

MC1751 –UNIX and Network programming

Short questions and answers

1. Define directory and filename

A **directory** is a file that contains directory entries. The directory entry containing a file name along with structure of information describing the attributes of the file. The names in a directory are called **filename**. The only two characters that cannot appear in a file name are / and null character.

2. Define file descriptor

File descriptors are nonnegative integers that the kernel uses to identify the file being accessed by a particular process. Whenever the kernel opens an existing file or creates a new file it returns the file descriptor.

3. Define program and process

A program is a executable file residing in a disk file. An executing instance of a program is called a process. Every UNIX process have a unique numeric identifier called the process ID. The process ID is always a non negative integer.

4. What are the two different time values supported by UNIX?

Calendar time: This value counts the number of seconds which is 00 : 00 : 00 January 1, 1970, coordinated universal time. The data type `time_t` hold this time values.

Process time: It is also called CPU time and measures the central processor resources used by process. Process time is measured inc clock ticks. The data type `clock_t` hold these time values.

5. Define user CPU time, system CPU time.

The time attributed to user instructions are referred as user CPU time. The time attributed to kernel, when it executes an behalf of th process is called system CPU time.

6. What are the kernel data structures are used for open files?

- Process table entry
- File table entry
- V node structure

7. Define atomic operation.

An operation composed of multiple steps. If the operation is performed atomically either all the steps are performed or none is performed.

8. How will you retrieve the file status?

The file status information are retrieved by using the functions `stat`, `fstat`, `lstat`. The `stat` function returns a structure of information about the named file. The `fstat`

function obtains information about the file that is already open. The lstat function similar to stat but when the named file is a symbolic link.

9. List down the different file types

- Regular file
- Directory file
- Character special file
- Block special file
- FIFO
- Symbolic link
- Socket

10. What are the macros available for identifying the file type?

- S_ISREG() for regular file
- S_ISDIR() for directory file
- S_ISCHR() for character special file
- S_ISBLK() for block special file.
- S_ISFIFO() For FIFO file
- S_ISLNK() for symbolic link
- S_ISOCK() for socket file

The argument to these macros is the st_mode member from the stat structure.

11. What are the options available to determine the group ID of the new file?

- The group ID of the new file can be the effective group ID of the process
- The group ID of the new file can be the group ID of the directory in which the file is being created.

12. Define sticky bit

The bit S_ISVTX is known as sticky bit. If this bit is set for an executable program file, then the first time the program was executed a copy of the programmer's text was saved in the swap area when the process terminated. The advantage of using this is the program is loaded into memory faster the next time.

13. Define symbolic link.

A symbolic link is an indirect pointer to a file unlike the hard links. There are no file system limitations on a symbolic link and what it points to and any one can create a symbolic link to a read directory. Symbolic links are typically used to move a file or an entire directory hierarchy to some other location on a system.

14. What are the different time fields maintained for each file

- Last access time of file data
- Last modification time of file data
- Last change time of i_node status

15. Define the purpose and format of **utime** function

The access time and modification time of a file can be changed with the **utime** function. The general format is

```
# include <sys/types.h>
```

```
# include <utime.h>
```

```
int utime (const char * pathname, const struct utimbuf *times);
```

Returns 0 if OK -1 on Error

16. What are the directory routines available in POSIX . 1

```
# include <sys/types.h>
```

```
#include <dirent.h>
```

```
DIR *opendir ( const char *pathname)
```

Returns pointer if OK NULL on error

```
struct dirent *readdir(DIR *dp);
```

Returns pointer if OK , Null at end of directory or error

```
void rewinddir(DIR *dp);
```

```
int closedir(DIR *dp);
```

Returns 0 if OK -1 on error

17. What is the major goal of buffering?

To use the minimum number of read and write calls. It tries to do its buffering automatically for each I/O stream

18. What are the different types of buffering is supported by UNIX?

Fully buffered : In this case the actual I/O takes place when the standard I/O buffer is filled

Line buffered: In this case the standard I/O library performs I/o when a new line character is encountered on input or output.

Un buffered: The standard I/O library does not buffer the characters.

19. What are the different types of unformatted I/O?

Character at a time I/O

Line at a time I/O

Direct I/O

20. How will you position a standard I/O stream?

By using the functions ftell , fseek and fgetpos and fsetpos.

ftell, fgetpos: to retrieve the current byte position

fseek ,fsetpos: to move the file pointer to specified place

21. List down the contents of password file.

User name

Encrypted password

Numerical user ID and group ID
Comment field
Initial working directory
Initial shell

22. what do you mean by shadow password?

