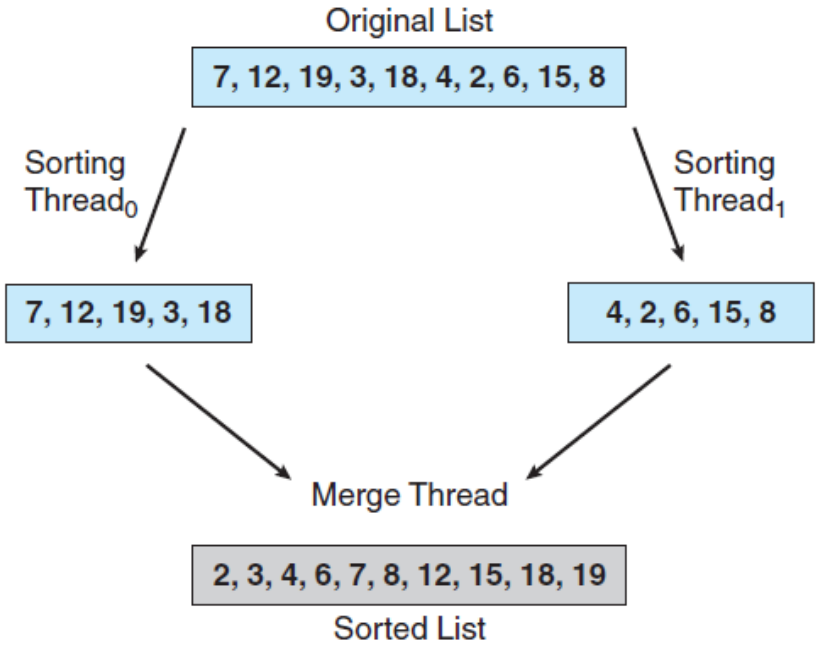
Assignment 6: Multithreaded and MultiProcess Sorting

Deadline: 13th September 2015, 9:00 pm

Goal: To implement sort using multithreading and multiprocessing. Then compare their performance.

Problem:
Write a program to sort the elements of an array using divide two equal parts, sort each sublist and then merge them to get sorted list.

The implementation using Threads:
A list of integers is divided into two smaller sublist of equal size. Two separate threads knows as ***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. As 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. The figure below shows the graphical representation of 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. The parent thread will output the sorted array once all sorting threads have exited.



Multithreaded sorting.

The implementation using Processes:
Implement the same task as above using processes though.

Report Description:
The report that you submit with this assignment, should describe in detail how you performed the sorting using threads and then processes. Make sure that your report is technically sound and readable.

The report should also compare the performance of the sort using processes and threads. Please have a graph in the report which compares the performance as follows: x-axis: varying the size of input (unsorted) array from 100 to 1000; y-axis: the time taken (in micro/milliseconds) to complete the sort.

The graph will have two curves. One for the time taken using threads and the other for time taken using processes.

Deliverables:

You have to submit the following

- A report describing your implementation
- A readme file which describes how to execute your program.
- Source codes of your programs. Name the sorting program using threads as: SortThread-<RollNo>.c. Similarly, name the sorting program using processes as: SortProcess-<RollNo.c>

Combine all these documents into a single zip file as: Assgn6-<RollNo>.zip and then upload them.

Evaluation Criteria:

Report Description: 50%
Program Execution: 40%,
Code Readability (comments and indentation): 10%

For code readability, you have to ensure that your code is easy to read. It is well commented and indented.

Deadline: 13th September 2015, 9:00 pm

Reading Materials and other Useful Links:

1. Operating System Concepts, Ninth Edition by Avi Silberschatz,Peter Baer Galvin,Greg Gagne, Chapter 4, Page 199.