ANISH ADNANI
D17B/01

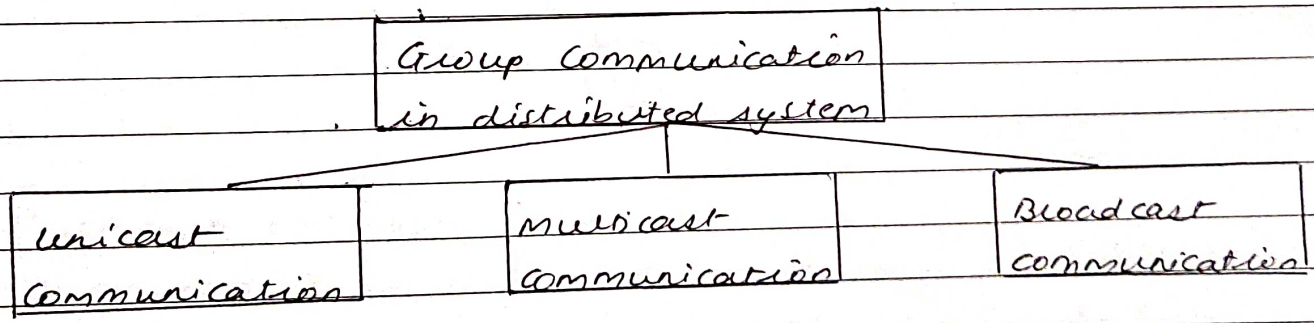
EXPERIMENT 4

Aim: To implement group communication in distributed systems

Theory:

Group Communication

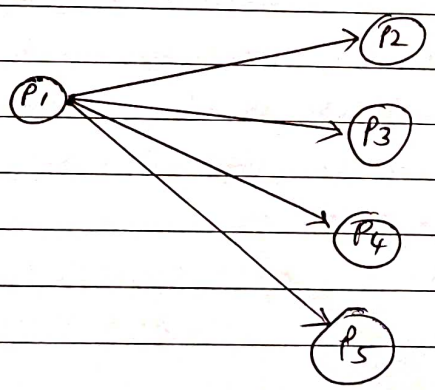
Communication between two processes in a distributed system is required to exchange various data, such as code or a file, between the processes. When one source process tries to communicate with multiple processes at once, it is called as Group Communication. A group is a collection of interconnected processes with abstraction. This abstraction is to hide the message passing so that the communication looks like a normal procedure call. Group communication also helps the processes from different hosts to work together and perform operations in a synchronized manner, therefore increases the overall performance of the system



Types of Group Communication in a Distributed system

- Broadcast communication

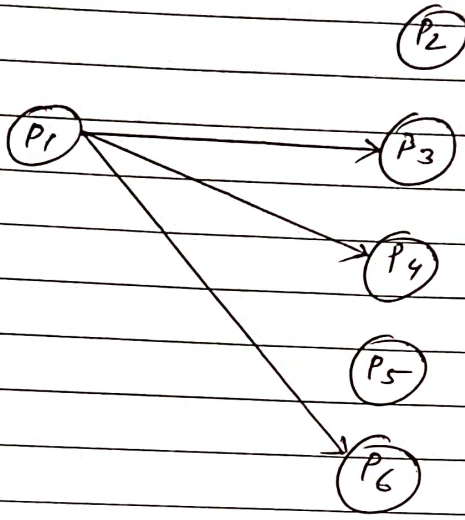
When the host process tries to communicate with every process in a distributed system at same time. Broadcast communication comes in handy when a common stream of information is to be delivered to each and every process in most efficient manner possible since it does not require any processing whatsoever, communication is very fast in comparison to other modes of communication. However, it does not support a large number of processes and cannot treat a specific process individually



