Computer File Systems

File systems https://en.wikipedia.org/wiki/File_system

A file system organises files on a storage device so users and software can find a specific file.

An example of a hierarchical system from botany:

Kingdom	Phylum	Class	Order	Family	Genus	Species
Animalia	*Chordata*	*Mammalia*	*Primata*	*Hominidae*	*Homo*	*Sapiens*

Computer file systems organise files in a similar manner called a "path".
Using the botany example, the file on John Smith would be found at this pathname:

[drive:]
\Animalia\Chordata\Mammalia\Primata\Hominidae\Homo\Sapiens\JohnSmith.txt

The concept of hierarchical filesystems is derived from paper filing systems:

- XYZ Company Office building
  - Departments within the company – Sales, Warehouse, Accounting
    - Filing cabinets for Orders (within the Sales Dept.)
      - Drawers – labeled A through Z
        - Folders – by Customer name
          - Sales orders

Each company has a file system using the hierarchical concept, but each company establishes its own hierarchy and naming convention according to its needs.

A computer's file system uses folders (aka directories) and sub-folders:

- XYZ Company Office building
- Computer server
- `/` [root of the file system]
- `/Sales` [Sales dept. folders and files]
- `/Sales/Orders` [Sales dept.'s Orders folder]
- `/Sales/Orders/ABCinc` [Sales Orders of ABCinc customer]
- `/Sales/Orders/ABCinc/123450.txt` [ABCinc customer order file]
- `/Sales/Orders/ABCinc/123451.txt` [ABCinc customer order file]
- `/Sales/Orders/ABCinc/123452.txt` [ABCinc customer order file]

The full "path" of an order file is `/Sales/Orders/ABCinc/123450.txt`

Operating systems enable file access, directory/folder operations, and authorization security via the **logical file system** which connects software applications with the data within files. The logical file system provides a uniform application program interface (API) for file operations — OPEN, CLOSE, READ, etc., and passes the requested operation to the physical layer it for processing on the storage device.

Computer File Systems

IBM, Microsoft, Apple, and Linux all use their own proprietary methods at the Operating Systems level. Despite this, most file systems look the same above the OS level to developers and software.

File Systems have different underlying structure and logic, properties for files (e.g. maximums for filename and pathname length, allowable characters for names and case sensitivity, max file size), security attributes, and more.

Optical discs usually use ISO 9660 file system because optical discs were designed for "cross-platform compatibility". USB flash drives are formatted (as FAT32 by default) for cross-platform compatibility.

Mounting Drives

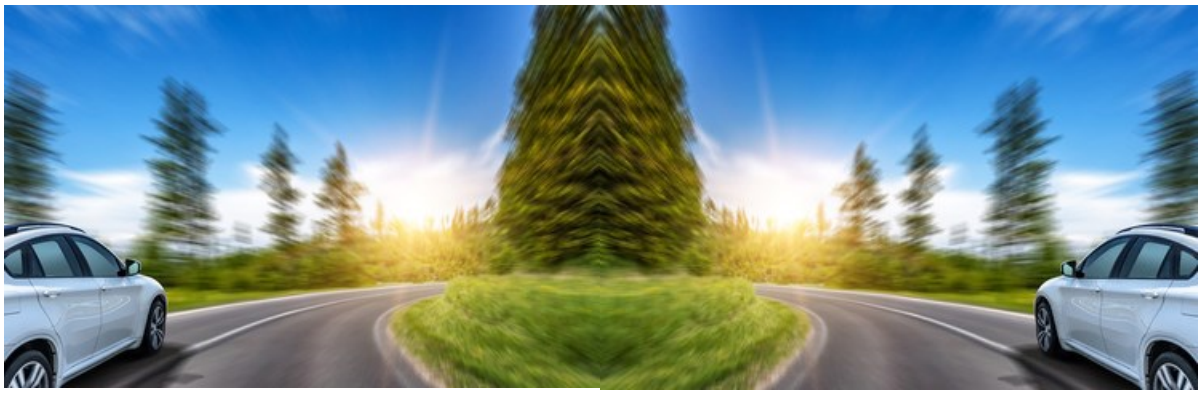
Each operating system (OS) has its own method for handling files even though almost all file systems look the same to developers and software. *Some* OSs are able to read *some* other OSs file systems with limitations but that should not be assumed.
An analogy: a driver who knows the rules of the road (operating system) and can operate a vehicle (storage device with a file system).

In most parts of the world and in most cars, a driver gets into a car through the front door on the left side expecting to see a steering wheel and controls in the same locations as in almost every other car.

Imagine a Canadian getting into a car England. 'What the heck is the steering wheel doing *over there*?' or '*over here?*' if you are a passenger. To use a car well, the car's controls must be suited to the country where the car is driven. In computer terms, what is the file system of the car? It is either for left-hand or right-hand drive.

Now that you are driving a car, you come to a roundabout: do you go clockwise or counterclockwise? It depends on which side of the road they drive in that country. Who has the right-of-way: the traffic already circling the roundabout or traffic coming into it? (it should be the former – the latter caused lots of problems.)

The driver must know the rules of the road in order to operate the car safely. An empty road looks much the same in Canada as in England; there are few, if any, clues on the road itself as to which side to drive on. Knowing how to use the road is important so as not to crash into others.



Tourists in London, England are injured all the time as they cross the road; they look left, see no cars coming, and step off the curb into the road where the cars come from the right.

