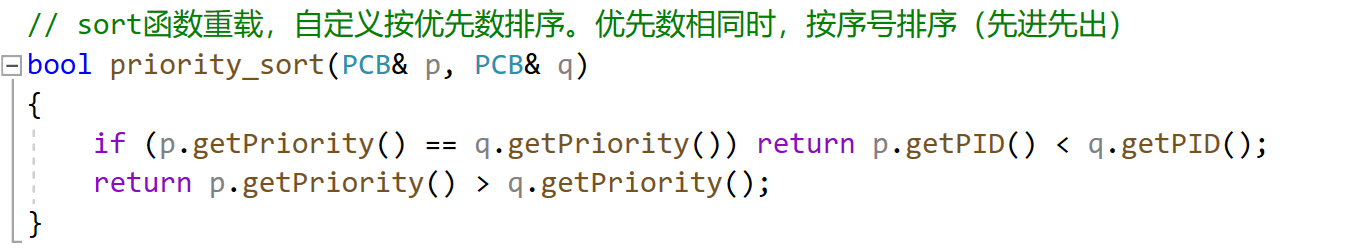
**b.时间调度算法的实现**

**图5时间片轮转调度算法代码实现 显示了该调度算法的具体实现**



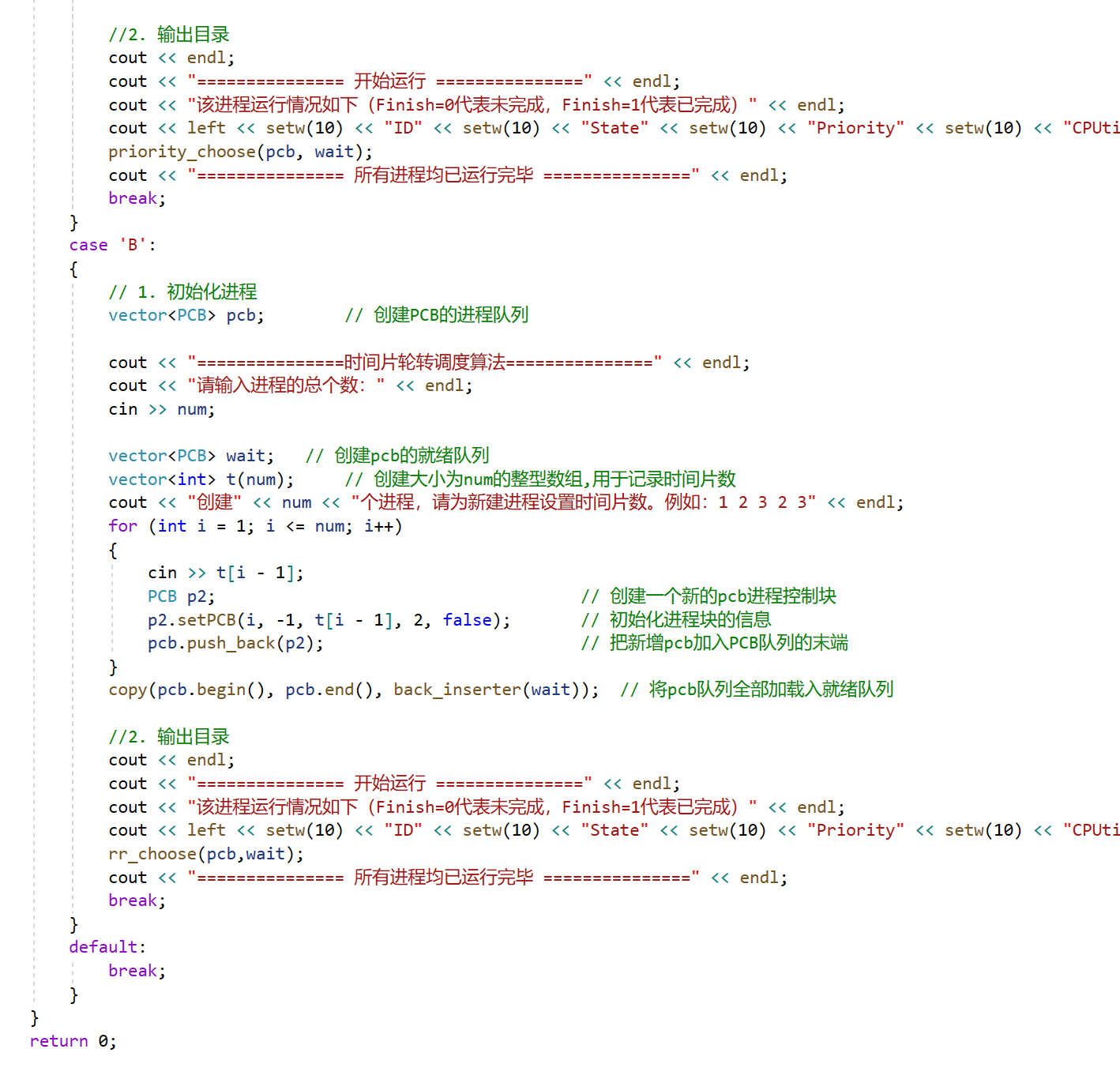
图 5时间片轮转调度算法代码实现

**4.优先级调度算法的sort函数重载**



**5. 主程序设计**





**5. 初始条件设置**

**a. 