Operating System

Unit – 6
(Part - B)
File Management

Mayank Mishra
School of Electronics Engineering
KIIT-Deemed to be University

File-System Mounting

- Just as a file must be opened before it is used, a file system must be mounted before it can be available to processes on the system.
- A directory structure can be built out of multiple volumes. These must be mounted to make them available within the file-system name space.
- **Mounting is a process by which the OS makes the files and directories on a storage device available for users to access within the file system.**

QUICK OVERVIEW

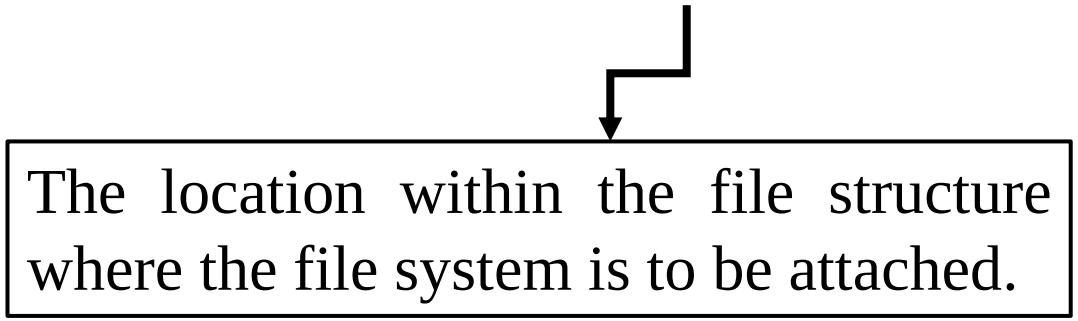
- A **file system** is a method and data structure that an operating system uses to manage how data is stored and retrieved on storage devices like hard drives, SSDs, or USB drives. It defines how data is organized, stored, and accessed, enabling both the OS and the user to interact with files in a consistent and efficient way.

- Without a file system, the computer would have no way to tell where one file ends and another begins, or even how to retrieve the data for a given file. The file system determines things like file names, organization (directories/folders), permissions, and metadata (like file size, date created, or last modified).

- A file system is essential for organizing, storing, and managing files, ensuring the efficient use of storage resources, and providing a way for users and applications to interact with data.

File-System Mounting (Contd.)

- The operating system is given the name of the device and the mount point.



- The OS verifies that the device contains a valid file systems.
- The OS notes in its directory structure that a file system is mounted at the specified mount point.

File-System Mounting (Contd.)

- There can be many file systems in an operating system. Each file system is constituted on a logical disk, i.e., on a partition of a disk.
- Files contained in a file system can be accessed only when the file system is mounted.
- The mount operation is what “connects” the file system to the system’s directory structure.
- An unmount operation disconnects a file system. The mount and unmount operations are performed by the system administrator. These operations provide an element of protection to files in a file system.
- Mounting does not permanently alter the directory structure. Its effect lasts only until the file system is **unmounted** or until the system is booted again.

File-System Mounting (Contd.)

- **Steps to Mount a File System (Example):**
  - **Plug in the Device**: You connect your external device (like a USB drive or a second hard disk) to the computer.
  - **Mount the Device**: When you mount a device, the operating system (OS) takes that external storage and makes it **appear as part of the computer's file system**. It assigns a **"mount point"**—which is just a folder on your computer where the contents of the external device will appear.
  - For example, if you mount a USB drive, its files might show up in a folder like /mnt/usb or /media/usb on a Linux computer.
  - **Access the Files**: Once mounted, you can access the files on the USB drive just like you would any file on your computer's internal storage.
  - **Unmount the Device**:When you're done using the device, you **"unmount"** it. This just means that the system stops showing the device's contents in the file system.
  - **Unmounting** is important because if you remove the device without unmounting it first, you could **lose data** or **damage the files**.

File-System Mounting (Contd.)

- **Mounting Example (Linux):**

  - **Insert the USB Drive:** You plug in your USB stick.
  - **Mount the Drive:**

    Open a terminal and run the command: **mount /dev/sdb1 /mnt/usb**

    - /dev/sdb1 is the device (the USB drive).
    - /mnt/usb is the folder (the mount point) where the device will appear.

File-System Structure

- Hard disks have two important properties that make them suitable for secondary storage of files in file systems:
  - (1) Blocks of data can be rewritten in place, and (2) they are direct access, allowing any block of data to be accessed with only (relatively) minor movements of the disk heads and rotational latency.

- File systems organize storage on disk drives, and can be viewed as a layered design.

