实验二 生产者消费者

班级： 07812101 学号： 1820211062 姓名： 洪子翔

一、实验目的

独立设计并编写程序实现生产者-消费者问题，加深对进程通信的理解。

二、实验内容

1. 在Windows系统下实现生产者-消费者问题。

2. 具体要求如下：

(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。

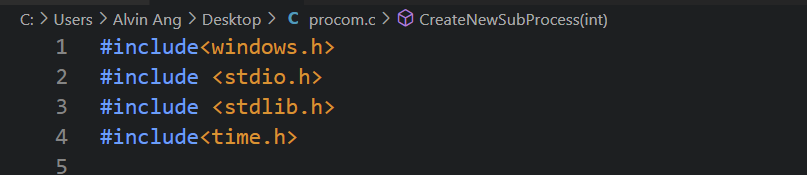
(3) 创建3个消费者进程，要求每个消费者进程：

* 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。

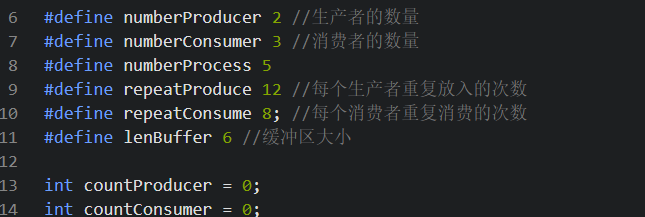
(4) 输出显示每次添加或读取数据的时间，以及缓冲区的映像。

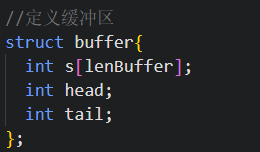
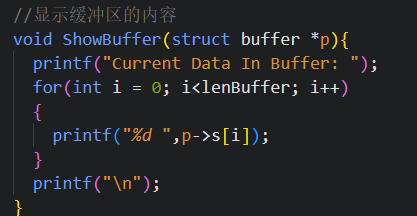
三、程序设计与实现

