Multithreaded Sorting Application

Write a multithreaded sorting program that works as follows: A list of integers is divided into two smaller lists of equal size. Two separate threads (which we will term sorting threads) sort each sublist using a sorting algorithm of your choice. The two sublists are then merged by a third thread—a merging thread—which merges the two sublists into a single sorted list.

Because global data are shared across all threads, perhaps the easiest way to set up the data is to create a global array. Each sorting thread will work on one half of this array. A second global array of the same size as the unsorted integer array will also be established. The merging thread will then merge the two sublists into this second array. Graphically, this program is structured as in Figure 4.27.

original list

7, 12, 19, 3, 18, 4, 2, 6, 15, 8

/ \

sorting thread 0 / \ sorting thread 1

/ \

7, 12, 19, 3, 18 4, 2, 6, 15, 8

\ /

merge thread

\ /

2, 3, 4, 6, 7, 8, 12, 15, 18, 19

sorted list

Figure 4.27 Multithreaded sorting.

This programming project will require passing parameters to each of the sorting threads. In particular, it will be necessary to identify the starting index from which each thread is to begin sorting. Refer to the instructions in Project 1 for details on passing parameters to a thread.

The parent thread will output the sorted array once all sorting threads have exited.