# UNIT I
# ELEMENTARY TCP SOCKETS

1) Compare TCP and UDP

| TCP | UDP |
|---|---|
| It is connection oriented protocol. | It is a connectionless protocol. |
| TCP is sophisticated reliable byte stream protocol | UDP is simple unreliable datagram protocol. |

2) What is the use of ARP and RARP?

ARP (Address Resolution Protocol)

It is used to find the physical address if the IP address is known.

RARP (Reverse Address Resolution Protocol)

It is used to find the IP address if the physical address is known.

3) What is IANA?

Internet Assigned Number Authority. IANA divides the port address.

4) What is mean by well known port?

In well known port, port no and the port address are 16 bits. The range is from 0 to 1023. It is assigned and controlled by IAN. It is used in server process.

5) Compare physical, logical and port address.

| PHYSICAL | LOGICAL | PORT |
|---|---|---|
| It belongs to data link layer | It belongs to network layer | It belongs to transport layer. |
| The size and the format depends upon the network | This is a 32 bit address and is uniquely identified between the node and internet. | This is a 16 bit address. |

6) Write a program to print its own process ID.

```
#include<stdio.h>
#include<stdio.h>
int main(void)
{
pid_t pid;
pid=getpid();
printf("My pid = %d",pid);
}
```

7) Write a program to print the process ID of parent and child.

```
#include<stdlib.h>
#include<stdio.h>
```

```
int main(void)
{
pid_t pid;
pid=fork();
if(pid==0)
printf("Child pid = %d\n",pid);
pid=getppid();
printf("My parent pid = %d\n",pid);
return 0;
}
```

8) Define socket. Give structure of socket.

In client server model two application programs one running on local system(client) and another running on remote system(srever) need to communicate with one another. The communication structure that we need in socket programming is socket.

9) Define the types of socket.

- Stream socket(SOCK_STREAM)
- Datagram socket(SOCK_DGRAM)
- Raw socket(SOCK_RAW)

10) Explain the fields present in sockaddr_in structure.

```
struct sockaddr_in
{
        uint8_t            sin_len;          /*length of the socket*/
        sa_family_t        sin_family;       /*AF_INET*/
        in_port_t          sin_port;         /*16 bit port addr*/
        struct in_addr     sin_addr;         /*32 IPV4  addr*/
        char               sin_zero[8];      /*unused field*/
}

struct int_addr
      {
        in_addr_t    s_addr;      /*32 bit IPV4 addr*/
       };
```

11) Specify the fields in sockaddr_in6.

```
struct int_addr
      {
        uint8_t    sa_addr[16];        /*128 bit IPV6 addr*/
       };
#define SIN6_LEN
struct sockaddr_in6
{
        uint8_t              sin6_len;          /*length of the socket*/
        sa_family_t          sin6_family;       /*AF_INET6*/
```

```
                in_port_t         sin6_port;      /*16 bit addr*/
                uint32_t          sin6_flowinto; /*flow control*/
                struct in6_addr   sin6_addr;      /*128 IPV6 socket addr*/
            };
```

12) What are the data types used in client server program.

The data types used in client server program are

| DATATYPE | DESCRIPTION | HEADER |
|---|---|---|
| int8_t | signed 8-bit integer | <sys/types.h> |
| uint8_t | usigned 8-bit integer | <sys/types.h> |
| int16_t | signed16-bit integer | <sys/types.h> |
| uint16_t | usigned16-bit integer | <sys/types.h> |
| int32_t | signed 32-bit integer | <sys/types.h> |
| uint32_t | usigned 32-bit integer | <sys/types.h> |
| sa_family_t | Address family of socket address structure | <sys/socket.h> |
| socklen_t | length of socket address structure, normally uint32_t | <sys/socket.h> |
| in_addr_t | IPV4 address, normally uint32_t | <netinet/in.h> |
| in_port_t | TCP or UDP port, normally uint16_t | <netinet/in.h> |

13) What is network byte order?

Network byte order chooses its own byte order by the protocols used. It may be either LEBO or BEBO.

14) What is the purpose of inet_aton function?

The purpose of inet_aton function is to convert the dotted decimal format to 32 bit binary format in network byte order.

GENERAL SYNTAX:

```
int  inet_aton(const char *strptr, struct in_addr *addrptr);
```

15) What are the function designed for byte order transformation?