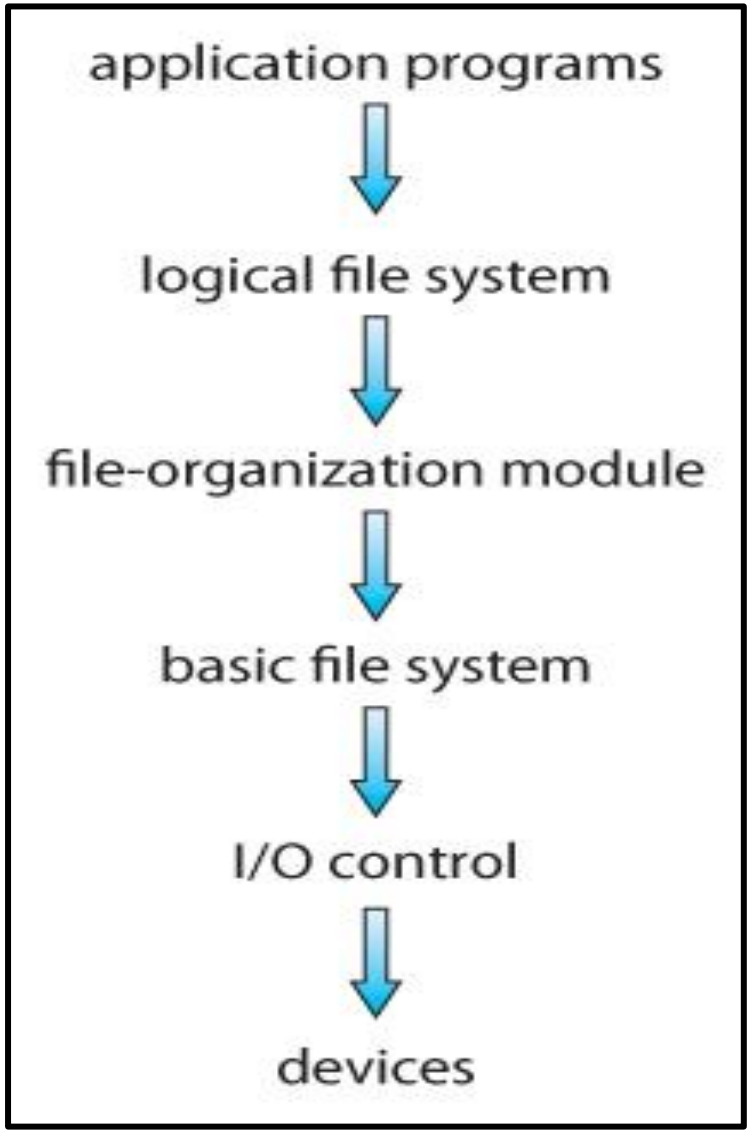


Fig. Layered file system

File-System Structure (Contd.)

- The ***logical file system*** deals with all of the meta data associated with a file, i.e. everything about the file except the data itself. This level manages the directory structure and the mapping of file names to ***file control blocks, FCBs***, which contain all of the meta data as well as block number information for finding the data on the disk.

- The **file organization module** knows about files and their **logical** blocks, and how they map to **physical** blocks on the disk. In addition to translating from logical to physical blocks, the file organization module also maintains the list of free blocks, and allocates free blocks to files as needed.