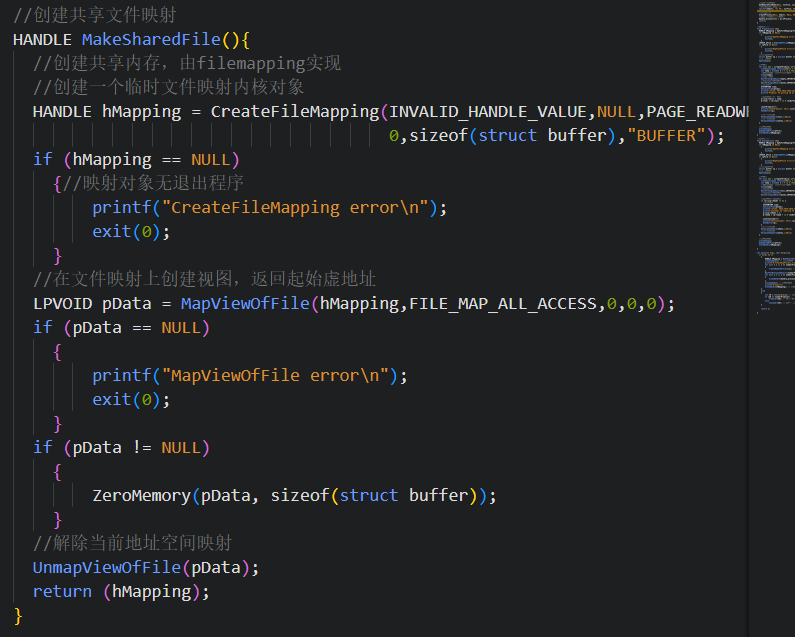
实验环境：Windows11，Visual Studio Code,C语言

首先，导入必要的库，以便利用Windows API实现多进程、共享内存和信号量机制。 Windows API的具体使用方法已在代码注释中提供详细解释。

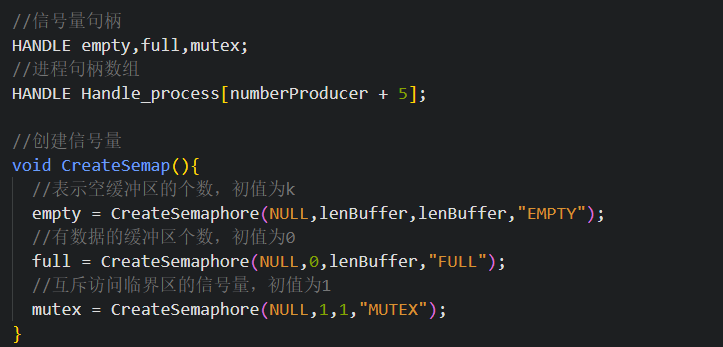
在代码中，我们提前设置了生产者的数量、消费者的数量、进程的数量以及特定操作的次数进行初始化。这种初始化的方式使得我们可以轻松地更改这些数值，以观察其对程序行为的影响。此外，我们还引入了两个全局变量 countProducer 和 countConsumer，用于追踪每个进程重复执行放入或消费操作的次数，以确保每个进程都成功完成了其任务。

以下是这些进程共享一个缓冲区,以下是程序的主要组成部分：

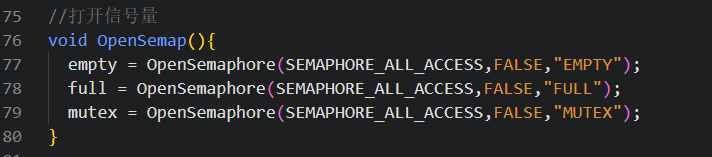
**struct buffer**：缓冲区数据结构，用于存储字符串。以及显示缓冲区函数

