Gebze Technical University
## Department of Computer Engineering
CSE 244 Systems Programming
Final Project, Due on Monday May 29 @ 8:55

In this project, you are expected to write a client-server based application that solves the linear equation of the form $Ax = b$ with $A \in R^{m \times p}$ being the input matrix, $x \in R^p$ is the vector of unknowns and $b \in R^m$ is the output vector using 3 different methods. You are also expected to evaluate the server performance for *(i)*thread-per-request, and *(ii)*worker pool before request based implementations (server program needs to be implemented twice). The overall structure of the system is expected to have the following form:
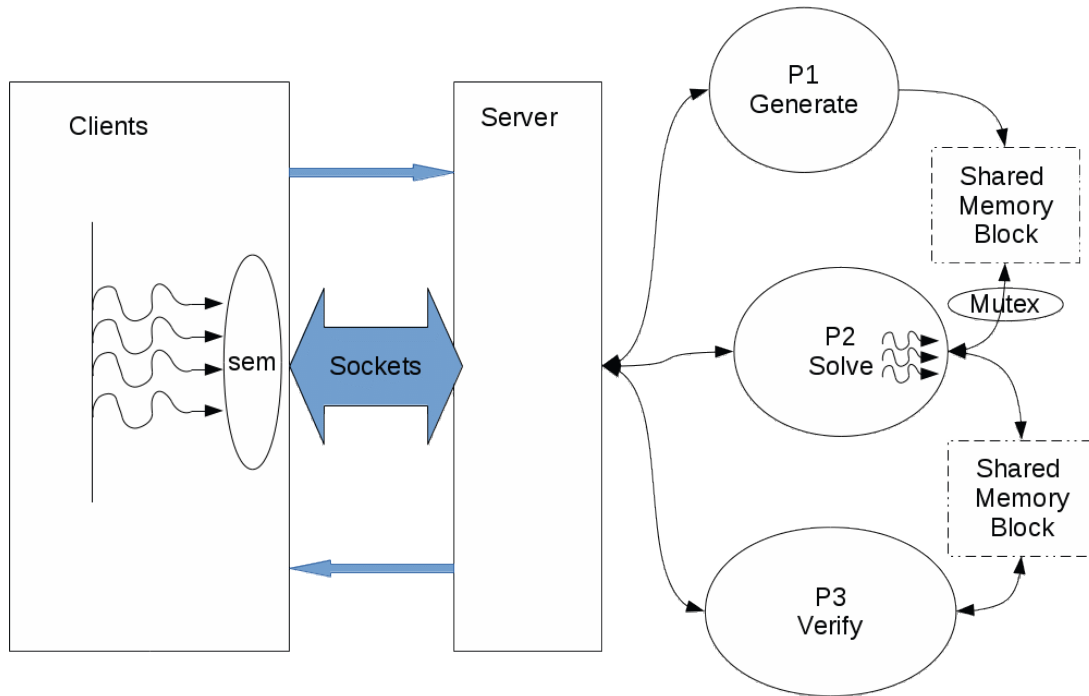
The client side of the project should be of the form

```
clients <#of columns of A, m> <#of rows of A, p> <#of clients, q>
```

this program generates `q` clients that race to connect to the Server side, each client would request the Server to generate the matrices A and b, the answer x and the norm of the errors generated by each method will be the return values of from the server (the clients should not block the connection port of the Server site). Each client will log the matrices generated (A and b), the return values from the server and the time of the termination signal if received before the end of execution. The program `clients` should log the average connection time, $\bar{x}$, and the standard deviation, s, of the overall run.

Note that

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n},$$

$$s = \sqrt{\frac{\sum_{i=1}^{n} x_i^2 - n\bar{x}}{n-1}}$$

error term can be calculated as
$$e = Ax_d - b$$

and the norm of the error term server will return is calculated as
$$|e| = \sqrt{e^T e}$$

The server side of the project is expected to be in the form

```
server   <port #, id> <thpool size, k >
```

For each request the server will initiate P1 which uses a random number generator to form the entries of the matrices A and b , P2 that solves for the unknown vector x, using three different methods SVD, QR factorization and pseudo-inverse method, and P3 which will calculate the error and norm of error terms using the formulations given above.

Server prints the number of clients currently being served at the screen and creates a log file for each client served with the matrices A and b, the answer x_i

Test your code using *(i)*thread-per-request, and *(ii)*worker pool before request based implementations for the server for different values for `<#of clients, q>` (i.e. 5, 10, 20, 50), multiple clients program running at the same time and evaluate the server performance for both cases with different `<thpool size, k >`. **Write a report using the results you obtained from your test cases.**

Note that clients should exit if no server is running. When ^C occurs on the client side server side should not be effected (clients should terminate if the server side receives a ^C). Beware of the semaphore and mutex usages shown in the figure. The project requires parallel executions, Solutions using sequentail execution approaches will not be graded.

Best Wishes.