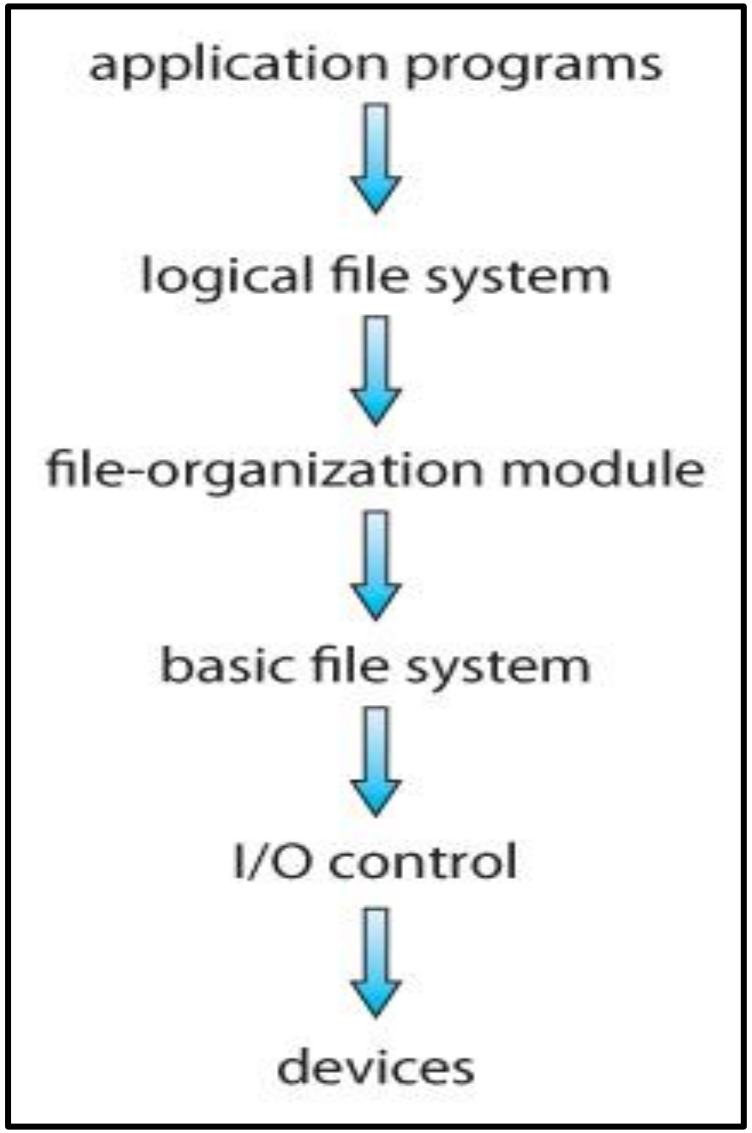


Fig. Layered file system

File-System Structure (Contd.)

- The **basic file system** level works directly with the device drivers in terms of retrieving and storing raw blocks of data, without any consideration for what is in each block. Depending on the system, blocks may be referred to with a single block number, (e.g. block #234234), or with head-sector-cylinder combinations.

- ***I/O Control*** consists of ***device drivers***, special software programs (often written in assembly) which communicate with the devices by reading and writing special codes directly to and from memory addresses corresponding to the controller card's registers. Each controller card (device) on a system has a different set of addresses (registers, a.k.a. ***ports***) that it listens to, and a unique set of command codes and results codes that it understands.

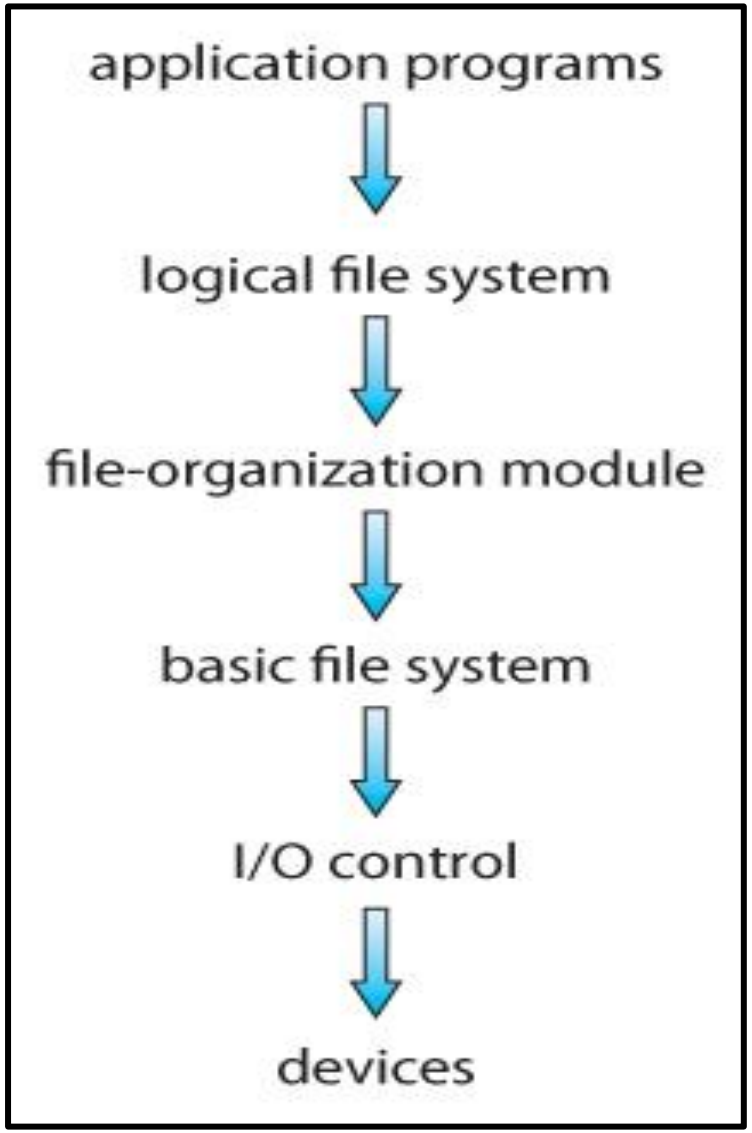


Fig. Layered file system

File-System Structure (Contd.)

