

EXP: BROADCAST MESSAGE FROM SERVER TO CLIENTS

Aim

Write a program to implement broadcasting

Program description

Server program running on one machine, is sending messages to more than one client machines.

Description of broadcasting

Algorithm

Create a class client1

main function

create a socket

create a BufferedReader object, which is used to read data from the socket

do

 read data from the socket

 write data to the screen

end do if data is equal to 'quit'

end of main function

end of class client1

Create a class server1

 create a array of sockets and serversocket

 define thread t1 and t2

 main function

 create object from class server1 and create thread t1 and t2 from that object

 start thread t1 and t2

 end of main function

 function run

 if current thread is t1 then do

 loop

 read the socket of newly connected client and store it into s[i]

 end loop

 end if

 if current thread is t2 then do

 loop

 read data from the keyboard

 for j=1 to I

 create a printwriter object which can be used to write data to the socket s[j]

 write data to s[j] using printwriter object

 end for

 end loop

 end if

 end of run

end of class server1

Program

Server side program

```
import java.net.*;
```

```
import java.io.*;
```

```

import java.lang.*;
class Server1 implements Runnable
{
int i=0,j=0;
String msg;
ServerSocket ser;
Thread t1=null,t2=null;
Socket soc[] =new Socket[20];
BufferedReader br;
PrintWriter pw;
public Server1()
    {
        try{
            ser= new ServerSocket(8000);
            br=new BufferedReader(new InputStreamReader(System.in));
        }
        catch(Exception e){}
    }
public void run()
    {
        if (Thread.currentThread()==t1)
            {try {for(;;)
                {i=i+1;
                soc[i]=ser.accept();
                System.out.println("client"+i+"connected");
                }
            } catch(Exception e){}
        }
        else
            {
                try{do
                    {msg=br.readLine();
                    for(j=1;j<i;j++)
                        {pw=new PrintWriter(soc[j].getOutputStream(),true);
                        pw.println(msg);
                        }
                    } while(!msg.equals("quit"));
                } catch(Exception e){}
            }
        System.exit(0);
    }
    public static void main(String args[]) {
        Server1 ser =new Server1();
        ser.t1=new Thread(ser);
        ser.t2=new Thread(ser);
        ser.t1.start();
        ser.t2.start();
    }
}

```

Client side program

```
import java.io.*;
import java.net.*;
import java.lang.*;
class client1 {
    String msg;
    Socket s;
    BufferedReader br;
    public client1(){
    try{
        s=new Socket("localhost",8000);
        System.out.println("Client started");
        System.out.println("message from the server");
        br=new BufferedReader(new InputStreamReader(s.getInputStream()));
        do{
            msg=br.readLine();
            System.out.println(msg);
        } while(!msg.equals("quit"));
    } catch(Exception e){}
    }
    public static void main(String args[]){

        client1 cl=new client1();

    }
}
```

}

Test cases

Case 1:

test the system with one server and one client

Case2:

test the system with one server and more than one client

Case 3:

Run the server program, run one client program, then send message from server to client. After that again run one more client program. Again send message from server to these 2 clients

OUTPUT

```
C:\WINDOWS\system32\cmd.exe

E:\JAVA\bin>java Server1
client1connected
hai client1
client2connected
now we have 2 clients
client3connected
now we have 3 clients
quit

E:\JAVA\bin>
```

```
C:\WINDOWS\system32\cmd.exe

E:\JAVA\bin>java client1
Client started
message from the server
hai client1
now we have 2 clients
now we have 3 clients
quit

E:\JAVA\bin>
```

```
C:\WINDOWS\system32\cmd.exe

E:\JAVA\bin>java client1
Client started
message from the server
now we have 2 clients
now we have 3 clients
quit

E:\JAVA\bin>
```

```
C:\WINDOWS\system32\cmd.exe

E:\JAVA\bin>java client1
Client started
message from the server
now we have 3 clients
quit

E:\JAVA\bin>
```