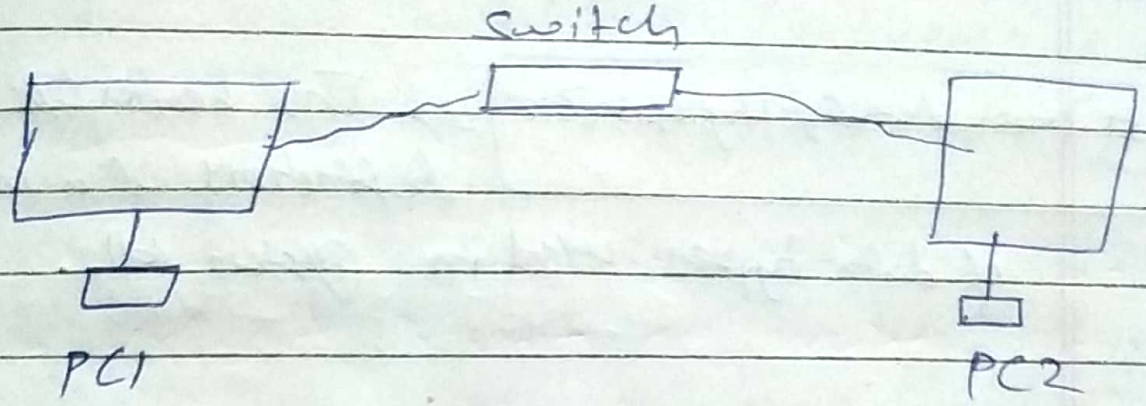
Socket Programming



Uses :-

1) Chat App
2) Client - Server Calculator
3) File transferring

Files includes in Socket Programming

```
# include <stdio.h>
# include <sys/types.h>
# include <sys/socket.h>
# include <netinet/in.h>
# include <stdlib.h>
```

in C-lang.

NDPC
Page No.:
Date : / /20

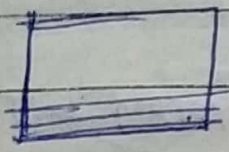
`#include<sys/types.h>` ⟹ This headerfile contains definitions of a number of data types used in system calls.

`#include <sys/socket.h>` ⟹ The header file socket.h includes a number of definitions of structures needed for sockets. Eg:- defines the sockaddr structure

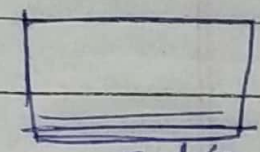
`#include <netinet/in.h>` ⟹ The headerfile in.h contains constants and structures needed for internet domain addresses. Eg:- sockaddr_in (we will be using this structure)