There are four functions designed for byte order transformation. They are as follows:

∗ htons (host to network short)

```
uint16_t htons(uint16_t  host16bitvalue);
```

∗ ntohs (network to host short)

```
uint16_t ntohs(uint16_t  net16bitvalue);
```

∗ htonl (host to network long)

```
uint32_t htonl(uint32_t  host32bitvalue);
```

∗ ntohl (network to host long)

```
uint32_t ntohl(uint32_t  net32bitvalue);
```

16) How a socket is uniquely identified?

The socket function returns a small non-negative integer value, similar to a file descriptor. So we call this to be socket descriptor,

GENERAL SYNTAX:

```
int socket(int family, int type, int protocol);
```

17) What is the purpose of bind and connect?

BIND: Bind function is used to assign the local socket address to already created socket.

GENERAL SYNTAX:

```
int bind(int sockfd, const struct sockaddr *myaddr, socklen_t addrlen);
```

CONNECT: Connect function is used by TCP client to establish active connection with TCP server.

GENERAL SYNTAX:

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

18) What is the need for listen and accept system calls?

LISTEN: Listen function enforce operating system that the server is ready to accept the connection through the socket.

GENERAL SYNTAX:

```
int listen(int sockfd, int backlog);
```

ACCEPT: Accept function is called by a TCP server to return the next completed connection from the front of the completed connection queue.

GENERAL SYNTAX:

```
int accept(int sockfd, struct sockaddr cliaddr, sockaddrlen_t addrlen);
```

19) Specify address conversion functions.

There are five conversion functions. They are as follows:

- inet_aton

  ```
  int  inet_aton(const char *strptr, struct in_addr *addrptr);
  ```

- inet_ntoa

  ```
  char *inet_ntoa(struct in_addr  inaddr);
  ```

- inet_addr

  ```
  in_addr_t inet_addr(const char *strptr);
  ```

- inet_pton

  ```
  int inet_pton(int family, const char *strptr, void *addrptr);
  ```

- inet_ntop

  ```
  const char *inet_ntop(int family, const void *addrptr, char
  *strptr, size_t len);
  ```

20) Explain the operation of fork() system call.

Fork() causes the process to split into two identical process one retains the original process identification number(PID) and is called parent. The other called child gets a new PID.

21) What are the functions used for byte manipulation operations?

Two groups of functions are for byte manipulation

- ★ b (byte)
  - | bzero
  - | bcopy
  - | bcmp
- ★ mem (memory)
  - | memset
  - | memcopy
  - | memcmp

22) What is socket abstraction?

Socket is an abstraction through which an application may send and receive data, in much the same way as an open file allows an application to read and write data to stable storage.

23) Compare various socket address structure.

IPV4
SOCKADDR_IN{}

| | |
|---|---|
| Length | AF_INET |
| 16 bit port# | |
| 32 bit IPV4 addr | |
| Unused 8 bytes | |

IPV6
SOCKADDR_IN6{}

| | |
|---|---|
| Length | AF_INET6 |
| 16 bit port# | |
| 32 bit FLOW LABEL | |
| 128 bit IPV6 addr | |

UNIX
SOCKADDR_UN{}

| | |
|---|---|
| Length | AF_LOCAL |
| Pathname(upto 104 bytes) | |

DATALINK
SOCKADDR_DL{}

| | |
|---|---|
| Length | AF_LINK |
| Interface index | |
| Type | namelen |
| addrlen | sel len |
| Interface name and link layer addr | |

24) Write short notes on sendto and recvfrom system call.

SENDTO:

This function is used by a UDP process to send a message to another process running on remote machine.

GENERAL SYNTAX:

```
ssize_t sendto(int sockfd, const void *buf, size_t buflen, int flags, const struct
                    sockaddr *toaddr, socklen_t toaddrlen);
```

RECVFROM:

This function is used by a UDP process  to send a reply to the sender.

