1. What is mean by file system?
2. Ans:In Any Operating System the Concept of file is defined as ”Unformatted Uniform stream of Byte”.
3. Which file systems are used by Linux and Windows operating systems?
4. Ans: **Windows mainly support FAT** (File Allocation Table) and NTFS (New Technology File system). Windows NT 4.0, Windows 200, Windows XP, Windows .NET server and Windows workstation use NTFS as their preferred file system. Still, FAT can be used with floppy disks and older Windows versions (for multi-boot systems). FAT is the initial file system used in Windows. FAT was used with DOS, and its three versions are FAT12, FAT16 and FAT32. The number of bits used to identify a cluster is the number that is used as the suffix in the name. FAT12, FAT16 and FAT32 have 32MB, 4GB and 32GB as the maximum partition sizes.
   * + 1. A variety of files systems can be sued with Linux. Commonly used file systems are ext\* family (ext, ext2, ext3 and ext4) and XFS. Silicon Graphics developed XFS, which is a journaling system with high performance. The ext (extended file system) was developed in early 1990’s. It was the first file system used in Linux operating system. Remy Card developed it by getting inspiration from the UFS (UNIX File System).
5. What are the parts of the file system?
6. Ans:- A file system is a process of managing how and where data on a storage disk, which is also referred to as file management or FS
7. NTFS.
8. FAT.
9. exFAT.
10. HFS Plus.
11. EXT.
12. Explain UAREA and its contents.
    1. Ans: In addition to the text, data, and stack segments, the operating system also maintains for each process a region called the u area (user area). The u area contains information specific to the process (e.g., open files, current directory, signal actions, accounting information) and a system stack segment for process use. If the process makes a system call (e.g., the system call to write in the function main in Program 1.1), the stack frame information for the system call is stored in the system stack segment. Again, this information is kept by the operating system in an area that the process does not normally have access to. Thus, if this information is needed, the process must use special system calls to access it. Like the process itself, the contents of the u area for the process are paged in and out by the operating system.
13. Explain the use of the File Table and its contents.
    1. Ans:In the traditional implementation of Unix, file descriptors index into a per-process file descriptor table maintained by the kernel, that in turn indexes into **a system-wide table of files opened by all processes**, called the file table.
14. Explain the use of InCore inode Table and its use.
15. Ans:- What is incore inode table?
16. In - core inode refers to **inode which is present in the main memory**. The kernel uses it whenever a process wants to manipulate a file in the secondary memory.It is a dynamic entity. On- disk inode is a static entity & is present only in the Secondary memory. HTH.
17. Use of Incore inode table:
18. An inode is the basis of a Linux file system. It **manages file and directory metadata** and is essential for the functioning of a file system. The inode is required to check the file system object locations, modification dates, size, and other relevant information
19. What is mean by inode?
20. Ans:- The **inode** (index node) is a data structure in a [Unix-style file system](https://en.wikipedia.org/wiki/Unix_filesystem) that describes a [file-system](https://en.wikipedia.org/wiki/File_system) object such as a [file](https://en.wikipedia.org/wiki/Computer_file) or a [directory](https://en.wikipedia.org/wiki/Directory_(computing)). Each inode stores the attributes and disk block locations of the object's data.[[1]](https://en.wikipedia.org/wiki/Inode#cite_note-1) File-system object attributes may include [metadata](https://en.wikipedia.org/wiki/Metadata) (times of last change,[[2]](https://en.wikipedia.org/wiki/Inode#cite_note-2) access, modification), as well as owner and [permission](https://en.wikipedia.org/wiki/File_system_permissions) data.[[3]](https://en.wikipedia.org/wiki/Inode#cite_note-3)
21. A directory is a list of inodes with their assigned names. The list includes an entry for itself, its parent, and each of its children.
22. What are the contents of Superblock?
23. Ans:- The superblock contains **information on the size of the file system, the number of inodes, the number of data blocks, the free and used inodes, and the block size for the file system**. The superblock is kept in memory and in multiple locations on disk for each file system.
24. What are the types of files?
25. Ans:There are 2 types of files 1. Regular File
26. 2. Special File .
27. What are the contents of the inode?
    1. Ans: Contetnt of Inode :File Name, Inode Number, File Size, File Actual Size, File Type ,Buffer,Link Count, Refeernce Count,Permission, next Pointer
28. What is the use of a directory file?

Ans:A directory is a container that is used **to contain folders and files**. It organizes files and folders in a hierarchical manner.

