**UNIT -07**

**Soket Programming**

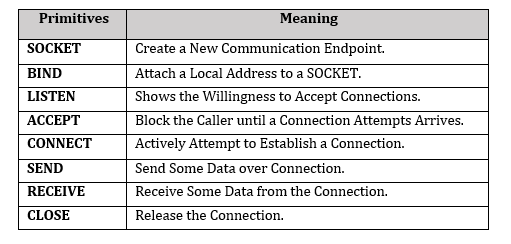
**Socket:**

1. Sockets are a service provided by transport layer.
2. A socket is one endpoint of a two-way communication link between two programs running on the network.

**Berkeley Socket:**

1. Berkeley sockets is an application programming interface (API) for Internet sockets and UNIX domain sockets.
2. It is used for inter-process communication (IPC).
3. It is commonly implemented as a library of linkable modules.
4. It originated with the 4.2BSD UNIX released in 1983.

**Primitive used in Berkeley Socket:**



**Socket Programming:**

**I) Server side:**

 Server startup executes SOCKET, BIND & LISTEN primitives.

 LISTEN primitive allocate queue for multiple simultaneous clients.

 Then it use ACCEPT to suspend server until request.

 When client request arrives: ACCEPT returns.

 Start new socket (thread or process) with same properties as original, this handles the request, server goes on waiting on original socket.

 If new request arrives while spawning thread for this one, it is queued.

 If queue full it is refused.

**II) Client side:**

 It uses SOCKET primitives to create.

 Then use CONNECT to initiate connection process.

 When this returns the socket is open.

 Both sides can now SEND, RECEIVE.

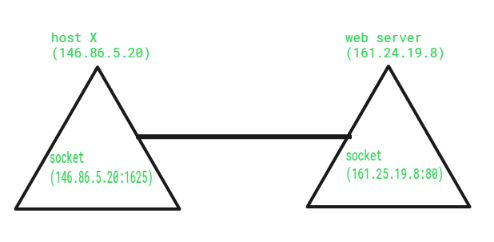
 Connection not released until both sides do CLOSE.

 Typically client does it, server acknowledges.

A socket is one endpoint of a two way communication link between two programs running on the network. The socket mechanism provides a means of inter-process communication (IPC) by establishing named contact points between which the communication take place.

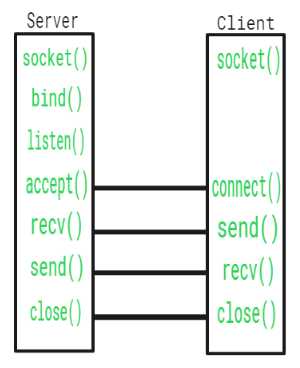
Like ‘Pipe’ is used to create pipes and sockets is created using ‘socket’ system call. The socket provides bidirectional FIFO Communication facility over the network. A socket connecting to the network is created at each end of the communication. Each socket has a specific address. This address is composed of an IP address and a port number.

Socket are generally employed in client server applications. The server creates a socket, attaches it to a network port addresses then waits for the client to contact it. The client creates a socket and then attempts to connect to the server socket. When the connection is established, transfer of data takes place.



Types of Sockets :  
There are two types of Sockets: the datagram socket and the stream socket.

1. Datagram Socket :  
   This is a type of network which has connection less point for sending and receiving packets. It is similar to mailbox. The letters (data) posted into the box are collected and delivered (transmitted) to a letterbox (receiving socket).
2. Stream Socket  
   In Computer operating system, a stream socket is type of [**interprocess communications**](https://www.geeksforgeeks.org/inter-process-communication-ipc/) socket or network socket which provides a connection-oriented, sequenced, and unique flow of data without record boundaries with well defined mechanisms for creating and destroying connections and for detecting errors. It is similar to phone. A connection is established between the phones (two ends) and a conversation (transfer of data) takes place.



| **Function Call** | **Description** |
| --- | --- |
| Create() | To create a socket |
| Bind() | It’s a socket identification like a telephone number to contact |
| Listen() | Ready to receive a connection |
| Connect() | Ready to act as a sender |
| Accept() | Confirmation, it is like accepting to receive a call from a sender |
| Write() | To send data |
| Read() | To receive data |
| Close() | To close a connection |

## Socket addresses

An application can communicate with a remote process by exchanging data with TCP/IP by knowing **the combination of protocol type, IP address, and port number.** This combination is often known as a *socket address*. It is the network-facing access handle to the network socket. The remote process establishes a network socket in its own instance of the protocol stack, and uses the networking API to connect to the application, presenting its own socket address for use by the application.