43139

Assignment 1C

Distributed application using Java RMI

* Problem Statement:

Power calculation: Design a distributed application which consists of a client server communication using TCP, UDP & RMI techniques in Java. Multiple clients can simultaneously connect to the server and send messages of the format -> (a, b) where a and b are integers and server returns the value a^b (a raised to b)

* Objectives:

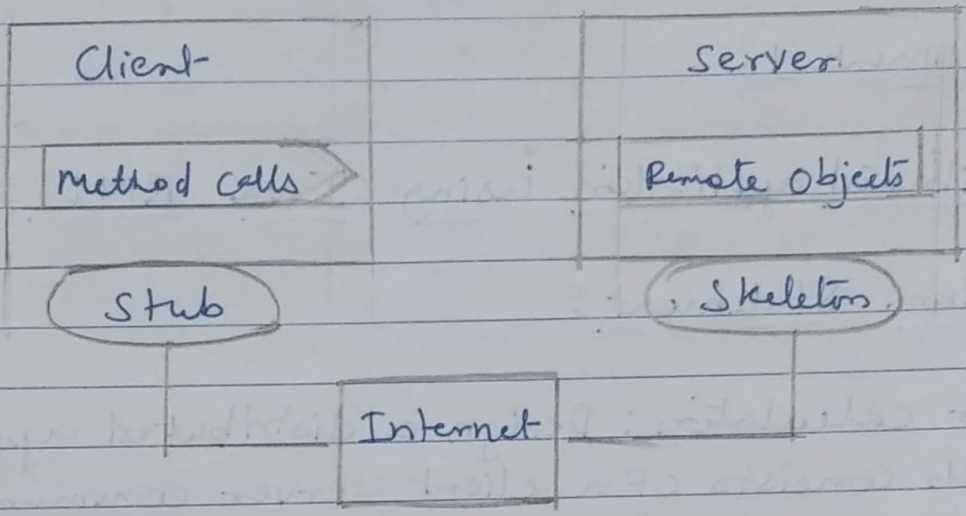
To learn to implement any distributed applications based on RMI

* Theory:

RMI

RMI (Remote Method Invocation) is an API used to access objects running on another JVM (server-side). It is mainly used for the creation of distributed systems and is provided in Java Rome. Stub and skeleton are the two objects used for handling communication between client and server. The following figure shows an overview of RMI

43139



Stub Object :

The stub object on the client machine builds an information block and this information to the server.

Skeleton Objects :

The skeleton object passes the request from the stub object to the remote object. RMI contains a rmiregistry that hold all the server objects. The server binds all the objects to the registry & then the client fetches the object from the respective registry after which the client invokes the methods using the fetched objects.

Steps to calculate a^b using RMI

1) To create remote interfaces we need to extend the remote interface and the prototype within the interface should throw the Remote Exception.

43139

2) To implement the remote interface, the class should extend to the unicast Remote object class of the Java.rmi package. Also a default constructor needs to be created to throw the Java.rmi.RemoteException from its parents constructor.

3) Create and execute the server application program.

4) Create and execute the client application program.

5) Compile all the java program
To compile all the java programs
Command : javac *.java

6) Create a stub and skeleton
The rmic tool is used to invoke the rmi compiler that creates the stub and skeleton objects.
Its prototype is
rmic . classname

7) start the registry service by the rmiregistry tool
Now start the rmic regestry service by using rmiregistry tool. We need to specify port number. default port no is 5259.

43139

* Conclusion :

We have successfully implemented distributed application through client-server communication based on Java - RMI.