GENERAL SYNTAX:

```
ssize_t recvfrom(int sockfd, const void *buf, size_t buflen, int flags,  struct
                       sockaddr *fromaddr, socklen_t fromaddrlen);
```

25) Differentiate iterative and concurrent server.

| Iterative server | Concurrent server |
|---|---|
| The server run’s only one client at a time. | The server run’s two or more clients at the same  time. |

## 16 MARKS QUESTIONS

1. Explain in detail TCP/IP protocols.
   - Physical & datalink  - all std. Protocols.
   - Network  -  IP, ARP, RARP, ICMP, IGMP
   - Transport – TCP, UDP
   - Application – SMTP, FTP, DNS, SNMP, NFS, RPC, TFTP

2. (i)Compare physical, logical and port address.
   - Physical
   - Logical
   - port

   (ii)Explain byte manipulation functions.
   - B(byte)
     - bzero
     - bcopy
     - bcmp
   - Mem(memory)
     - memset
     - memcpy
     - memcmp

3. (i) What are the data types used in client-server programs?
   - Datatypes
   - Explanation

(ii) What is the purpose of inet_addr and inet_pton functions.

- Syntax
- Explanation

4. (i) What is socket abstraction? Explain.
    - Definition
    - Explanation

   (ii) Compare various socket address structures.
    - IPV4
    - IPV6
    - UNIX domain
    - Datalink

5. (i) How a socket is uniquely identified?
    - Socket function
    - Explanation

   (ii)Explain address conversion functions.
    - inet_aton
    - inet_ntoa
    - inet_addr
    - inet_pton
    - inet_ntop

6. Explain the following system calls.
    a. write
    b. read
    c. recvfrom
    d. listen
        - syntax and explanation

7. Explain the following system calls.
    a. connect
    b. bind
    c. sendto
    d. accept
        - syntax and explanation

8. Explain the required socket functions for implementing a UDP client / server application?
    - Socket functions and explanation
    - Flowchart
    - UDP Client
    - UDP server

9. Write a socket program to implement TCP client server?

- Socket functions and explanation
- Flowchart
- TCP Client
- TCP server

### UNIT-II

## APPLICATIONDEVELOPMENT

1. Define posix?

It's not a single standard; it's a group of standard developed by IEEE. UNIX standardization is done in posix and open group posix is adopted as ISO & IEC.

2. Define signal?

Signal is notification a process that an event has occur. It's also called as software interrupt. The signal can occur asynchronously .The process won't know the time exactly when the signal has occur.

3. Mention the ways to send a signal?

The signal can send by 2 ways,

- Process to Process
- Kernel to process

4. Define Disposition?

Disposition is action associated with the signal. We set the disposition of a signal by calling the sigaction function.

5. What are the three choices for disposition?

1. We can provide a function that is called whenever a specific signal occurs. This action is called catching the signal.
2. Ignore the signal by setting its disposition to SIG-IGN
3. We can set the default disposition for a signal by setting its disposition to SIG_DFL.

6. Specify general format for signal function?

A pointer to a signal handling function is the second argument to the function, as well as the return value from the function.

Sigfunc *signal (int signo, sigfunc *func);
Signo -> signal name or signal no.
Func-> signal handling function (or) SIG_IGN/SIG-DFC

7. What is Handling SIGCHLD signal?

Whenever a process finishes execution its parent will be modified by the kernel by a SIGCHLD. The signal is generated automatically. The parent process will be interrupted. The parent process may choose either ignore, go to system default or catch and handle the signal.

7. What is slow system call?

System call that can block for an undetermined period of time. This is called slow system call.

8. What is meant by wait?

This function is used by parent process to wait for the child process to complete or to terminate. The purpose of this wait is to avoid the child becoming the Zombie state.

```
          pid_t wait(int *statloc);
                return      process id if ok
                            0 or -1 if error
```

wait () based SIGCHLD signal handler

```
          void sig_chld (int signo)
          {
          pid_t  pid;
          int stat;
          pid =wait (&stat);
          printf (pid);
          return;
}
```

9. What is meant by waitpid?

Specify which process to wait for and whether and not to block.

```
          Pid_t waitpid (pid_t pid, int *statloc, int options);
                   return   process id if ok
                             0 or -1 if error
```

10. What is client termination?

When a child want to terminate the connection it send FIN segment to signal child. Whenever child is terminated the kernels automatically generate the SIGCHLD to inform to the parent that child is terminated.

