Assignment 1

Roll No : CS18BTECH11001

PLAGIARISM STATEMENT

I certify that this assignment/report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment/report has not previously been submitted for assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarised the work of other students and/or persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honour violations by other students if I become aware of it.

Name:A. Venkata Sai Mahesh

Date:20/1/2020

Signature: Venkata Sai Mahesh Abburi

Tasks :

Multithreaded Matrix Multiplication :

1. First I check whether it is in interactive mode or non-interactive then i call the respective functions.
2. I created crows no. of threads, such that each tread computes one row of matrix C. This row is computed in the Multiplication function which each thread performs and exits.
3. There would be a context switch between the threads but there will not be any synchronization issues as for different threads we are accessing different elements of the matrices.
4. The used pthread_join for the Main thread to wait until all the other threads completes their work.
5. After the wait of Main thread all the values in the C matrix will be computed and we print this in the output using output_matrix function.

Multiprocess Matrix Multiplication :

1. First I have sent the C matrix to the shared memory using shmat function of shm library so that every process can be able to edit the C matrix(not creating copies).
2. Then I created 1 worker process which again creates another worker process which computes the first half and exits and the first created worker process computes the next half of the matrix.
3. Here i didn’t use more number of processes because it takes much time and the context switching also computes much more time.

Measurements of CPU time :

For the sake of easy understanding let us take all the dimensions of the matrices to be equal (i.e.,N)

Input Size	**T(Single Process)**	**T(Multi Process)**	**T(Multi Thread)**	**SpeedUp (Multi process)**	**SpeedUp (Multi Thread)**
3	1 us	661 us	118 us	0.00 x	0.01 x
10	12 us	673 us	237 us	0.02 x	0.05 x
30	292 us	981 us	451 us	0.30 x	0.65 x
50	1430 us	1165 us	690 us	1.23 x	2.07 x
100	4536 us	2479 us	1953 us	1.83 x	2.32 x
400	155406 us	129569 us	31358 us	1.20 x	4.96 x
800	1380599 us	1067719 us	269440 us	1.29 x	5.12 x
1000	2658843 us	2105751 us	638595 us	1.26 x	4.16 x

Observations :

1. On increasing the Input size the speed up of multi process increases and decreases but it is becomes greater than the single process. This says that multiprocess makes the work much less time than using a single process.
2. On increasing the Input size the speed up of Multithread increases and decreases. This is due to the context switching between the threads which computes some time and increases as the number of threads increases.
3. I have tried using constant number of threads but the speedup is much less than using crows no. of threads.