实验2 生产者-消费者问题

班级： 07112002 学号： 1120200822 姓名： 郑子帆

一、实验目的

在Windows系统下独立设计并编写C++程序来实现生产者-消费者问题。通过创建几个流程来模拟生产者和消费者，并使用共享内存实现生产者和消费者之间的共享缓冲区；设置信号量完成同步、互斥过程。通过此实验加深对进程通信的理解。

二、实验内容

1. 创建一个包含6个缓冲区的缓冲池，初始为空，每个缓冲区能存放一个长度为10个字符的字符串。
2. 创建2个生产者进程，要求每个生产者进程：

随机等待一段时间后，向缓冲区中添加数据；

若缓冲区已满，则等待消费者取走数据后再继续添加数据；

重复添加数据12次。

1. 创建3个消费者进程，要求每个消费者进程：

随机等待一段时间后，从缓冲区中读取数据；

若缓冲区为空，则等待生产者添加数据后再继续读取数据；

重复读取数据8次。

1. 输出显示每次添加或读取数据的时间，以及缓冲区的映像。

三、程序设计与实现

按照上课所讲的内容，生产者-消费者问题通过设置信号量来完成生产数据，放入缓冲区以及从缓冲区取数据并消耗。设置同步信号量empty表示为空的缓冲区的个数，初值为6，最大值为6；full表示非空的缓冲区个数，初值为0，最大值为6。通过课堂的学习，得知P操作同步信号量要在生产者、消费者子进程的伪代码表示如下。

producer(){

    while(1){

        生产一个数据;

        P(empty);

        P(mutex);

        把数据放入缓冲区

        V(mutex);

        V(full);

    }

}

consumer(){

    while(1){

        P(full);

        P(mutex);

        从缓冲区取数据

        V(mutex)

        V(empty)

        消耗数据

    }

}

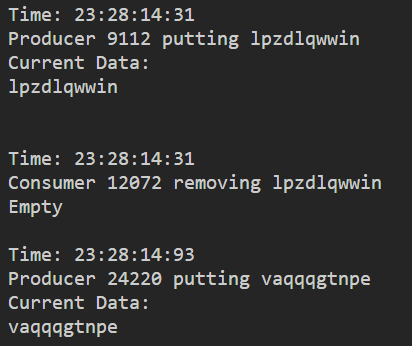
本实验分成三个程序，m.cpp、p.cpp和c.cpp，分别表示make\_processing（创建子进程）、producer和consumer。代码详见附件源代码。

这里用到了一些函数如下：

1. CreateFileMapping()用于创建一个文件映射内核对象，返回一个文件映像的句柄。
2. MapViewOfFile()用于将一个文件映射对象映射到当前程序地址空间。
3. UnmapViewOfFile()用于停止当前程序的一个内存映射。
4. OpenFileMapping()用于打开一个已经存在的文件映射对象。
5. CreateSemaphore()用于创建一个同步信号量。
6. CreateMutex()用于创建一个互斥信号量。

具体思路为在创建进程的程序中完成对于2个生产者子进程和3个消费者子进程的创建，并且创建了文件映射作为共享内存文件，以及创建了同步信号量和互斥信号量。在生产者进程程序中，设置了随机种子，在随机时间唤醒进程并生产随机字符串进行P、V操作，放入缓冲区。同理，消费者进程程序的原理与生产者类似，具体区别见上面伪代码。

四、实验结果及分析

上图为运行结果的部分截图。

下面展示了部分运行结果的文本表示：

Producer 1 created.

Producer 2 created.

Consume 1 created.

Time 232331600

Producer 20396 putting wbodhwvhzt

Current Data

wbodhwvhzt

Consume 2 created.

Consume 3 created.

Time 232331724

Producer 20396 putting gvtviofbub

Current Data

wbodhwvhzt

gvtviofbub

简单分析上图，两个生产者进程先被创建，然后三个消费者进程被创建，在第2、3个消费者进程被创建之前，一个生产者进程向缓冲区放了一个随机生成的字符串；再所有进程均被创建之后，同一个生产者进程又向缓冲区放了一个随机生成的字符串，当前缓冲池共有两个区域被占据。

Time: 23:47:57:19

Producer 24584 putting cxipmvhtby

Current Data:

cxipmvhtby

Time: 23:47:57:19

Consumer 23124 removing cxipmvhtby

Empty

再比如这一段运行结果，一个生产者进程放了一个数据，之后有一个消费者进程取走了这个数据，此时缓冲池为空，所以会返回Empty。

由于运行结果太长，故将完整的运行结果保存在压缩包中的运行结果全.docx文件中。

五、实验收获与体会

这次实验我先对上课的内容做了一个总结，从理论部分明白了生产者-消费者的实现原理。然后通过实验书和网上的资料学习了c++有关于进程创建、信号量创建的API函数接口，并完成了程序编写。通过这次实验我加深对进程通信的理解，并且提高了利用C++代码编写的能力。

附录：程序清单及说明

m.cpp：创建进程源程序

p.cpp：生产者进程操作源程序

c.cpp：消费者进程操作源程序

运行结果全.docx：一次完整的运行结果

运行结果全.jpg：一次完整的运行结果截图

1120200822-郑子帆-实验2 运行过程录屏.mp4：实验要求的两次演示录屏（Full和Empty）