Max Halbert

Lab5 writeup

Step 1:

The threads are not running in sequence as they are created. It shows thread scheduling on the CPU is nondeterministic. The program behavior can completely change when you rerun the program.

The number that passed to each thread is overwritten by the main program and all the threads share the same variable, means not thread safe. That is why the passing argument output is wrong in the output.

Step 2:

The program in step 1 is fixed by giving each thread its own global variable, so not overwritten by the main thread or shared by other threads.

Step 3:

Matrix multiplication program is running fine in the itcd4.csumb.edu Linux if number of threads is less than 13. However, the system not letting the program to create threads more than 13.

The program is fine when run it in my virtual Linux window even with 1024 threads created to compute a matrices 1024 rows.