1. How the operating system maintains security for files?

Ans: Security refers to providing a protection system to computer system resources such as CPU, memory, disk, software programs and most importantly data/information stored in the computer system. If a computer program is run by an unauthorized user, then he/she may cause severe damage to computer or data stored in it. So a computer system must be protected against unauthorized access, malicious access to system memory, viruses, worms etc. We're going to discuss following topics in this chapter.

1. Authentication
2. One Time passwords
3. Program Threats
4. System Threats
5. Computer Security Classifications
6. What happens when a user wants to open the file?
7. What happens when a user calls lseek system call?
8. Ans: **lseek()** system call repositions the read/write file offset i.e., it changes the positions of the read/write pointer within the file. In every file any read or write operations happen at the position pointed to by the pointer. lseek() system call helps us to manage the position of this pointer within a file.
9. What is the difference between library function and system call?

|  |  |
| --- | --- |
| 1. **SYSTEM CALL** | 1. **LIBRARY CALL** |
| 1. 1. | 1. A system call is a request made by the program to enter into kernel mode to access a process.. | 1. A library call is a request made by the program to access a library function defined in a programming library. |
| 1. 2. | 1. In kernel mode the programs are directly accessible to the memory and hardware resources. | 1. In user mode, the programs cannot directly accessible to the memory and hardware resources. |
| 1. 3. | 1. In system call, the mode is executed or switches from user mode to Kernel mode. | 1. In library call, the mode is executed in user mode only. |

1. What is the use of this project?
2. Ans: Used to create Virtual file system on RAM .
3. What are the difficulties that you faced in this project?
4. Ans: This Project operating on Regular files only but special file system not supported by this project.
5. Is there any improvement needed in this project?
6. **Explain the internal working of below system calls**
7. **(Write the solutions in the documentation)**
8. Open: Optional and relevant only when creating a new file, defines the [file permissions](https://en.wikipedia.org/wiki/File_permissions). These include read, write or execute the file by the owner, group or all users. The mode is masked by the calling process's [umask](https://en.wikipedia.org/wiki/Umask" \o "Umask): bits set in the umask are cleared in the mode.
9. Close:   
   A **close system call** is a [system call](https://en.wikipedia.org/wiki/System_call) used to close a [file descriptor](https://en.wikipedia.org/wiki/File_descriptor) by the [kernel](https://en.wikipedia.org/wiki/Kernel_(operating_system)). For most [file systems](https://en.wikipedia.org/wiki/File_system), a [program](https://en.wikipedia.org/wiki/Computer_program) terminates access to a [file](https://en.wikipedia.org/wiki/File_(computing)) in a filesystem using the close system call.
10. read : In modern [POSIX](https://en.wikipedia.org/wiki/POSIX) compliant [operating systems](https://en.wikipedia.org/wiki/Operating_systems), a program that needs to access data from a [file](https://en.wikipedia.org/wiki/Computer_file) stored in a [file system](https://en.wikipedia.org/wiki/File_system) uses the **read system call**. The file is identified by a [file descriptor](https://en.wikipedia.org/wiki/File_descriptor) that is normally obtained from a previous call to [open](https://en.wikipedia.org/wiki/Open_(system_call)). This system call reads in data in [bytes](https://en.wikipedia.org/wiki/Byte), the number of which is specified by the caller, from the file and stores then into a buffer supplied by the calling process.
11. The read system call takes three arguments:
12. The file descriptor of the file.
13. the buffer where the read data is to be stored and
14. the number of bytes to be read from the file.
15. Write: The **write** is one of the most basic [routines](https://en.wikipedia.org/wiki/Subroutine) provided by a [Unix-like](https://en.wikipedia.org/wiki/Unix-like) [operating system kernel](https://en.wikipedia.org/wiki/Operating_system_kernel). It writes data from a buffer declared by the user to a given device, such as a file. This is the primary way to output data from a program by directly using a system call. The destination is identified by a [numeric code](https://en.wikipedia.org/wiki/File_descriptor). The [data](https://en.wikipedia.org/wiki/String_(computer_science)) to be written, for instance a piece of text, is defined by a [pointer](https://en.wikipedia.org/wiki/Pointer_(computing)) and a size, given in number of bytes.
16. write thus takes three arguments:
17. The file code ([file descriptor](https://en.wikipedia.org/wiki/File_descriptor) or **fd**).
18. The pointer to a [buffer](https://en.wikipedia.org/wiki/Data_buffer) where the data is stored (**buf**).
19. The number of bytes to write from the buffer (**nbytes**).
21. lseek : **lseek (C System Call)**: lseek is a system call that is used to change the location of the read/write pointer of a file descriptor. The location can be set either in absolute or relative terms.  
    **Function Definition**
22. *off\_t lseek(int fildes, off\_t offset, int whence);*
23. stat : Stat system call is a system call in Linux to check the status of a file such as to check when the file was accessed. The stat() system call actually returns file attributes. The file attributes of an inode are basically returned by Stat() function. An inode contains the metadata of the file. An inode contains: the type of the file, the size of the file, when the file was accessed (modified, deleted) that is time stamps, and the path of the file, the user ID and the group ID, links of the file, and physical address of file content.
24. Chmod: **chmod**() changes the mode of the file specified whose pathname
25. is given in *pathname*, which is dereferenced if it is a symbolic
26. link.
27. 6.unlink : In [Unix-like](https://en.wikipedia.org/wiki/Unix-like) operating systems, **unlink** is a [system call](https://en.wikipedia.org/wiki/System_call) and a [command line](https://en.wikipedia.org/wiki/Command-line_interface) utility to delete files. The program directly interfaces the system call, which removes the file name and (but not on [GNU](https://en.wikipedia.org/wiki/GNU) systems) directories like [rm](https://en.wikipedia.org/wiki/Rm_(Unix)" \o "Rm (Unix)) and [rmdir](https://en.wikipedia.org/wiki/Rmdir" \o "Rmdir).[[1]](https://en.wikipedia.org/wiki/Unlink_(Unix)#cite_note-1) If the file name was the last [hard link](https://en.wikipedia.org/wiki/Hard_link) to the file, the file itself is deleted as soon as no program has it open.[[2]](https://en.wikipedia.org/wiki/Unlink_(Unix)#cite_note-2)

