

You are required to write a routine in C using OpenMP that performs matrix multiplication using the blocked algorithm:

```
MatMul( int m, int n, int p, int b, double alpha, double beta,  
        double *A, double *B, double *C )
```

More precisely, the routine computes $C = \alpha \times \text{trans}(A) \times B + \beta \times C$, where alpha and beta are scalars of type double, A is a pointer to the start of a matrix of size $n \times m$ doubles, B is a pointer to the start of a matrix of size $n \times p$ doubles, C is a pointer to the start of a matrix of size $m \times p$ doubles, and b is the block size.

You can assume that m, n, p are powers of 2 and are ≥ 64 . The routine should work for values of b = 1, 2, 4, 8, 16, 32, 64.

The routine will be tested with large matrices and with varying number of threads (e.g. 1, 2, 4, 8, 16, 32). The routine should use openmp directives directly as illustrated in the test code.

Two files main.c and MatMul.c are provided as reference (the provided file MatMul.c does not contain a correct code and is just provided for reference):

<Substitute path to main.c>

<Substitute path to MatMul.c>

You are required to replace MatMul.c with your version of the code. **You have to finally submit the file Matmul.c only.** This file will be compiled with Main.c as follows:

```
$ gcc -fopenmp main.c MatMul.c (no additional compile time flags will be used)
```

The marks for this assignment will be based on the performance of the code for large matrices and threads. Therefore you have to try to make the code as efficient as possible

FYI: The code will finally be run with different versions of main.c by changing some of the parameters.

You should start writing the OpenMP code and test it on your system. The recommended way to do this is to install VirtualBox on your system (<https://www.virtualbox.org/>) and use the Linux Lite image (<http://www.osboxes.org/linux-lite/#linux-lite-36-vbox>) that already has the OpenMP compatible gcc compiler setup. To create the image, carefully follow the instructions at <https://www.linuxliteos.com/forums/tutorials/how-to-install-test-and-use-linux-lite-in-oracle-virtualbox/> step by step.

Alternatively, you can install gcc or any other OpenMP supporting compiler on your system. Please note that final submission will be compiled with gcc and therefore should be gcc compatible.

Lecture 7 of Week 3 will be covering Matrix multiplication using blocked algorithm in detail. The lecture will be made available on 9th September. The submission date of the assignment is 15th September. This assignment is released early so that you can setup virtualbox and try

out an OpenMP non-blocked version of the matrix multiplication code in the meanwhile (you should target to complete this before 9th September).

Please read the instructions in this Assignment carefully. Note that:

- **No marks will be awarded if the multiplication code is not blocked**
- **No marks will be awarded if A is not transposed**
- **Do not submit main.c or any file containing the main function**
- **The example code provided is not correct. It is only provided as a reference to help you understand how to access elements of the arrays.**