Assignment NO: 1

Title :- Implement multi-thread client/server process communication using RMI

Questions for theory :-

1) What is RPC, LRPC, RMI ?

→ ① RPC -

- Remote procedure call (RPC) is a communication technology that is used by the one program to make a request to another program for utilizing its service on a network without even knowing the network's details. A function call or a subroutine call are other terms for a procedure call.

② LRPC -

- Lightweight Remote procedure call is a communication facility designed & optimized for cross-domain communications in microkernel operating systems. For achieving better performance than conventional RPC systems, LRPC uses the following four techniques : Simple control transfer, simple data transfer, simple stubs, & design for concurrency.

③ RMI -

- RMI stands for Remote method Invocation. It is a mechanism that allows an object residing in one system (JVM) to access/invoke an object running on another JVM.

- RMI is used to build distributed applications; it provides remote communication between Java programs.
- It is provided in the package java.rmi.

2) How stub is generated in RPC.

→ Stubs can be created in two different ways:-

1> Manual Generation of Stub:

2> Automatic Generation of Stub:

1> Manual Generation : In the manual generation of stubs, the RPC implementer provides a collection of translation functions from which a user can create their own stubs using this way.

2] Automatic Generation: In the automatic generation of stubs, client & server interfaces are defined using Interface Definition Language (IDL). An interface specification, for example, contains information indicating whether each argument is input, output, or both; only input arguments must be passed from client to server, while only output elements must be copied from server to client.

3] Explain call semantics in RPC & RMI invocations?

→ Types of call semantics:-

- perhaps or possibly call semantics: It is the

weakest one, here, the caller is waiting until a predetermined timeout period & then continues with its execution. It is used in services where periodic updates are required.

- Last-one call semantics: Based on timeout, retransmission of call message is made. After the elapsing of the timeout period, the obtained result from the last execution is used by the caller. It sends out orphan calls. It finds its application in designing Simple RPC.

- Last-of-many call semantics: It is like last-one call semantics but the difference here is that it neglects orphan calls through call identifiers. A new call-id is assigned to the call whenever it is repeated. It is accepted by the caller only if caller id matches the most recent repeated call.

4) How applications are developed in RMI?

→ To write an RMI Java application, you would have to follow the steps given below.
- Define the remote interface.
- Develop the implementation class (remote object)
- Develop the server program
- Develop the client program

- compile the application
- Execute the application

5] Advantages & disadvantages of RMI

→ • Advantages of RMI:
- Simple & clean to implement that leads to more robust, maintainable & flexible applications.
- Distributed systems creations are allowed while decoupling the client & server objects simultaneously.
- It is possible to create zero-install client for the users.

• Disadvantages of RMI:-
- Less efficient than socket objects.
- Assumming the default threading will allow ignoring the coding, being the servers are thread-safe & robust
- cannot use the code out of the scope of java.
- security issues need to be monitored more closely.