

- **Multithreading use and evaluation**

Matrix multiplication is one of the applications that can benefit much from the multithreading approach of parallelism. Data distribution is an important implementation detail that, if abstracted out of a module interface, can facilitate code reuse, each of which can be given to a separate independent thread. The multiplications of rows and columns can be divided over different thread workers, all of which can be joined together to make up the result matrix.

Different approaches can be followed, one of which is to create a thread for each `Row X Column`. Another can be based on dividing each matrix into sub-matrices; multiplying these and combining their results. However, this assignment is asking you to **re-write the program given in HW2 to apply the Pthreads under Linux** and to repeat the same set of experiments on multiplying the two matrices.

- **Specifications /Algorithm:**

**Your approach must create a thread that works on multiplying one Row of the first matrix (A) with every column of the second matrix (B) and generate one row in the resulted matrix (C).**

- **Implementation:**

- You must provide an implementation for your program and demonstrate its work under **Linux**. Your program should implement the specification/algorithm given above
- First, your program should be written C (not C++), and compiled with the (gcc) compiler under Linux. This program should make use of the Pthreads library
- Second, you must compare the calculations time with the time of the sequential multiplication approach on the same set of matrices that you achieved in HW#2.

- **Experimentation**

You should make your programs dynamic and configurable with command line arguments to build different matrices each time with different sizes, these matrices can be filled with random numbers (integers) that are generated automatically at each run.

- a. You need to experiment and find the time for five different square matrices of sizes:

- i. 10 X 10
- ii. 100 X 100
- iii. 500 X 500
- iv. 1000 X 1000
- v. 2000 X 2000

- b. You should record the time for this thread based approach and for each run.
- c. Provide a chart diagram for each program and compare the time with the one obtained from HW2.

- **What to submit:**

1. Upload your programs along with your documentations (data & figures) into the JUST e-learning system as one compressed file (\*.zip or \*.rar). After submitting your work, you should schedule an appointment to discuss your submission. Your grade will be given based on your discussion.