

提交时间 12 月 26 日

作业递交方式:

1: 上课时提交文本文档 (.doc, .pdf)

需要给出 1: 问题的描述; 2: 概要设计; 3: 详细介绍避免死锁的策略; 4: 实验结果; 5: 分析

2: 网上提交实现代码到 [homeworkecnu@163.com](mailto:homeworkecnu@163.com)

需要给出: 源代码 (需要注释关键代码), makefile 等, 需打包

项目描述:

1: 为了加深大家对并发程序的理解, 本次项目将围绕著名的哲学家问题进行相应的开发。细节请查看

[http://en.wikipedia.org/wiki/Dining\\_philosophers\\_problem](http://en.wikipedia.org/wiki/Dining_philosophers_problem)

2: 编程语言必须使用 IEEE Posix Thread 库与 C 语言实现。需要采用多线程实现。哲学家的刀叉由于是共享变量, 所以需要作为临界区用互斥变量、信号量等机制实现互斥。本实验规定采用信号量实现互斥。

3: 本次哲学家问题主要想考察大家对于死锁形成机制的理解。希望大家实现的一个能够导致死锁的 primitive 哲学家问题。程序的输入可以指定哲学家个数。每个哲学家分为 3 个状态 {thinking, trying, eating}, 分别代表思考, 尝试获得叉子, 吃面三个状态。规定 thinking 于 eating 的时间在区间 [0.01, 0.1] 秒之间。所谓死锁就是大家都处于 trying 的状态。请给出相应的状态输出序列。

4: 为了避免死锁, 有许多的解决方法。请你列举 2 种方法, 并用程序实现, 输出状态序列。

测试方法 (输入格式)

有三种模式

-normal 存在死锁的模式, -method1 第一种解决方法; -method2 第二种解决方法;

哲学家的个数可以指定

-n 哲学家个数

例如

./philosophoer -normal -n 哲学家个数

./philosophoer -method1 -n 哲学家个数

./philosophoer -method2 -n 哲学家个数