**操作系统课程设计实验报告**

实验名称： 生产者消费者问题

姓名/学号： 张惟振/1120170117

1. **实验目的**
2. 学习掌握Windows和Linux操作系统中进程之间的通信
3. 理解并掌握使用信号量机制进行多进程之间互斥访问共享内存区域的控制
4. 学习进程的创建和控制，共享内存区域的创建、使用和删除，信号量的创建使用和删除
5. **实验内容**

（1）一个大小为3的缓冲区，初始为空

（2）2个生产者

随机等待一段时间，往缓冲区添加数据

若缓冲区已满，等待消费者取走数据后再添加

重复6次

（3）3个消费者

随机等待一段时间，从缓冲区读取数据

若缓冲区为空，等待生产者添加数据后再读取

重复4次

说明：

显示每次添加和读取数据的时间和缓冲区里的数据

生产者和消费者都用进程模拟

1. **实验环境**

1、软件环境

Windows10操作系统、Ubuntu 18.04.3 LTS

2、硬件环境

Intel® Core™ i5-7200U CPU @ 2.50GHz×4

1. **程序设计与实现**

1、实验思路：设置三个信号量：mutex，full，empty。创建信号量 mutex 作为互斥信号量，用于控制各个生产者和消费者进程互斥访问缓冲区，其初值设为1；信号量full用于记录当前缓冲区满的位置的个数，初值为0；信号量empty用于记录当前缓冲区中空的位置的个数，初值设为3。