The duplicate copy of password file is called password file. Minimally this file has the user name and the encrypted password. Other information relating to the password is also stored here. For example system with shadow passwords often required to choose a new password at certain intervals.

23. What are the different functions for system's data file?

- A get function that reads the next record, opening the file if necessary. These functions normally return a pointer to the structure. A null pointer is returned when the end of file is reached.
- A set function opens the file and rewinds the file.
- An end entry that closes the data file.

24. How will you identify the system in POSIX.1?

The system is identified by using the function uname. This function return information on the current host and operating system.

25. List down the different time conversion functions?

gmtime
local time
mktime
asctime
ctime
strftime

26. List down the functions for process termination

The normal terminations are return from main, calling exit, calling _exit. The abnormal terminations are by calling abort, terminated by a signal

27. Write down the contents of C program in memory layout.

Text segment, initialized Data segment, uninitialized, stack, heap

28. What is the purpose of memory allocation functions and what are the functions available for memory allocation.

The memory allocation functions are used to allocate, reallocate, and deallocate memory locations.

The functions available for memory allocation are malloc, calloc, realloc and free.

29. What are the rules govern the changing of resource limits?

- A soft limit can be changed by any process to a value less than are equal to its hard limit

- Any process can lower its hard limit to a value greater than or equal to its soft limit
- Only a super user process can raise a hard limit.

30. Write down the functions those who are return a process ID

The functions return process ID is

Getpid, getppid, getuid, geteuid, getgid, getegid

31. Define zombie process

The process that has terminated but whose parent has not yet waited for it called zombie process.

32. What is the difference between wait and waitpid functions

Wait can block the caller until a child process terminates while waitpid has no option that prevents it from blocking.

Waitpid doesn't wait for the first child to terminate- it has a number of options that control which process it waits for.

33. What are the additional features available for waitpid comparing to wait

- ✓ Waitpid function waits for one particular process
- ✓ Waitpid provides a non blocking version of wait
- ✓ Waitpid supports job control

34. What is the purpose of exec functions?

When a process calls one of the exec functions that process is completely replaced by the new program. The new program starts execution from main function. The process does not change across an exec because a new process is not created. But this function replaces the current process with new program from disk.

35. Write down the rules for changing user ID?

- ✓ The process must have super user privileges
- ✓ If the process does not have super user privileges, but uid equals either the real user ID or the saved set user ID

36. What are the uses of interpreter files?

- ✓ Interpreter files hide the fact that certain programs are scripts in some other language.
- ✓ Interpreter scripts provide an efficiency gain
- ✓ Interpreter scripts let us write shell scripts using shells other than / bin/ sh

37. Define process group

A process group is a collection of one or more processes. Each process group has a unique process ID. A function getpgrp returns the process group id of the calling process. Each process group have a leader. The leader is identified by

having its process group ID equal its process ID. A process joins an existing process group or creates a new process group by calling `setpgid`.

38. Define session.

A session is a collection of one or more process groups. A process establishes a new session by calling `setsid` function. This function returns process group id if OK.

39. What do you mean by signal?

Signals are software interrupts. Signals provide a way of handling asynchronous events: a user at a terminal typing the interrupt key to stop a program or the next program in the pipeline terminating prematurely.

40. What are the conditions to generate a signal?

- ✓ The terminal generated signals occur when users press some terminal keys.
- ✓ The hardware exceptions generate signals
- ✓ The `kill(2)` function allows a process to send any signal to another process or process group.
- ✓ The `kill(1)` function allows a process to send
- ✓ Software conditions generate signals

41. Define pending signal

A signal is generated for a process when a process event that causes the signal occurs. The signal is delivered to a process when the action for a signal is taken. During the time between the generation of signal and its delivery is referred as pending.

42. Define the purpose of `kill` and `raise` functions

The `kill` function sends a signal to a process or a group of processes. The `raise` function allows the process to send a signal to itself.

43. What is the purpose of `alarm` and `pause` functions

The `alarm` function allows to set a timer that will expire at a specific time in the future. When the timer expires the `SIGALRM` signal is generated. If we ignore or don't catch this signal its default action is terminate the process.

