VE482 Lab11

Kexuan, Weili Dec 17, 2021

1. What is a file system?

File system is a method and data structure that the operating system uses to control how data is stored and retrieved (wikipedia).

File systems provide an abstraction, both to the user of a computer system and to the programmer. They present a uniform hierarchical view of data, even though this data may actually be spread across various areas of the disk, or may even be spread across multiple disks or multiple computer systems. A file system is composed of a set of files and directories. Files store the actual data while directories are lists of files and other directories.

2. How is linux VFS working?

Idea: abstract out that part of the file system that is common to all file systems and put that code in a separate layer that calls the underlying concrete file systems to actually manage the data. All system calls relating to files are directed to the virtual file system for initial processing. These calls, coming from user processes, are the standard POSIX calls, such as open, read, wr ite, lseek, and so on. Thus the VFS has an "upper" interface to user processes and it is the well-known POSIX interface.

The VFS also has a "lower" interface to the concrete file systems, which is labeled VFS interface. This interface consists of several dozen function calls that the VFS can make to each file system to get work done. Thus to create a new file system that works with the VFS, the designers of the new file system must make sure that it supplies the function calls the VFS requires. An obvious example of such a function is one that reads a specific block from disk, puts it in the file system's buffer cache, and returns a pointer to it. Thus the VFS has two distinct interfaces: the upper one to the user processes and the lower one to the concrete file systems. (From textbook)

3. What is FUSE?

FUSE: file system in user space. An open source framework for implementing file systems in userspace.

https://www.cs.cmu.edu/~fp/courses/15213-s07/lectures/15-filesys/index.html

1. A program, such as ls, mkdir, or perflab makes a call to a file system routine. For example, open("/test/fuse/file1.txt"). This call gets sent to the kernel.
2. If this file is in a FUSE volume, the kernel passes it on to the FUSE kernel module, which then passes it on to the implementation of that file system (this is the portion we will be writing in this lab).
3. The implementation of open then refers to the actual data structures that represent the file system and returns a file handle. It is open's job to take a concrete view of data (bits stored on a hard drive) and present an abstract view (a hierarchically organized file system).
4. The kernel returns the result of the open function to the program that originally made the call.