ANISH ADNANI
D17B/01

DC

Page No. 1
Date

EXPERIMENT 3

Aim: Write a program to demonstrate inter-process communication in client server environment

Theory:

Interprocess communication: It is a process of exchanging the data between two or more independent process in a distributed environment interprocess communication on the internet provides with both datagram and stream communication

→ Examples of interprocess communication:

1) N number of applications can communicate with the x-server through network protocols
2) Servers like Apache spawn child process to handle requests

→ It has two functions:

1) Synchronization: Exchange of data is done synchronously which means it has a single clock pulse

2) Message Passing: When processes wish to exchange information, message passing takes several forms such as: pipes, FIFO, shared memory and message queues.

Teacher's Sign.:

Characteristics of interprocess communication

→ There are mainly five characteristics of interprocess communication in a distributed environment/ system

1) Synchronous system calls: In the synchronous system call both sender and receiver use blocking system calls to transmit the data which means the sender will wait until the acknowledgement is received and receiver waits until the message arrives

2) Asynchronous system calls: In the asynchronous system calls, both sender and receiver use non-blocking system calls to transmit the data which means the sender doesnt wait for the reciver acknowledgement

3) message Destination: A local port is a message destination within a computer, specified as an integer. A port has exactly one receivers but many senders processes may use multiple ports from which to receive messages. Any process that known the number of ports can send message to it.

4) Reliability: It is defined as validity and integrity
. Integrity: message must arrive without corruption and duplication to the destination.

- Validity: point to point message services are defined as reliable if message are guaranteed to be delivered.

5) Ordering: It is the process of delivering message to the receiver in a particular order

Sockets:

→ Sockets allow communication between 2 different processes on the same or different machines
A socket is used in a client-server application framework. A server is a process that performs same functions on request from a client. most of the application level protocols like FTP, SMTP, make use of socket to exchange data.

Socket type:

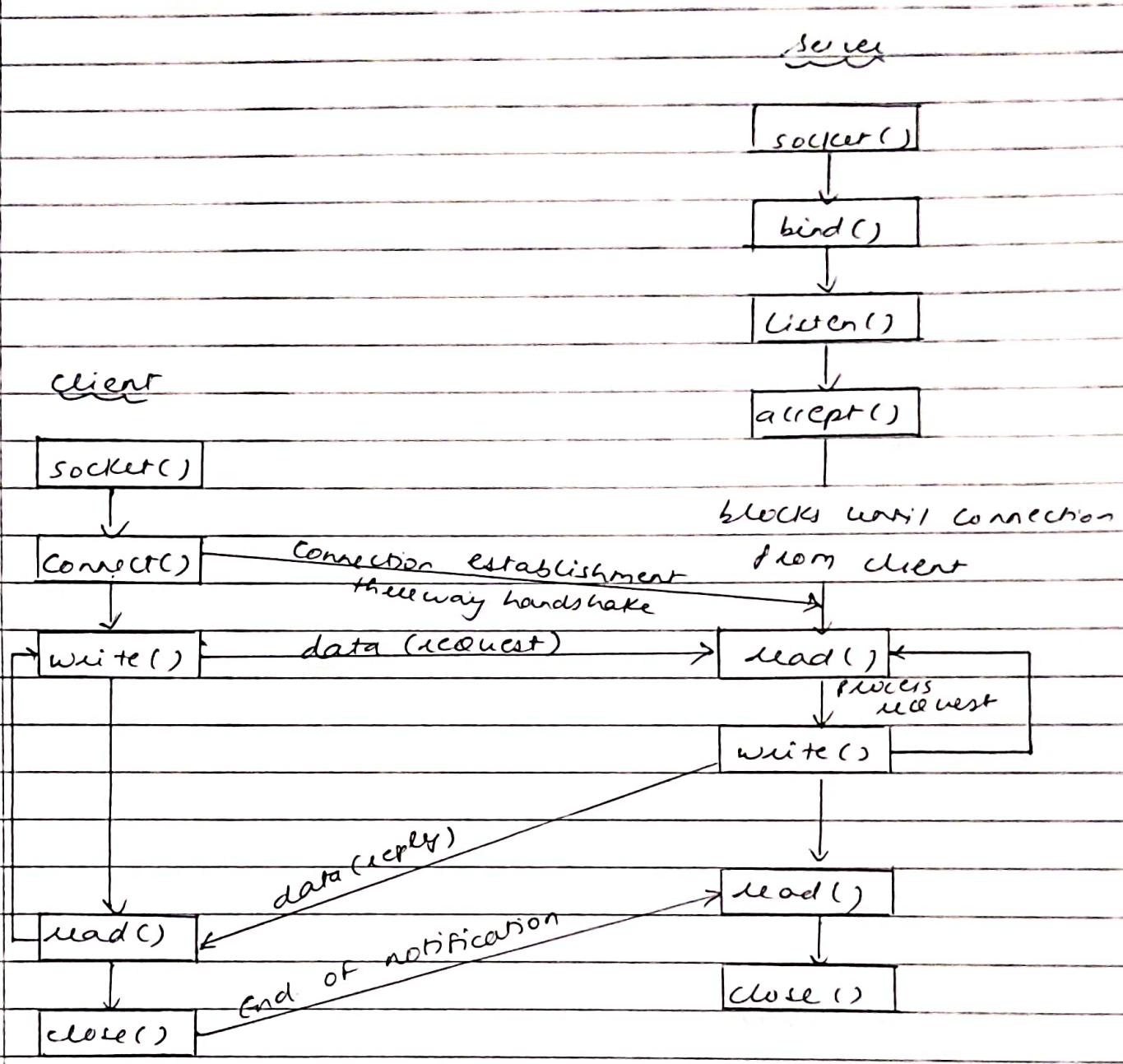
a) Stream sockets: Delivery in a networked environment is required/guaranteed. if you send through the stream socket three items "A, B, C", they will arrive in same order "A, B, C"