11. Specify termination of server process?

1. Crashing of TCP server process
2. Crashing of TCP server host

12. What is the crashing of TCP server host?

1. Server host crashes it is unreachable
2. Server host gets rebooted
3. Sever host is shutdown by system administrator

13. What are the boundary conditions for crashing of TCP server host?

1. The client host crashes
2. The client process crashes
3. The server process crashes
4. The server host crashes

5. Network connectivity is lost

14. What is server shutdown?

Server is shutdown by the operator, when a UNIX system is shutdown, the init process normally sends the SIGTERM signal to all process waits some fixed amount of time.
Then send SIGKILL signal to any processes still running.

15.What is meant by polling?

An application that lies in a loop calling recvfrom on a non-blocking I/O model is called polling

16. What is I/O multiplexing model?

***** Use any one of the function as select / poll

1. Application process call select system call.
2. Wait for the datagram socket to be readable.
3. When select return that socket is readable, we then call recvfrom to copy the datagram into application buffer.
4. recvfrom returns ok and then process the datagram.

17. Write any one advantage of I/O multiplexing model?

Using select we can wait for more than one descriptor to be ready.

18. Write any one disadvantage of I/O multiplexing model?

Instead of one system call we use the system calls select and recvfrom.

19. Write any two types of I/O operation?

* Synchronous I/O
* Asynchronous I/O

20. What is meant by synchronous I/O with example?

It causes requesting process to be blocked until that I/O operation completes.

E.g.: Blocking, Non- blocking, I/O multiplexing, signal driven I/O Model.

21.What is meant by Asynchronous I/O with example?

It doesn’t cause requesting process to be blocked.

E.g.: Asynchronous I/O model.

22. What is select function?

Select function monitor several sockets at the same time. This function return ready socket. select function allows the process to instruct the kernel to wait for any one of multiple events to occur and to wake up the process only when one or more of the events occurs or when a specified amount of time has passed.

23. Describe the general format of select function?

The general format is,

```
int select (int maxfd1, fd_set *readset, fd_set *writeset, fd_set *exceptset,
        const struct timeval *timeout);
              return     positive count of ready descriptors
                              0    if on timeout
                              -1   if error
```

24. Define wait forever?

Wait forever is that, it returns only when one of the specified descriptor is ready for I/O.

25. What is Donot wait at all?

Do not wait at all states that, return immediately after checking the descriptor. Timeout point to timeval structure. Timeval value must be zero.

26. What are the conditions of the descriptor to become ready?

The conditions of the descriptor to become ready,

- If the socket is ready for reading
- If the socket is ready for writing
- Socket has an exception pending

27. What are the steps involved for socket has exception pending?

The steps involved for socket has exception pending are,

- There exist an out of band data for the socket.
- Socket is in the out of band mark.

28. What is the difference between shutdown and close function?

# With shutdown, we can initiate TCP's normal connection termination sequence regardness of reference count.

In this case of close function, decrement the reference count and closes the socket only if the count reaches 0.

# Shutdown function tell the other end that we have finish sending eventhough that end might have to send more data's.

In this case of close function, terminates both directions of data transfer reading and writing.

29. What is poll function?

Poll function test for events on multiple socket simultaneously. It should include the following header file.

```
#include<poll.h>
int poll (struct pollfd *fdarray, unsigned long *nfds, int timeout);
```

returns count of ready descriptors
0 on timeout
-1 on error.

## 16 MARKS QUESTIONS

1. Explain the steps involved in crashing of server process and server host?
   - crashing of server process
   - crashing of server host

2. i) What is the use of I/O multiplexing.
   - Uses

   ii) Explain the types of I/O model in detail.
   - Blocking I/O
   - Nonblocking I/O
   - I/O Multiplexing
   - Signal driven I/O
   - Asynchronous I/O

3.Write a socket program to implement TCP echo client /server application?

- Socket functions and explanation
- Block diagram
- TCP echo client
- TCP echo server

4. Explain the concept of posix signal handling?
   - Signal
   - Disposition
   - Handling SIGCHLD signal
   - Handing interrupted system calls

5. Write a socket program to implement TCP echo client /server application with multiplexing?

- Socket functions and explanation
- Block diagram
- TCP echo client with multiplexing
- TCP echo server with multiplexing

# UNIT-III

## SOCKET OPTIONS, ELEMENTARY UDP SOCKETS

1. What are the various ways to get and set the options that affect a socket?
   - The getsockopt and setsockopt functions
   - The fcntl function, and
   - The ioctl function

