SIT315- Seminar 4- Multithreading

Overview

Form a group of 5-6 students and work on the following activities.

Submission Details

Each student should submit the answers in the onTrack individually. Please write the names of all of your group members in the first page of your submission.

Activity 1- Decomposition techniques

This week's lecture discussed different decomposition techniques, namely Input and Output Data Decompositions, Recursive decomposition, and Exploratory Decomposition. Identify what decomposition techniques can be used for the following problems and explain the decomposition for each variation. For some of the problems, you might be able to apply more than one decomposition technique.

1. Solving a sliding 15-puzzle (https://en.wikipedia.org/wiki/Sliding_puzzle)
2. Find the frequency of usage of {'ch','de','des','th','es', 'ci'} in the sliding 15-puzzle Wikipedia page.
3. Binary search (https://en.wikipedia.org/wiki/Binary_search_algorithm) [assume the input array is sorted]

Activity 2 - Parallel Vector Addition

In the resources, you have been provided with a simple vector addition program.

1. Complete the code by adding appropriate comments in the designated lines.
2. Compile and run the program.
3. Develop a roadmap to parallelise this program. You should start with decomposition of the program/problem into sub-tasks - i.e. partitioning data/tasks. Document your list of sub-tasks or activities you plan to do in parallel vs activities that need to be in sequence.
4. Implement your parallel algorithm in C or C++ using pthread or std::thread library.
5. Evaluate the performance of your program (using execution time as a metric), to assess the speed up achieved. Compare the results with the sequential program.
6. Varry the partition size and analyse how it can impact the executing time of the program.

After finishing these activities, use the remaining time of the seminar to work on TaskM2.T1p individually. You can find the task description from OnTrack.