ASSIGNMENTS 1-2

The two following assignments must be implemented in a distributed fashion by utilizing the Java RMI programming framework. The students must study the Java RMI slides, watch the relevant lectures, and study the Java RMI tutorials that have been uploaded in the course's Web page.

According to our lectures, these assignments are not mandatory. However, the students that complete any of them will get a bonus in their final grade. More specifically, the successful completion of assignment 1 gives a bonus of 1 point. Similarly, assignment 2 gives a bonus of 1 point. Consequently, a student that successfully completes both of them will receive a bonus of 2 on his/her final grade.

The deadline for both assignments is set to 31/12/2020.

ASSIGNMENT 1

The Server plays the role of a matrix calculator. It is considered to be a computational machine with great processing power, much greater than that of the Clients.

So:

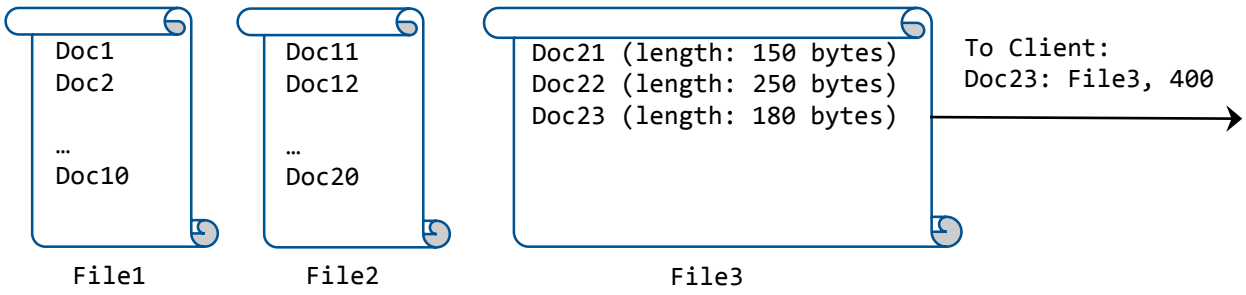
1. The Client sends two matrices to the Server, and the operation to be performed on them.
2. The Server performs the requested operation and returns the result to the Client.
3. Supported Operations:
    a. Summation (element-wise).
    b. Subtraction (element-wise).
    c. Multiplication (element-wise).
    d. Standard matrix multiplication.

ASSIGNMENT 2

A crawler is a distributed system employed by a Web search engine with the aim of fetching Web pages and files. The files are then stored into a local repository, and indexed by the engine's text indexers.

This assignment is an extremely simplified implementation of a Web crawler.

1. The Client reads a local file* that contains URLs and saves them into a list.
2. The Client sends the first URL of the list to the Server.
3. The Server downloads the content of the Web page under this URL.
4. The Server stores the HTML content into a file. The writing must be in "append" mode. The file may contain at most 10 Web pages. If this limit is reached, the file is closed and another file is opened.



5. The Server sends back to the Client the filename where the document has been saved and the location in the file where the document has been saved.
   In this example, the Client has asked from the server to download the 23rd file. The Server downloads the file and saves it in File 3 (Files 1 and 2 are full). Then, it responds to the Client that the Web page has been stored in File 3, in location 400 (=150+250), which is the starting position of Doc23 inside File3.
6. The Client sends the next URL to the Server to be downloaded.

* Download the file from https://people.iee.ihu.gr/~lakritidis/downloads/courses/dsys/urls.txt