CSCI 176 (Parallel Processing) Spring 2021 Program 2 (20 pts) Due: March 02 (T)

Shared memory programming with Pthreads

Task: Write a Pthreads program with C++ for parallel global_sum computation.

Please study "Hello-correct.cpp" for basic Pthreads technology.

In the global area,
  declare a 1-D array of integers (please use size 500,000,000) and global_sum;

in main( ),
  get the number of threads from command line argument;
  fill the array elements: use a[i] = i+1;
  define the needed number of threads based on the input (number of threads);
  create threads by passing the slave function and parameter (thread_id);
  join all the threads;
  display the final global sum (of the array elements);
  display the total execution time;

in the slave function,
  declare partial_sum with value 0;
  compute start_index and end_index of the array; //assume that the array size is evenly divisible by P
  compute partial_sum (in each thread);
  display thread_id, start_index, end_index, and partial_sum;
  update the global sum with partial_sum;

Note that you should use mutex variable(s) as needed for synchronization. In fact, two mutex variables are needed, one for cout statements and the other for updating the global sum.

Run your program with number of threads = 1, 2, 4, 8 and check/show the execution time for each run. For this, you need to use time checking codes in your program (in main()).

Submission:

Include good documentations (global and each function head) and submit source code and run time output (all 4 runs).