18CSC302J (Computer Networks Lab)
Lab session 2 - BASIC FUNCTIONS OF SOCKET PROGRAMMING

Name :- Puneet Sharma
Reg. No. :- RA1911003010331
Class :-CSE F1

1. **socket()** – – create an endpoint for communication.

   int **socket**(int domain,int type,int protocol);

- ❖ Socket creates an endpoint for communication and returns a descriptor.
- ❖ The domain parameter specifies a common domain this selects the protocol family which will be used for communication.
- ❖ These families are defined in <sys/socket.h>.
- ❖ This call creates an unnamed socket and returns a file descriptor to the calling process. usage : int socket(int domain, int type, int protocol)
  Eg: **sockfd=socket(AF_INET,SOCK_STREAM,0);** here AF_INET means the the communication is over the internet domain.SOCL_STREAM indicates its a stream type of communication and 0 indicates the protocol used is TCP/IP.

2. **connect()** – initiate a connection on a socket.

   int connect(int sockfd,const (struct sockaddr*)serv_addr,socklen_t addrlen);

- ❖ The file descriptor sockfd must refer to a socket.
- ❖ If the socket is of type SOCK_DGRAM then the serv_addr address is the address to which datagrams are sent by default and the only addr from which datagrams are received.
- ❖ If the socket is of type SOCK_STREAM or SOCK_SEQPACKET , this call attemptsto make a connection to another socket.
- ❖ **connect()** – is a function used by the client to establish a connetion to the server. It takes 3 arguements: usage: connect(sockfd, host_to_which_itconnects, sizeof_addr);
  Eg: **connect(sockfd,(struct sockaddr*)&serv_addr,sizeof(Serv_addr))<0);**

3. **accept() -**accept/reject job is sent to a destination.

- ❖ accept instructs the printing system to accept print jobs to the specified destination.
- ❖ The –r option sets the reason for rejecting print jobs.
- ❖ The –e option forces encryption when connecting to the server.
- ❖ **accept()** is a system call that causes the process to block until the client connects to the server. it returns a new descriptor and all communication should be carried out using the new file descriptor. usage: int accept(sockfd,pointer_to_address_of client, addr_storing_size_of_client_address); Eg: **newsockfd= accept(sockfd, (struct sockaddr *)&cli_addr,&clilen);** here **cli_len=sizeof(cli_addr);** so the newsockfd has the new socket address which will be used for communication. So, this command blocks until the read() of data is complete that is till the client has finished its write().

5. bind()-

- It is a system call that binds a socket to an address. Here, the address would be the IP address of the current machine and the port number.
- usage : bind(socket_fd, pointer_of_address_its_bound_to, size_of_address);
- Eg: bind(sockfd, (struct sockaddr *)&serv_addr,sizeof(Serv_addr)_); On failure, it returns a value less than zero.

6. Listen()

- This call allows a process to listen on socket for communication.

- usage: listen(socket_fd, no_of_waiting_connections);
- so it takes in a socket file descriptor and the no. of connections waiting while the process is handling a particular connection. so they wait in a blocking queue.
- Eg : listen(sock_fd,5); so 5 connections can wait at the max.

7. recv (Socket-Desc, Data-Buffer, Data-Buffer-length, Flags)

- This function is used to receive data.

- Socket-Desc is the socket descriptor that receives the data.
- Data-Buffer specifies where the data received is stored.
- Data-Buffer-Length is the size of the buffer, in bytes.
- Flags are optional; they affect the way data is received.

8. send (Socket-Desc, Data-Buffer, Flags)

- This function is used to send data.

- Socket-Desc is the socket descriptor used to send data.
- Data-Buffer specifies where the data to be sent is stored.
- Flags are optional; they affect the way data is sent.

9. shutdown (Socket-Desc, How)

- This function is used before the close function in case of a connection-oriented socket because there might be information ready to be sent or received.
- Socket-Desc specifies the socket descriptor to be closed.
- If the How value is 0, receiving bytes is disabled in this socket. If it is 1, sending bytes is disabled on this socket. If it is 2, both sending and receiving bytes is disabled on this socket.

10. close (Socket-Desc)

- This function is used to close the socket and free the Socket-Desc.
- Socket-Desc is the socket descriptor representing the socket to be closed.
- This function does not seem to work in Perl, but it is a socket system call in C under UNIX.

11. select(Socket-Desc/s)

- The select function is used to check the status of a socket descriptor to determine, for example, whether it is ready for reading or writing, or whether it has an error condition pending.