Name : Ajay Shitkar
Roll No : 58
Experiment No. 03
Implement Group Communication
Date of Performance: 01/03/24
Date of Submission: 15/03/24

Aim: To implement Group Communication.

Objective: Develop a program to implement Group Communication.

Theory:

Group communication refers to the process of exchanging information among a group of processes in a distributed system. In a group communication scenario, a message is sent by a process to a group of processes, and all the processes in the group receive the message. Group communication is used in a wide range of distributed applications, such as group chat, distributed file systems, and distributed databases.

Multicast is a popular group communication protocol that involves the transmission of a message to a group of processes using a single send operation. The message is delivered to all the processes in the group, which allows for efficient dissemination of information among multiple

processes. Multicast can be implemented using various network protocols, such as User Datagram Protocol (UDP) and Transmission Control Protocol (TCP).

The multicast datagram socket class is useful for sending and receiving IP multicast packets. A MulticastSocket is a (UDP) DatagramSocket, with additional capabilities for joining "groups" of other multicast hosts on the internet.

A multicast group is specified by a class D IP address and by a standard UDP port number. Class D IP addresses are in the range 224.0.0.0 to 239.255.255.255, inclusive. The address 224.0.0.0 is reserved and should not be used.

To join a multicast group, first create a MulticastSocket with the desired port, then invoke the joinGroup(InetAddress groupAddr) method.

Code :

GroupSender.java

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.InetAddress;
import java.net.MulticastSocket;
```

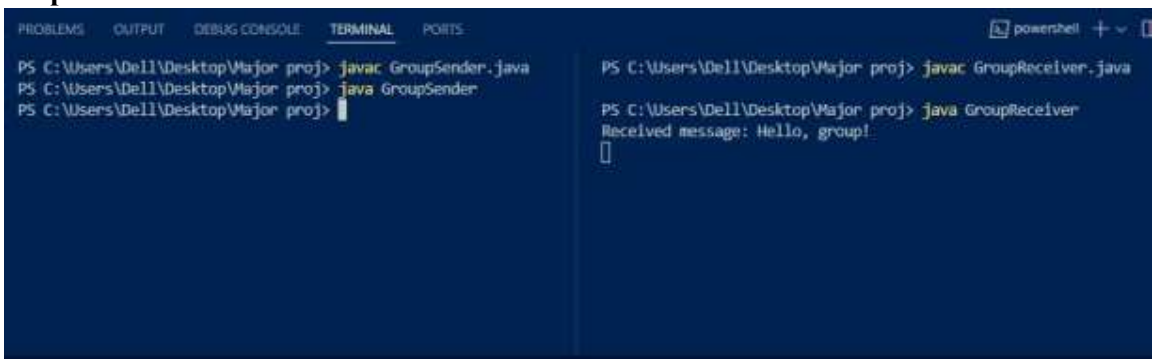
```
public class GroupSender {
public static void
main(String[] args) throws
IOException {
        // Create a multicast socket
        MulticastSocket socket = new MulticastSocket();
        // Set the address and port of the multicast group
        InetAddress address = InetAddress.getByName("230.0.0.1");
        int port = 5000;
        // Create the message to send
        String message = "Hello, group!";
        byte[] data = message.getBytes();
        // Create a DatagramPacket to send the message
        DatagramPacket packet = new DatagramPacket(data, data.length, address, port);
        // Send the message
        socket.send(packet)
        ; // Close the socket
        socket.close();
    }
}
```

GroupReceiver.java

```
import java.io.IOException; import
java.net.DatagramPacket; import java.net.InetAddress; import
java.net.MulticastSocket; public class GroupReceiver { public
static void main(String[] args) throws IOException {
        // Create a multicast socket
        MulticastSocket socket = new MulticastSocket(5000);
        // Join the multicast group
        InetAddress address =
        InetAddress.getByName("230.0.0.1");
        socket.joinGroup(address); while (true) {
            // Create a buffer to store incoming messages
            byte[] data = new byte[1024];
           // Create a DatagramPacket to receive the message
            DatagramPacket packet = new DatagramPacket(data, data.length);
```

```
                // Wait for a message to arrive
                socket.receive(packet);
                // Extract the message from the packet
                String message = new String(packet.getData(), 0, packet.getLength());
                // Print the message
                System.out.println("Received message: " + message);
            }
            // Close the socket (unreachable code)
            // socket.close();
        }
}
```

Output:



Conclusion: This experiment focused on implementing group communication using multicast in Java, achieved through the development of `GroupSender` and `GroupReceiver` programs. Leveraging the MulticastSocket class streamlined multicast communication, enabling efficient message exchange among multiple processes in a distributed system. Demonstrating the process of joining multicast groups and utilizing Datagram Packet for message transmission, the experiment underscored the effectiveness of multicast for distributing information among processes. Successful execution validated the efficacy of the implementation, offering valuable insights into group communication concepts and practical experience in multicast communication within distributed systems.