This lab aims at learning how to do multithreaded programming in Linux using **pthreads** (POSIX threads) and synchronize between threads using semaphores.
Task 10.1

Use POSIX threads to solve the following problem.

Given two matrices $A$ and $B$, where $A$ is a matrix with $m$ rows and $k$ columns and matrix $B$ contains $k$ rows and $n$ columns, the matrix product of $A$ and $B$ is matrix $C$, where $C$ contains $m$ rows and $n$ columns. The entry in matrix $C$ for row $i$ and column $j$, $C[i][j]$, is the sum of the products of the elements for row $i$ in matrix $A$ and column $j$ in matrix $B$. That is,

$$C[i][j] = A[i][0]*B[0][j] + A[i][1]*B[1][j] + ... + A[i][k-1]*B[k-1][j]$$

For this task, calculate each element $C[i][j]$ in a separate thread. This will involve creating $M \times N$ threads. The main - or parent - thread will initialize the matrices $A$ and $B$ and allocate sufficient memory for matrix $C$, which will hold the product of matrices $A$ and $B$. 
Task 10.2

Implement the following synchronization patterns with semaphores using POSIX threads.

1. Signaling
2. Rendezvous
3. Mutual exclusion
4. Multiplex
Task 10.3

Implement the **Barrier** synchronization patterns with semaphores using POSIX threads.
Task 10.4

Implement the **Reusable Barrier** synchronization patterns with semaphores using POSIX threads.
Task 10.6*

- If you complete this task your points will be \textit{doubled} for the whole Lab 10.

- Solve one of the following synchronization problems with semaphores using POSIX threads and semaphores.
  - Dining Savages Problem
  - Building H$_2$O Problem
  - River Crossing Problem
  - Refer \textit{The Little Book of Semaphores by Allen B. Downey}. 