**MakeSharedFile**：创建共享文件映射并初始化缓冲区。

**信号量**

**CreateSemap**：创建信号量，包括empty、full和mutex，用于同步生产者 和消费者之间的操作。

**OpenSemap**：打开现有的信号量。



**CloseSemap**：关闭信号量。

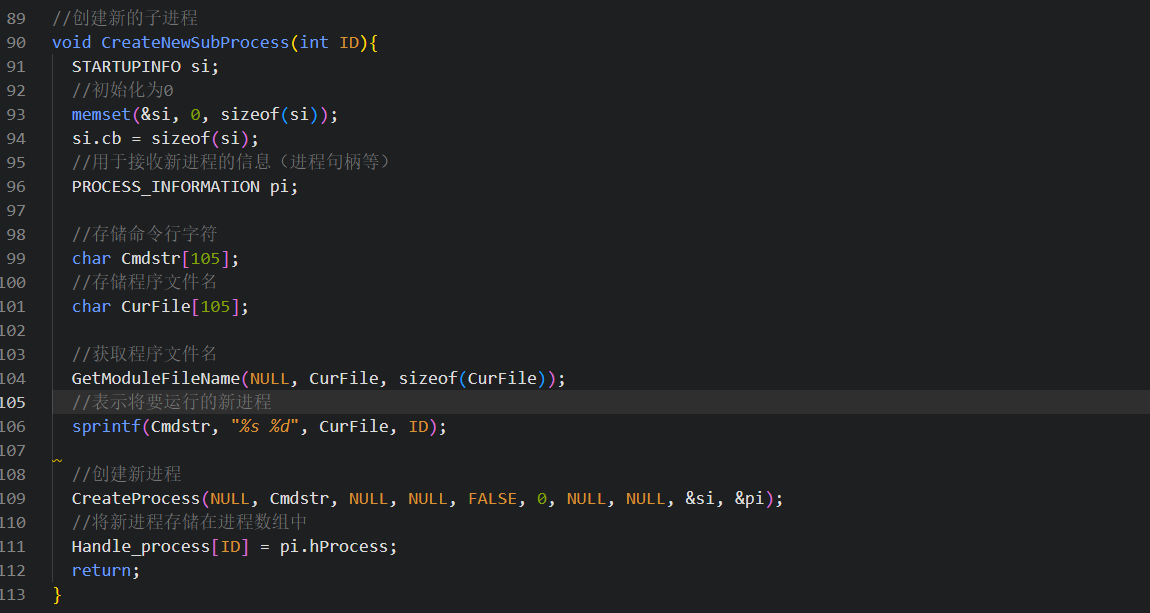


