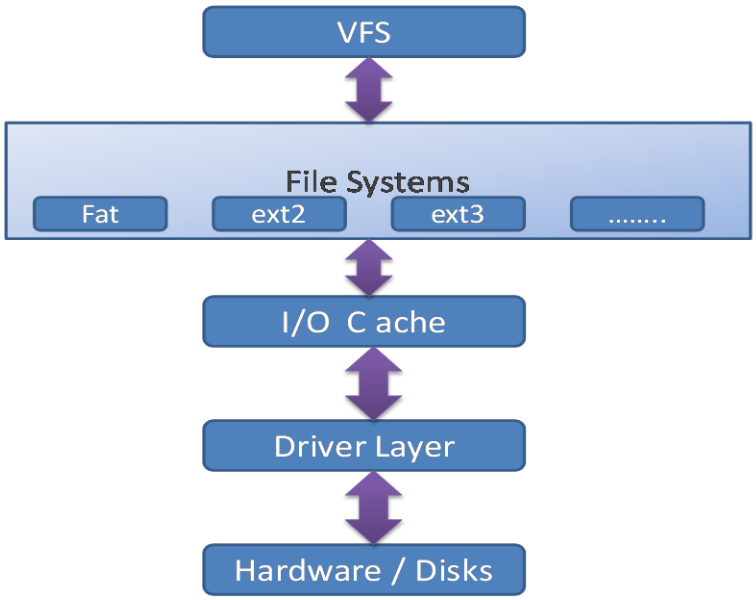
Linux I/O Architecture

Class Notes

Linux I/O Architecture

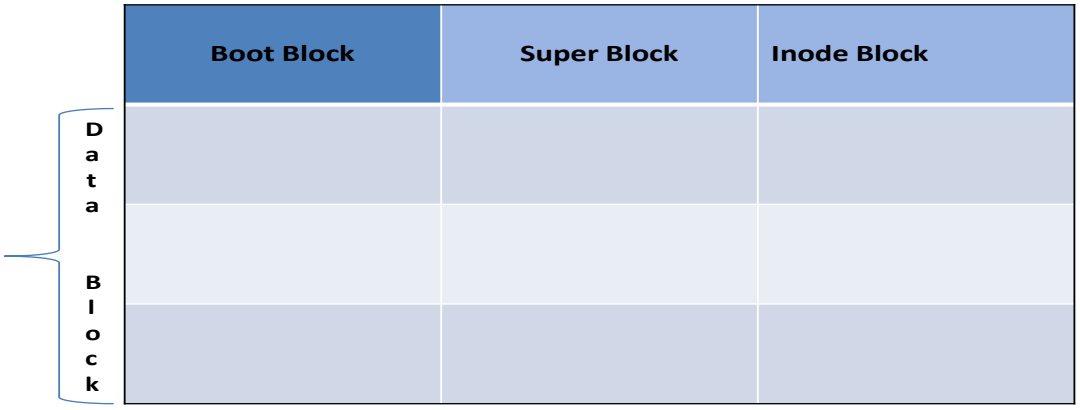
- Linux I/O architecture provides a common API interface for the applications to initiate I/O transfers on various soft and hard resources.



File systems

- File systems organized disk into four
  1) Boot block
  2) Super block
  3) Inode block
  4) Data block
- Mounting is the process of loading super block and inode block from disk to memory which is done by compatible file system.
- For a mount operation to succeed a compatible file system service that can understand the format of the super block and inode table is required.
- Command for mounting in Linux

  mount -t<fs type> **<**source device**>** **<**mount point**>**



- From Boot block, a set of blocks (super block & inode block) are reserved for use of file system called FS block.