2.Give the general format for getsockopt.

```
int getsockopt (int sockfd, int level, int optname, void  *optval,
                socklen_t  *optlen);
```

sockfd->socket descriptor
level->integer that specifies the protocol specific code that interprets the option
optname->integer that specifies name of the option
optval->pointer to the variable into which the current value of the option is stored
optlen->identifies the length of the option value

3.Give the general format for setsockopt.

```
int setsockopt(int sockfd, int level, int optname, const void  *optval,
               socklen_t  optlen);
```

sockfd->socket descriptor
level->integer that specifies the protocol specific code that interprets the option
optname->integer that specifies name of the option
optval->pointer to the variable into which the current value of the option is stored
optlen->identifies the length of the option value

4.What are the following socket options inherited by a connected TCP socket from the listening socket?

The following socket options inherited by a connected TCP socket from the listening socket are:

SO_DEBUG,SO_DONTROUTE,SO_KEEPALIVE,SO_LINGER,
SO_OOBINLINE,SO_RCVBUF and SO_SNDBUF.

5.What is meant by generic socket option?

They are protocol independent options and are handled by the protocol independent code within the kernel.

6.What is the purpose of SO_BROADCAST option?

It supports the datagram sockets and they enable or disable the ability of process to send broadcast messages. The application must set the option before sending the broadcast messages.

7.Give the characteristics of SO_DEBUG.

Only the TCP socket supports this. When this option is enabled, kernel keep track of detailed information about all packets sent or received by TCP for the socket.

8.What is the use of SO_DONTROUTE.

It specifies that the outgoing packets are to bypass the normal routing mechanisms of the underlying protocol.

9.Write notes on SO_ERROR socket option.

When an error occurs on a socket, the protocol module set a variable name SO_ERROR for that socket. That error is called as pending error. The process can obtain the value of SO_ERROR by using the socket error option.

10.Specify the function of SO_LINGER.

It specifies how the close( ) function operate for a connection-oriented protocol.

11.What is the purpose of SO_OOBINLINE.

When this option is set, out of band data will be placed in the input queue.

12.What is the function of SO_REUSEPORT.

This socket option allows duplicate binding but only if each socket that wants to bind the same IP address and port specify this socket option.

13.What are called IPV4 socket options?

These options are processed by IPV4 and have a level of IPPROTO_IP.

14.What do you mean by IP_HDRINCL socket.

It is set for the raw IP socket. By using this, we can construct our own IP header for all the datagrams being sent by the raw socket.

15.What are called ICMPV6 socket options?

This socket option is processed by ICMPV6 and has a level of IPPROTO_ICMPV6. This sets the ICMPV6 filter structure. This structure specifies the 256 possible ICMP messages. They are passed to a socket through a raw socket.

16.What are the options of IPV6?

They are
IPV6_ADDRFORM,IPV6_CHECKSUM,IPV6_DSTOPTS,IPV6_HOPLIMIT,
IPV6_HOPOPTS,IPV6_NEXTHOP,IPV6_PKTINFO,IPV6_PKTOPTIONS
,
IPV6_RTHDR,IPV6_UNICAST_HOPS.

17.What are the options of TCP socket?

The options are
TCP_KEEPALIVE,TCP_MAXRT,TCP_MAXSEG,TCP_NODELAY,
TCP_STDURG

18. What is the function of ICMP6_FILTER?

This option specifies which of the 256 possible ICMPV6 message types are passed to the process on a raw socket.

19.Define Nagle's algorithm.

If a given connection has outstanding data, then no small packets will be sent on the connection until existing data is acknowledged.

20.Define a Domain Name System.

The Domain Name System maps between hostnames and IP addresses.

21.What are called resource records?

Entries in the DNS are called resource records.

22.What are the types of resource records?

There are five types of resource records. They are

- A
- AAAA [quad A]
- PTR
- MX
- CNAME

23.Give the syntax of gethostname function.

This returns the current name of the host.

General Format:

```
int gethostname(char *name, size_t namelen);
```

24. Specify the functions of gethostbyname function.

This function takes the host name and finds the corresponding binary IP address from the hostent structure.

General Format:

```
struct hostent  *gethostbyname (const char *hostname);

        struct hostent
        {
        char *h_name;
        char *h_aliases;
        int h_addrtype;
        int h_length;
        char **h_addr_list;
};
```

25.Define gethostbyaddr function.