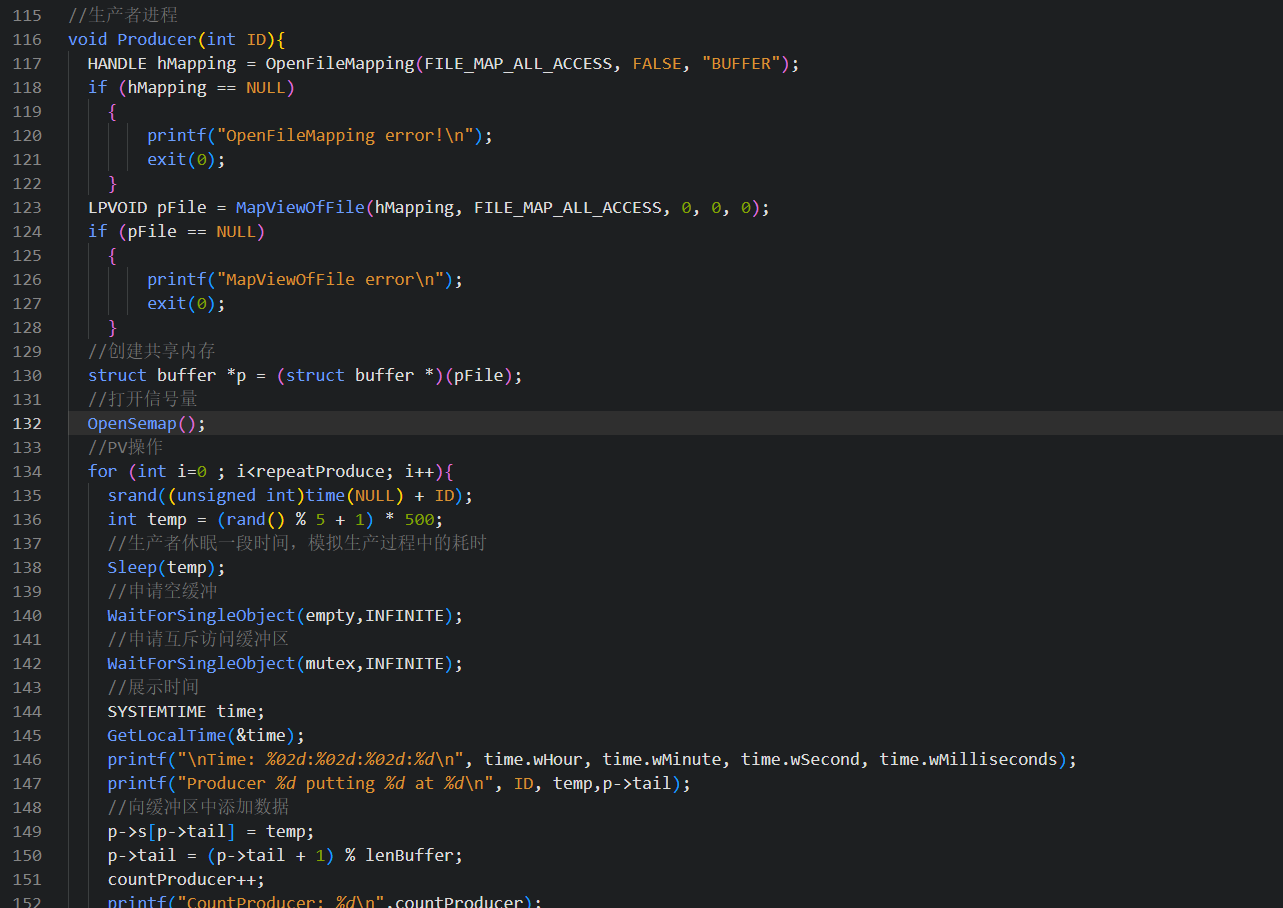
这里使用信号量来实现生产者和消费者之间的同步。具体来说，使用了三个信号量：

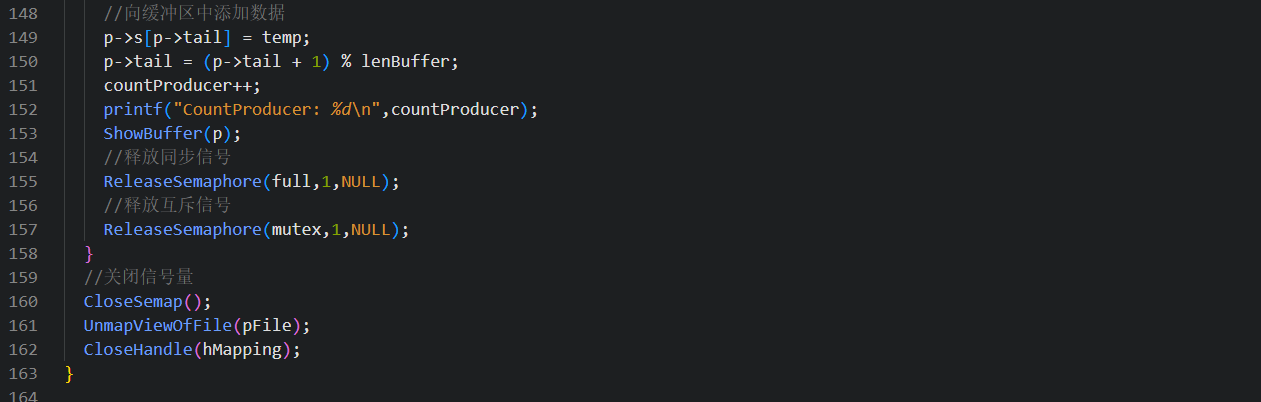
* empty：表示空缓冲区的数量，初值为缓冲区大小（6）。
* full：表示有数据的缓冲区的数量，初值为0。
* mutex：用于互斥访问临界区的信号量，初值为1。