2、Windows系统实现中的API

（1）CreateFileMapping（）创建一个文件映射对象

实验使用如下：

（2）MapViewOfFile（）将一个文件映射对象映射到当前程序地址空间

实验使用如下：



（3）OpenFileMapping（）打开一个已经存在的文件映射对象

实验使用如下：



（4）UnmapViewOfFile（）停止当前程序的一个文件映射对象的映射

实验使用如下：



（5）CreateSemaphore（）创建一个信号量

 实验使用如下：

（6）OpenSemaphore（）打开一个已有的信号量

 实验使用如下：

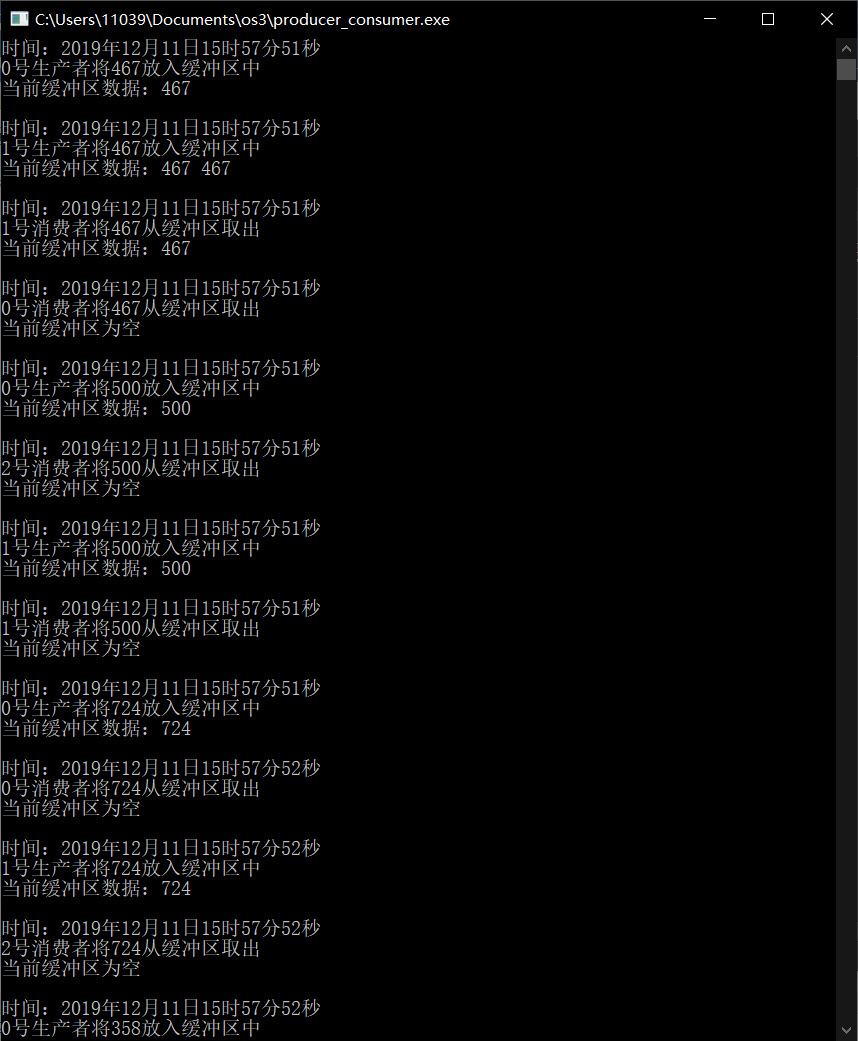
（7）ReleaseSemaphore（）给信号量增加指定值

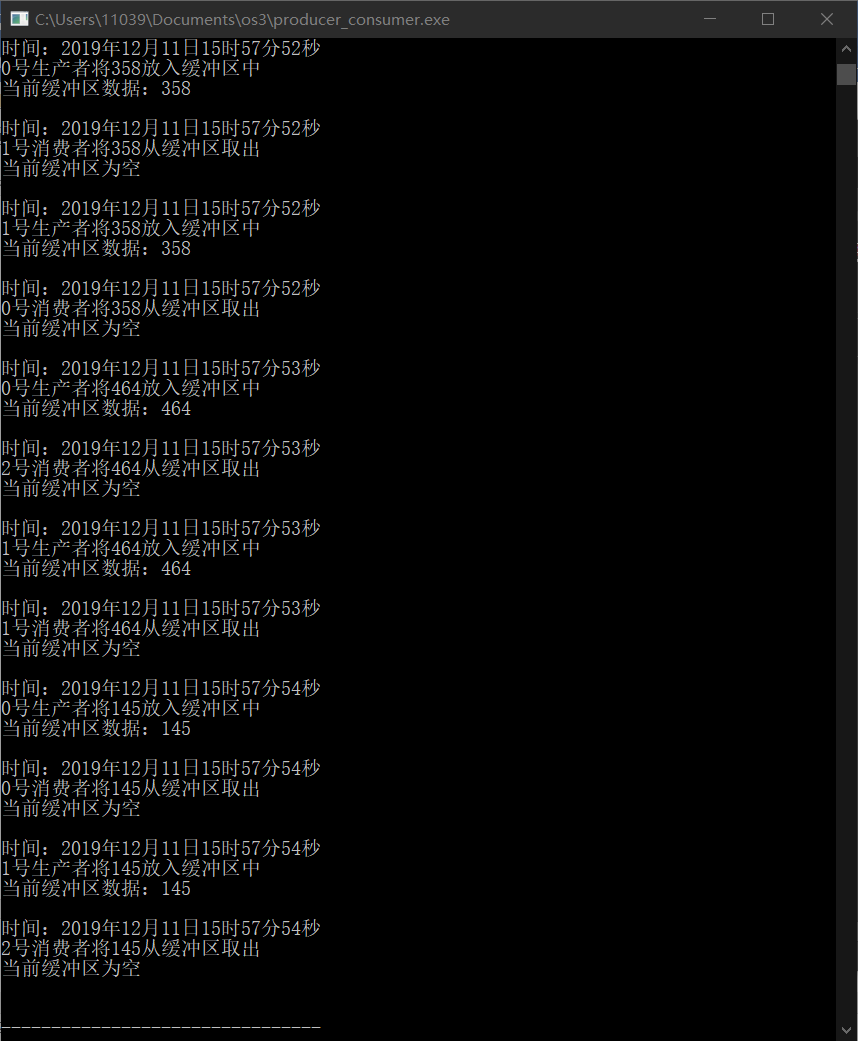
 实验使用如下：

（8）CloseHandle（）关闭已打开的句柄

 实验使用如下：

3、Windows下实验结果



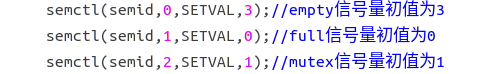


1. Linux系统实现中的API
2. semget（） 创建一个新的信号量集

实验使用如下：

2019-12-11 23-46-22 的屏幕截图（2）semctl（）执行在信号量集上的控制操作

实验使用如下：



1. shmget（）创建共享内存对象

实验使用如下：

2019-12-11 23-51-07 的屏幕截图

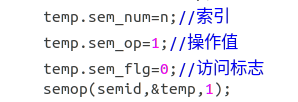