This function takes the binary IP address and finds the host name corresponding to that address from the hostent structure.

General Format:

```
struct hostent  *gethostbyaddr (const char *addr, size_t len, int family);

    struct hostent
    {
    char *h_name;
    char *h_aliases;
    int h_addrtype;
    int h_length;
    char **h_addr_list;
    };
```

26.Give the syntax for getservbyname function.

Getservbyname function maps from the service name to the port number.

General format:

```
struct servent *getservbyname (const char *servname, const char *protoname);
    struct servent
    {
    char *s_name;
    char **s_aliases;
    int s_port;
    char *s_proto;
    };
```

27. Specify the functions of getservbyport system call.

Getservbyport looks up a service name given its port number.

General Format:

```
struct servent * getservport (int port, const char *protoname);
struct servent
{
char *s_name;
char **s_aliases;
int s_port;
char *s_proto;
};
```

## **16 MARKS QUESTIONS**

1. Write a socket program to implement UDP echo client /server application?

- Socket functions and explanation
- Block diagram
- UDP echo client
- UDP echo server

2. What is meant by generic socket option? Explain in detail.
   - Definition
   - Explanation of all so_ option.

. 3. Explain the following socket options.

(i) SO_REUSEADDR
(ii) SO_ERROR
(iii) IP_TOS
(iv) IPV6_CHECKSUM
(v) TCP_MAXSEG

- Purpose and explanation

4. i)Write a program to display the port number from a given service name?

- Getservbyname – syntax
- Program

ii) Write a program to find the IP address from a given host name?

- Gethostbyname – syntax
- Program

5. Explain the following socket options?

i) SO_KEEPALIVE
ii) SO_REUSEPORT
iii) IP_TTL
iv) IPV6_ADDRFORM
v) TCP_NODELAY

Purpose and explanation

## UNIT-IV
## ADVANCED SOCKETS

1. What is a dual stack?

Hosts' running on both IPV4 protocol stack and IPV6 protocol stack i called dual stack.

2. What is the use of IPV6 address testing macros?

Twelve macros are defined to test an IPV6 address for certain properties. The first seven macros test the basic type of IPV6 address. The final five macros test the scope of an IPV6 multicast address.

3. What is the need of IN6_IS_ADDR_V4MAPPED macro?

An IPV6 client could call the IN6_IS_ADDR_V4MAPPED macro to test the IPV6 address returned by the resolver. An IPV6 server could call this function to test the IPV6 address returned by accept or recvfrom.

4. What is IPV6_ADDRFORM socket option?

The IPV6_ADDRFORM socket option can change a socket from one type to another, subject to the following restrictions

1. An IPV4 socket can always be changed to an IPV6 socket. Any IPV4 addresses already associated with the socket are converted to IPV4-mapped IPV6 addresses.
2. An IPV6 socket can be changed to an IPV4 socket only if any addresses already associated with the socket are IPV4 mapped IPV6 addresses.

5. Why thread is called light weight processes?

Threads are sometimes called light weight processes since a thread is lighter in weight than a process. That is, thread creation can be 10-100 times faster than process creation.

6. What is a Pthread?

Posix threads are also called Pthreads. These were standardized in 1995 as part of the posix.1c standard and most versions of UNIX support them. Pthread functions begin with pthread_.

7. How is a thread created?

When a program is started by exec, a single thread is created , called the initial thread or main thread. Additional threads are created by pthread_create.

```
Syntax:  int pthread_create(pthread_t *tid, const pthread_attr_t *attr,
                            void*(*func)(void *), void *arg);
```

8. How is a thread terminated?

One way for a thread to terminate is to call pthread_exit.

```
Syntax: void pthread_exit(void *status);
```

The other 2 ways are

- The function that started the thread can return. Since this function must be declared as returning a void pointer, that return value is the exit status of the thread.
- If the main function of the process returns or if any thread calls exists, the process terminates including threads.

9. What is a common technique for making the function thread safe?

The common technique for making the function thread safe is to define a new function whose name ends in _r. Two of the functions are thread safe only if the caller allocates space for the result and passes that pointer as the argument to the function.

