

# Linux kernel-VFS

---

Here's a summary of the Virtual File System (VFS) in Linux, along with an example:

## What is VFS?

The Virtual File System (VFS) is a layer of abstraction between the file system and the operating system. It provides a common interface for different file systems to interact with the kernel, allowing them to access files in a uniform way.

## Key Features:

1. **Abstraction:** VFS abstracts away the details of each file system, making it possible to access files on different types of storage devices (e.g., hard drives, USB sticks, network shares) using a common interface.
2. **Portability:** By providing a standardized API for accessing files, VFS makes it easier to port applications between different Linux distributions and architectures.
3. **Flexibility:** The VFS layer allows new file systems to be added without modifying the kernel or existing code.

## VFS Components:

1. **Superblock:** A data structure that contains metadata about a file system (e.g., inode cache, block allocation).
2. **Inode Cache:** A cache of inodes (file descriptors) to reduce lookup times.
3. **File System Operations:** Functions that perform basic file system operations (e.g., open, close, read, write).

## Example: Accessing a File using VFS

Suppose we want to access a text file named `example.txt` located on an ext4 file system.

1. **User Space Application:** An application like `cat` or `vim` requests access to the file `example.txt`.
2. **VFS Interface:** The application uses the VFS interface (e.g., `open()`, `read()`) to request access to the file.
3. **File System Driver:** The VFS layer identifies the ext4 file system driver and passes the request to it.
4. **ext4 File System Driver:** The ext4 driver performs the necessary operations (e.g., reading the file's metadata, allocating blocks) to grant access to the file.
5. **VFS Interface:** The VFS layer provides the requested data (the contents of `example.txt`) back to the application.

In this example, the VFS layer abstracts away the details of accessing a file on an ext4 file system, making it possible for user space applications to interact with files on different types of storage devices using a common interface.

---

*Curated by Brajesh Kumar*