- The **basic file system** level works directly with the device drivers in terms of retrieving and storing raw blocks of data, without any consideration for what is in each block. Depending on the system, blocks may be referred to with a single block number, (e.g. block #234234), or with head-sector-cylinder combinations.

- ***I/O Control*** consists of ***device drivers***, special software programs (often written in assembly) which communicate with the devices by reading and writing special codes directly to and from memory addresses corresponding to the controller card's registers. Each controller card (device) on a system has a different set of addresses (registers, a.k.a. ***ports***) that it listens to, and a unique set of command codes and results codes that it understands.

- At the lowest layer are the physical devices, consisting of the magnetic media, motors & controls, and the electronics connected to them and controlling them.

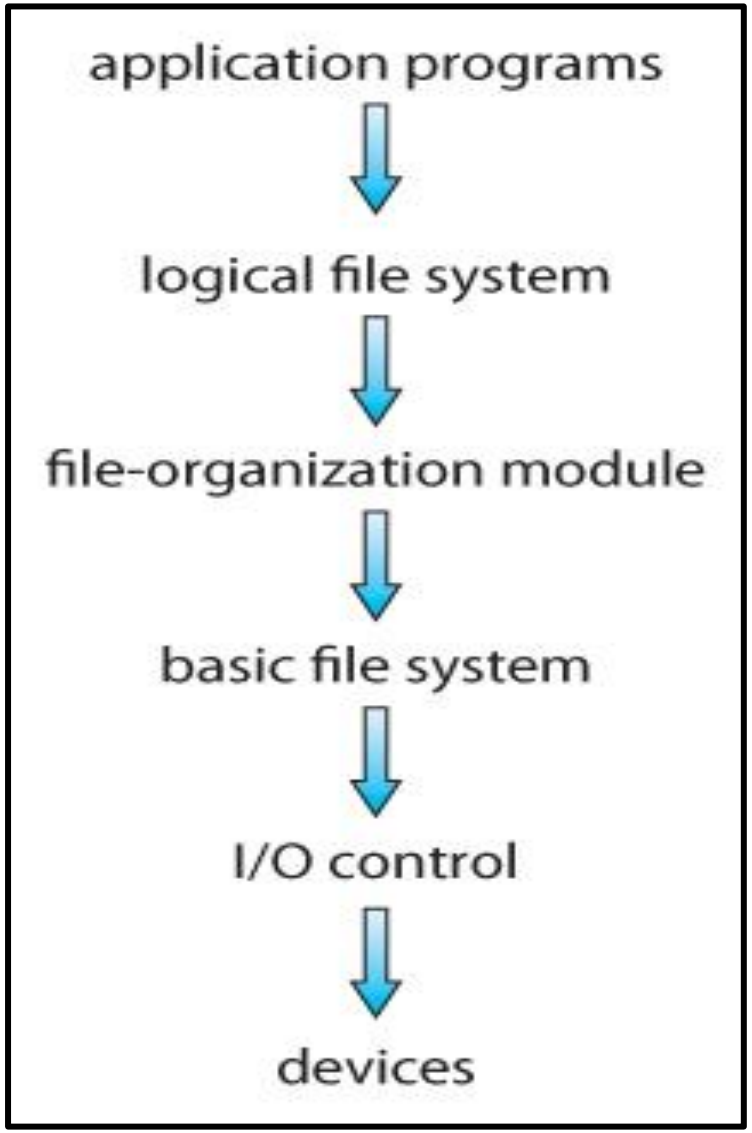


Fig. Layered file system

References

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, “Operating System Concepts,” Eleventh Edition (Willey).

2. Andrew S. Tanenbaum, “Modern Operating Systems”, Fourth Edition (Pearson Publications), 2014.

3. https://www.geeksforgeeks.org/

4. https://www.javatpoint.com/

5. https://www.tutorialspoint.com/

6. https://www.nesoacademy.org/

7. https://www.tpointtech.com/