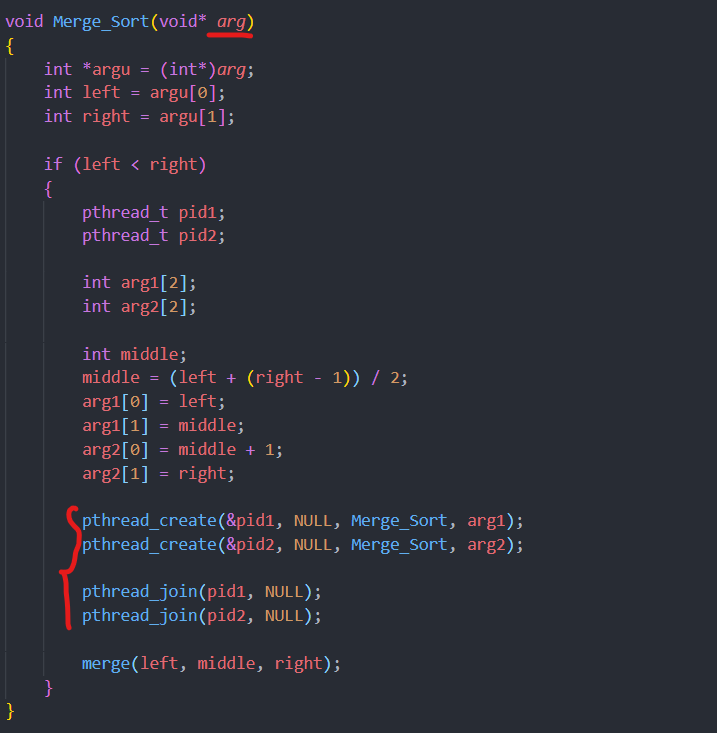
**PA2实验报告**

191300073 AI 杨斯凡 191300073@smail.nju.edu.cn

1. 归并排序：

因为上个学期算法课我写过归并排序，我对上个学期的代码进行了微改

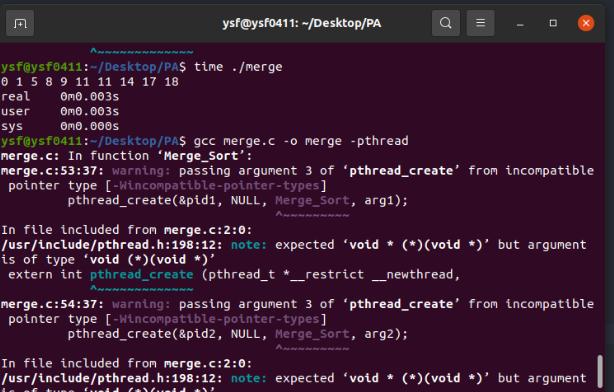


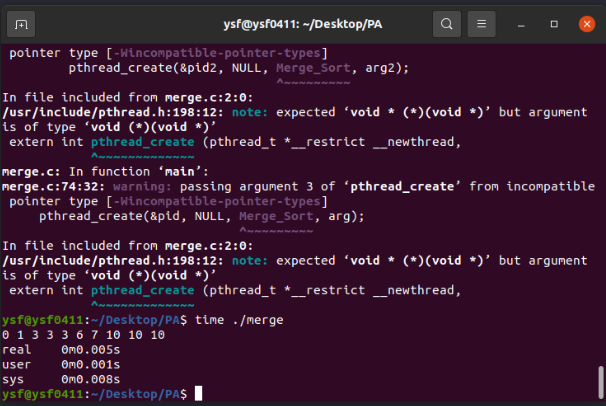
将数组起始位置参数设置成数组传入，而进行mergesort的时候，不再递归调用，

而是创建新的线程去执行，只需等待线程完成即可。

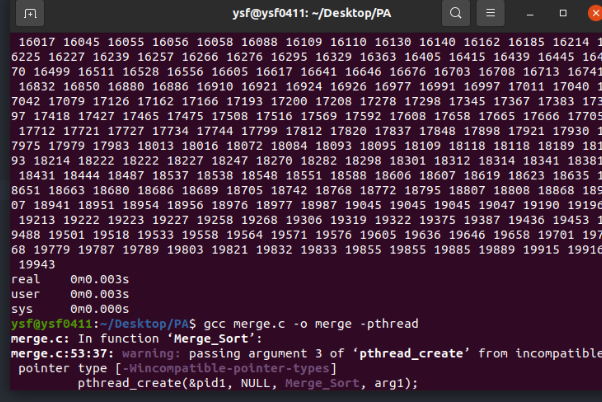
而为了比较时间,我在单线程的时候使用了递归调用.