b) Datastream sockets: Delivery in a networked environment is not guaranteed. They are connectionless because you dont need to have an open connection as in stream socket

c) Raw sockets: These provide user access to the underlying communications protocols which support socket abstraction

d) sequence packet sockets: They are similar to a stream socket, with the exception that record boundaries are preserved.

Primitive	Meaning
• SOCKET	Create new communication and point
• BIND	Attach a local address to socket
• LISTEN	Announce willingness to accept connections
• ACCEPT	Block the caller until a connection attempt arrives
• CONNECT	Actively attempt to establish a connection
• SEND	send some data from the connection
• RECEIVE	Receive some data from the connection
• CLOSE	Release the connections



Conclusion: interprocess communication in client-server environment is successfully implemented.

Client Server Program to sort an array provided at client side

Server.java

```
import java.io.*;
import java.net.*;
import java.util.*;
class Server
{
	public static void main (String args[]) throws Exception
	{
		ServerSocket ss = new ServerSocket(5000);
		Socket s=ss.accept();
		System.out.println("connected.....");
		DataInputStream din=new DataInputStream(s.getInputStream());
		DataOutputStream dout=new DataOutputStream (s.getOutputStream());
		int r, i=0;
		int n=din.readInt();
		int a[]=new int[n];
		System.out.println ("data :");
		int count=0;
		System.out.println ("Receiving Data....");
		for(i=0;i<n;i++)
		{
			a[i] =din.readInt ();
		}
		System.out.println("Data Received is ....");
		for(i=0;i<n;i++)
		{
		System.out.println(a[i]+" ");
		}
		Arrays.sort(a);
		System.out.println("Data Sorted");
		System.out.println("Sending Data.....");
		for(i=0;i<n;i++)
		{
			dout.writeInt(a[i]);
			System.out.print(a[i]+" ");
		}
		System.out.println("\nData Sent successfully");
		s.close();
		ss.close ();
	}
}
```

Client.java

```
import java.io.*;
import java.net.*;
import java.util.Scanner;
public class Client {
	public static void main(String[] args) throws Exception
	{
			Socket s=new Socket("127.0.0.1",5000);
			if(s.isConnected())
			{
				System.out.println("Connected to server");
			}
			System.out.println("Enter size of array:");
			Scanner scanner=new Scanner(System.in);
			int n=scanner.nextInt ();
			int a[]=new int[n];
			System.out.println("Enter element to array:");
			DataOutputStream dout=new DataOutputStream(s.getOutputStream());
			dout.writeInt(n);
			for(int i=0;i<n;i++)
			{
				int r=scanner.nextInt();
				dout.writeInt(r);
			}
			System.out.println("Data Sent");
			DataInputStream din=new DataInputStream(s.getInputStream());
			int r;
			System.out.println("Receiving Sorted Data....");
			for(int i=0;i<n;i++)
			{
				r=din.readInt();
				System.out.print(r+" ");
			}
			s.close ();
	}
}
```

Step1 : Compile the java files

C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\User\Desktop\sem8-exps-anish\DC\exp3\sort>javac *.java

C:\Users\User\Desktop\sem8-exps-anish\DC\exp3\sort>
```

Step2 : Run Server

C:\Windows\System32\cmd.exe - java Server

```
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\User\Desktop\sem8-exps-anish\DC\exp3\sort>javac *.java

C:\Users\User\Desktop\sem8-exps-anish\DC\exp3\sort>java Server
```

Step3 : Run Client

C:\Windows\System32\cmd.exe - java Server

```
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\User\Desktop\sem8-exps-anish\DC\exp3\sort>javac *.java

C:\Users\User\Desktop\sem8-exps-anish\DC\exp3\sort>java Server
connected..........
```

C:\Windows\System32\cmd.exe - java Client

```
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\User\Desktop\sem8-exps-anish\DC\exp3\sort>java Client
Connected to server
Enter size of array:
```

Output

Client Side

C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\User\Desktop\sem8-exps-anish\DC\exp3\sort>java Client
Connected to server
Enter size of array:
4
Enter element to array:
4
5
2
1
Data Sent
Receiving Sorted Data....
1 2 4 5
C:\Users\User\Desktop\sem8-exps-anish\DC\exp3\sort>
```

Server Side

C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\User\Desktop\sem8-exps-anish\DC\exp3\sort>javac *.java

C:\Users\User\Desktop\sem8-exps-anish\DC\exp3\sort>java Server
connected..........
data :
Receiving Data....
Data Received is ....
4
5
2
1
Data Sorted
Sending Data........
1 2 4 5
Data Sent successfully

C:\Users\User\Desktop\sem8-exps-anish\DC\exp3\sort>
```