

并行计算

第一次课作业

姓名 刘恒星 学号 2022229044

请简要回答以下问题：

1、线程与进程的区别是什么？

1. 线程是进程上下文的执行代码序列，是调度执行的实体，而进程是资源分配的实体
2. 进程拥有自己的系统资源，线程只拥有不可缺少的少量资源，但是可以共享进程的资源
3. 进程创建和撤销的时候，系统都要分配和回收资源，而线程的切换只需要保存少量的寄存器内容，无需存储管理器的操作，线程的切换，同步，通信都不需要操作系统内核干预

2、多线程程序运行期，在什么情况下会出现竞态条件？

当两个或者多个线程同一时刻试图访问共享内存，或者读写某些数据，而最后的结果取决于执行的精准时序，会出现竞态条件

3、Busy-waiting 是否能从根本上解决竞态条件问题？为什么？

可以。他可以限制访问临界区的线程只有一个，不会让多个线程同时进入临界区

4、根据第 62/63 页 PPT 的代码，用文字描述当 i 从 1 到 4 时两个线程加锁解锁的动作时间序列。

当 i=1 的时候

1. 若 thread1 先执行加锁，此时 $i \% 3 \neq 0$ ，输出 thread1: 1，并解锁
 2. 若 thread2 先执行加锁，就会调用 `pthread_cond_wait`，解锁并挂起等到条件触发
- 所以输出 thread1: 1

当 i=2 的时候

1. 若 thread1 先执行加锁，此时 $i \% 3 \neq 0$ ，输出 thread1: 2，并解锁
 2. 若 thread2 先执行加锁，此时仍在挂起等到条件触发
- 所以输出 thread1: 2

当 i=3 的时候

1. 若 thread1 先执行加锁，此时 $i \% 3 == 0$ ，调用 `pthread_cond_signal`
 2. 若 thread2 先执行加锁，会等待 thread1 的条件触发，等到之后，释放线程，输出 thread2:3，并解锁
- 所以输出 thread2: 3

当 i=4 的时候情况与 i=1 相似，所以输出 thread1: 4