优先数调度算法的设置**

**选择调度算法：A 选择优先数调度算法**

**设置进程个数 4**

**设置优先级 4 2 1 3**

**设置时间片数 5 3 4 5**

**b. 时间片轮转算法的设置**

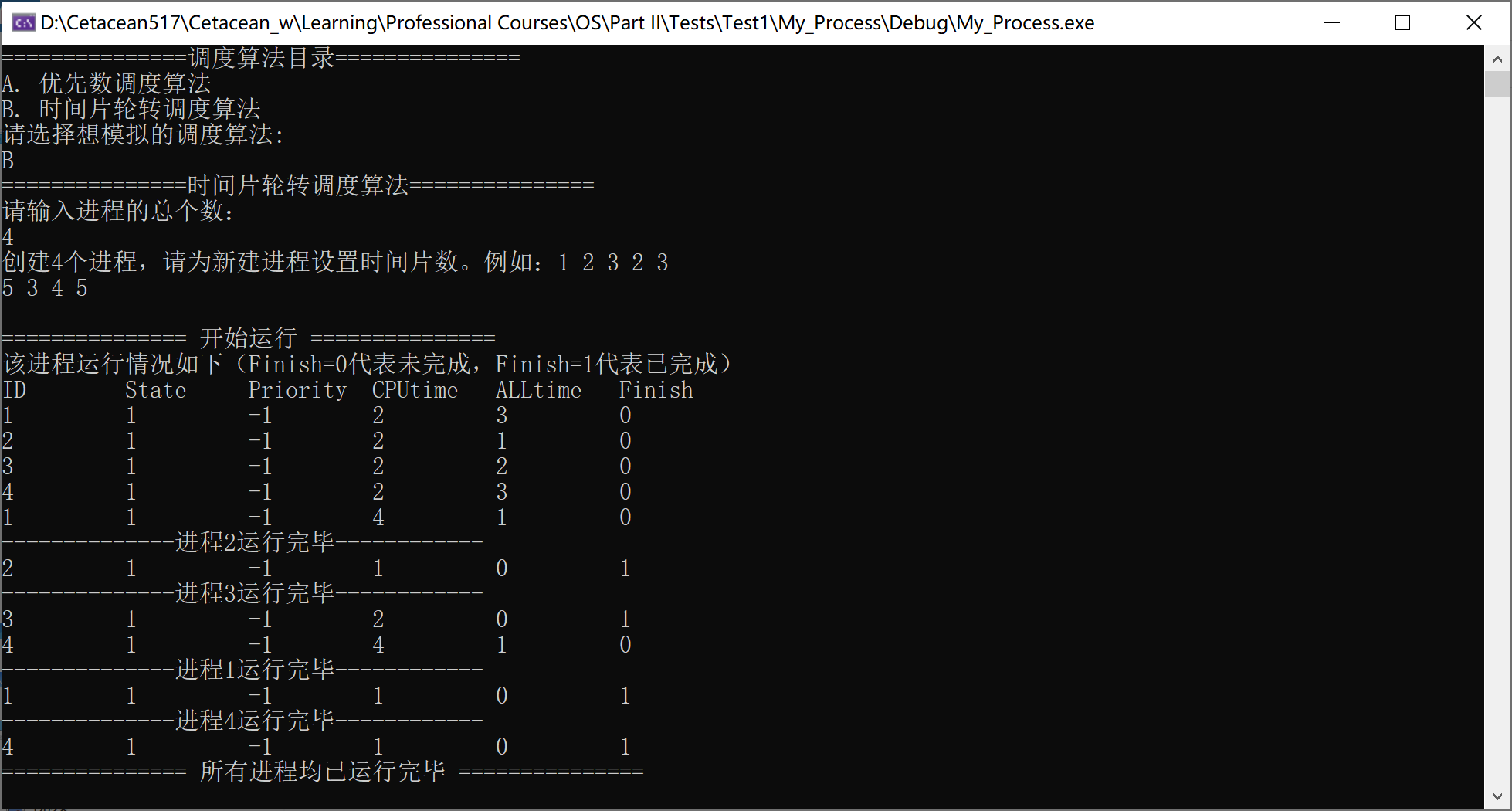
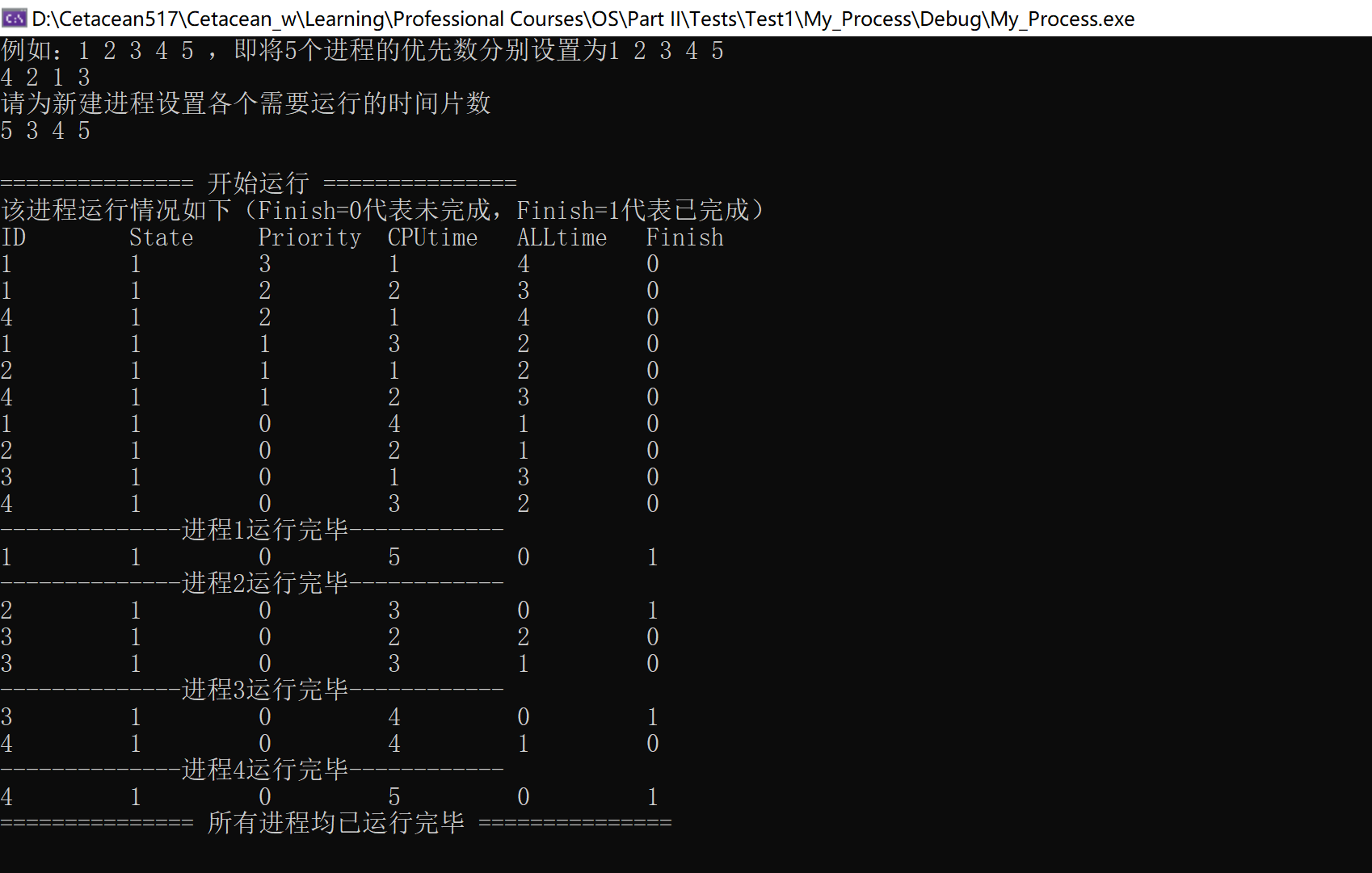
**选择调度算法：B 选择时间片轮转调度算法**

