操作系统实验指导书

同步问题

# 教师奠定工程基础

## 生产者消费者问题(工程基础：producer\_consumer)

### 环形队列的实现

在单线程环境下完成对Buffer类的单元测试

### 用互斥量保护数据结构

#### 不足

当缓冲区已空时，get就会白跑一趟；当缓冲区已满时，put就会白跑一趟。

#### 实现互斥量(mutex)的三种选择

有Windows临界区、Windows互斥量、Windows信号量三种方式。前两种都可视为一种针对特定情况的优化。建议使用Windows信号量来实现互斥量，以便强化对互斥量本质的认识。

* Windows临界区：

类型CRITICAL\_SECTION

加锁EnterCriticalSection

解锁LeaveCriticalSection

创建InitializeCriticalSection

销毁DeleteCriticalSection

* Windows互斥量

类型HANDLE

加锁WaitForSingleObject

解锁ReleaseMutex

创建CreateMutext

销毁CloseHandle

* Windows信号量

类型HANDLE

加锁WaitForSingleObject

解锁ReleaseSemaphore

创建CreateSemaphore

销毁CloseHandle

### 用信号量实现同步

两个信号量，一个代表“空格子”资源，一个代表“满格子”资源。

# 学生的工程目标

看懂教师编写的例子程序（一个生产者，一个消费者，一个缓冲区），然后在生产者-消费者程序的基础上，实现下述功能：

1. 三个线程，两个缓冲区

第一个线程往缓冲区a中put；第二个线程从缓冲区a中get，然后put到缓冲区b中；第三个线程从缓冲区b中get。第一个线程相当于纯生产者，第三个线程相当于纯消费者,第二个线程相当于既是生产者又是消费者。

注意：创建信号量时，最后一个参数写NULL，不要给信号起名字。起了名字你可能用的就是系统中其他进程或者你本进程创建的同名信号量了，会产生干扰。

1. 用C++类封装信号量相关的API函数，实现一个名为Semaphore的类，提供两个成员函数：p()和v()；

Semaphore s(8);

s.p();

s.v();

1. 用两个锁分别保护head和tail