# Network Programming Assignment II

## Problem

Design and implement a multi-threaded server that allows concurrent clients to search the meaning(s) of a word, add a new word, and remove an existing word.

This assignment has been designed to demonstrate the use of two fundamental technologies that have been discussed during the lectures:

*Sockets*
*Threads*

Hence, the assignment must make an **EXPLICIT** use of the two above. By explicit, we mean that in your application, sockets and threads must be the lowest level of abstraction for network communication and concurrency.

## Architecture

The system will follow a client-server architecture in which multiple clients can connect to a (single) multi-threaded server and perform operations concurrently.

## Interaction

All communication will take place via sockets. These sockets can be either TCP or UDP, however, keep in mind that all communication between clients and server is required to be reliable.

## Failure Model

It is expected that, on both the server and the client side, errors (by means of exception handling) are properly managed. The errors include the following:

Input from the console for what concerns the parameters passed as command line. Network communication (address not reachable, bad data...).

I/O to and from disk (cannot find the dictionary file, error reading the file, etc...).
Other errors you might come up with.

The application will be tested and validated against all these errors.

## Functional Requirements

### Query the meaning(s) of a given word

The client should implement a function that is used to query the dictionary with the following minimum (additional input/output parameters can be used as required) input and output:

Input: Word to search

Output: Meaning(s) of the word

Error: The client should clearly indicate if the word was not found or if an error occurred. In case of an error, a suitable description of the error should be given to the user.

### Add a new word

Add a new word and one or more of its meanings to the dictionary. For the word to be added successfully it should not exist already in the dictionary. Also, attempting to add a word without an associated meaning should result in an error. A new word added by one client should be visible to all other clients of the dictionary server. The minimum input and output parameters are as follows:

Input: Word to add, meaning(s)

Output: Status of the operation (e.g., success, duplicate)

Error: The user should be informed if any errors occurred while performing the operation.

### Remove an existing word

Remove a word and all of its associated meanings from the dictionary. A word deleted by one client should not be visible to any of the clients of the dictionary server. If the word does not exist in the dictionary then no action should be taken. The minimum input and output parameters are as follows:

Input: Word to remove
Output: Status of the operation (e.g., success, not found)

Error: The user should be informed if any errors occurred while performing the operation.