Filesystems

Sumner Evans and Sam Sartor

November 9, 2017

Mines Linux Users Group

Introduction

What are Filesystems?

- Filesystems manage the storage and retrieval of files from storage media.
- Filesystems are an abstraction layer between storage media (SSDs, HDDs, disk drives, even tape drives).
- Filesystems exist on *partitions*, physically contiguous segments of the disk.

Filesystems are Responsible for...

- **Space management:** filesystems allocate and manage space in discrete chunks. Filesystems must keep track of what data is stored at each chunk.
- **Filenames:** identify a storage location in the file system. Can be case sensitive (ext4) or case insensitive (HFS, NTFS).
- **Directories (folders):** group files into separate collections. Modern filesystems allow arbitrary nesting of directories.
- **Metadata:** filesystems store book-keeping information about their contents (e.g. file sizes, last accessed date, owner and permissions, etc.).
- **Access Control:** prevent unauthorized access to files on disk.
- **Data Integrity:** filesystems must be resilient to failure, some are better at this than others.

Current Filesystems

Linux

ext4

Windows & mac

NTFS

HFS and HFS+

Apple has made a ton of filesystems with varying degrees of terribleness.

- **HFS:** Hierarchical File System — Introduced in 1985 with the first Apple computer with a hard drive. Had a limitation of 65,535 files and every file had to take up at least 1 / 65,535th of the disk.
- **HFS+:** Released in 1998 to fix some of the issues with HFS. the core of the filesystem uses case-insensitive NFD Unicode strings, which led Linus Torvalds to say that "HFS+ is probably the worst file-system ever".

APFS

APFS: Apple Filesystem — Introduced in June 2016 to replace HFS+ and is optimized for SSDs. It fixes some of the problems of HFS+. Basically it replicates the work of other modern filesystems which are actually maintained by large communities.

Apple forcibly upgraded all computers to APFS in macOS High Sierra.

Flashdrives

FAT32

Other Options

Alternative Filesystems

Btrfs

B-tree **f**ile **s**ystem (Btrfs) pronounced “Butter FS” or “better FS” or “b-tree FS” was developed starting in 2007 by Oracle.

Pros

- Copy on Write
- Mostly self-healing
- Can convert from `etx*` to Btrfs

Cons

- It’s being depricated by Oracle. RHEL 7.4 includes it, but they are transitioning away. The SUSE project will still use and maintain it.

ZFS

TFS

TFS is a work-in-progress filesystem for the Redox operating system. Intended as a modern alternative to ZFS, the feature list is mouth-watering.

Pros

- Concurrent & non-blocking
- Lightweight full-disk compression
- Zero-overhead revision history
- Automatic corruption detection
- $O(1)$ recursive directory copies
- Designed for solid state drives
- Perfectly resilient to sudden power loss

Cons

- Not implemented yet

Network Filesystems

What is a network filesystem?

You can access remote storage devices over the internet using a *network filesystem*.

NFS

Samba

Virtual Filesystems

What is a virtual filesystem?

A *virtual filesystem* is an abstraction layer that takes in some other source of data and represents it as a mountable structure of files and directories.

Colloquially, any filesystem that is associated with a physical disk is called a virtual filesystem. These might still be backed by other kind of storage, or they might be purely procedural.

tmpfs

`tmpfs` is a filesystem stored in RAM. It appears as a mounted filesystem, but all data is stored in volitile memory. By default, the `/tmp` directory is a `tmpfs`.

Pros

- Useful for storing temporary files such as downloads and program files
- Saves unnecessary disk I/O
- Secure storage for decrypted data, since files are never written to disk

Cons

- Everything in a `tmpfs` will be "deleted" on reboot.

procfs

In *Nix, everything is a file. This includes things such as process information. The way to access this data is through the `proc` filesystem which provides a convenient and standardized method for dynamically accessing process data held in the kernel.

In Linux, `procfs` contains more than just process information including memory information, network utilization statistics, etc.

FUSE

Filesystem in Userspace (FUSE) is an interface for creating filesystems without writing any kernel-level code, which makes it incredibly useful for creating virtual filesystems. It is an available in Linux, FreeBSD, OpenBSD, NetBSD, OpenSolaris, Minix 3, Android, and macOS.

`libfuse`...

- is a C library
- provides a high-level interface for FUSE
- makes creating new filesystems really easy
- has bindings for Python, Rust, etc

sshfs

`sshfs` is a network filesystem implemented through libfuse. You can use it to mount remote directories via ssh.

Pros

- Very easy & quick to setup (one command)
- Remote machine only needs ssh installed

Cons

- Intended to be temporary
- Generally slower than NFS

Questions?

References

- https://en.wikipedia.org/wiki/File_system
- http://www.tldp.org/LDP/sag/html/filesystems.html
- https://arstechnica.com/gadgets/2008/03/past-present-future-file-systems/2/

Copyright Notice

Colorado School of Mines
Linux Users Group