Linux Directory Structure

Tristan Miller

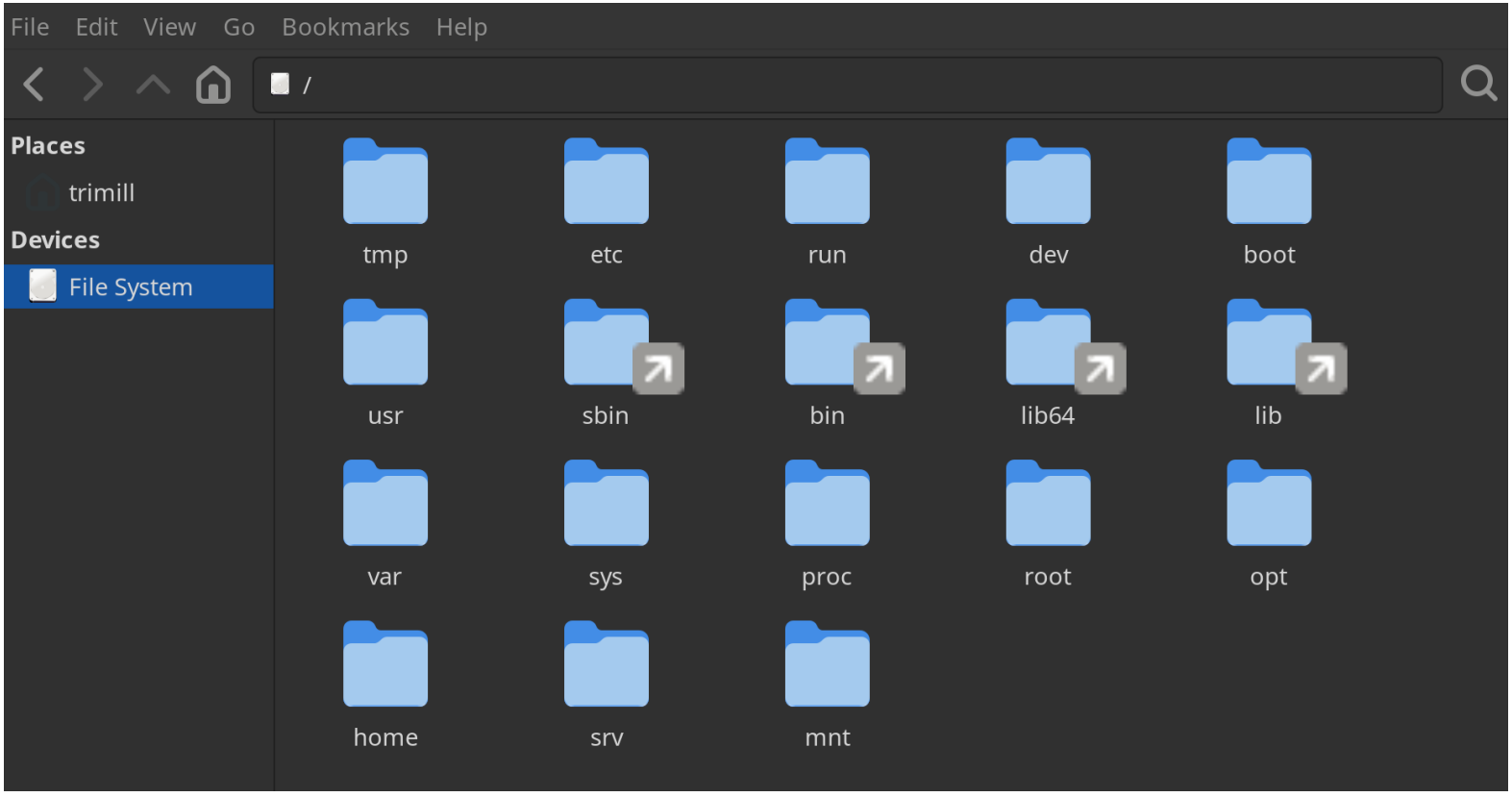
September 2025

The Linux filesystem

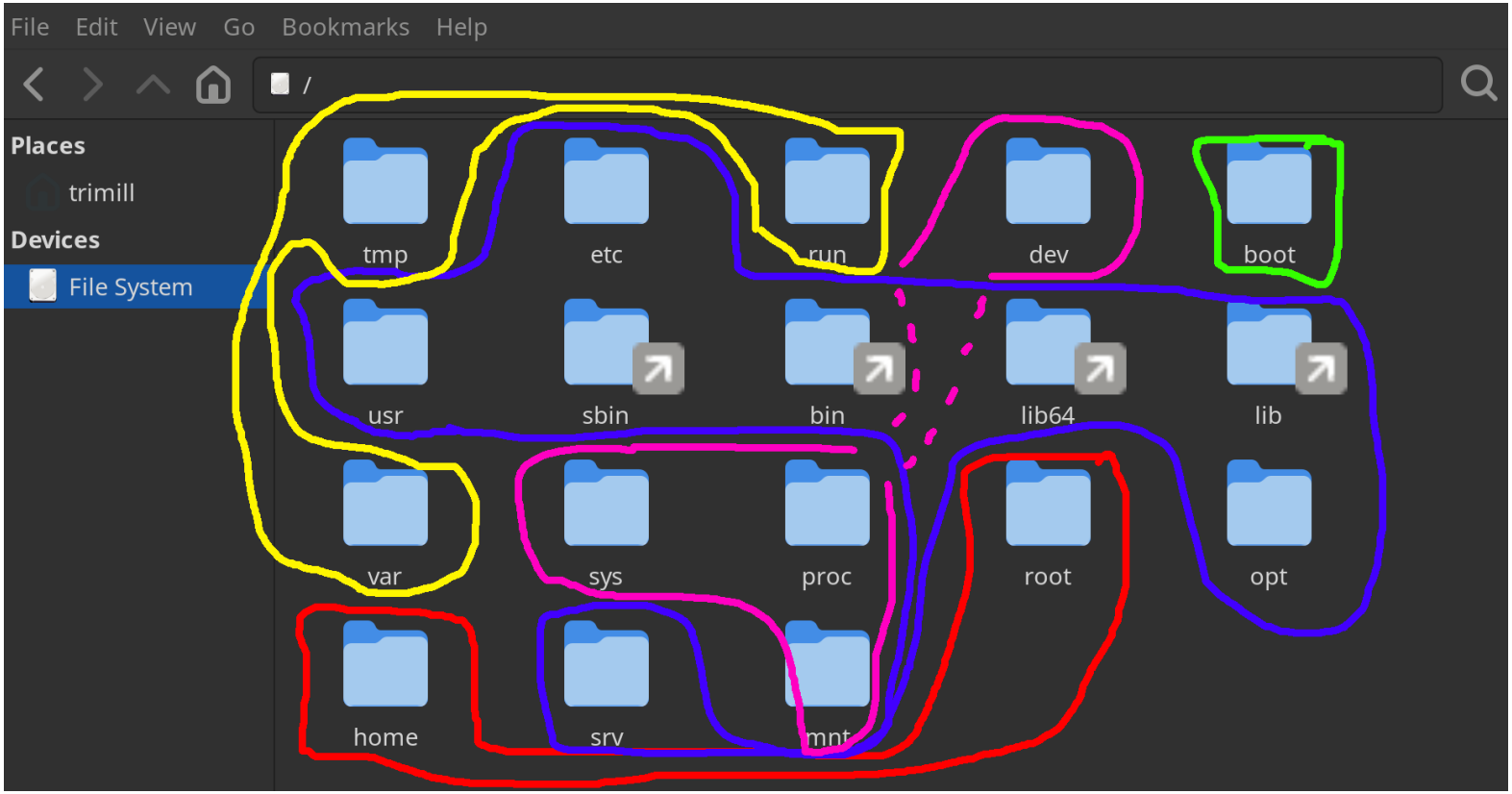
- "Everything is a file"
- All files exist in a single hierarchy, starting at / (root directory)
- Files can be stored on disk, in memory, or not correspond to storage at all

1. What's going on in /?

What's going on in /?



