UNIT 1

1. What is network protocol? With a neat block diagram explain the network application for client and server.
2. List out the various approaches used to handle multiple clients at the same time.
3. With a neat block diagram, explain the client and server communication on Local Area Network using TCP.
4. List and explain the steps involved in simple daytime client.
5. Develop the ‘C’ program to implement simple daytime client.
6. Comment on the Protocol Independence. Modify the day time client program for IPv6.
7. What are wrapper functions? Develop the wrapper function for the following:
   1. Socket function
   2. Pthread\_mutex\_lock
8. List and explain the steps involved in simple daytime server.
9. Develop the ‘C’ program to implement simple daytime server.
10. Write a note on Unix errno value.
11. Explain with a neat block diagram the layers of OSI model and Internet protocol suite.
12. List and explain the features of UDP Protocol in detail.
13. List and explain the features of TCP Protocol in detail.
14. Explain with a neat diagrams the following:
    1. TCP connection establishment
    2. TCP data transfer
    3. TCP connection termination
15. Explain the TCP State Transition diagram with a neat diagram.

Unit 2: (Sockets Introduction)

1. What are Value- Result Arguments? Explain the scenario with a neat block diagram.
2. Explain with a neat diagram the various byte ordering functions.
3. Write a note on the Byte Manipulation Functions.
4. Develop a ‘C’ program to determine host byte order.
5. Illustrate the significance of socket functions for elementary TCP client/server with a neat block diagram.
6. Explain the following arguments of the socket function:
   1. Family
   2. Type
   3. Protocol
7. Explain the following functions of TCP socket:
   1. connect
   2. bind
   3. listen
   4. accept
   5. close
8. With a neat block diagram explain the queues maintained by TCP for a listening socket. Also show the packets exchanged during the connection establishment with these two queues.
9. Illustrate the significance of fork and exec functions.
10. Outline the typical concurrent server with the help of pseudocode.
11. Demonstrate the status of client/ server before and after call to *accept* returns with a neat block diagram.
12. Demonstrate the status of client/ server after fork returns with a neat block diagram.
13. Comment on the significance of *getsockname* and *getpeername* functions.
14. Develop the pseudocode that returns the address family of a socket.

**Self-Learning Topics:**

1. Develop the ‘C’ program to demonstrate the TCP echo server: main function
2. Develop the ‘C’ program to demonstrate the TCP echo client: str\_cli function

**Unit 3: (Elementary UDP Sockets)**

1. Illustrate the significance of socket functions for UDP TCP client/server with a neat block diagram.
2. Explain the following functions of UDP socket:
   1. recvfrom
   2. sendto
3. List and explain with a neat block diagram the steps associated with simple UDP echo client and server.
4. Develop the ‘C’ program to demonstrate the UDP echo server: main function
5. Develop the ‘C’ program to demonstrate the UDP echo server: dg\_echo function
6. Develop the ‘C’ program to demonstrate the UDP echo client: main function
7. Develop the ‘C’ program to demonstrate the UDP echo client: dg\_cli function
8. Outline the summary of TCP client/server with two clients.
9. Outline the summary of UDP client/server with two clients.
10. Develop the ‘C’ program for dg\_cli function that verifies returned socket address.
11. Outline the summary of UDP client/server from client’s perspective with a neat block diagram.
12. Outline the summary of UDP client/server from server’s perspective with a neat block diagram.
13. Develop the ‘C’ program to demonstrate the UDP dg\_cli function that calls connect.
14. Develop the ‘C’ program to demonstrate the UDP dg\_cli function that writes a fixed number of datagrams to the server.

**Self-Learning Topics:**

1. Explain the simple SCTP streaming echo client and server with a neat block diagram.

**Unit 4: (Advanced Sockets-I)**

1. With a neat block diagram explain IPv6 server on dual stack host serving IPv4 and IPv6 clients.
2. Explain with a neat block diagram how the received IPv4 and IPv6 datagrams are processed depending on the type of receiving socket.
3. Explain with a neat block diagram how the client requests are processed depending on the address type and socket type.
4. List and explain the numerous ways to start a daemon.
5. List and explain the actions on startup for syslogd Daemon.
6. With a function prototype explain the syslog Function.

**Self- Learning Topics:**

1. Explain the significance of daemon\_init function.
2. List and explain the steps performed by of inetd Daemon.
3. With a neat block diagram explain the inetd descriptors when connection request arrives for TCP port.
4. Explain the following with a neat block diagram:
   1. inetd descriptors in child
   2. inetd descriptors after dup2
5. Develop the pseudocode for daemon\_inetd function to daemonize process run by inetd.

**Unit 5: (Advanced Sockets-II)**

1. Outline the important points with respect to unicasting, multicasting and broadcasting.
2. List and explain the routing protocols which makes use of multicasting.
3. Differentiate between unicast and broadcast.
4. Make use of UDP datagram to understand unicasting.
5. Make use of UDP datagram to understand broadcasting.
6. Explain with a neat block diagram how IPv4 and IPv6 multicast addresses are mapped to Ethernet addresses.
7. What is the format of IPv6 multicast addresses?
8. Illustrate the scope of multicast addresses.
9. Illustrate with a neat block diagram, multicast example of a UDP datagram.
10. Illustrate the NTP packet format and definitions with the help of ntp.h header.