You have to know how to use the physical hardware (the car) *and* the logical operation of it (rules of the road). If you plug a metaphorical car into a United Kingdom road OS, the OS will recognize cars only with right-hand drive; plug into a Canadian road OS and only left-hand drive cars are recognized.

The operating system knows how to recognize and use the physical hardware (the storage device via low level software called a "driver" coincidently) and the logical operation of it to find and open a file and then read and write data to the file.

The process of the OS recognizing a storage device so it can be used is called ***mounting***. (The spare wheel & tire in the car's trunk/boot does not become useful until it is mounted in place of the wheel with the flat tire. Snow tires are not useful until they are mounted on wheels and the wheels mounted on your car. The car's engine drives the wheels.) When the OS mounts a drive, it recognizes the logical file system on the attached physical storage device. The way the file system is presented to the end user through the file browser looks very much the same on most systems – a hierarchical organization beginning at the drive letter or root with multiple levels of directories or folders containing sets of files.

https://en.wikipedia.org/wiki/Mount_(computing)

The **physical file system** layer is concerned with the mechanical/electrical operation of the storage device. It processes physical blocks being read or written. It handles buffering and memory management and is responsible for the physical placement of blocks in specific locations on the storage medium. When the storage device is attached

to the host computer, it is "mounted". The Operating System loads a device driver to interact with device and its physical file system. Then the OS logical file system can interact with the device's physical file system to access directories (folders) and files. The OS then makes the data within files available to software.

Location of storage devices:

- DAS: Direct Attach Storage – local devices (internal HDD & SSD, external plug-ins like USB and optical) attached through an internal bus adapter/driver. DAS has a one-to-one relationship between each storage device and the host (PC or server). For non-removable internal devices, the OS must provide formatting and a file system to enable file-level access. Most removeable external devices are preformatted with a cross-platform compatible filesystem.
- SAN - SANs separate computers from their storage. Storage Area Network is a large extensible block-level data storage system (made up of a whole lotta disks). SAN storage appears to a computer *as if* an allocation of SAN blocks were a disk. (think Lego blocks assembled into whatever size and shape is needed) The computer OS provides a file system on that virtual disk. A SAN is more similar to DAS than NAS. The difference between SAN and DAS is the attachment of the storage device: SAN through a connection (iSCSI or Fibre Channel) to external storage, DAS through a local connection (SCSI) to internal storage.
  - high performance, flexibility, scalability, and extensibility for large enterprises and cloud services
- NAS: Network Attached Storage is file-level computer data storage through a network protocol which effectively appears to the OS as a native file system, but the actual implementation is hidden inside the NAS appliance – essentially a small server. NAS will support different file systems for multiple OSs over a standard network.
  - e.g. a home office NAS will store a file and make it accessible to both Windows and macOS systems even though those OSs use different file systems. Most NAS will provide file access over the Internet, again using different access methods.
  - Used for data sharing among many users on heterogeneous systems on the same intranet in Small to Medium sized Enterprises (SMEs).
- NAS and SAN can provide automatic backup and snapshotting, storage reliability and redundancy, and scalability. DAS requires the host server and administrator to provide those functions.
- cloud providers can provide both NAS and SAN services remotely.
- https://serverfault.com/questions/81723/what-is-the-difference-between-san-nas-and-das

Critical features of contemporary OS file systems *except FAT32, exFAT*

- File Metadata: Windows Properties / Linux `stat` / macOS `xattr`

- Security, Data Integrity (controlling multiple applications attempting to access the same file), Crash protection (journaling), Encryption, Compression
- Snapshot capability for regressing system to previous state. For efficiency, snapshots record only the incremental changes made since the previous snapshot. Flat Backup systems take advantage of this.

Filesystem	**MS Windows**	**Apple macOS**	**Linux**
NTFS	Native	native read-only, 3rd party utility for writing and management	
APFS	3rd party utility	Native	3rd party utility
ZFS, **ext4**	Paragon ExtFS (3rd party utility)		Native
FAT32	Native read/write but 4 GB file size limit, slower I/O. USB storage constraints tolerated for "cross-platform compatibility"		
exFAT **32 vs ex**	Native, MS proprietary format		3rd party utility and/or licensing
Optical	ISO 9660 designed for "cross-platform compatibility"		

Enterprise and high-performance computing file systems:

- IBM Spectrum Scale
- Lustre – Linux Cluster
- Apache Hadoop

https://en.wikipedia.org/wiki/Comparison_of_file_systems

Storage devices

https://www.backblaze.com/b2/hard-drive-test-data.html
https://www.backblaze.com/blog/

https://www.zdnet.com/article/ssd-reliability-in-the-real-world-googles-experience/
evidence that consumer grade HDD and SSD are as reliable as enterprise/server grade.

	Reliability reduced by	**Failure risk**
HDD Hard Disk Drive Low capital cost per GB	Usage	Drive loss > Data loss addressed with RAID redundancy and MAID Massive Array Of Idle

High operating cost due to drive motor and waste heat to be cooled		Drives in large storage systems.
SSD Solid State Drive High capital cost per GB Low operating cost due to no moving parts and minimal heat emissions	Age - need filesystem with data checksum. IBM Spectrum/GPFS, Linux ZFS. Diagnostic utilities needed on Win10, macOS, Linux which, ironically, can reduce the life of drives due to additional intensive use.	Data loss > Drive loss SSD back up *more* important than with HDD