作业报告

生产者-消费者问题是一个经典的并发编程问题，它涉及到多个线程访问共享数据的问题。程序使用python语言实现了一个简单的生产者-消费者应用程序，其中一个线程代表生产者，另外两个线程代表消费者。生产者生成数据并将其添加到共享队列中，消费者从队列中获取数据并进行处理。为了确保线程安全，程序使用条件变量和锁来保护队列的访问。

实现细节

全局共享数据

应用程序的全局共享数据是一个简单的队列，它用于存储生产者生成的数据。队列使用python中的列表结构：  


线程同步技术

条件变量

在应用程序中，消费者只有在队列非空时才会尝试获取数据，当队列为空时消费者将会等待生产者线程的通知，以便在新数据添加到队列时提醒，使用python中条件变量实现。



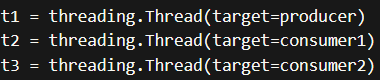
生产者在添加数据到队列中时，会获取条件变量并发送通知给所有等待的消费者线程。

锁：  
队列是一个共享数据结构，多个线程同时访问它可能会导致竞争条件。为了避免这种情况的发生，使用锁来保护队列的访问。

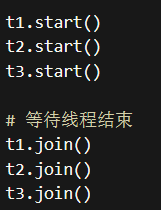


在生产者添加新数据到队列时，会获取锁并修改队列的内容。在消费者获取数据时，也会获取锁并修改队列的内容。

程序使用了三个线程，producer函数，comsumer1函数，comcumer2函数



程序运行时，启动三个线程并等待他们结束



程序源代码：



