Operating Systems, Spring 2020

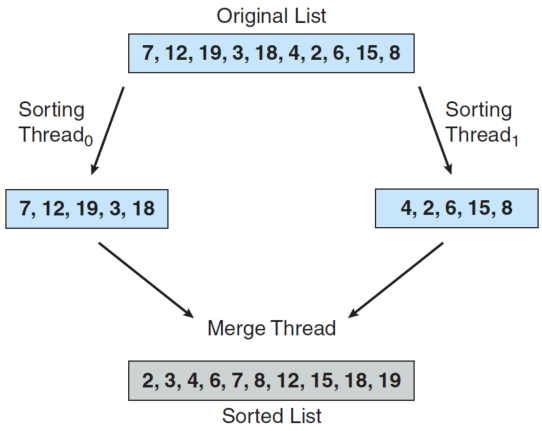
Instructor: Rajeev Kumar Singh

Assignment 2 : Multithreaded Sorting
Submission date: March 19, 2020, 1:00 PM

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 cross 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 according to the following figure.



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 below for details on passing parameters to a thread. The parent thread will output the sorted array once all sorting threads have exited.

Passing parameters to a thread

The easiest approach for passing parameters to a thread is to create a data structure using a struct. For example, a structure to pass the two indices where a thread must begin and end sorting would appear as follows:

```
/* structure for passing data to threads */
typedef struct {
        int start;
        int end;
} parameters;
```

Pthreads will create worker threads using a strategy similar to that shown below:

```
parameters *data = (parameters *) malloc(sizeof(parameters));
data->start = s;
data->end = e;
/* Now create the thread passing it data as a parameter */
```

The data pointer will be passed to the pthread_create() function, which in turn will pass it as a parameter to the function that is to run as a separate thread.

Submission Instructions

- Submit a well-commented code (c file) on Blackboard.
- Make sure to add your and your teammates' names at the top of the code.
- Even if team-work, all members need to submit code separately on Blackboard