**进程**

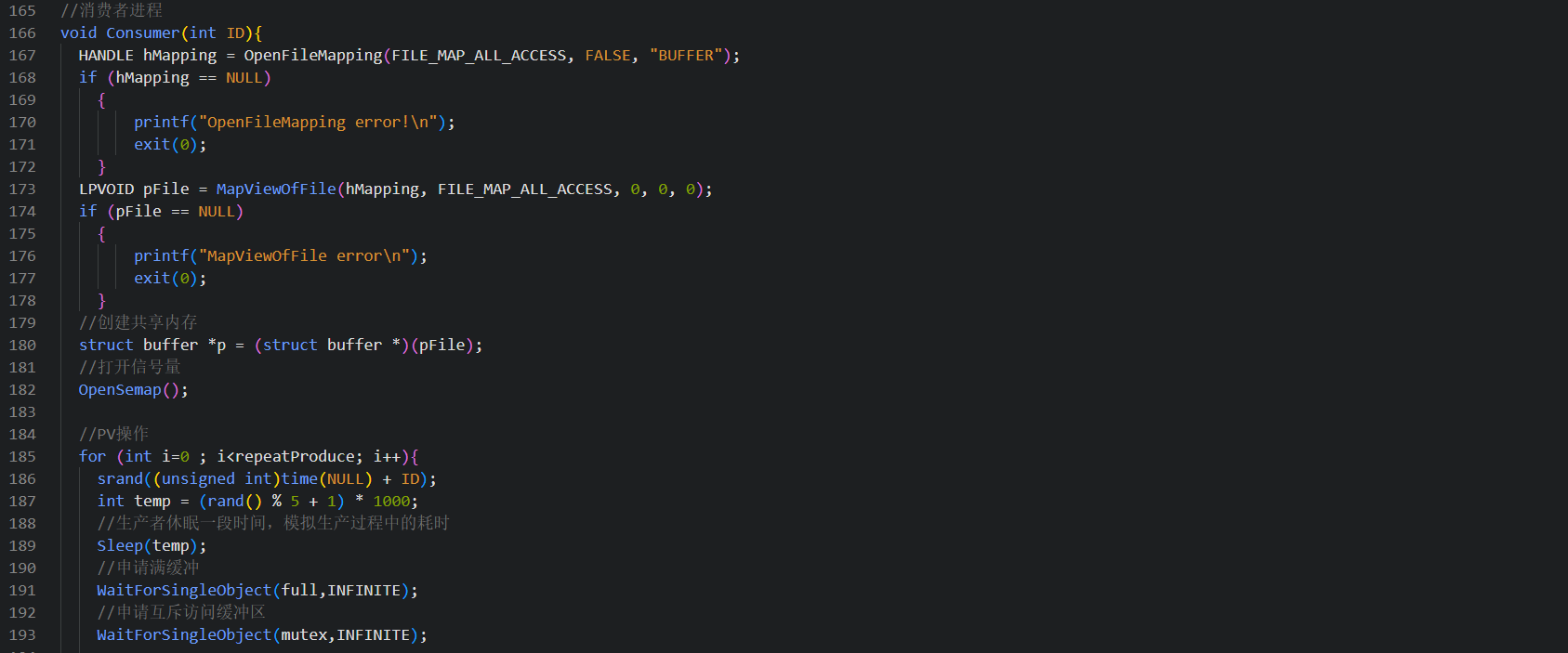
**CreateNewSubProcess**：创建新的生产者或消费者进程。



**Producer**：生产者进程的主要函数，生产并放置数据到缓冲区。



**Consumer**：消费者进程的主要函数，从缓冲区中取走数据。





生产者：

生产者进程在随机等待一段时间后，将字符串添加到缓冲区中。

如果缓冲区已满，生产者等待，直到消费者取走数据后继续添加。

每个生产者重复执行12次。

消费者：

消费者进程在随机等待一段时间后，从缓冲区中读取数据。

如果缓冲区为空，消费者等待，直到生产者添加数据后继续读取。

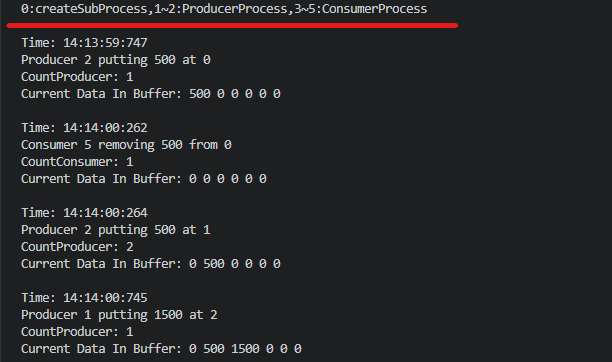
每个消费者重复执行8次。

**Main函数**

 首先，main函数检查是否有命令行参数传递给程序，如果没有，它会创建共享文件映射和所需的信号量，然后循环创建生产者和消费者进程，最后等待所有进程执行完毕，并关闭相关资源。如果有命令行参数传递，main函数会解析参数来确定是启动生产者进程还是消费者进程，然后调用相应的函数来执行相应的操作。通过这种方式，main函数有效地管理了生产者和消费者的并发操作，确保它们按照实验要求进行交互和同步。

四、实验结果及分析

运行开始

 运行结果，可以看到每个进程都完成它们的要求，运行成功



五、实验收获与体会

这个实验的大部分时间都投入在学习 Windows API 上，同时在思考 PV 操作的时机以及输出的形式上也经过了不少思考。此外，还需要确定每个进程的重复操作次数是否达到要求，并记录了每个操作的时间。虽然学习 Windows API 花费了相当多的时间，但也积累了许多宝贵的经验和知识。

附录：程序清单及说明

procom.c-源代码

procom.exe-可执行文件