**DEVELOPING A SIMPLE FILE SYSTEM WITH USER SPACE UTILITIES**

-Rahul S

-2016103568

**Abstract**

File system is a core component of a functional operating system. Traditional File system development has been confined to the kernel space. A customized, purpose-built, and user-driven. File system development involves extensive knowledge of kernel internals, tools and processes. Alternatively, user-space File systems are preferred over the kernel space File system, for ease of development, portability and developing prototypes File systems, particularly for intuitive abstraction of “non-file” objects.

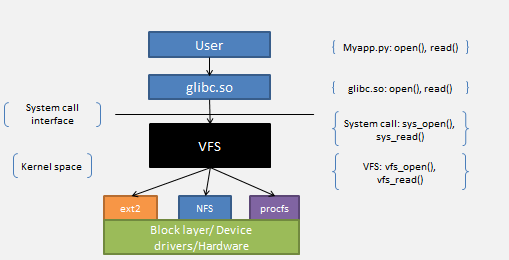
In UNIX kernel, a File system implementation is abstracted with a virtual File system (VFS).VFS is an umbrella that acts as an interface to all available (mounted) File systems on a computer system. VFS itself has generic implementation of File system operations. Its major task is to:

- Decouple File system operation from the interface

- Manage File system ‘mount’

- Figure out the target File system for an file I/O request and route the request

- Provide a consistent interface (POSIX) to user application for file I/O



An illustration of how VFS communicates with actual file system

FUSE is a loadable kernel module (fuse.ko) and acts as a File system to the VFS. It registers itself with the VFS and opens a special device “/dev/fuse”. FUSE module (fuse.ko) is an interface between the fuse device and the VFS. It receives file I/O requests from the VFS and writes those requests to “/dev/fuse” device. The use space library “lifuse.so” polls the device, and read

“/dev/fuse” device.

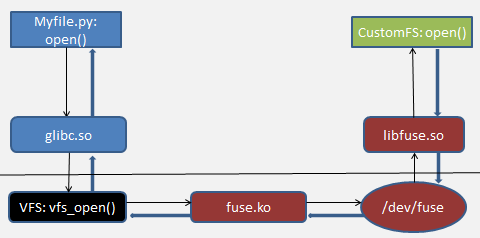


Illustration showing the working of a FUSE file System

FUSE also provides a utility called **fusermount** to unmount the filesystem without special privileges.

My project intends to use the FUSE API to develop a file system that stores files

along with their contents encrypted. Reading files will perform decryption operations on the file contents and write operations automatically encrypt their content and store them in the files.

This project will be implemented in Python with fuse bindings for Python and **fusepy** python library.

The encryption of file contents will be performed with some python libraries that implement standard encryption algorithms(SHA).