The `pause` function suspends the calling function until signal is caught.

44. Define the different modes of terminal I/O

Canonical mode input processing. In this mode terminal input is processed as lines. The terminal driver returns at most one line per read request. Default mode

Non canonical mode input processing. The input characters are not assembled into lines. Used by programs that manipulate the screen.

45. What are the different problems used in `fork` function?

`Fork` is expensive. Because memory and all descriptors are duplicated in the child.

Inter process communication is required to pass information between the parent and the child after the fork.

46. What do you mean by thread?

Threads are light weight processes. All threads within the process share the same global memory. This makes communication simple but necessitates synchronization. All threads within a process share process instructions, data, open files, signal handlers and signal descriptors, current working directory, user and group IDs.

47. Define pipe

Pipes were the first widely used inter process communication method available both within programs and from the shell. The problem with pipe is that they are usable only between processes that have an ancestor, this problem is solved with the named pipes or FIFOs

48. List down the types of IPC.

- Message Queues
- Semaphore
- Shared memory

49. Explain the purpose of ftok function with format

This function is used to generate an IPC key. The general format for this function is

```
#include <sys/ipc.h>
key_t ftok(const char *pathname, int id);
```

This function takes information derived from the pathname, the file's inode number and the lower order 8 bits of id.

50. Explain the purpose of popen and pclose functions with format

The popen function creates a pipe and initiates another process that either reads from the pipe or writes to the pipe.

51. Define message queue

Message queue is a linked list of messages. It is used to make a communication between the related or unrelated processes

52. What are the attributes in the message on the message queue.

- An unsigned integer priority
- The length of the data portion of the message
- The data

53. What are the differences between POSIX message queue and system V message queue?

POSIX message queue always returns the oldest message of the highest priority, But the system V message queue can return a message of any desired priority.

Posix message queues allow the generation of a signal or the initiation of a thread when a message is placed onto an empty queue, whereas nothing similar is provided by system V message queues.

54. what is the purpose of mq_open, mq_close functions?

The mq_open functions open an existing Posix message queue or create a new message queue.

The mq_close function is used to close the opened message queue.

55. What is the purpose of mq_unlink function?

The mq_unlink function is used to delete a Posix message queue permanently from the system. But the message queue is deleted only the reference count for the queue reached 0.

56. How will you send and receive an message to the POSIX message queue?

The posix message queue can pass a message to the queue by calling the function mq_send with the parameters are the descriptor, character array pointer, length of the message and priority of a message.

The posix message queue can retrieve a message to the queue by calling the function mq_receive with the parameters are the descriptor, character array pointer, length of the message and priority of a message.

57. What are the information available in system V message queue?

Read write permissions, pointer to first message, pointer to last message, number of bytes currently in queue, Number of messages in the queue, maximum number of bytes allowed, process id of last message send, process id of last message receive, process time of last message send, time of last message receive, time of last message control function called.

58. List down the system calls related to system V message queue with description.

msgget- to create a new message queue or open an existing message queue.

msgsnd – to send a message to a queue

msgrcv – to receive a message from a message queue.

msgctl – to provide a variety of control operations in a message queue.

59. what are the features provided by type field in the system V message queue?

- Used to identify a message, allowing multiple processes to multiplex messages onto a single queue.
- Used as a priority field

60. What is the purpose of synchronization? List the building blocks of synchronization?

Synchronization normally needed to allow the sharing of data between threads and processes. Mutexes and condition variables are the building blocks of synchronization.

61. What is the use of Mutex?

Mutex is used to protect a critical region, to make certain that only one thread at a time executes the code within the region.

62. List down the function formats for lock and unlock a region.

```
#include<pthread.h>
int pthread_mutex_lock(pthread_mutex_t *mptr);
int pthread_mutex_trylock(pthread_mutex_t *mptr);
int pthread_mutex_unlock(pthread_mutex_t *mptr);
```

63. Define bounded buffer problem or producer- consumer problem.

One or more producers are creating data items which are then processed by one or more consumers.

64. What are the rules for allocating read- write locks?

- Any number of threads can hold a given read-write lock for reading as long as no thread holds the read write lock for writing.
- A read-write lock can be allocated for writing only if no thread holds the read-write lock for reading or writing.

65. What do you mean by critical region?

The critical region access or update one or more pieces of data that are shared between threads.

