

# 设计思想

使用递归的方法和回溯的思想完成问题的求解。

单线程可抢占式执行单元(也就是打印机)的执行特点是,高优先级的任务可以打断低优先级的任务,完成后才能执行低优先级的任务。

这个特点说明:

当高优先级的任务在放入执行序列中时,只会根据更高优先级的任务调整自己的实际开始执行时间和实际结束执行时间。

因此,一旦高优先级的任务的实际开始·结束执行时间已知后,低优先级的任务的实际开始·结束行时间都可以计算得出。

**所以可得出结论:**一旦知道所有任务的优先级,一定可以得到所有任务的实际开始执行时间和实际结束执行时间。方法是从最高优先级的任务开始,一个个**贴入**执行序列中。

**我们可以得出计算优先级的方法:**我们不妨将未知优先级的任务优先级先设为最高,在放入执行序列后判断调整后的实际结束执行时间是否为设定的时间。如果不符合,则说明该优先级不符合未知任务的实际优先级,此时我们需要逐渐降低未知任务的优先级,并把高优先级的任务贴入执行序列;当降到某一个优先级时,未知任务的实际结束执行时间符合设定的时间,那么这个优先级符合该任务的实际优先级。实际优先级的解可能不止一个,因为假设共有优先级 1,优先级 4,和未知优先级三个任务,假设未知优先级任务实际优先级为 3,那么就算把该优先级设为 2 时,它的实际结束执行时间也会符合预设时间。

# 算法描述：

## 核心算法描述：

①将未知优先级的 Task(后称 UnknowTask)的优先级设为待执行序列（包含所有未进行计算的 Task）中的最高优先级。

将 UnknowsTask 使用(% 贴入算法 %)贴入执行序列中(根据序列中已有的 task 调整 UnknowTask 的实际开始执行时间和实际结束执行时间，并将 task 放在执行序列中)

判断 UnknowTask 的实际开始执行时间和实际结束执行时间是否符合设定的数据。

如果不符合预设的实际结束执行时间，则将 UnknowTask 从执行序列中取出（由于高优先可抢占的特性，取出时将不影响任何已经贴入的高优先级任务）

从待执行序列中取出一个具有最高优先级的 Task，贴入执行序列中

回到①

如果符合设定的数据，则将待执行序列中的剩余所有任务逐一贴入执行序列中。

返回执行序列（即为最终的结果）

## 贴入算法：

因为主算法是将 Task 从优先级最高的开始从高到低逐步安排执行，因此新进入的 Task 一定比已贴入的 Task 具有更低的优先级

Task 的进入、调整 and 安排的过程不会修改已安排好的任务（因为 Task 的优先级最低）

最后就算把 Task 从安排好的序列中拿出，也不会影响别的已安排任务

## 算法描述：

得到一个新的 Task，获取它的到达时间和执行时间，设定它的开始执行时间和结束执行时

间

① 根据 Task 的开始执行时间和结束执行时间，使用(% [冲突检测算法](#) %)查看这段时间

内有没有和已安排的任务发生冲突

如果有

则根据不同的情况将 Task 的开始执行时间或结束执行时间向后调整

回到①

如果没有

将 Task 贴入序列

返回这个 Task

## 冲突检测算法：

当 newTask 到来时，可能会和已安排的任务发生 4 种可能的重合情况

重合第一种情况：



需要调整为：( newTask 的实际结束执行时间+ = 高优先任务的执行时间 )



**重合第二种情况（完全被包含）：**



需要调整为：（修改 newTask 的实际开始执行时间为高优先任务的实际结束执行时间）



**重合第三种情况：**

注：newTask 针对每一个高优先任务只会调整一次，因此不可能出现经过重合第一种情况

调整后，重新判断为这一种情况。



需要调整为 : ( newTask 的实际结束执行时间+ =高优先任务的执行时间 )



重合第四种情况：



需调整为 : ( newTask.实际开始执行时间=高优先任务.实际结束执行时间 )



# 模块概述

本程序共分为 4 个模块，分别是 Inputer, Printer, PrintQueue 和 Outputer，核心模块为 Printer 和 PrintQueue.

| 模块名        | 模块职责   | 实现简介  |
|------------|--|---|
| Inputer    | 将计算需要的数据按照需求读取,并转换成 Printer 需要的格式                          | 从文件中读取数据，将数据转换成 LinkedList<Task>，并将具有-1 priority 的那一行的数据转换成 UnknowTask 一并放入 LinkedList<Task>中，交给 Printer                            |
| Printer    | 完成程序的核心功能，求出待求解的 task 的 priority，计算出所有 task 的 endPrintTime | 使用递归的方法，用回溯法的思想求出 task 的 priority，需要 PrintQueue 的帮助，具体的思想已在本文档的设计思想部分给出。  |
| PrintQueue | 接收一个 task，计算这个 task 的开始执行时间和结束执行时间，并把这个 Task 记录在执行序列中；     | 当打印序列中新添加一个 task 的时候，通过这个 task 和已有 task 的冲突检测，调整这个 task 的开始执行时间和结束执行时间，并把调整玩的这个 task 写入执行序列中 ;允许 Printer 按写入执行序列的顺序逐一将 task 从序列中释放。 |
| OutPuter   | 将 Printer 得出的结果按照需   | 从 Printer 中读取 UnknowTask，从  |

|  |     |   |
|--|-----|---|
|  | 求给出 | Inputer 中读取最初的数据读取顺序<br>( Printer 中的结果是按时间排序的 ), 并<br>将数据输出到本地文件中 |
|--|-----|---|