可以看出, 当数组数较少的时候, 递归次数少, 线程少, 两者时间相差无几.

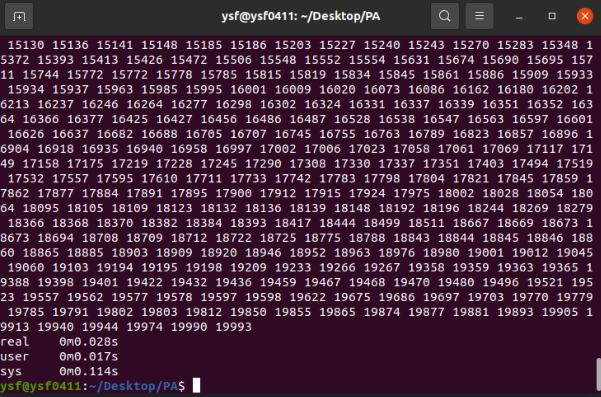




而当数组大小很大的时候, 多线程在线程调度上就得花费很多时间



单线程



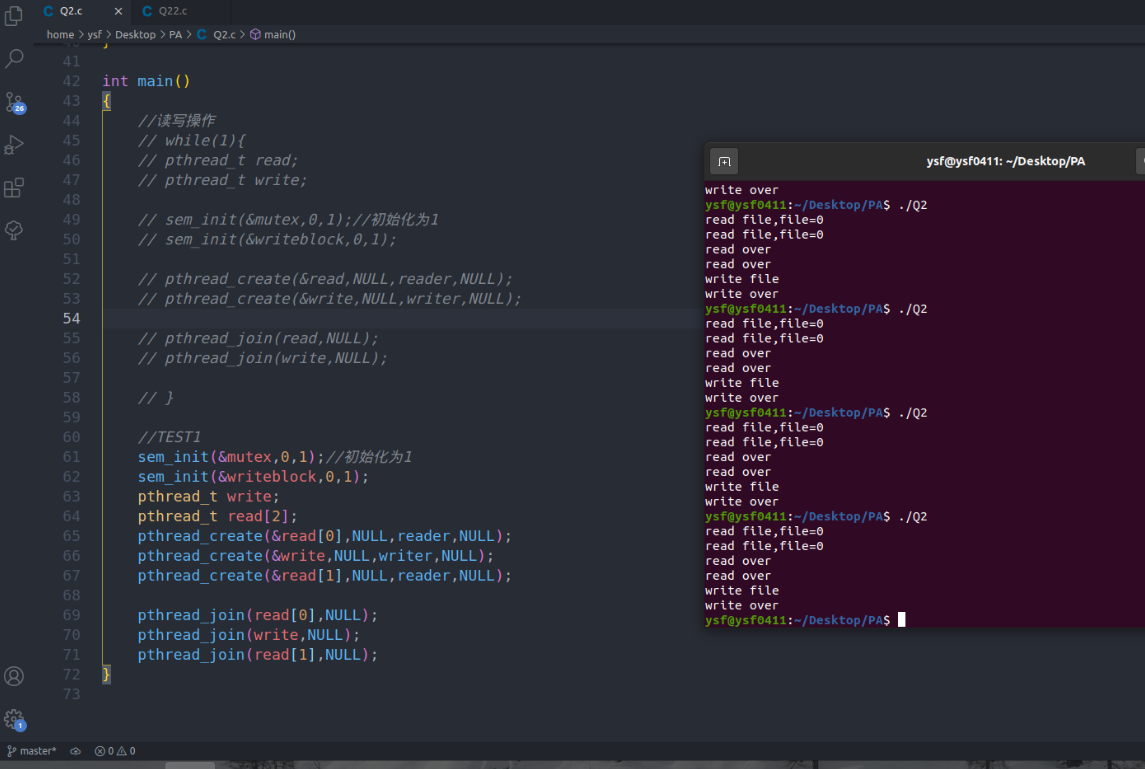
多线程

1. 对于读者写者，我按照课本上的进行PV操作，得以正常实现，而使用读写锁，更加