- Multicast Communication

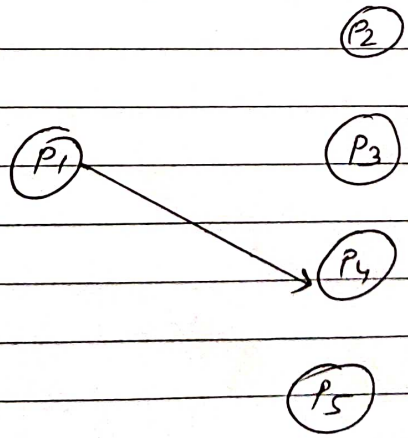
When the host process tries to communicate with a designated group of processes in a distributed system at the same time. This technique is mainly used to find a way to address problem of a high workload on host system and redundant information

from process in system. Multitasking can significantly decrease time taken for message handling



- Unicast Communication

When the host process tries to communicate with a single process in a distributed system at the same time. Although, same information may be passed to multiple processes - This works best for two processes communicating as only it has to treat a specific process only



Conclusion: Thus we have successfully implemented group communication using various ways.

****

Server.java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Vector;
public class Server {
private static Vector<PrintWriter> writers = new Vector<PrintWriter>();
public static void main(String[] args) throws Exception {
ServerSocket listener = new ServerSocket(9001);
System.out.println("The server is running at port 9001.");
while (true)
new Handler(listener.accept()).start();
}
private static class Handler extends Thread {
private Socket socket;
public Handler(Socket socket) {
this.socket = socket;
}
public void run() {
try {
BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
out.println("SUBMITNAME");
String name = in.readLine();
System.out.println(name+" joined");
writers.add(out);
while (true) {
String input = in.readLine();
for (PrintWriter writer : writers)
writer.println("MESSAGE " + name + ": " + input);
}
} catch (Exception e) {System.err.println(e);}
}
}
}
```

****

****

master.java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;
public class master
{
public static void main(String[] args) throws Exception {
Scanner sc = new Scanner(System.in);
Socket socket = new Socket("localhost", 9001);
BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
System.out.print("Enter your name: ");
String name = sc.nextLine();
while (true) {
String line = in.readLine();
if (line.startsWith("SUBMITNAME")) out.println(name);
else if (line.startsWith("MESSAGE"))
System.out.println(line.substring(8));
if(name.startsWith("master")){
System.out.print("Enter a message: ");
out.println(sc.nextLine());
}
}
}
}
```