NDPC
Page No.:
Date : / / 20

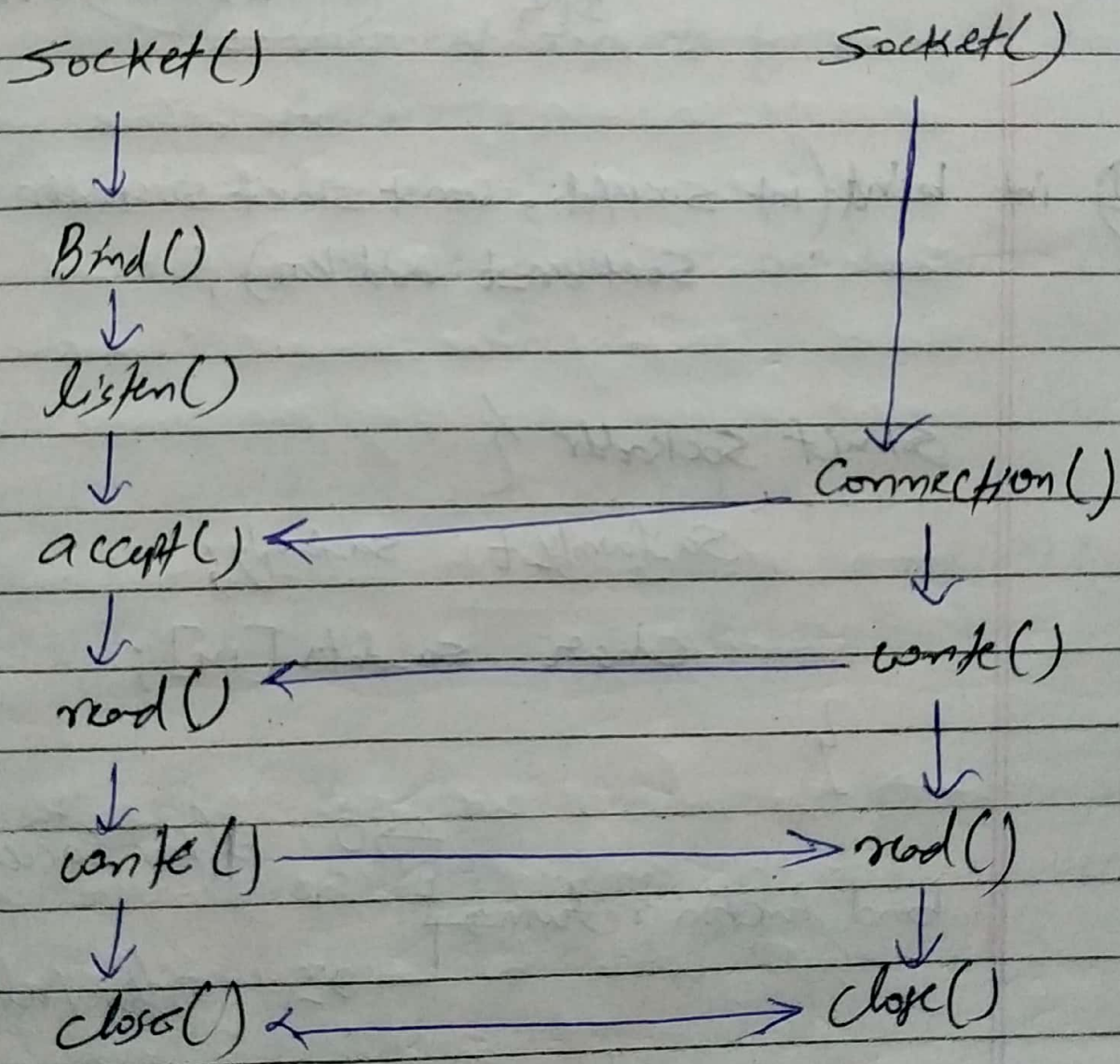
client - server model # (Sys/socket.h functions)



Server



Client



NDPC
Page No.:
Date : / /20

(1) int sockfd = socket(int domain, int type, int protocol)

0 (default for TCP)

AF_INET → IPv4

SOCK_STREAM → TCP
or
SOCK_DGRAM → UDP

(2) int bind(int sockfd, const struct sockaddr *addr, socklen_t addrlen);

```
struct sockaddr {
    sa_family_t  sa_family;
    char         sa_data[14];
};
```

bind function return → 0 ⇒ (for success)
→ -1 ⇒ (for failure)

→ we use bind function to associate an address with a socket.

NDPC
Page No.
Date: / /20

(3) int listen (int sockfd, int backlog)

↳ No. of Connections a system can Handle at single time

→ A server announces that it is willing to accept connect requests by calling the listen function.

→ Returns 0 if ok, 1 on error

(4) newsockfd = accept (sockfd, (struct sockaddr*) &addr, &addrlen)

return: file(socket)des, if ok else return 1 on error

→ file descriptor returned by accept is a socket descriptor that is connected to the client that called connect.

→ once a server has called listen, the socket used can receive connect requests, we use the accept function to retrieve a connect request and convert that into a connection.

Page No.:
Date : / /20

(5) int Connect(int sockfd, ~~const struct~~
Const struct sockaddr *addr,
socklen_t len);

→ If we are dealing with a connection-oriented ~~network~~ service (SOCK_STREAM or SOCK_SEQPACKET), then before we can exchange data, we need to create a connection between the socket of the process requesting the service (the client) and the process providing the service (the server).

→ We Connect function to Create a Connection.

→ The address we specify with connect is the address of the server with which we wish to communicate.
↳ If sockfd is not bound to an address, Connect will bind a default address for the caller.

→ Returns: 0 if OK, 1 on error

NDPC
Page No.:
Date: / / 20

(6) int read(newsockfd, buffer, buffer_size)

(7) int write(newsockfd, buffer, buffer_size)

# struct SockAddr_in {
    u_short sin_family; // always AF_INET
    u_short sin_port; // the service port
    struct in_addr sin_addr; // the IP address
    char sin_zero[8]; // unused (reserved for extension)
};

struct in_addr {
    in_addr_t s_addr; // the IP address in network byte order.
};

NDPC
Page No.:
Date : / /20

struct hostent {
    char *h_name;
    char **h_aliases;
    short h_addrtype;
    short h_length; // length in bytes of each address
    char **h_addr_list;
};

The hostent structure is used by functions to store information about a given host, such as host name, IPv4 address,