**Operating Systems**

**Threads Lab**

Consider the problem of matrix multiplication, described here in Wikipedia, <https://en.wikipedia.org/wiki/Matrix_multiplication>, and here at <https://www.mathsisfun.com/algebra/matrix-multiplying.html>.

It is a task that can be distributed among many threads, and serve as an interesting simple example of cooperating threads. Try the following tasks:

Part 1:

* Hard-code two operand matrices and one result matrix globally.
* Write a sequential solution to this problem.
* Time the computation: you may use the function call gettimeofday, which can give you microseconds (read the man page), to get the time before and after your computation is finished.

int m1[2][3] = {{1, 2, 3},

{4, 5, 6}};

int m2[3][2] = {{1, 2},

{3, 4},

{5, 6}};

int result[2][2] = {{0, 0},

{0, 0}};

Part 2:

* Divide the matrix multiplication task, assigning one thread per **row** of the first matrix.
* Output the result matrix and check your solution.
* Does this division of work require any synchronization mechanisms? Is it faster than sequential?