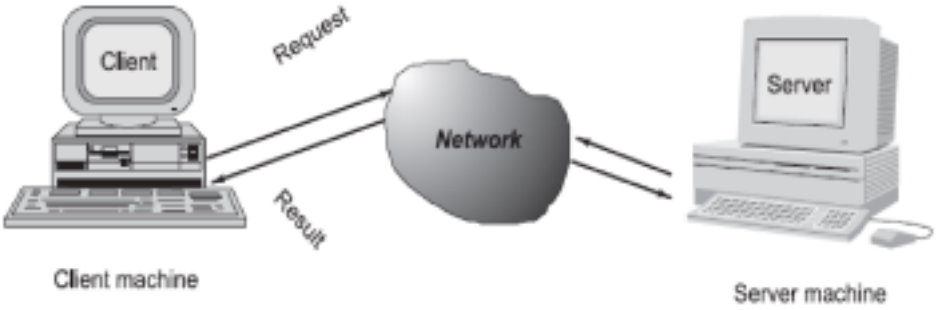
PRACTICAL NO. 1
Remote Process Communication

LOB1: Understand the basics of remote process communications and Mutual Exclusion.
LO1: Develop Remote Process Communication and Mutual Exclusion programs.

1. **Theory:**

- **Client-server communication:**

At a basic level, network-based systems consist of a server, client, and a media for communication. A computer running a program that makes a request for services is called client machine. A computer running a program that offers requested services from one or more clients is called a server machine. The media for communication can be wired or wireless network.



Generally, programs running on client machines make requests to a program (often called as server program) running on a server machine. They involve networking services provided by the transport layer, which is part of the Internet software stack, often called TCP/IP (Transport Control Protocol /Internet Protocol) stack. The transport layer comprises two types of protocols, TCP (Transport Control Protocol) and UDP (User Datagram Protocol). The most widely used programming interfaces for these protocols are sockets. TCP is a connection-oriented protocol that provides a reliable flow of data between two computers. Example applications that use such services are HTTP, FTP, and Telnet. UDP is a protocol that sends independent packets of data, called datagrams, from one computer to another with no guarantees about arrival and sequencing.

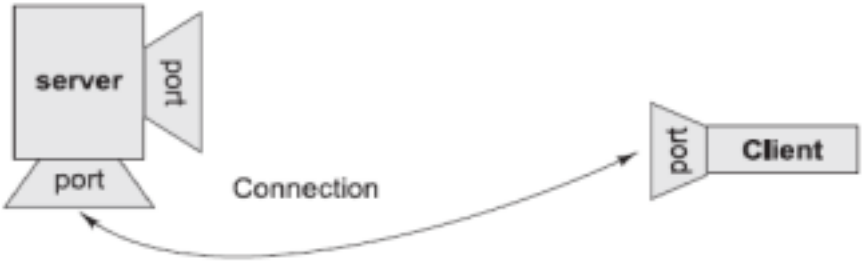
- **Sockets and Socket-based Communication:**

Sockets provide an interface for programming networks at the transport layer. Network communication using Sockets is very much similar to performing file I/O. In fact, socket handle is treated like file handle. The streams used in fi le I/O operation are also applicable to socket-based I/O. Socket-based communication is independent of a programming language used for implementing it. That means, a socket program written in Java language can communicate to a program written in non-Java (say C or C++) socket program.

A server (program) runs on a specific computer and has a socket that is bound to a specific port. The server listens to the socket for a client to make a connection request (see Fig.).

If everything goes well, the server accepts the connection (see Fig.). Upon acceptance, the server gets a new socket bound to a different port. It needs a new socket (consequently a different port number) so that it can continue to listen to the original socket for connection requests while serving the connected client.

[a]: a client making a connection request to the server



[b]: session established with temporary ports used for two way communication.

- **Socket Programming and Java.Net Class:**
  - **A socket is an endpoint of a two-way communication link between two programs running on the network.** Socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent.
  - Java provides a set of classes, defined in a package called **java.net**, to enable the rapid development of network applications.
  - Key classes, interfaces, and exceptions in java.net package simplifying the complexity involved in creating client and server programs are:

The Classes

- ContentHandler
- DatagramPacket
- DatagramSocket
- DatagramSocketImpl
- HttpURLConnection
- InetAddress
- MulticastSocket
- ServerSocket
- Socket
- SocketImpl
- URL
- URLConnection
- URLEncoder
- URLStreamHandler

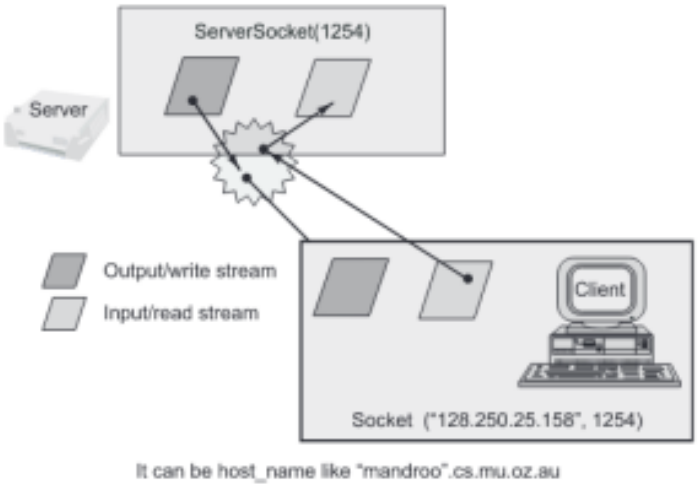
The Interfaces

- ContentHandlerFactory
- FileNameMap
- SocketImplFactory
- URLStreamHandlerFactory

Exceptions

- BindException
- ConnectException
- MalformedURLException
- NoRouteToHostException
- ProtocolException
- SocketException
- UnknownHostException
- UnknownServiceException

- **TCP/IP Socket Programming:**



- The two key classes from the java.net package used in creation of server and client programs are:
  1. ServerSocket
  2. Socket
- A server program creates a specific type of socket that is used to listen for client requests (server socket), In the case of a connection request, the program creates a new socket through which it will exchange data with the client using input and output streams.
- The socket abstraction is very similar to the file concept: developers have to open a socket, perform I/O, and close it.

- **Exercise:**
  1. Write a java program to implement client server application using TCP.
  2. Develop a JAVA program for multi-client chat server using Socket.