

Operating System Feature Comparison: Filesystems and VFS

by Sinan Topkaya

CS 444 - Spring 2016

Abstract: This paper will examine how Windows and FreeBSD implements filesystems and VFS, then this information will be used to compare these operating systems to Linux Kernel.

WRITING ASSIGNMENT - FILESYSTEMS AND VFS COMPARISON

INTRO

A filesystem is the methods and data structures that an operating system uses to keep track of files on a disk or partition; that is, the way the files are organized on the disk. VFS or virtual filesystem which is an abstraction layer on top of a more concrete file system. The purpose of a VFS is to allow client applications to access different types of concrete file systems in a uniform way.

File systems organize storage on disk drives, and can be viewed as a layered design, at the At the lowest layer are the physical devices, consisting of the magnetic media, motors and controls, and the electronics connected to them and controlling them. Modern disk put more and more of the electronic controls directly on the disk drive itself, leaving relatively little work for the disk controller card to perform. I/O Control consists of device drivers, special software programs which communicate with the devices by reading and writing special codes directly to and from memory addresses corresponding to the controller card's registers. Each controller card on a system has a different set of addresses that it listens to, and a unique set of command codes and results codes that it understands.

The basic file system level works directly with the device drivers in terms of retrieving and storing raw blocks of data, without any consideration for what is in each block. Depending on the system, blocks may be referred to with a single block number, , or with head-sector-cylinder combinations.

The file organization module knows about files and their logical blocks, and how they map to physical blocks on the disk. In addition to translating from logical to physical blocks, the file organization module also maintains the list of free blocks, and allocates free blocks to files as needed.

The logical file system deals with all of the meta data associated with a file (UID, GID, mode, dates, etc), everything about the file except the data itself. This level manages the directory structure and the mapping of file names to file control blocks, FCBs, which contain all of the meta data as well as block number information for finding the data on the disk.

Common file systems used include UFS, FFS, FAT, FAT32, NTFS, CD-ROM, EXT2, EXT3...

WINDOWS

Windows operating system supports CDFS, UDF, FAT12, FAT16, FAT32, exFAT and NTFS file system formats and each of these formats is best suited for certain environments.

Windows supported formats

CDFS or CD-ROM file system, is a read-only file system driver that supports a superset of the ISO-9660 format as well as a superset of the Joliet disk format. It has couple restrictions: A maximum file size cannot exceed 4GB, and it may only have 65,535 directories.^[1]

The Windows UDF file system implementation is OSTA UDF-compliant. OSTA defined UDF in 1995 as a format to replace the ISO-9660 format for magneto-optical storage media, mainly DVD-ROM. UDF is included in the DVD specification and is more flexible than CDFS. This also has some restrictions such as the file/directory names cannot exceed 254 ASCII, files can be sparse and their sizes are specified with 64bits. UDF provides read-write support for Blu-ray, DVD-RAM.

Windows supports the FAT file system primarily for compatibility with other operating systems in multiboot systems, and as a format for flash drives or memory cards^[1]

Simply FAT12 is 12-bit cluster identifier, FAT16 is 16-bit and FAT32 is 32-bit meaning that in logical order they allow more clusters as their bits increase.

Designed by Microsoft, the Extended File Allocation Table file system (exFAT, also called FAT64) is an improvement over the traditional FAT file systems and is specifically designed for flash drives. The main goal of exFAT is to provide some of the advanced functionality offered by NTFS, but without the metadata structure overhead and metadata logging that create write patterns not suited for many flash media devices.^[1]

FAT64 meaning that it supports 64-bit and therefore can have that many sectors.

And lastly NTFS file system is the native file system format of Windows. NTFS uses 64-bit cluster numbers. This capacity gives NTFS the ability to address volumes of up to 16 exaclusters; however, Windows limits the size of an NTFS volume to that addressable with 32-bit clusters, which is slightly less than 256 TB (using 64-KB clusters).

The algorithm used by NTFS is also used by FAT, CDFS and third-party file systems to generate long file name:

1- Remove from the long name any characters that are illegal in MS-DOS names, including spaces and Unicode characters. Remove preceding and trailing periods. Remove all other embedded periods, except the last one.

2- Truncate the string before the period (if present) to six characters (it may already be six or fewer because this algorithm is applied when any character that is illegal in MS-DOS is present in the name); if it is two or fewer characters, generate and concatenate a four-character hex checksum string. Append the string `n` (where `n` is a number, starting with 1, that is used to distinguish different files that truncate to the same name). Truncate the string after the period (if present) to three characters.

3- Put the result in uppercase letters. MS-DOS is case-insensitive, and this step guarantees that NTFS won't generate a new name that differs from the old only in case.

4- If the generated name duplicates an existing name in the directory, increment the `n` string. If `n` is greater than 4, and a checksum was not concatenated already, truncate the string before the period to two characters and generate and concatenate a four-character hex checksum string.

FREEBSD

Since both FreeBSD and Linux have their origins in UNIX, their operating systems coexistence have similar file systems. It handles multiple types of files by hiding the implementation details of single file types behind a software layer, the Virtual File System (VFS). This allows for a lack of duplication due to layering with object-oriented design principles. Both systems incorporate the Virtual File System, which makes all the files on the devices appear to exist in a single hierarchy.

The Extended Filesystem(EXT2) is one of the most widely used file system in FreeBSD, it supports standard Unix file types such as regular files, directories, device special files and symbolic links. FreeBSD preferred native file system for local disks is the FFS, this format supports file systems up to 16TB and depending on some parameters files up to several terabytes. File systems update their structural information called metadata by synchronous writes. Each metadata update may require many separate writes, and if the system crashes during the write sequence, metadata may be in inconsistent state. At the next boot, the file system check utility called `fsck` must walk through the metadata structures, examining and repairing them.

This operation takes a very long time on large filesystems. Moreover, the disk may not contain sufficient information to correct the structure. This results in misplaced or removed files.[2]

CONCLUSION

Although there are many similarities the Linux directory structure does not just use different names for folders, it uses an entirely different layout. One important thing is that the filesystems implemented by Windows are not case-sensitive meaning that a folder named file and also FILE does not exist in the same directory. In Windows partitions and devices are exposed as drive letters, even if you have multiple hard drives, or even removable devices, each of them is available under its own drive letter. While Linux just makes these devices or other filesystems accessible at arbitrary directories also in Linux everything is represented as a file meaning that the dcd drive will be called a file cdrom. Windows uses a lot of exclusive access locks while Linux does not.

FreeBSD and Linux support encrypted file systems, quotas and RAID but FreeBSD also supports AFSM Coda, ReiserFS, UFS, XFS and FF which are not supported by Linux or they aren't exactly the same. Since EXT2 is widely used for FreeBSD and EXT2 is used by Linux, it appears that although both operating systems are really similar, Linux is more efficient.

REFERENCES

- [1] D. S. Mark Russinovich and A. Ionescu, *Windows Internals, Part 1* 6/e. Microsoft Press, 2012.
- [2] M. K. McCusick and G. V. Neville-Neil, *Design and Implementation of the FreeBSD Operating System* 2/e. Addison-Wesley, 2015.