Lab2-Multithreading 2

1- Write a multithreaded program to calculate vector-matrix multiplication. Do not use mutex or semaphore. Each thread multiplies the vector by a set of columns of the matrix. No need to use Mutex. Vary the array size and the number of threads and report the runtime in each case and compare it with the sequential runtime. Compute the speedup in each case.
2- Repeat Q1, but using Matrix-Matrix multiplication. Do not use square matrix, use matrices in the general form.
3- Work the producer-consumer problem and solve the critical section problem using Mutex and RAII.
4- Assume that we need to sum the elements of a huge array using multithreading. In Lab1, we did solve it using another temporary array to save the local sums of each thread. Now, we would like to enhance the solution by using Mutex without using a temporary array. Only a global variable that will have the accumulative sums by each thread. Think of an efficient solution that uses Mutex without having to lock the thread in every summation.
5- Implement the ticket reservation problem and test it using different configurations.
6- Think of a problem to use semaphore to control the access to the shared resource pool using counting_semaphore.

Good luck

Dr. Samir Elsagheer