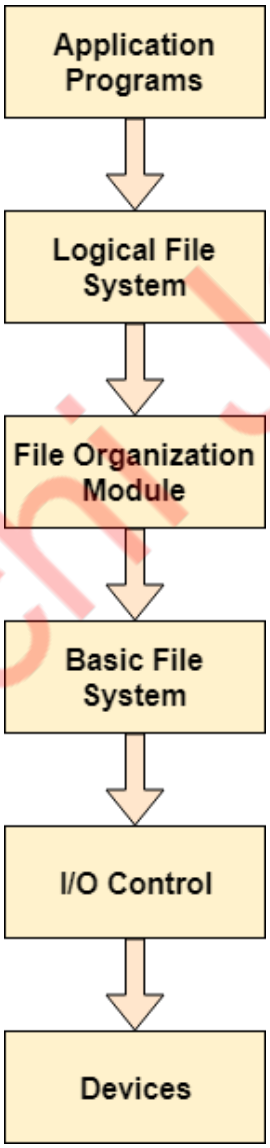
File System Structure

- File System provides efficient access to the disk by allowing data to be stored, located and retrieved in a convenient way.
- A file System must be able to store the file, locate the file and retrieve the file.
- Operating Systems use a layering approach for file systems.
- Every layer of the file system is responsible for some activities.



- When an ***application program*** asks for a file, the first request is directed to the ***logical file system***.

- The ***logical file system*** contains the ***Metadata*** of the file and directory structure. If the application program doesn't have the required permissions of the file then this layer will throw an error. Logical file systems also verify the path to the file.
- In order to store and retrieve the files, the logical blocks need to be mapped to physical blocks. This ***mapping*** is done by the ***File organization module***. It is also responsible for free space management.
- Once the File organization module decides which physical block the application program needs, it passes this information to the basic file system. The ***basic file system*** is responsible for issuing the commands to I/O control in order to fetch those blocks.
- ***I/O controls*** contain the codes by using which it can access hard disk. These codes are known as device drivers. I/O controls are also responsible for handling interrupts.

File System Implementation

File system can be **implemented** by using two types data structures :

1. **1. On-disk Structures –**

   Generally they contain information about total number of disk blocks, free disk blocks, location of them and etc. Below given are different on-disk structures :

   1. **Boot Control Block –**

      It is usually the first block of volume and it contains information needed to boot an operating system.

   2. **Volume Control Block –**

      It has information about a particular partition ex:- free block count, block size and block pointers etc.

   3. **Directory Structure –**

      They store file names and associated inode numbers.

   4. **Per-File FCB –**

      It contains details about files and it has a unique identifier number to allow association with directory entry.

2. **In-Memory Structure :**

   They are maintained in main-memory and these are helpful for file system management for caching. Several in-memory structures given below :

   1. **Mount Table –**

      It contains information about each mounted volume.

   2. **Directory-Structure cache –**

      This cache holds the directory information of recently accessed directories.

   3. **System wide open-file table –**

      It contains the copy of FCB of each open file.

   4. **Per-process open-file table –**

      It contains information opened by that particular process and it maps with appropriate system wide open-file.

File System Mounting

- Each filesystem has its own *root directory*.
- The filesystem whose root directory is the root of the system's directory tree is called *root filesystem*.
- Other filesystems can be mounted on the system's directory tree; the directories on which they are inserted are called *mount points*.
- A mounted filesystem is the *child* of the mounted filesystem to which the mount point directory belongs.
- For instance, the /proc virtual filesystem is a child of the root filesystem (and the root filesystem is the *parent* of /proc).