（5）shmat（）把共享内存区对象映射到调用进程的地址空间

实验使用如下：

2019-12-11 23-52-44 的屏幕截图

（6）semop（）改变信号量的值

实验使用如下：



（7）shmctl（）对共享内存进行控制

实验使用如下：

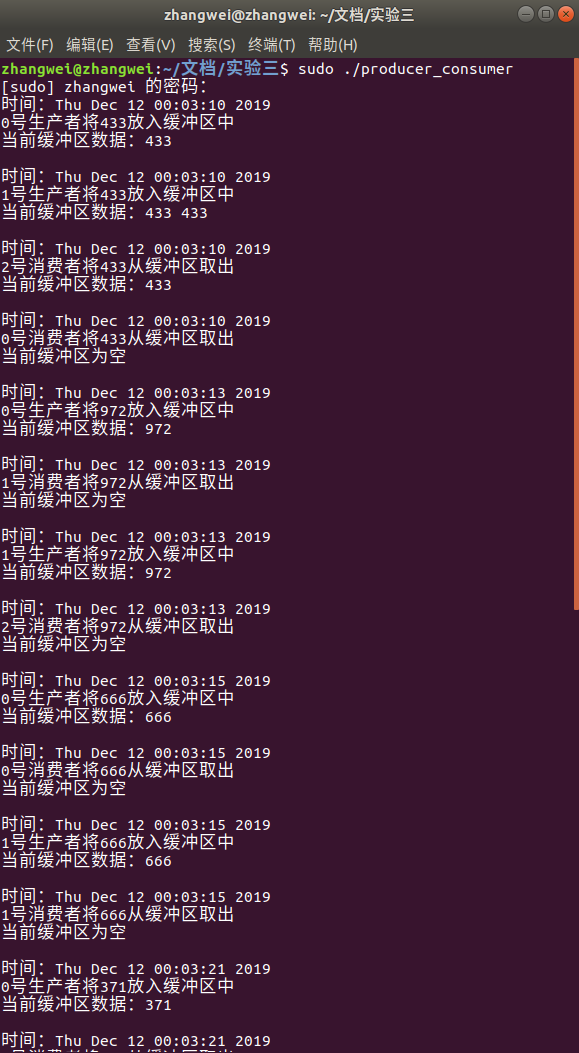
2019-12-11 23-59-16 的屏幕截图

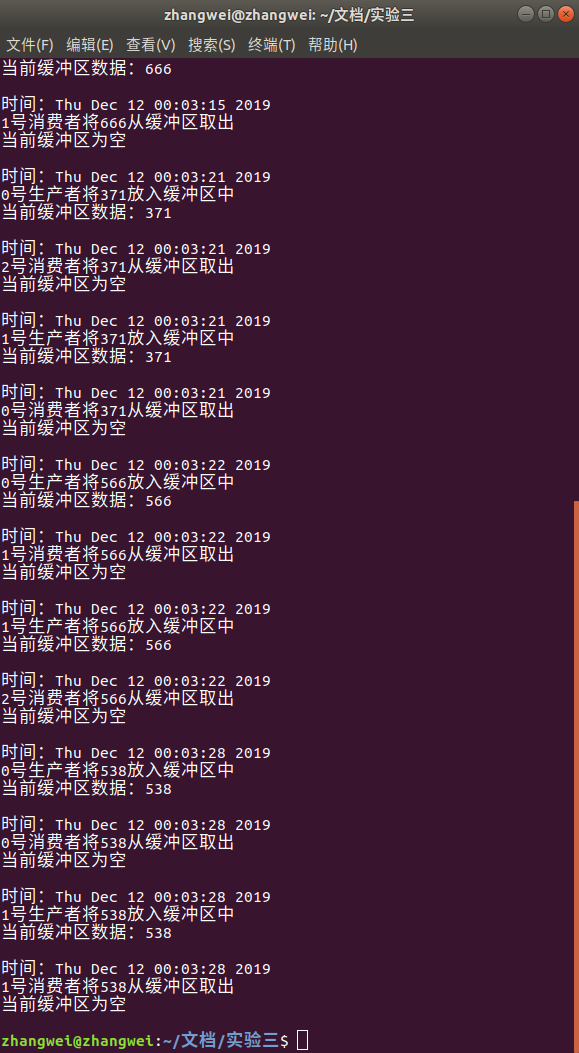
1. shmdt（）断开共享内存连接

实验使用如下：

2019-12-12 00-02-08 的屏幕截图

5、Linux下实验结果





1. **实验收获与体会**

通过这次实验，我对windows 系统和Linux系统中进程的创建与控制有了更加深刻的了解。生产者消费者这个问题在操作系统原理课程中曾经学习过，该问题主要需要解决信号量的创建与控制，PV操作及PV操作的顺序问题。在本实验中还需要了解共享缓冲区的创建方法，各种API的调用，对他们的参数也需要更加了解。

总的来说，在Windows系统中实现生产者消费者问题比在Linux系统中实现要困难一些，在Windows系统中需要创建一个文件映射内核对象，然后再把文件映射到内存地址空间中，然后才可以操作控制共享内存区。而在Linux中使用共享内存区则要更为方便一些。此外，有一点需要注意的是，信号量与共享内存区创建和使用之后必须要进行删除操作，否则将会遗留共享内存区域和共享信号量。

生产者消费者问题是进程控制通信中的一个比较基本的问题，其中所包含的缓冲区创建，信号量机制等知识对于以后实现更加完善的进程控制有着非常重要的作用，需要好好掌握。