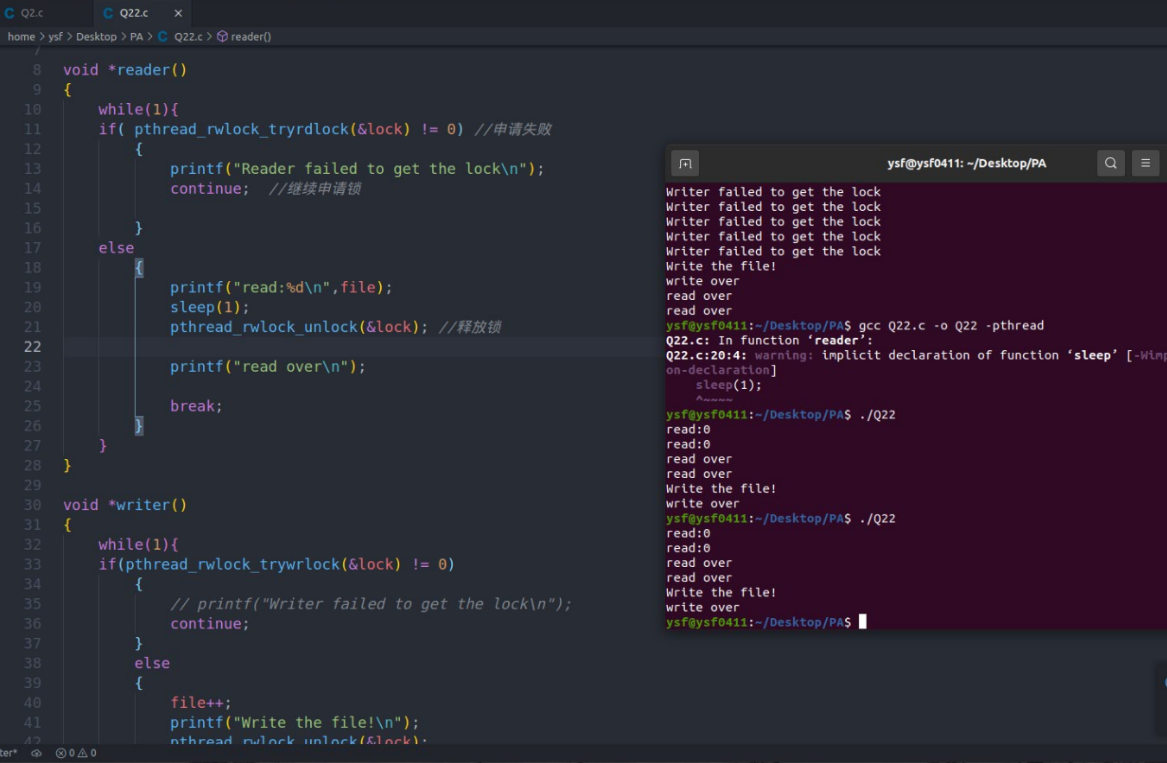
容易得实现了读者写者问题。

而两者在同步上也有着区别。为了使得结果更加明显，我使用了printf和sleep来使

得读写请求的间隔更大，最后结果反映两者都是读优先。



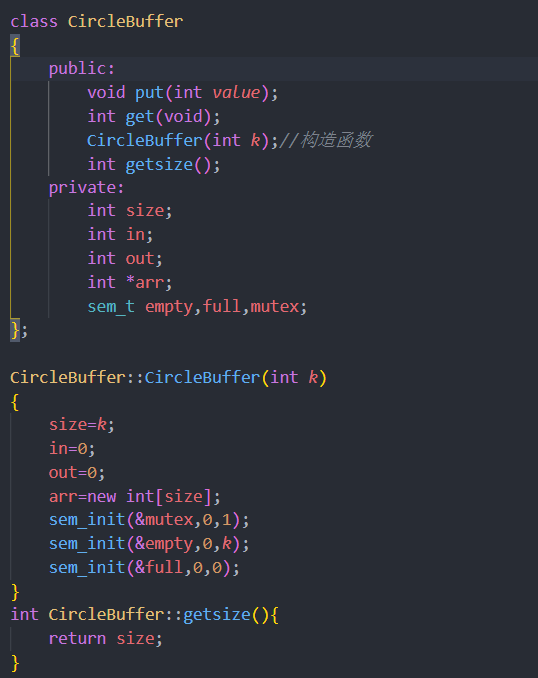
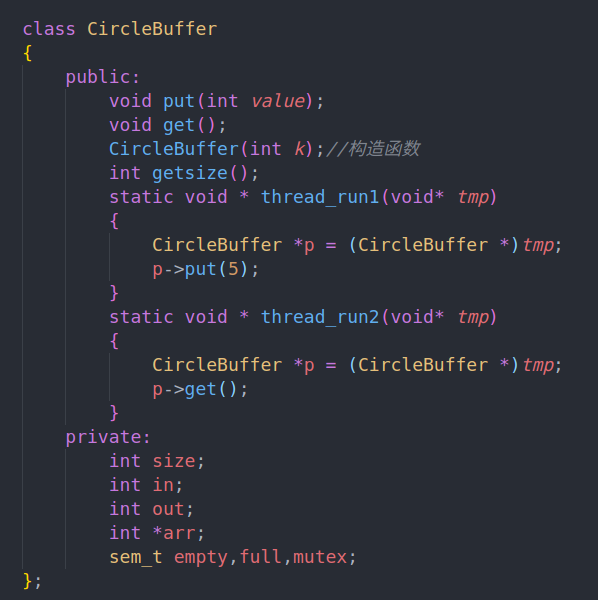
（PV操作）