66. What do you mean by semaphore? List down different types of semaphore.

- Semaphore is a primitive used to provide synchronization between various processes or between the various threads in a given process.
- The types of semaphore are
 - Posix named semaphores
 - Posix memory based semaphore
 - System V semaphores.

67. List down the operations performed on a semaphore by semaphore.

Create a semaphore
Wait for a semaphore
Post to a semaphore

68. Write down the differences between the semaphore and mutexes and condition variables.

The thread that locked the mutex, whereas a semaphore the same thread that did the semaphore wait need not perform post, must always unlock a mutex..

A mutex is either locked or unlocked.

69. Define shared memory.

It is the fastest form of IPC. It is a memory shared by unrelated processes. Once the memory is mapped into the address space of the process that are sharing the memory region no kernel involvement occurs in placing data between the processes

70. List down the purpose for using the function mmap.

- To provide a memory mapped I/O
- To provide anonymous memory mappings
- To provide shared memory between unrelated processes

71. What are the different ways to share memory between unrelated processes?

- By using memory mapped files
- By using shared memory objects

72. Define sockets

A socket is a construct to provide a communication between computers. It hides the underlying networking concepts and provides us with an interface to communicate between computers.

73. Define value result arguments

The way in which the length of the structure passed is depends on which direction the structure is being passed from process to kernel or vice versa. The size parameter is passed by value when the function is called and passed by result when the function returns. This type of argument is called value result argument.

74. Define little byte endian order, big byte endian order

- ✓ Little byte endian order: in sixteen bit integer number the low order byte at the starting address.
- ✓ Big byte endian order: in sixteen bit integer number the high order byte at the starting address.

75. Write down the different byte manipulation functions

bzero, bcopy, bcmp, memset, memcpy, memcmp

76. What is the purpose of address conversion functions

The functions inet_aton, inet_ntoa and inet_addr convert IPv4 address between dotted decimal string and its 32-bit network byte ordered binary value. The functions inet_pton, inet_ntop handle both IPv4 and IPv6 address. The letters p and n stands for numeric and presentation.

77. What are the different options and type values available for creating a socket?

The different option values are AF_INET, AF_INET6, AF_LOCAL, AF_ROUTE, AF_KEY. The different type values are SOCK_STREAM, SOCK_DGRAM, SOCK_RAW

78. What is the purpose of connect and bind function.

The connect function is used by a TCP client to establish a connection with a TCP server. The bind function assigns a local protocol address to a socket.

79. What are the actions performed by listen function.

The listen function is called by TCP server and it performs the following actions

- ✓ It converts the unconnected socket into a passive socket, indicating that the kernel should accept incoming connection requests directed to this socket
- ✓ It specifies the maximum number of connections that the kernel should queue for this socket.

80. Define concurrent servers and iterative servers.

The servers that can handle multiple clients simultaneously are called concurrent servers. The servers that can handle multiple clients serially are called concurrent servers.

81. What are the uses of fork functions.

- A process makes a copy of itself so that one copy can handle one operation does another task.
- A process wants to execute another program, that is done by one of the processes by calling exec to replace itself with the new program.

82. List down the different I/O models

- Blocking I/O
- Nonblocking I/O
- I/O multiplexing
- Signal driven I/O
- Asynchronous I/O

83. Define synchronous and asynchronous I/O operations

A synchronous I/O operations causes the requesting process to be blocked until I/O operations complete

An asynchronous I/O operation does not cause the requesting process to be blocked.

84. Write down any 8 generic socket options

SO_BROADCAST, SO_DEBUG, SO_ERROR, SO_DONTROUTE, SO_KEEPALIVE, SO_LINGER, SO_RCVBUF, SO_SNDBUF

85. Define UDP sockets.

UDP is a connection less unreliable, datagram protocol. In UDP the client does not establish connection with a server. Instead the client just sends a datagram to the server.

86. What is the purpose of DNS?

The domain name system or DNS is used primarily to map between host names and IP addresses. DNS entries are known as resource records (RRs)

87. What is the purpose of resolvers?

The client and server applications contact a DNS server by calling functions in a library known as resolver. The common resolver functions are `gethostbyname` and `gethostbyaddr`.

88. What are the different Resource records(RR)?

The different RR format are A, AAAA, PTR, MX, CNAME

89. Define TCP sockets.

TCP sockets provide a simple and effective way to provide connection oriented client server networking.