**Explain use of below commands**

**(Write the solutions in the documentation)**

ls: Running **ls command** with no option list files and directories in a bare format where we won’t be able to view details like file types, size, modified date and time, permission and links, etc.

1. ls – l : Here, **ls -l** (**-l** is a character, not one) shows file or directory, size, modified date and time, file or folder name and owner of the file, and its permission.
2. ls – a: List all files including hidden files starting with ‘.‘.
3. rm : The rm command removes the entries for a specified file, group of files, or certain select files from a list within a directory. User confirmation, read permission, and write permission are not required before a file is removed when you use the rm command.
4. Cat: The cat (short for “concatenate“) command is one of the most frequently used commands in Linux/Unix-like operating systems. cat command **allows us to create single or multiple files, view content of a file, concatenate files and redirect output in terminal or files**.
5. Cd: cd command in linux known as **change directory command**. It is used to change current working directory.
6. Chmod: The chmod (short for change mode) command is **used to manage file system access permissions on Unix and Unix-like systems**. There are three basic file system permissions, or modes, to files and directories: read (r) write (w) execute (x)
7. cp: cp command **copies files (or, optionally, directories)**. The copy is completely independent of the original. You can either copy one file to another, or copy arbitrarily many files to a destination directory. In the first format, when two file names are given, cp command copies SOURCE file to DEST file.
8. Df: The df command (short for disk free), is **used to display information related to file systems about total space and available space**. If no file name is given, it displays the space available on all currently mounted file systems.
9. Find: The Linux **find command** is one of the most important and frequently used command command-line utility in Unix-like operating systems. The find command is used to search and [locate the list of files and directories](https://www.tecmint.com/install-locate-command-to-find-files-in-centos/) based on conditions you specify for files that match the arguments.
10. Grep : Grep is an acronym that stands for Global Regular Expression Print. Grep is **a Linux / Unix command-line tool used to search for a string of characters in a specified file**. The text search pattern is called a regular expression. When it finds a match, it prints the line with the result
11. Ln : The ln command is **used to create links to files or directories**.
12. Mkdir: **allows users to create or make new directories**.
13. Pwd : The pwd command **writes to standard output the full path name of your current directory (from the root directory)**. All directories are separated by a / (slash). The root directory is represented by the first /, and the last directory named is your current directory.
14. Touch: The **touch** command's primary function is to modify a timestamp. Commonly, the utility is used for file creation, although this is not its primary function. The terminal program can change the modification and access time for any given file. The **touch** command [creates a file](https://phoenixnap.com/kb/how-to-create-a-file-in-linux) only if the file doesn't already exist.
15. Uname: **To display system information**, use the uname command. Displays the operating system name as well as the system node name, operating system release, operating system version, hardware name, and processor type.
16. stat : The stat is a command which **gives information about the file and filesystem**. Stat command gives information such as the size of the file, access permissions and the user ID and group ID, birth time access time of the file. Stat command has another feature, by which it can also provide the file system information.