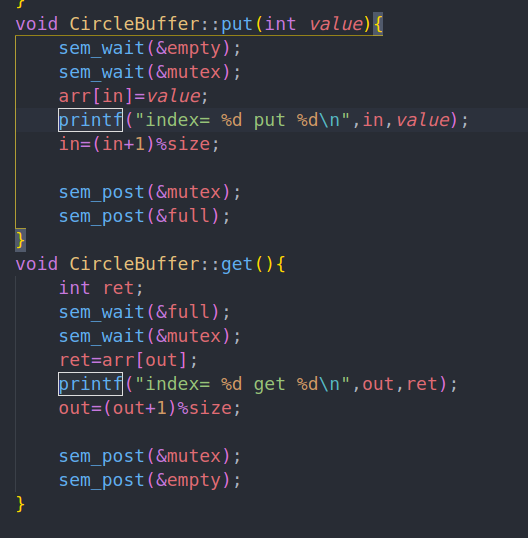


（读写锁）

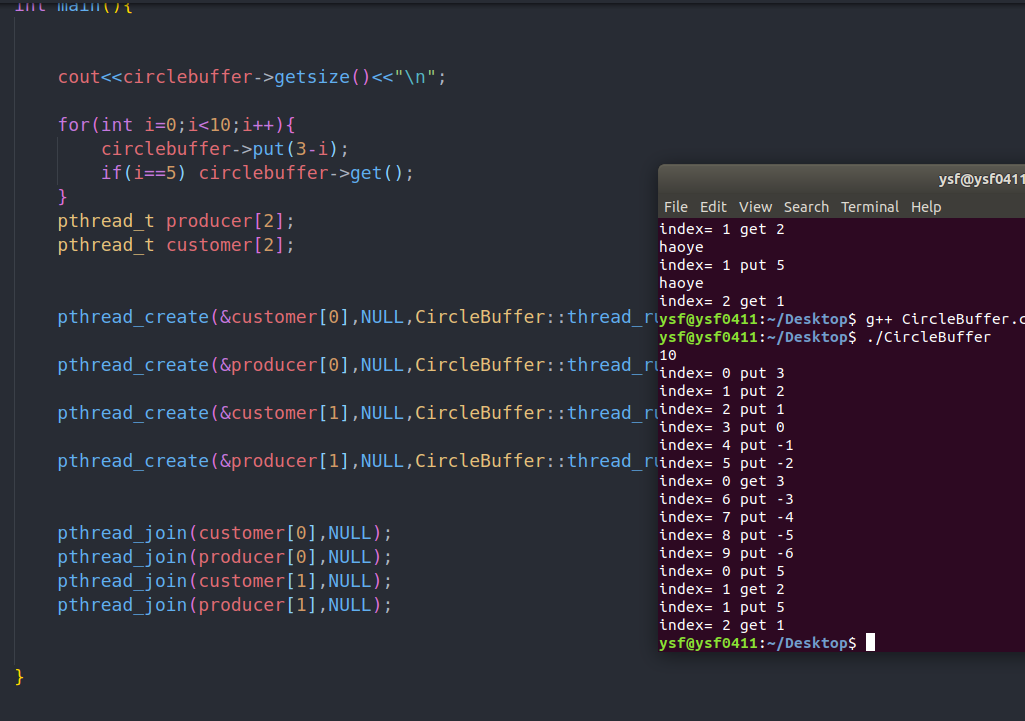
3.

我定义了一个类，然后使用构造函数对缓冲区的大小初始化，然后初始化信号量，

put，get函数我对照课本上的读者写者问题实现，使用了PV操作来实现。而后面发现在多线程中无法调用类中的函数，我对类进行了修改。



（修改后的函数）



（实验结果：可以正确同步）

4.死锁，我仿照了书上的哲学家问题，只不过我把人数降到了2，并且每次对一个信

号量进行P操作后，都sleep挂起当前线程，执行另一个线程，从而提高发生死锁概率，最后成功实现了死锁。

