**第二次作业: producer-customer problem**

**1511502 吴文竹**

1. **输出结果截图：**

输出当前生产者、消费者的线程id及其生产消费情况。



1. **实现过程：**
2. 设定生产或消费总数量MAX=50，缓冲区大小buffer\_MAX=10；
3. 设置了互斥量the\_mutex和条件变量condc、condp；
4. 生产者、消费者均通过the\_mutex互斥使用缓冲区。设置buffer变量反应当前缓冲区已使用情况：

对于生产者：当buffer=buffer\_MAX时，调用pthread\_cond\_wait阻塞当前线程并解锁互斥量。

对于消费者：当buffer=0时，调用pthread\_cond\_wait阻塞当前线程并解锁互斥量。

消费者数量设置为2、生产者为3，因各有多个，故在改变buffer值后均使用pthread\_cond\_broadcast来唤醒其他多个线程。

1. 生产者、消费者生产总数达到50后即退出各线程。
2. **其他说明：**
3. 刚开始未添加sleep函数，运行效果类似顺序运行，如：当consumer1整个运行完后才会运行consumer2；添加sleep之后显示出了多线程的效果。
4. 最终的结果中有编号超过50的产品，个人思考原因为：有的线程在total\_con,total\_pro未到50前完成判断并进入阻塞状态，等该线程再被唤醒时已不需再进行小于50的判断，将会继续执行下面的操作，故存在编号大于50的产品。

**※完整代码在hw02.c中。**