****

****

slave1.java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;
public class slave1
{
public static void main(String[] args) throws Exception {
Scanner sc = new Scanner(System.in);
Socket socket = new Socket("localhost", 9001);
BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
System.out.print("Enter your name: ");
String name = sc.nextLine();
while (true) {
String line = in.readLine();
if (line.startsWith("SUBMITNAME")) out.println(name);
else if (line.startsWith("MESSAGE"))
System.out.println(line.substring(8));
if(name.startsWith("master")){
System.out.print("Enter a message: ");
out.println(sc.nextLine());
}
}
}
}
```

****

****

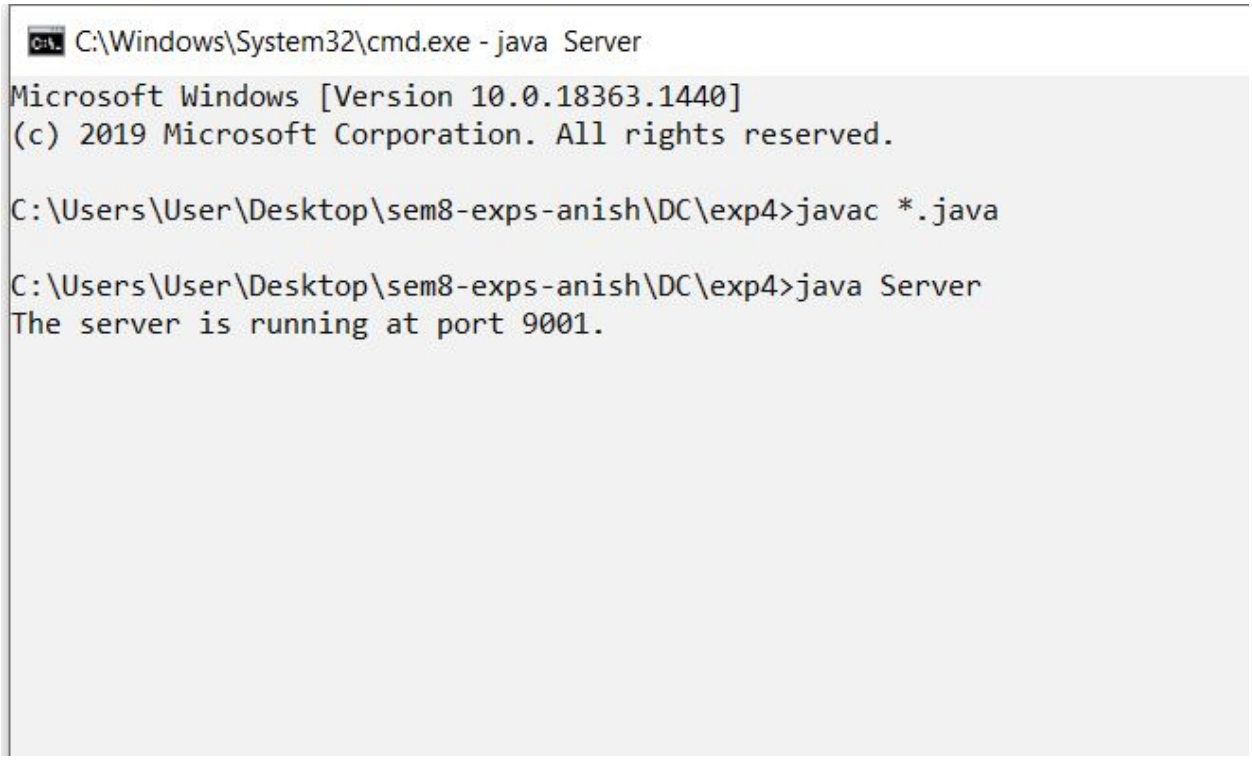
slave2.java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;
public class slave2
{
public static void main(String[] args) throws Exception {
Scanner sc = new Scanner(System.in);
Socket socket = new Socket("localhost", 9001);
BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
System.out.print("Enter your name: ");
String name = sc.nextLine();
while (true) {
String line = in.readLine();
if (line.startsWith("SUBMITNAME")) out.println(name);
else if (line.startsWith("MESSAGE"))
System.out.println(line.substring(8));
if(name.startsWith("master")){
System.out.print("Enter a message: ");
out.println(sc.nextLine());
}
}
}
}
```

