

# CS 5523: Operating Systems

## Homework 1

!!!! No late HW will be accepted !!!!!

(Check BlackBoard Learn for due date and submission)

### Objective

- Learn and practice developing a multithreaded application
- Use both Java **AND** Pthreads with C

### Description

Here is the description for Java version, but you will implement the same application using Pthreads with C, too.

**4.20** Write a multithreaded sorting program in Java that works as follows: A collection of items is divided into two lists of equal size. Each list is then passed to a separate thread (a *sorting thread*), which sorts the list using any sorting algorithm (or algorithms) of your choice. The two sorted lists are passed to a third thread (a *merge thread*), which merges the two separate lists into a single sorted list. Once the two lists have been merged, the complete sorted list is output. If we were sorting integer values, this program should be structured as depicted in Figure 4.23.

Perhaps the easiest way of designing a sorting thread is to pass the constructor an array containing `java.lang.Object`, where each `Object` must implement the `java.util.Comparable` interface. Many objects in the Java API implement the `Comparable` interface. For the purposes of this project, we recommend using `Integer` objects. To ensure that the two

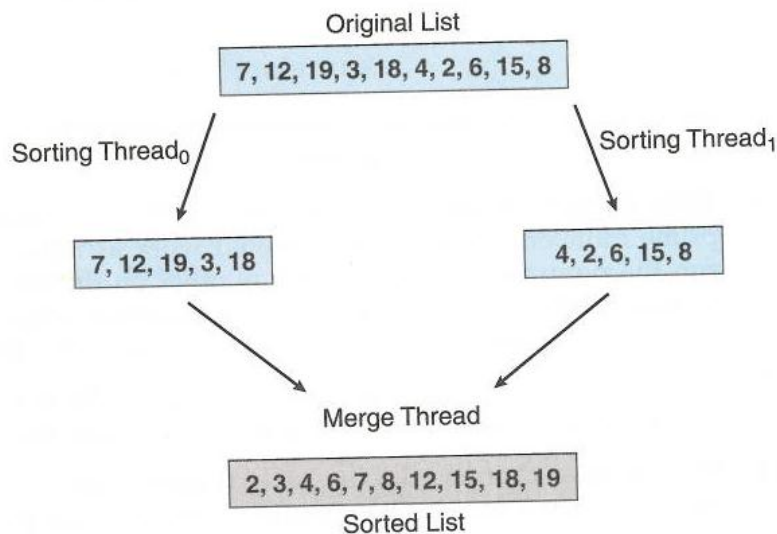


Figure 4.23 Sorting.

sorting threads have completed execution, the main thread will need to use the `join()` method on the two sorting threads before passing the two sorted lists to the merge thread. Similarly, the main thread will need to use `join()` on the merge thread before it outputs the complete sorted list. For a discussion of sorting algorithms, consult the bibliography.

```
/* Don't forget to include comments about the problem, yourself and each major  
step in your program! */
```

---

**Grading:** This is a 100-point homework.

50 for Java and 50 for C Pthreads with C.

Do all your work under a directory **lastname\_hw1**, which should include your source codes, instructions for compile/execute, and some output files showing your test results etc...

Zip **lastname\_hw1** and submit **lastname\_hw1.zip** through BB Learn

---

You must submit your work using Blackboard Learn and respect the following rules:

- 1) All assignments must be submitted as either a zip or tar archive file unless it is a single pdf file.
  - 2) Assignments must include all source code.
  - 3) Assignments must include an output.txt file which demonstrates the final test output run by the student.
  - 4) If your assignment does not run/compile, the output.txt file should include an explanation of what was accomplished, what the error message was that prevented the student from finishing the assignment and what the student BELIEVES to be the underlying cause of the error.
-