实验报告

实验名称：**生产者和消费者问题**

实验时间：2018/4/6

实验人员：李子强（姓名）11510352（学号）15（年级）

实验目的：掌握基本的同步互斥算法，理解生产者和消费者模型，理解读者写者问题。了解多线程的并发执行机制，线程间的同步和互斥。

实验环境：Linux

实验步骤：

1.读懂源程序

2.编辑修改源程序

实验陈述：

1、基础知识：

常用的几个API函数：1.pthread_create 2.pthread_join 3. pthread_mutex_lock 4.pthread_cond_wait 5.pthread_cond_signal 6. pthread_mutex_unlock

这些函数的作用：1. **创建线程**

2. **等待一个线程的结束,线程间同步的操作**

3. **线程调用该函数让互斥锁上锁**

4. **条件变量是利用线程间共享的全局变量进行同步的一种机制，主要包括两个动作：一个线程等待"条件变量的条件成立"而挂起；另一个线程使"条件成立"（给出条件成立信号）。**

5. **发送一个信号给另外一个正在处于阻塞等待状态的线程,使其脱离阻塞状态,继续执行.如果没有线程处在阻塞等待状态**

6. **解除锁定 mutex 所指向的互斥锁的函数**

2、生产者/消费者问题

消费者从缓冲区中读到的数据都是由同一个生产者生产的吗？不一定，生产者可以有多个。

消费者的读取操作和生产者的写入操作有什么先后关系 写入操作一定先于读取操作进行。

简述程序运行的结果 创建2个进程分别是生产者和消费者，3个互斥锁保证读写安全，生产者每次产生一个随机字母放入队列，消费者每次取出一个队列中的字母显示出来。

此程序采用了什么队列？有何特点？环形队列，实现FIFO，而且循环利用全部空间，不需要移动元素，每次只移动指针。

简述此程序的互斥机制？ 3个互斥锁保证读写安全，一个保证写入读取的时候另一个进程不在操作，一个保证队列空时不读，一个保证队列满时不写。

3、 读写者问题

简述此程序用了信号量的那些接口？ sem_open, sem_close, sem_unlink, sem_post,

sem_wait

这些接口的作用分别是什么？ 创建和初始化有名称信号量，关闭信号量，清除有名信号量，对信号量 post 操作，对信号量 wait 操作，

读者可以同时读吗？在代码中如何体现 可以，

```
void *reader(int *buffer) {
    sem_wait(rc);
    if (0 == readcount) {
        sem_wait(db);
        readcount++;
    }
    sem_post(rc);
    printf("\nReader Inside..%d\n", *buffer);
    sem_wait(rc);
    readcount--;
    if (0 == readcount) {
        sem_post(db);
    }
    sem_post(rc);
}
```

写者可以同时写吗？在代码中如何体现 不可以，

```
void *writer(int *buffer) {
    sem_wait(db);
    *buffer += 1;
    printf("write ::%d\n", *buffer);
    sem_post(db);
}
```

读者并发读的表现是什么？ 多次读出同样的 buffer。

写者写操作之后可以没有读者读就执行下一个写者的写操作吗？为什么 可以，因为是覆盖操作，不存在空间问题。

实验总结：

学习信号量的使用。