10. Name some of the thread safe functions.

asctime_r, ctermid, ctime_r, getgrid_r, getgrnam_r, getlogin_r, getpwnam_r, getpwiud_r, gmtime_r, localtime_r etc.

11. What is concurrent or parallel programming?

We call threads programming as concurrent or parallel programming since multiple threads can be running concurrently, accessing the same variables.

12. What are the functions that is used to lock and unlock the mutex.?

Syntax:

```
int  pthread_mutex_lock(pthread_mutex_t  *mptr);
int pthread_mutex_unlock(pthread_mutex_t  *mptr);
        Both return    0 if ok
                       positive Exxx value on error.
```

13. Why is a mutex always associated with a condition variable?

The condition is normally the value of some variable that is shared between the threads. The mutex is required to allow this variable to be set and tested by the different threads. So mutex is always associated with a condition variable.

14. Define proto structure

we use proto structure to handle the difference between Ipv4 and Ipv6. This structure contains two function pointers, two pointers to socket address structures, the size of the socket address structures, and the protocol value for ICMP. The global pointer pr will point to one of these structures that we will initialize for either IPv4 or IPv6.

15. Establish signal handler for SIGALRM

We establish our function sig_alrm as the signal handler for SIGALRM because each time we send a UDP datagram we wait 3 seconds for an ICMP message before the next probe.

16. Differentiate IPV4 and IPV6 raw socket .

1. All fields in the protocol handler send or receive on a raw IPV6 socket in network byte order

2.There is no similar to IPV4 socket option

3.Checksum on a IPV6 raw are handle differently

17. Explain the concept of Raw Socket Input.

1. Raw socket input means the kernel pass the received IP datagram to raw socket

2. Received UDP packet and receive TCP packet are never passed to the raw socket
3. If a process wants to read IP datagram containing UDP or TCP packet the packet must be read at the data link layer.
4. Most ICMP packets are passed to the raw socket after the kernel has finished processing the ICMP packets.

18. Define latency

Latency means time it takes a parent to reach its destination is called latency.

19. What is the function of PING( Packet Internet Grosper)

The ping program checks the responsiveness between two host, it is use to check the connectivity.

20. What are the information return by the Response of Ping program.

1. The IP address of where the echo reply came from
2. No of bytes of data send
3. Round Trip Time(RTT)

   RTT = sending time + receiving time
4. Time To Live(TTL)

21.Kernel receive IP datagram and perform three test

1. If a nonzero protocol is specified when the raw socket is created then the receive datagram protocol field must match this value.
2. If a local IP address is bound to a raw socket by byte then the destination IP address of the receive datagram must match this bound address.
3. If a foreign IP address was specified for the raw socket by connect then the source IP address of received datagram must match the connected address.

22.What is the function of traceroute utility.

It will check responsiveness as well as trace the path a IP datagram follow from source to the destination.

23. What are the two sockets used in traceroute program.

We need two sockets: a raw socket on which we read all returned ICMP messages and a UDP socket on which we send the probe packets with the increasing TTLs.

24. Define mutex.

- Mechanism for implementing thread synchronization
- Act like a lock protecting access to a shared data resource.

## 16 MARKS QUESTIONS

1. i) Define thread. Give the syntax for creating the thread?
   - Definition
   - Thread creation – syntax

   ii) Write a TCP based echo client / server program using thread?
   - Thread functions and explanation
   - TCP echo client using thread
   - TCP echo server using thread

2. i) Explain the steps involved in creating a raw socket?
   - Steps

   ii) Explain the operation of ping program?
   - Ping def.
   - Operation
   - ICMPV4 echo request and echo reply message format.

3. Explain the concept of IPV4 and IPV6 interoperability.
   - IPV4 client and IPV6 server
   - IPV6 client and IPV4 server
   - IPV6 address testing macros
   - Source code portability

4.Explain the concept of mutual exclusion.

- Mutex – def.
- Sequence of steps
- Mutex functions

5.Explain the operation of traceroute program.

- Definition
- Operation

## UNIT V
## SIMPLE NETWORK MANAGEMENT

1. Define SNMP.

SNMP refers to a set of standards for network management including a protocol, a database structure specification and a set of data objects.

2. Define network management system.

A network management system is a collection of tools for network monitoring and control that is integrated in the following senses:

- It is a single operator interface with a powerful but user friendly set of commends for performing most or all network management tasks.
- A minimal amount of separate equipment is necessary.

