Operating Systems–1: CS3510
Autumn 2021
Programming Assignment 2: Merge Sort using threads
Submission Date: 9th December 2021, 9:00 pm

Goal: Implement merge sort using multiple threads. In this program, you will be using multiple threads to divide the array into multiple sub-arrays (no. of threads) and sort the sub-arrays.

Details:
Merge sort is one of the most efficient sorting algorithms. It works on the principle of Divide and Conquers. Merge sort repeatedly breaks down an array into several sub-arrays until each sub-arrays consists of a single element and merges those sub-arrays in a manner that results in a sorted list.

In this program instead of dividing until the last element, take the number of threads (N) as input and divide the array into N sub-parts. Then the main thread will create N sub-threads. These N threads will sort these N sub-arrays parallelly. Once the sub-array is sorted, the threads will end their execution. Then the main thread will merge the sorted subarrays.

Input:
As input, the program will take the following in a file: (a) Number of threads (b) Size of array (c) the input array.

A sample input file is as follows:
2
5
4 -54 234 -94 18

Output:
Your program should output the following: (i) Time taken to sort (ii) The Sorted array in a form of text file

Report Details:
As a part of this assignment, you have to prepare a report. The report will describe the low-level design of your program.

This report should contain a comparison of the performance of sequential and parallel execution in the form of graph plots. The comparison must consist of the following graphs:

1. Varying the number of threads: (a) x-axis: number of threads varying as - 1, 2, 4 and 8, 16. (b) y-axis: time taken to sort a random array of size 50000.

2. Varying the input array size: (a) x-axis: the array size varying as - 10000, 20000, 30000, 40000, 50000. (b) y-axis: time taken to sort with the total number of threads as 16.

Note that for execution with one thread, you can consider the normal sequential execution code.

Submission Format

You have to upload:
(1) The source code in the following format: Assgn2Src-<RollNo>.c
(2) Readme: Assgn2Readme-<RollNo>.txt, which contains the instructions for executing the program.
(3) Report: Assgn2Report-<RollNo>.pdf which will contain the report as described above.

Zip all the above documents. Name the zipped document as: Assgn2-<RollNo>.zip. Please follow this naming convention. Otherwise, your assignment will not be graded.

Please submit the zipped document on the classroom page by the above mentioned deadline.

Grading Policy:

The policy for grading this assignment will be -
(1) Design as described in report and analysis of the results: 50%;
(2) Execution of the tasks based on the description in the readme: 40%
(3) Code documentation and indentation: 10%.

As mentioned in the class, this is an optional assignment. The best of assignment 1 and 2 will be considered.