Virtual File System (VFS)

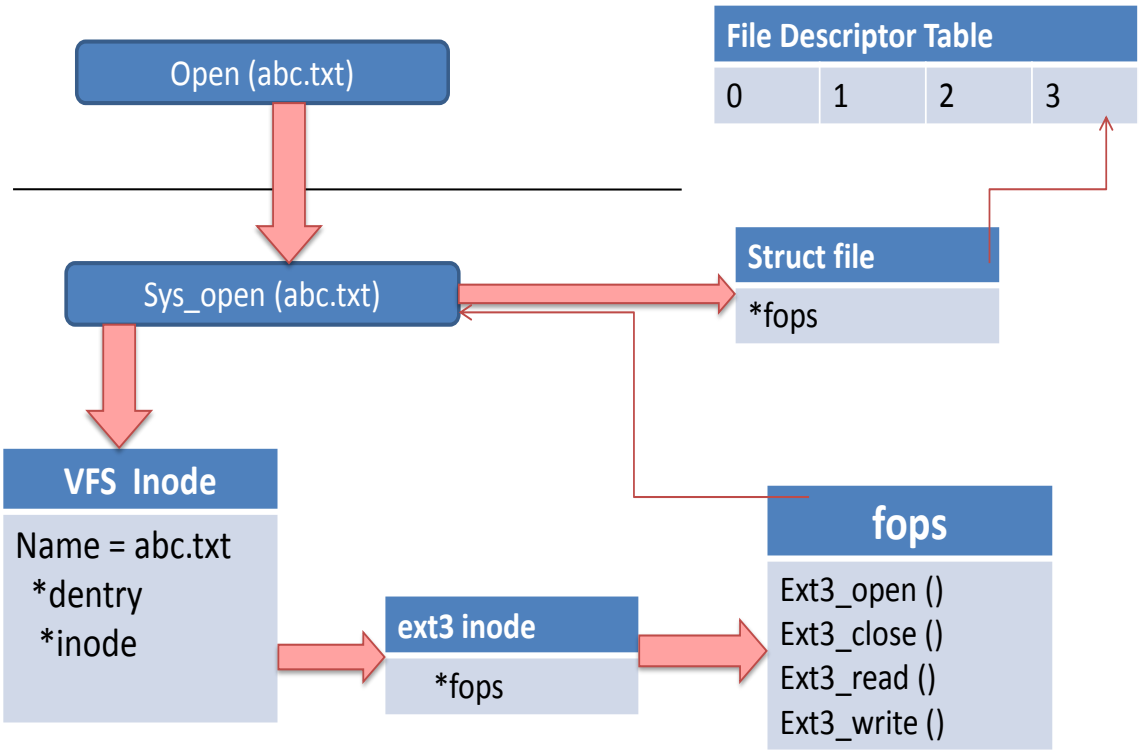
- VFS is a kernel abstraction layer that hides file system operations from the applications.
- This method of abstraction provides the following benefits for the application developers and users.
    1) A limited common API interface to initiate operations on any type of file (irrespective of file system).
    2) All mounted files (disks) can be unified into a single file tree (instead of drives).
    3) Changes to file system layer will not affect existing applications.
    4) Support for new file systems will not require changes to application layer.
- VFS can be used as an application interface to provide other services like IPC, device access, provided respective services register with the VFS.

VFS Request Switch

- When a disk is mounted, inode tree is copied from disk into appropriate file system cache. (Term **cache** referred to reserved region of memory for a specific kernel service usage).

- Virtual file system maintains a file tree in the VFS cache made up of virtual inode instances. Each virtual inode represents a file system inode in VFS cache.

File Systems



- When file system open call gets invoked, it verifies the caller applications request to access file and returns zero or a negative number based on application privilege validation.

- When file system open routine returns zero, sys_open allocates a new file descriptor instance and stores the reference of file operations.
- File descriptor address is stored in the caller process's file descriptor table (Table is implemented as an array).
- The offset number of the file descriptor table is returned to the caller.

File descriptor

- It is an instance of a structure that contains pointer to relevant file operations.
- It also holds application specific file access attributes.
- The address of the file descriptor is stored in the process PCB file descriptor table.
- File descriptors are process local data structures.