**AIM -** Implement multi-threaded client/server Process communication using RMI.

**Name -** Yogita Sunil Girigosavi

**RevNumServerIntf.java**

import java.rmi.\*;

public interface RevNumServerIntf extends Remote

{

int reverseNum(int n) throws RemoteException;

}

**RevNumServerImpl.java**

import java.rmi.\*;

import java.rmi.server.\*;

public class RevNumServerImpl extends UnicastRemoteObject implements RevNumServerIntf

{

public RevNumServerImpl() throws RemoteException

{

}

public int reverseNum(int n) throws RemoteException

{

int reverse = 0;

while(n!= 0)

{

int remainder = n % 10;

reverse = reverse \* 10 + remainder;

n = n/10;

}

return reverse;

}

}

**RevNumServer.java**

import java.net.\*;

import java.rmi.\*;

public class RevNumServer

{

public static void main(String args[])

{

try

{

RevNumServerImpl revnumServerImpl = new RevNumServerImpl();

Naming.rebind("REV-NUM-SERVER", revnumServerImpl);

}

catch(Exception e)

{

System.out.println("Exception: " + e);

}

}

}

**RevNumClient.java**

import java.util.\*;

import java.rmi.\*;

public class RevNumClient extends Thread implements Runnable

{

public void run()

{

System.out.println("Thread is running...");

System.out.println(Thread.currentThread().getName());

try

{

String revnumServerURL = "rmi://localhost/REV-NUM-SERVER";

RevNumServerIntf revnumServerIntf = (RevNumServerIntf)Naming.lookup(revnumServerURL);

Scanner sc= new Scanner(System.in);

int n;

System.out.println("Enter number : ");

n = sc.nextInt();

System.out.println("The reverse number is: " + revnumServerIntf.reverseNum(n));

}

catch(Exception e)

{

System.out.println("Exception: " + e);

}

}

public static void main(String args[])

{

RevNumClient thread1=new RevNumClient();

RevNumClient thread2=new RevNumClient();

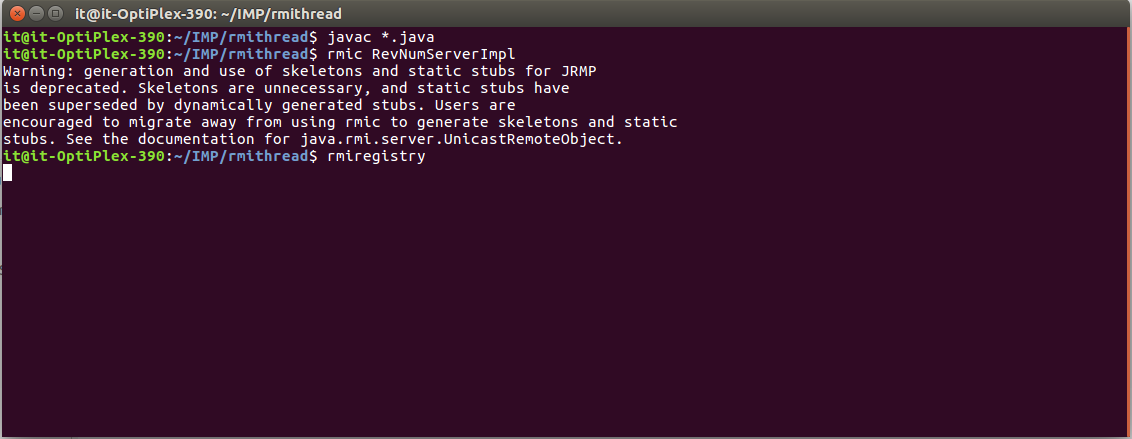
thread1.start();

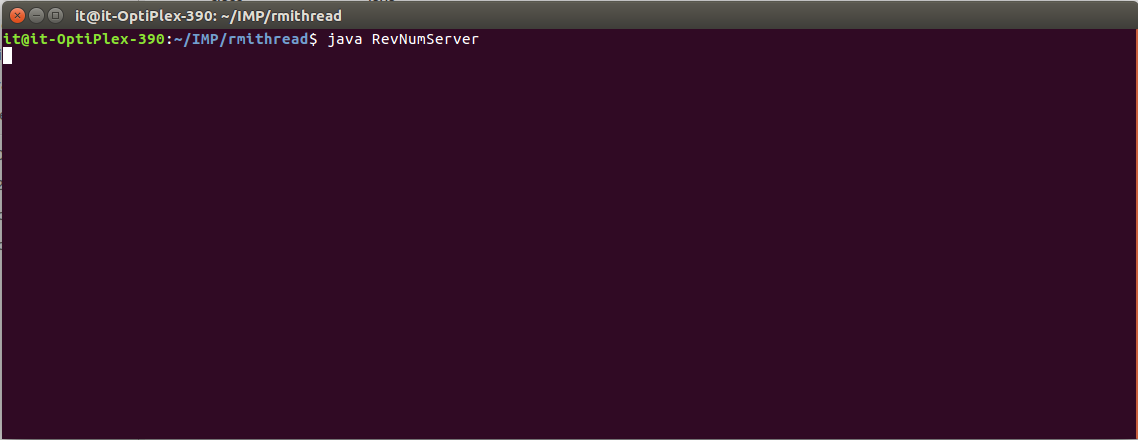
thread2.start();

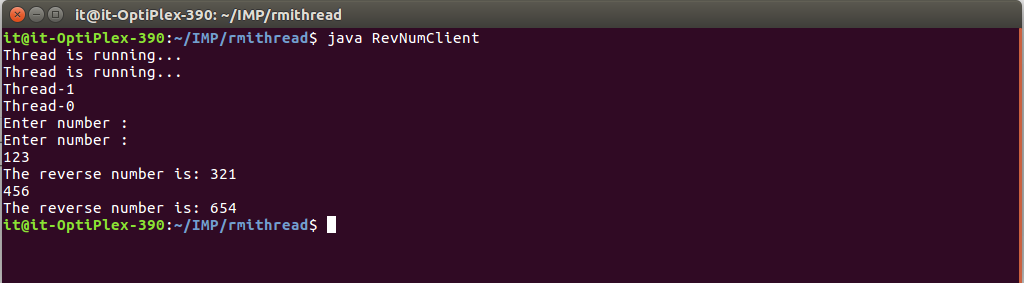
}

}

**Output –**

****

****

****