DISTRIBUTED SYSTEMS (Lab) Exercise #1 2022-23 Date Delivery: 30/4/2023

You are asked to create in C a concurrent1 server (server process) which as a service task will perform the following calculations (taking as inputs a real number *r* and two integer vectors *X (x1,x2,...,xn), Y (y1,y2,…,yn)* of length *n* where *n* will be defined by the user, and which can be repeatedly sent by one or more clients2 / client processes):

1. The inner product of the two vectors *X ÿ Y* (return: an integer)
2. The average value of each vector: *ÿX, ÿÿ* (return: an array of 2 real numbers)
3. The product *r*(X+Y)* (return: a vector-array of real numbers of length n)

Communication should be done via TCP AF_INET (Internet Domain) sockets. Each socket-client process will read from the keyboard (repeatedly, until the user indicates that it does not wish to continue) (a) the selection of the computation that the user wishes to be performed (1,2,3) and (b) the respectively-required case-by-case data *(n, X, Y,* r), will pipe it to the socket-Server process and wait to receive from it the result to print it on the screen.

The socket-Server process will accept the data to be processed **from the socket-Clients processes, and will produce the corresponding result** NOT through its own (local) function-calculation BUT **through an appropriate Remote Procedure Call** that you will implement with the help of the ONC RPC implementation . In other words, the socket-Server process (operating simultaneously as an RPC-Client) should call (depending on the calculation value sent by the user - 1,2,3) the corresponding routine from an RPC-Server and to wait for the corresponding result from it (in order to then channel it to the corresponding socket-Client).

As for the RPC-based part of the communication, you should first properly define the required ('.x') interface file (defining within it three separate function-procedures (one for each of the three computations requested above), then generate *automatically* through the *rpcgen* utility (and based on what you learned in the workshop) both the required system modules (RPC-server-stub
module and RPC-client-stub module) for the implementation of the requested RPCs, as well as the ready-made templates for the two application modules of your application (RPC-server-application module and RPC-client-application module), and as follows:

(a) properly complete the RPC-server-application module (which will perform the basic service tasks on the data-parameters that will be sent remotely by the RPC-Client/socket-Server) and

(b) also properly complete the RPC-client-application module (through which
the actual remote process call will be made) by integrating/merging it with the main socket-Server process among others
described above.

Deliverables: code, commentary/documentation, test runs

[1] that is, with the possibility of serving multiple requests simultaneously.
[2] the execution process of which you are also asked to make.