作业3

1. 实验内容

写一个使用多线程进行排序的程序,工作流程如下:一个整数列表会被分成两个子列表,每个子列表包含原列表一半的元素。然后开启两个单独的线程(称为排序线程)来对每个子列表进行排序,可以使用任意一种排序算法。之后会开启第三个线程(称为合并线程)来将两个已经排序好的子列表合并成一个最终有序的列表。考虑到全局数据需要在所有线程之间共享,实现数据结构的最简单方式可能是使用一个全局数组。每个排序线程分别处理这个数组的一半元素。还需要一个与原始整数列表大小相同的第二个全局数组。合并线程将两个子列表合并到这个第二个数组中。这个项目需要向每个排序线程传入参数。具体来说,需要指定每个线程从数组的哪个索引开始进行排序。

二．实验目的

1.通过实验，加深对线程的知识点理解。

2.通过编写简单的程序，对操作系统中的创建线程和等待线程完成等过程有更深刻的理解。

1. 设计思路

1.首先定义一个全局数组用于存储原始待排序的数据。

2.创建两个线程用于排序:

1）线程1负责排序数组的前半部分数据:传入线程1的启动参数包括:数 组指针,起始索引(0),结束索引(数组长度的一半-1)。

2）线程2负责排序数组的后半部分数据:传入线程2的启动参数包括:数 组指针,起始索引(数组长度的一半),结束索引(数组长度-1)。

3.两个线程分别使用选择排序/冒泡排序等算法对指定范围内的数据进行排序。

4.创建一个线程用于合并:这个线程不需要传入任何参数它会申请一个新的临时数组,用于存储合并后的结果然后依次从两个已排序的子数组中取出最小元素,放入临时数组最后把临时数组中的数据拷贝回原始数组。