**设置进程个数 4**

**设置时间片数 5 3 4 5**

**结果：**

**1.输出运行结果**



**2.评价指标**

**平均周转时间 = （完成时间-到达时间）/ 进程个数**

**带权周转时间 = （周转时间/服务时间）/进程个数**

**3. 两个算法性能比较**

**a. 时间片轮转**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 进程号 | 服务时间 | 周转时间 | 带权周转时间 | 平均周转时间 | 平均带权周转时间 |
| 1 | **5** | **12** | **2.4** | **10.50** | **2.505** |
| 2 | **3** | **8** | **2.67** |
| 3 | **4** | **11** | **2.75** |
| 4 | **5** | **11** | **2.2** |

**b. 优先权**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 进程号 | 服务时间 | 周转时间 | 带权周转时间 | 平均周转时间 | 平均带权周转时间 |
| 1 | **5** | **6** | **1.2** | **10.25** | **2.5125** |
| 2 | **3** | **9** | **3** |
| 3 | **4** | **13** | **3.25** |
| 4 | **5** | **13** | **2.6** |

**总结：**

**在4个进程同时到达，且只有一个CPU为期服务的时候。优先数调度算法平均周转时间更短，性能更优。**

**体会：**

**本次实验在于模拟进程调度的算法调度的过程，同时进行性能的比较。在实验实现的过程中，才会发现理论掌握和实践实现的过程还是有很大的区别。最开始在设计PCB的数据结构的时候，会考虑是用结构体还是封闭性更好的类来实现。也会为了更好的展示进程运行结果设计用户友好的输出提示。最后，本次实验也帮我更好的掌握了两种调度算法的差异和效率高低。**