90. What are the additional facilities supported by raw sockets?

Raw sockets are used to read and write ICMPv4, IGMPv4, and ICMPv6 packets. With the raw socket a process can read and write IPv4 data grams with an IPv4 protocol field that is not processed by kernel.

With a raw socket a process can build its own IPv4 header using the `IP_HDRINCL` socket option

91. What are the differences between raw IPv6 sockets with ordinary raw output?

- ✓ All fields in the protocol send or received on a raw IPv6 socket are in network byte order
- ✓ There is no equivalent of `IP_HDRINCL` socket option with IPv6.
- ✓ Checksums on raw Ipv6 sockets are handled differently using the IPv6 socket option

92. What are the different level constants are available for socket option functions?

`SQL-SOCKET`, `IPPROTO_IP`, `IPPROTO_ICMPV6`, `IPPROTO_IPV6`, `IPPROTO_TCP`.

93. What are the functions available for set and retrieve the socket options?

The functions for set and retrieve socket options are `getsockopt` and `setsockopt`

94. What are the different system tracing tools are available/

- ✓ SVR streams based socket library
- ✓ BSD kernel sockets
- ✓ Solaris 2.6 kernel sockets

95. Describe different TCP/IP services.

| | |
|---------|---|
| echo | server return whatever the client sends |
| discard | server discards whatever the client sends |
| daytime | server returns the date and time in a human readable form |

96. What is the purpose of sock program

It is used to generate special case conditions that can help us visualize various scenarios without writing individual test programs.

97. List down the different operation modes for sock program.

Standard input, standard output client
Standard input, standard output server
Source client
Sink server

98. What are the different functions done by TCP echo client server program?

- ✓ The client reads a line of text from its standard input and writes it to the server
- ✓ The server reads the line from the network input and echoes the line back to the client
- ✓ The client reads the echoed line and prints it on its standard output

99. What is the purpose of trace route program?

The trace route program allows us to determine the path that IP datagrams follow from our host to some other destination.

100. What is the purpose of ping program?

The ping program is a utility program that is used to fetch echo replies from the server.

101. What is the purpose of netstat program?

- ✓ It shows the status of networking endpoints
- ✓ It shows the multicast groups that a host belongs to on each interface
- ✓ It shows the pre protocol statistics with the `-s` option
- ✓ It displays the routing table with the `-r` option and the interface information with the `-i` option

16 Mark Questions with key

1. Explain the different functions for FILE I/O.

- Explain about file descriptors
- Explain the functions with format and example
 - open
 - creat
 - read
 - write
 - lseek
 - close

2. Explain in detail about directory functions.
 - ✓ Explain the functions mkdir, rmdir, opendir, readdir functions with format
 - ✓ Explain the rewinddir, closedir functions with format
 - ✓ Explain the functions link, unlink, remove and rename functions with format
 - ✓ Explain symlink, readlink functions with format
3. Explain standard I/O library with necessary functions
 - ✓ Explain different buffering methods and explain the functions setbuf, setvbuf functions with format
 - ✓ Explain stream opening functions fopen, freopen, fdopen functions with format.
 - ✓ Explain reading and writing a stream functions with format
 - ✓ Explain line -at-a time I/O, binary functions with format
 - ✓ Explain formatted I/O functions with format
4. Explain in detail about system Data files
 - ✓ Explain password file with necessary fields in password file and the functions related to password file
 - ✓ Explain group file with necessary fields in group file and the functions related to group file.
 - ✓ Explain supplementary group file with necessary functions with example.
5. Explain in detail about Binary I/O and formatted I/O
 - ✓ Explain the functions fread and fwrite
 - ✓ Explain the functions for positioning a stream
 - ✓ Explain about formatted output functions
 - ✓ Explain in detail about formatted input functions
6. Explain in detail about process control.
 - ✓ Explain about process control related functions
 - ✓ Explain fork and vfork functions
 - ✓ Explain about exit function
 - ✓ Explain about wait and waitpid functions
 - ✓ Explain about exec functions
7. Explain in detail about signals . Define signals
 - Explain signal function
 - Explain some of the signals
8. Explain in detail about terminal I/O
 - ✓ Define the different modes of terminal I/O
 - ✓ Explain the functions for setting and getting terminal attributes
 - ✓ Explain in detail about canonical mode and non canonical mode
 - ✓ Explain terminal window I/O
9. Explain in detail about the thread
 - ✓ Define thread
 - ✓ Explain basic thread functions
 - ✓ Explain thread specific data
10. Explain in detail about the UNIX process Environment
 - ✓ . Explain about main function