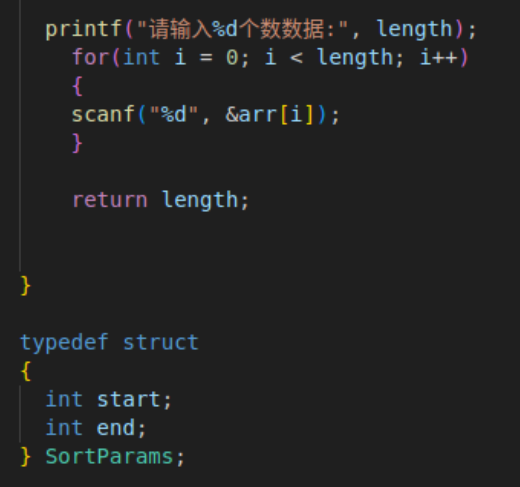
5.主线程等待两个排序线程和合并线程完成,然后输出排序后的结果。

1. 源代码

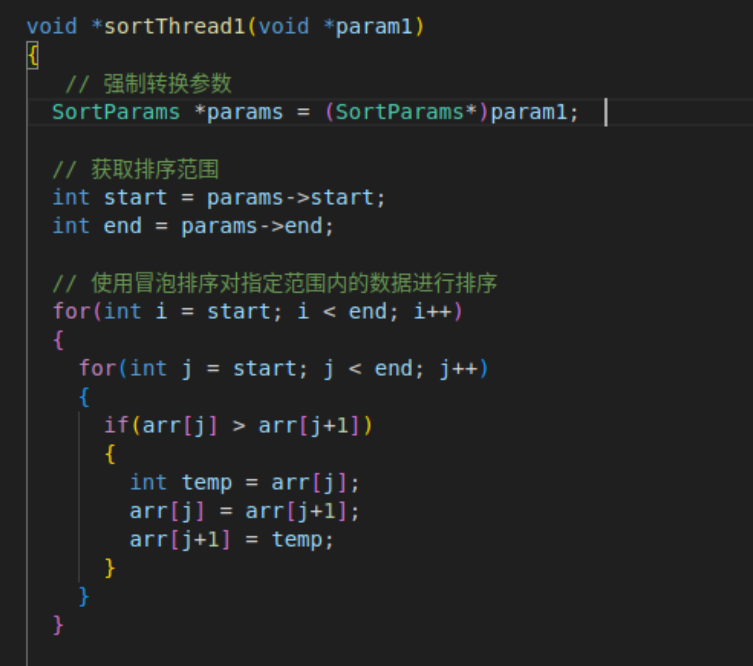
添加头文件，调用Pthread库

用户通过getArray（）输入数组。

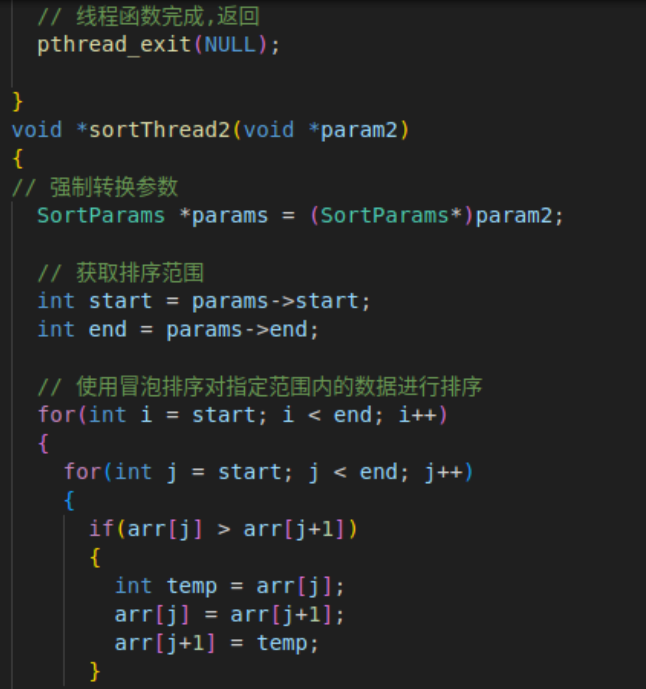




线程一：使用start和end作为标签对数组前半部分进行排序。

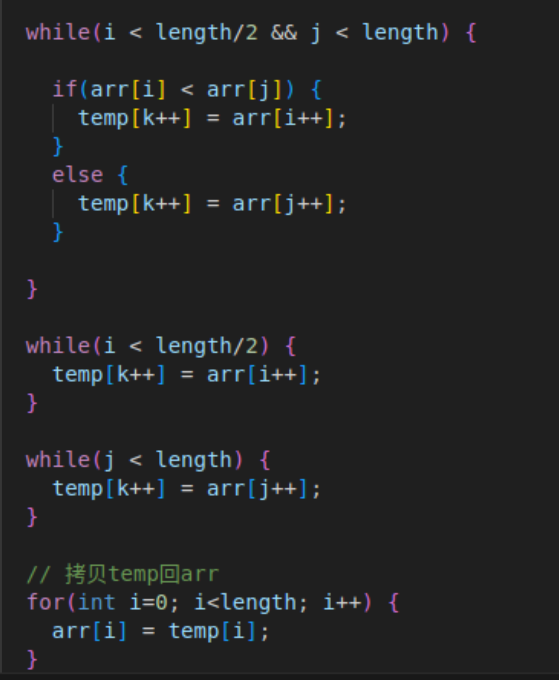


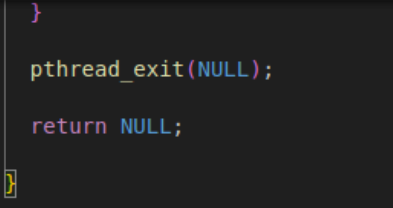
线程二：通过标签start和end对数组后半部分进行排序。



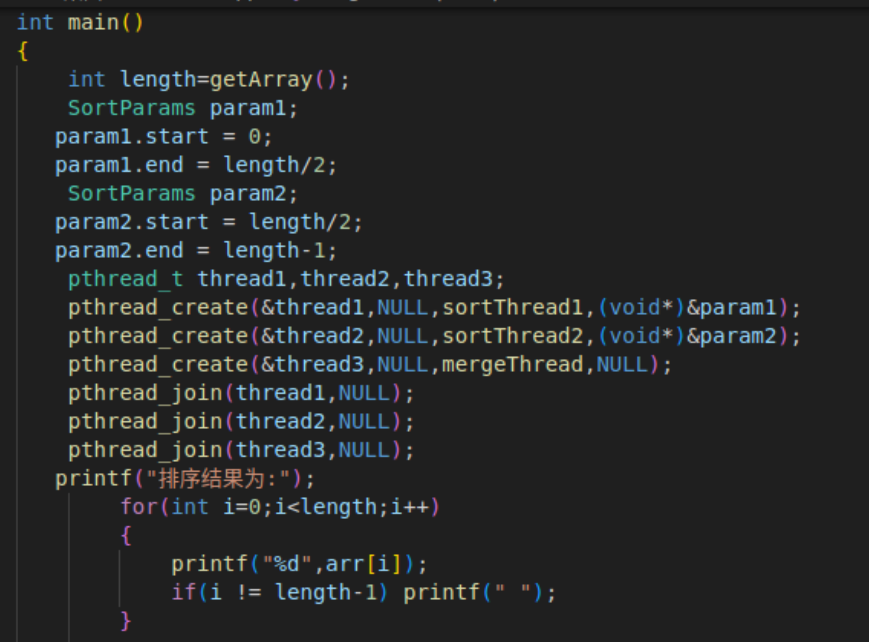
线程三：将排序好的线程一和线程二合并排序到temp数组中，并最后传到arr数组中。



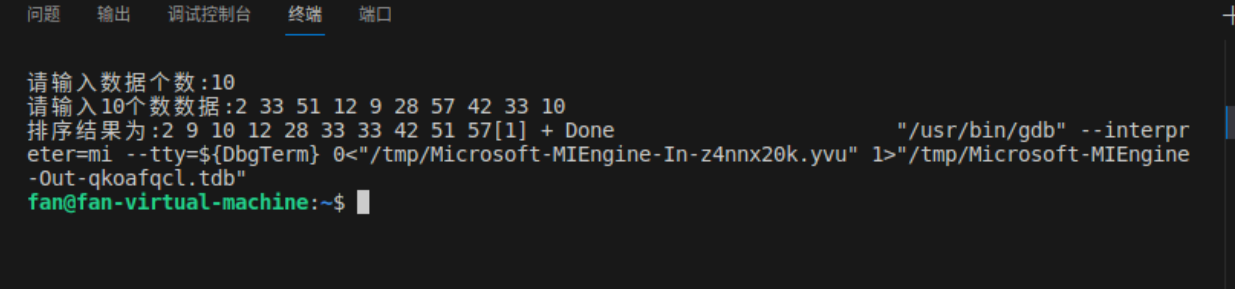




主函数：通过创建线程1、2、3，对数组进行排序操作，主进程等待三个线程完成后，按顺序输出排好序的arr数组。



1. 程序运行结果



1. 总结
2. 通过这次实验，对进程和线程的概念有了更深。
3. 更好地掌握了线程参数传递。需要向每个排序线程传递其工作范围的参数,这是线程编程一个重要内容。
4. 对多线程共享数据访问同步有了更深刻的理解。