****

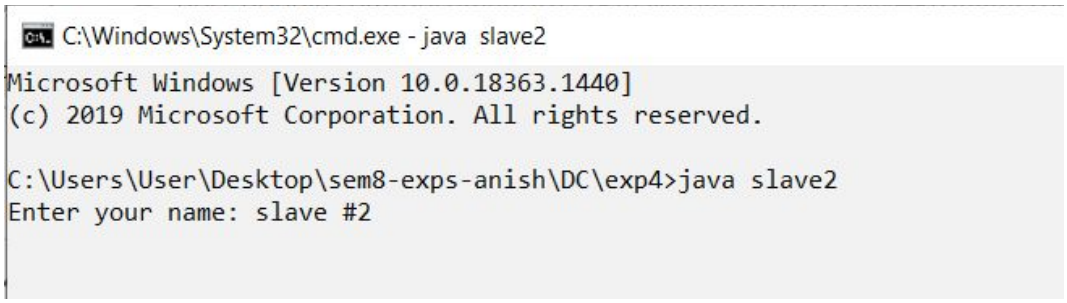
Compiling all java files

Running Server program

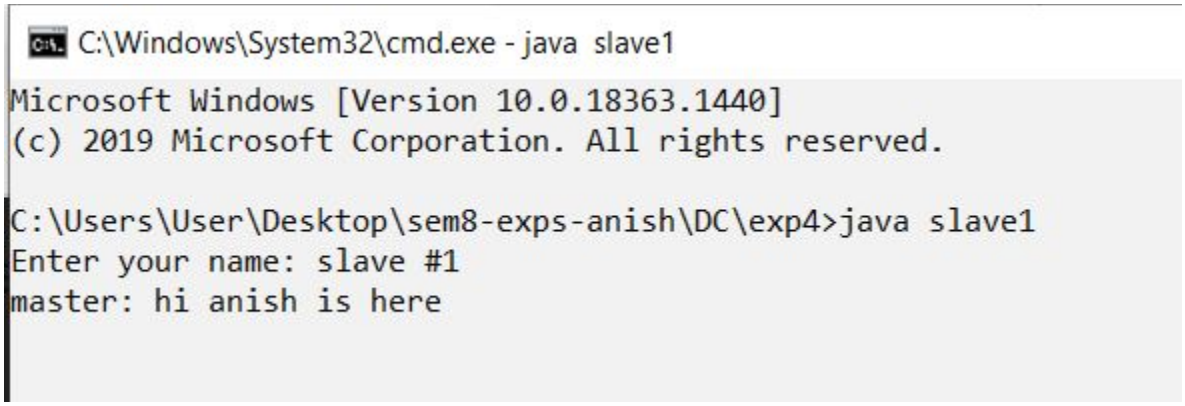


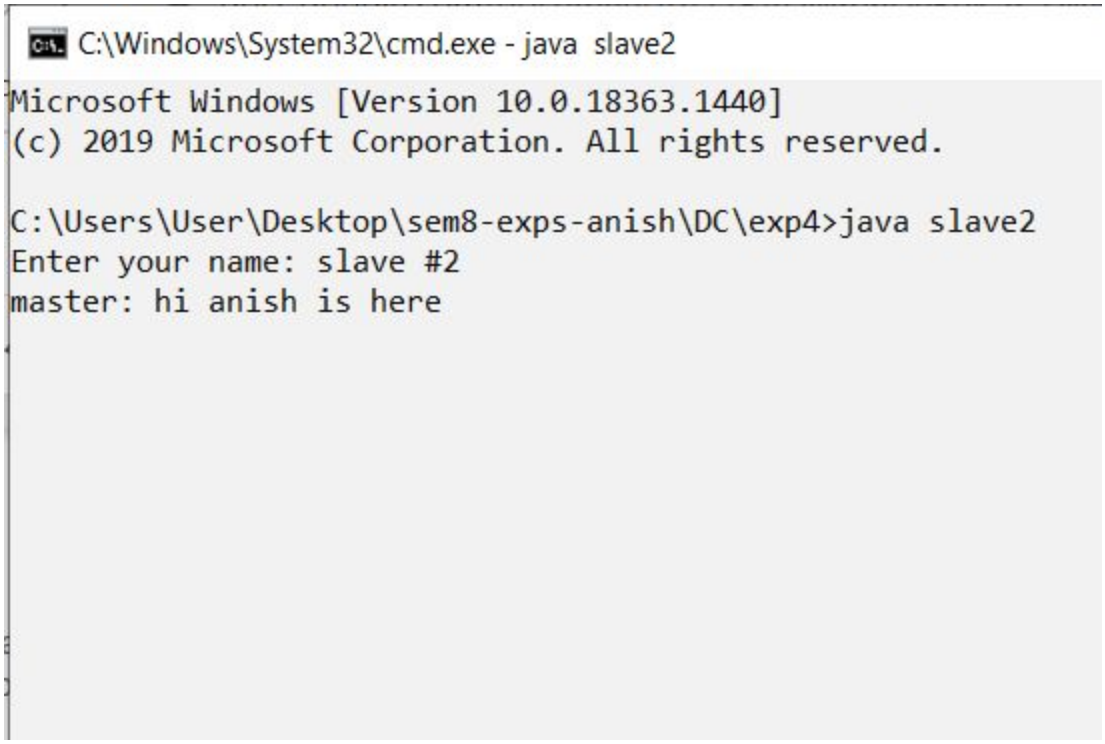
Running master

Running slave1 and slave 2



Master broadcasts the message within the group, which will be delivered at slave1 and slave2 terminals


C:\Windows\System32\cmd.exe - java slave1
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\User\Desktop\sem8-exps-anish\DC\exp4>java slave1
Enter your name: slave #1
master: hi anish is here


C:\Windows\System32\cmd.exe - java slave2
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\User\Desktop\sem8-exps-anish\DC\exp4>java slave2
Enter your name: slave #2
master: hi anish is here