

题目描述

假设你有一台计算机，而我们很多任务。这些任务需要提交给计算机来完成，我们希望你能写出一个合理的调度算法，来合理分配你拥有的资源，用这些资源来尽可能快的完成任务

条件

1. 任务

每个任务包含：

1. 任务id，一个任务的唯一标识，一个整数，范围1-999
2. cpu时间，这个任务需要一个cpu用多少时间片来完成，一个整数，int类型
3. 需要的资源，执行这个任务需要哪些资源，一个int型数组，里面包含需要的资源id

2. 资源

每个资源包含：

1. 资源id，一种任务的唯一标识，一个整数，范围1-128

环境模拟

1. 内存

内存是一块你可以自由读写的区域，你可以在这里存放pcb，bitmap以及其他可以辅助你进行调度的数据结构

2. 多个cpu

每个cpu有两种操作：

1. 运行一个任务，前提是这个任务所需要的资源都已经被放到资源内存里，且该任务已经获得了该资源的权限
2. 空闲，什么也不做

3. 时间片

我们模拟一个一直在运行的时间序列，每个时间片都会调用你的调度算法，获得当前cpu的动作。

cpu每个时间片可以执行1时间片的任务。

说明

1. 如果要运行一个任务，它所需要的资源不能被别的正在运行的任务使用，如果已经有正在执行的任务占用这个资源，那么新的任务不能获得这个资源的使用权，也就不能执行这个任务
2. 任务到达的时间是不一定的。每个时间片，系统都会调用你写的调度算法。算法的输入是在当前时间片到达的任务。（当然到来任务所需要的资源很可能还不能使用，或者没有一个cpu可以用来处理这个任务，那么你能只能挂起这个任务，直到在某一个时间片这个资源已经准备就绪了）
3. 一个任务在同一时刻只能由一个cpu完成
4. 一开始的时候，所有需要的资源都准备就绪。
5. 任务保证会按照顺序到达，即如果 $x < y$ ，那么任务x的到达时间一定小于等于任务y的到达时间

评比

1. 正确性：任务的执行必须符合条件，如果不符合条件，那么就会判定程序出错
2. 满意度：满意度是我们衡量一个调度策略的重要指标，每个任务都有条等待曲线。横坐标是时间，表示从任务到达最终完成花费的时间；纵坐标是得分，表示在特定等待时间下该任务的不满意得分。显然，得分越低越好。（下面的表是一个例子，未来会进行调整，不过原则不会变：如果一个任务等待太久，那么不满意度会增长的非常快）

横坐标x:时间	纵坐标y:不满意
$x < 5$	$y = 1$
$x \geq 5 \ \&\& \ x < 15$	$y = 2x + 1$
$x \geq 15$	$y = 2^{(x-11)}$

3. 内存访问次数：很显然，越少越好
4. cpu环境切换次数：如果cpu在当前时刻执行的任务和cpu在上一时刻执行的任务不同，那么就需要切换执行环境，很显然，这个次数越少越好

工程文件

请重点关注带*的类

测试类如main/Main, main/Test result包 等内容无需关心

在实现类(work包下)提供了main函数，可以供大家进行运行和调试以及看到结果

bottom

BottomMonitor: 底层监视器，用于统计以及正确性检查

BottomService: 底层接口的具体实现

Constant * : 一些常量信息，主要关注一下自由内存的大小

Task * : 任务类，可以直接读取任务的id，需要执行时间和所需资源

TaskState: 底层的任务状态记录

main

Main: 执行测试的主函数入口

Schedule * : 调度器类

Test: 测试用例

testFile

该包用于存放测试文件，每个测试文件中记录着一种任务队列，具体形式可参考该包下的testSample.txt

work

S161250xxx *: 你需要实现的类，该类必须继承main.Schedule，可以使用父类已经实现的方法

Sgreedy: 我写的一个简单的样例，如果你觉得看了我的代码会影响你的思路可以不看23333

result

该包用于存放测试结果文件

提交及加分

提交内容

1. 提交work包下的实现类，类名为S+你的学号，如S161250001.java
2. 提交说明文档，记录你的实现思路，可以写出你设计的比较巧妙的部分

加分

该加分会直接加在你的操作系统实验得分中

1. 所有提交代码，且通过查重检测和正确性检测的同学，可以获得基础加分（预计6分起步）
2. 在性能评比中表现出色的同学，可以获得额外加分（封顶10分）

Q&A

关于实验的问题：

实验及底层代码是我写的，所以有疏漏的地方欢迎指出来，我会及时调整

关于实验的问题，可以在QQ群提问或写邮件咨询

邮箱：151250145@smail.nju.edu.cn

工程代码可以在github上下载：<https://github.com/wshwbluebird/PMPlus>