并行计算

第三次课作业

姓名 陆子毅 学号 3022206045

请简要回答以下问题：

1、线程与进程的区别是什么？

线程是系统资源分配的基本单位，进程是CPU调度的基本单位。

线程不独立拥有资源，但是可以访问线程隶属的进程的资源，进程拥有由CPU调度的系统资源。

创建一个进程所耗费的系统资源大于创建一个线程耗费的系统资源，多线程更加轻量化。

线程之间共享资源，共享I/O设备，共享打开文件。而进程之间相对独立，无法互相访问。

线程之间的切换不会引起进程的切换，避免了进程切换带来的巨额系统开销。

一个进程可以有多个线程并行，从而提高进程的运算速度。

进程被封锁时，进程所提供的服务就会被终止，如果在一个进程下设置多个线程，当一个线程被封锁时，另一个线程仍然能够正常运行，仍然能够正常提供服务。

2、多线程程序运行期，在什么情况下会出现竞态条件？

竞态条件是两个线程同时读写共享数据，会导致最后的结果不确定（取决于线程的运行顺序），与操作系统所讲的进程管理相似。

一般是多个线程同时读写共享数据时出现竞态条件。

3、Busy-waiting是否能从根本上解决竞态条件问题？为什么？

不能，Busy-waiting所使用的while（Lock）与Lock=True并不是原子操作，当进程被中断后 再次运行时，依然可能导致两个线程同时访问临界资源，出现竞态条件。

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。

如果thread1先执行到加锁动作

i=1：thread1正常运行，输出thread1:1，thread2位于等待状态

thread1解锁，thread1 sleep，thread2得到锁。

thread2执行到pthread\_cond\_wait，等待条件改变并解锁。

i=2：thread1得到锁，输出thread1:2，thread2被挂起。

thread1解锁，thread1 sleep，thread2挂起。

i=3：thread1判断后改变cond条件，thread2被唤醒。

thread1解锁，thread2得到锁，thread2输出thread2:3，thread1 sleep。thread2解锁，thread2 sleep。（由于thread1先进入睡眠，所以先醒先抢到锁）

i=4：thread1得到锁，输出thread1:4，thread2位于等待状态。

thread1解锁，thread1 sleep，thread2得到锁。

thread2执行到pthread\_cond\_wait，等待条件改变并解锁。

后续重复上面的过程。Thread2拿到锁时如果i不是3的倍数就会被挂起，直到thread1检测到i变成3的倍数以后主动唤醒thread2。过程如下图（左侧thread1先锁，右侧thread2先）

图片包含 雷达图

描述已自动生成