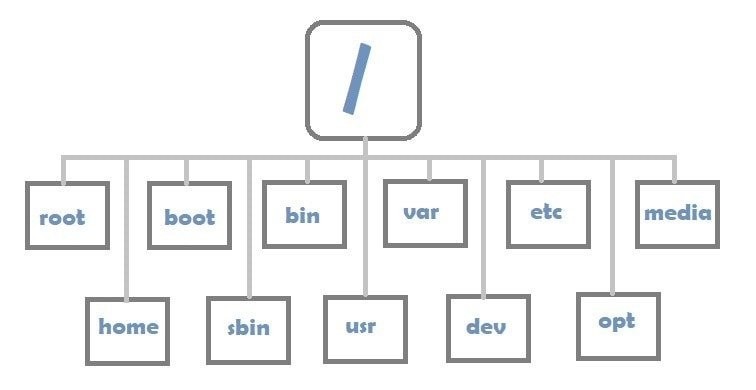
**LAB 4- File System**

Operating systems, the software that powers your computer, rely on a crucial element known as the file system. Think of it as a virtual organizational tool that manages, stores, and retrieves your data efficiently. In the Linux world, a diverse range of file systems has emerged, each crafted to address specific needs and preferences.



1. Linux File system hierarchy

The Linux file system is a complex structure made up of three key layers. At the base, the Logical File System acts as the interface between user applications and the file system, handling tasks such as opening, reading, and closing files. Above this, the Virtual File System allows multiple physical file systems to operate simultaneously, providing a standardized interface for compatibility. At the top, the Physical File System manages the actual storage and organization of data on disk, ensuring efficient allocation and retrieval of memory blocks. Together, these layers create a unified architecture that effectively manages data within the Linux operating system.

Linux File System Structure

A file system is primarily organized into three layers, from top to bottom:

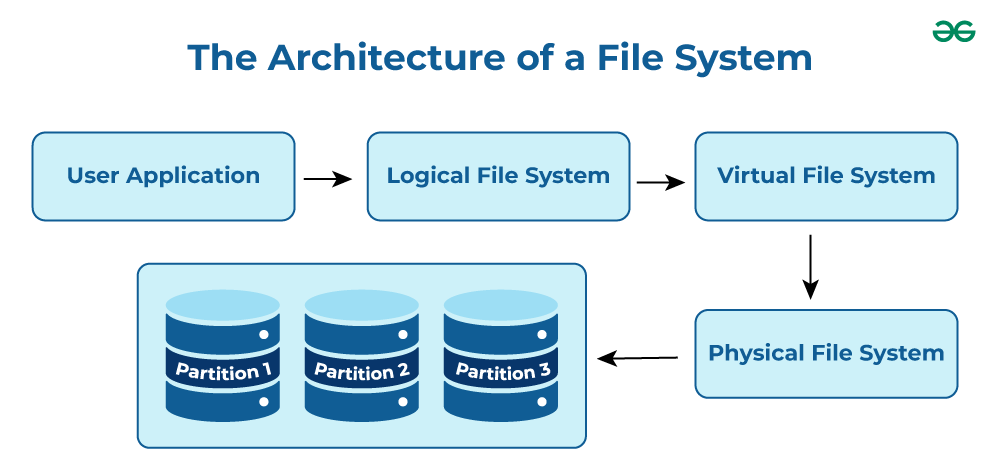
**1. Logical File System:**

The Logical File System serves as the interface between user applications and the file system itself. It manages fundamental operations like opening, reading, and closing files, acting as a user-friendly front-end that allows applications to interact with the file system in a manner that meets user expectations.

**2. Virtual File System:**

The Virtual File System (VFS) is a vital layer that enables multiple physical file systems to operate simultaneously. It provides a standardized interface that allows various file systems to coexist, abstracting the underlying complexities and ensuring compatibility among different implementations.

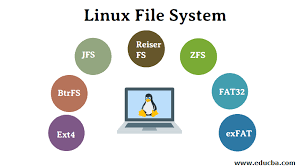
**3. Physical File System:**

The Physical File System is tasked with the direct management and storage of physical memory blocks on disk. It deals with the low-level aspects of data storage and retrieval, interacting directly with hardware components. This layer ensures efficient allocation and use of physical storage resources, enhancing the overall performance and reliability of the file system**.**

1. Architecture of Linux file system

**Characteristics of a File System**

* **Space Management**: how the data is stored on a storage device. Pertaining to the memory blocks and fragmentation practices applied in it.
* **Filename**: a file system may have certain restrictions to file names such as the name length, the use of special characters, and case sensitive-ness.
* **Directory**: the directories/folders may store files in a linear or hierarchical manner while maintaining an index table of all the files contained in that directory or subdirectory.
* **Metadata**: for each file stored, the file system stores various information about that file’s existence such as its data length, its access permissions, device type, modified date-time, and other attributes. This is called metadata.
* **Utilities**: file systems provide features for initializing, deleting, renaming, moving, copying, backup, recovery, and control access of files and folders.
* **Design**: due to their implementations, file systems have limitations on the amount of data they can store.



**The standard top-level Linux directories and their purposes:**

* **/** (root filesystem): The top-level directory that must contain all files necessary to boot the Linux system before any other filesystem is mounted. Other filesystems are mounted at specific points defined by this root structure.
* **/boot**: Contains the static kernel and bootloader configuration files needed to start the Linux system.
* **/bin**: Houses user-executable files.
* **/dev**: Contains device files for all hardware devices connected to the system. These files provide access to the devices but are not the device drivers themselves.
* **/etc**: Holds local system configuration files specific to the host system.
* **/lib**: Includes shared library files required to start the system.
* **/home**: The directory where user files are stored, with each user having a subdirectory under /home.
* **/mnt**: A temporary mount point for filesystems that administrators may use while working or repairing a filesystem.
* **/media**: Designed for mounting external removable media devices, such as USB drives.
* **/opt**: Contains optional files, including vendor-supplied application programs.
* **/root**: The home directory for the root user, distinct from the root filesystem (/).
* **/tmp**: A temporary directory used by the OS and applications for storing transient files. Users can also place files here, but note that files may be deleted at any time without warning.
* **/sbin**: Contains system binary files—executables used for system administration tasks.
* **/usr**: Includes read-only and shareable files, such as executable binaries, libraries, man pages, and various types of documentation.
* **/var**: Stores variable data files, including log files, database files, email inboxes, and web server data.