3. SNMP uses TCP. Justify.

No. SNMP depends on UDP, which is a connectionless protocol .
SNMP is itself connectionless.

4. Name the key elements of network management.

- Management station
- Management agent
- Management information base
- Network management protocol

5.. Define Management agent.

Active element in the network management system

6. Define Management information base.

Collection of objects is referred to as a MIB.

7. Specify the advantages of trap-directed polling.

- Saving of network capacity
- Agent processing time

8. Define Agent Function:

Agent function gathers and records Management information for one or more network elements and communicates the information to the monitor

9. Polling and events reporting

Polling is a request response interaction between the manager and agent. The manager can query any agent and request the values of various information elements, the agent responds with information from its MIB.

With event responding, the initiative is with the agent and the manager is in the role of listener, waiting for incoming information, event resorting is useful for Detecting problems.

10. Define proxy.

A proxy is an SNMP agent that maintains information on behalf of one or more non SNMP devices.

11. Define structure of Management information.

The structure of Management information defines the general framework within which a MIB can be defined and constructed.

12. What do you mean by Application_wide datatypes.

Application wide datatypes are relevant to a particular application.

13. What do you mean by object instance.

A object instance is a particular instance of an object type that has bound to a specific value.

14. What are the information stored in the tcpConnTable.

- State
- Local IP address
- Local port
- Remote IP address
- Remote port

15. What is the purpose of system group.

System group provides general information about the managed system.

16. What is the uses of ip group.

Ip group provides information related to the implementation and execution experience of IP on this system.

17. What are the operations supported by SNMP.

- Get
- Set
- Trap

18. What is the getrequest PDU?

The GetRequest PDU is issued by an SNMP entity on behalf of a network management station application. The sending entity includes the following fields in the PDU: PDU type, request-id, variable bindings. The receiving SNMP entity responds to a GetRequest PDU with a GetResponse PDU containing the same request-id.

19. What are the limitations of using SNMP?

- SNMP may not be suitable for the management of truly large networks because of the performance of polling.
- SNMP is not suited for retrieving large volumes of data, such as an entire routing table.
- SNMP traps are unacknowledged.
- The basic SNMP standard provides only trivial authentication.
- SNMP does not directly support imperative commands
- The SNMP MIB model is limited and does not support applications that support applications that make sophisticated queries.
- SNMP does not support manager-to-manager communications.

20. What is a Trap PDU?

The Trap PDU is issued by an SNMP entity on behalf of a network management agent application. It is used to provide the management station with an asynchronous notification of some significant event.

21. What are the types of Threats?

Interruption: An asset of a system is destroyed or becomes unavailable or unusable. This is a threat to availability.

Interception: An authorized party gains access to an asset. This is a threat to secrecy.

Modification: An authorized party not only gains access but also tampers with an asset. This is a threat to integrity.

Fabrication: An authorized party inserts counterfeit objects into the system. This is a threat to integrity.

22. Name the Network Monitoring Information:

Static:

This is information that characterizes the current configuration

and the elements in the current configuration.

Dynamic:

This information related to events in the network.

Statistical:

This is information that may be derived from dynamic information.

23. What is MTBF?

Component failure is expressed by *mean time between failure* (MTBF).

The availablity, A, is based on MTBF and can be expressed as

A=MTBF/MTBF+MTTR,

Where MTTR is the *mean time to repair* following a failure.

24.What are the key capabilities of SNMP?

The SNMP includes the following key capabilities:

1. get: enables the management station to retrieve the value of objects at the agent.
2. set: enables the management station to set the value of objects at the agent.
3. trap: enables an agent to notify the management station of significant events.

25. State the features of SNMPV3.

- Security
- Administrative framework.

# 16 MARKS QUESTIONS

1 Define SNMP. Describe the architecture of SNMP?

- Definition
- Architecture

2. Explain the message format of SNMP?

- SNMP message format
- Explanation

3. What are the steps involved in transmission and reception of an SNMP message?

- Tansmission of an SNMP message
- Reception of an SNMP message

4.. Write short notes on SNMP Protocol Specification

- SNMP formats
- Transmission and reception of SNMP message
- Variable binding
- Protocol data unit

5.Describe the concept of standard MIBs.

- MIB- II
- Ethernet MIB