- ✓ . Explain about process termination
 - ✓ . Explain about command line arguments & memory layout of C program
 - ✓ . Explain about environment list and environment variables
11. Explain in detail about system V message queues
- . Define message queue
 - . Explain the functions with format and example
 - ✓ msgget
 - ✓ msgsnd
 - ✓ msgrcv
 - ✓ msgctl
12. Explain detail about mutex and condition variables
- . Define mutex
 - . Define critical region
 - . Write down the purpose of mutex and condition variables
 - . Explain the functions useful for lock, unlock, init and destroy functions for mutex
 - . Explain waiting and signaling for condition variables
 - Explain the producer consumer problem
13. Explain in detail about semaphores
- . Define semaphore and its types
 - . Explain the following system calls for posix named and memory based semaphores
 - ✓ sem-open(), sem_init(), sem - wait(), sem_trywait(), sem postO, sem getvalue(), sem-close(), sem - destroyO, sem- unlinkO
 - . Explain the binary semaphore and counting semaphore with necessary functions shmgetO, semopO,semctl()
14. Explain in detail about shared memory or fastest IPC mechanism
- . Define shared memory
 - . Explain the functions mmap, munmap, msync functions with format.
 - . Explain posix shared memory
 - ✓ Explain about memory mapped file
 - ✓ Explain shared memory objects
 - ✓ Explain the following functions with format
 - . shm_open
 - shm unlink
 - . ftruncate, fstat
 - . Explain system V shared memory
 - Explain the functions shmget, shmat, shmdt, shmctl with format
15. Explain in detail about record locking
- . Explain about read write locks
 - . Explain record locking versus file locking
 - .fcntl record locking
 - . Explain advisory locking
 - . Explain mandatory locking
 - . Explain about locking files

16. Explain in detail about TCP communication
 - . Define TCP communication
 - . Explain client server communication example with diagram
 - . Explain the system calls for TCP communication
 - . Explain about three way handshake algorithm (for connection)
17. Explain in detail about I/O multiplexing
 - . Define I/O multiplexing
 - . Explain the different I/O models
 - . Explain the following functions
 - Select
 - . Shutdown
 - . Pselect
 - Poll
18. Explain in detail about socket options
 - . Explain the functions for set socket option and get socket option functions .
 - Explain the following socket options
 - Generic socket options
 - IPv4 socket options
 - IPv6 socket options
 - TCP socket options
19. Explain in detail about UDP sockets
 - . Define UDP communication
 - . Explain client server communication example with diagram . Explain the system calls for UDP communication
20. Explain in detail about name and address conversion functions
 - . Explain about DNS
 - . Explain gethostbyname and gethostbyname2 functions
 - . Explain gethostname and uname functions
 - . Explain gethostbyaddr, getservbyname, getservbyport functions
21. Explain in detail about raw sockets
 - Explain about
 - . Raw socket creation
 - . Raw socket output
 - . Raw socket input
22. Explain in detail about debugging techniques
 - Explain about
 - . System call tracing
 - . Standard internet services
 - . Sock program
 - . Small test programs
 - . T cpdump program
 - . Netstat program
 - . Lsof program
23. Explain TCP echo client server

- . Define echo client server
 - . Write down the steps for TCP communication
 - . Write down the additional steps for implanting echo client server . Draw the diagram for client server communication
 - . Explain the above diagram
24. Explain TCP echo client server
- . Define echo client server
 - . Write down the steps for UDP communication
 - . Write down the additional steps for implanting echo client server . Draw the diagram for client server communication
 - . Explain the above diagram
25. Explain in detail about ping and trace route program
- . Write down the purpose of ping program
 - . Write down the operations done by ping program
 - . Write down the purpose of trace route program
 - . Write down the operations done by trace route program
26. Explain about file system
- ✓ Draw the diagram for file system
 - ✓ List the components and Explain
 - ✓ Explain working of file system with example
27. Explain in detail about file sharing
- ✓ Explain the data structures needed for file sharing
 - ✓ Explain the process during file sharing with example