33231

CLASSMATE
Date :
Page :

ASSIGNMENT 3

Title : Multithreading

Problem Statement : Implement multithreading for matrix multiplication using pthreads

Theory :

What is a thread ?

A thread is a single sequence stream within in a process. Because threads have some of the properties of processes, they are sometimes called lightweight processes.

What are the differences between process and thread ?

Threads are not independent of one other like processess as a result threads shares with other threads their code section, data section and OS resources like open files and signals. But, like process, a thread has its own Program Counter (PC), a register set and a stack space.

Unlike JAVA, multithreading is not supported by the language standard. POSIX Threds (or Pthreads) is a POSIX standard for threads. Implementation of pthread is available with gcc compiler.

Why Multithreading ?

Threads are popular way to improve application through parallelism. For example, in a browser, multiple tabs can be different threads. MS word uses multiple threads, one thread to format the text, other to process input, etc.

33231

CLASSMATE

Date :

Page :

Threads operate faster than processes due to the following reasons :

- Thread creation is much faster
- context switching between threads is much faster
- Threads can be terminated easily
- Communication between threads is faster.

POSIX Thread Libraries

- based thread API for C / C++
- allows to spawn a new concurrent process flow
- most effective on multi-processor or multi-core systems where the process flow can be scheduled to run on another processor thus gaining speed through parallel or distributed processing
- Threads require less overhead than "forking" or spawning a new process because the system does not initialize a new system virtual memory space and environment for the process.

33281

CLASSMATE
Date
Page

A3

Algorithm :

1) Input matrix 1
2) Input matrix 2
3) Validate rows and columns for multiplication to be possible
4) Create m x n threads
5) Call thread function to calculate one element of resultant matrix m x n times.
6) Join the threads
7) Display the resultant multiplication matrix.

Conclusion : I have successfully implemented multithreading for matrix multiplication using POSIX threads.