**DEPARTMENT OF INFORMATION TECHNOLOGY Experiment No. 2**

|  |  |
| --- | --- |
| Semester | T.E. Semester VI – Information Technology |
| Subject | Distributed Systems |
| Subject Professor In-charge | Prof. Ichhanshu Jaiswal |
| Assisting Teachers | Prof. Girish Wadhwa |
| Laboratory | L05B |

|  |  |  |
| --- | --- | --- |
| Student Name | Ashutosh Engavle | |
| Roll Number | 15101B0042 | |
| Grade and Subject Teacher’s Signature | A P V T |  |

|  |  |  |
| --- | --- | --- |
| Experiment Number | 2 | |
| Experiment Title | Client Server based program using RMI. | |
| Resources / Apparatus Required | Hardware:  Desktop | Software:  Gedit editor, Ubuntu OS |
| Objectives  (Skill Set / Knowledge Tested / Imparted) | * To know the RMI Concepts and Communication. | |
| Theory of Operation | Remote Method Invocation (RMI) is an API which allows an object to invoke a method on an object that exists in another address space, which could be on the same machine or on a remote machine. Through RMI, object running in a JVM present on a computer (Client side) can invoke methods on an object present in another JVM (Server side). RMI creates a public remote server object that enables client and server side communications through simple method calls on the server object.  Working of RMI  The communication between client and server is handled by using two intermediate objects: Stub object (on client side) and Skeleton object (on server side).  Stub Object  The stub object on the client machine builds an information block and sends this information to the server. The block consists of  An identifier of the remote object to be used  Method name which is to be invoked  Parameters to the remote JVM  Skeleton Object  The skeleton object passes the request from the stub object to the remote object. It performs following tasks  It calls the desired method on the real object present on the server.  It forwards the parameters received from the stub object to the method.[RMI in Java](http://d1hyf4ir1gqw6c.cloudfront.net/wp-content/uploads/RMI-in-Java.png)  Steps to run:   1. Compile Client and Server code. 2. Create stub/skeleton by running command--------rmic FibServer. 3. Start naming registry by running command-------start rmiregistry. 4. Run server program. 5. Run client program. | |
| Code | **Client Code:**  import java.rmi.\*;  public class MyClient{  public static void main(String args[]){  try{  Adder stub=(Adder)Naming.lookup("rmi://localhost:5000/Add");  System.out.println(stub.add(34,4));  }catch(Exception e){} } }  **SERVER CODE:**  import java.rmi.\*;  import java.rmi.server.\*;  public class AdderRemote extends UnicastRemoteObject implements Adder{  AdderRemote()throws RemoteException{  super();  }  public int add(int x,int y){return x+y;}  }  **INTERFACE CODE:**  import java.rmi.\*;  public interface Adder extends Remote{  public int add(int x,int y)throws RemoteException; } | |
| Output |  | |
| Conclusion: | Thus we successfully studied RMI communication. | |