



## Lab6 Threads: Matrix Multiplication

### Objectives:

1. Introducing threads concepts and POSIX threads library.
2. Implementing popular algorithms as multi-threaded ones.

### Problem Statement:

In this project, you will implement two variations of the matrix multiplication algorithm:

- Version 1)** The computation of each element of the output matrix happens in a thread.  
**Version 2)** The computation of each row of the output matrix happens in a thread.

### Codebase:

Your codebase consists of two source files and a Makefile.

- a) **project.c**. This file generates two matrices, and calls the function that multiplies both matrices. (Make your project, run `./lab6` to see what is happening). You may add any tests here (for functionality and speedup). However, you will not deliver this file. We will use our version, which contains our testcases. **Therefore, do not put any code here that you need in your implementation.**
- b) **matmult.c**. This file contains three functions:
  - `matmult`: **ALREADY IMPLEMENTED**
  - `matmult_v1`: to be implemented
  - `matmult_v2`: to be implemented
- c) **Makefile**. Do not modify it. You do not need to link except with `pthread`, which is already done, for you, over there.

### Hint:

- You may learn about POSIX threads [here](#).

### Deliverables:

- You will submit your **matmult.c** file ONLY, commented thoroughly and clearly.
- Append `.pdf` to the filename to be able to upload it. (`matmult.c.pdf`)

### Notes:

- Languages used: C.
- Students will work **individually**.
- You may talk together on the algorithms or functions being used, but are allowed to look at **anybody**'s code.
- *Revise the academic integrity note found on the class web page.*