What's going on in /?



Home directories

- `/home/[username]` and `/root`
- Store files specific to one user – documents, photos, downloads, configuration, data
  - more on this later
- The root user's home is `/root`

Boot directory

- `/boot`
- Files used by the bootloader – configuration, kernel, and initramfs
- `/boot/efi` – mount point for the EFI partition on systems using GPT

Programs, libraries, data, and configuration

- `/bin`, `/sbin`, `/usr/bin`, `/usr/sbin`
  - Executable programs
  - Most distros symlink all of these together
- `/lib`, `/usr/lib`
  - Shared libraries, same logic as
  - `lib32`, `lib64` variants

Directory	Essential for boot?	Root user only?
`/usr/bin`	No	No
`/usr/sbin`	No	Yes
`/bin`	Yes	No
`/sbin`	Yes	Yes
`/usr/lib`	No	
`/lib`	Yes	

Programs, libraries, data, and configuration

- `/usr/include` – C header files
- `/usr/share` – data files for installed programs
- `/usr/local` – for programs built by the user
  - `/usr/local/bin`, `/usr/local/lib`, `/usr/local/share`, etc.
- `/opt` – programs from packages outside your package manager
- `/etc` – system-wide configuration
- `/srv` – files served by servers (HTTP, FTP, etc)

Temporary and runtime data

- `/tmp` - generic temporary files, not saved to disk
- `/run` - temporary runtime data, not saved to disk
- `/var` - runtime data, saved to disk
  - `/var/cache` – cache files
  - `/var/lib` – program state (databases, etc)
  - `/var/lock` – lockfiles
  - `/var/log` – program logs
  - `/var/mail` – user mailboxes
  - `/var/run` – usually symlinked to `/run`
  - `/var/spool` – print queues
  - `/var/tmp` – temporary files, saved to disk

Mount points

- `/media` – mount points created automatically
- `/mnt` – mount points created manually

Device/kernel interfaces

- `/dev`, `/proc`, `sys`
- We'll talk more about these later

2. What's going on in ~?

XDG User Directories

Environment variables defining common user directories – many programs respect these.

Environment variable	Default directory
XDG_DESKTOP_DIR	~/Desktop
XDG_DOCUMENTS_DIR	~/Documents
XDG_DOWNLOADS_DIR	~/Downloads
XDG_MUSIC_DIR	~/Music
XDG_PICTURES_DIR	~/Pictures
XDG_PUBLICSHARE_DIR	~/Public
XDG_TEMPALTES_DIR	~/Templates
XDG_VIDEOS_DIR	~/Videos

XDG Base Directory

Environment variable	Default directory	Description
`XDG_CONFIG_HOME`	`~/.config`	Configuration
`XDG_CACHE_HOME`	`~/.cache`	Cache
`XDG_DATA_HOME`	`~/.local/share`	Program data
`XDG_STATE_HOME`	`~/.local/state`	Program state
`XDG_RUNTIME_DIR`	none*	Runtime data

*not set by default, systemd used `/run/user/$UID`

Summary

	System package	Installed by admin	Installed by user
Executables	`/usr/bin` †	`/usr/local/bin` ‡	`~/.local/bin` *
Libraries	`/usr/lib` §	`/usr/local/lib`	`~/.local/lib` *
Data	`/usr/share`	`/usr/local/share`	`~/.local/share`

	System-wide	User-specific
Configuration	`/etc`	`~/.config`
Cache	`/var/cache`	`~/.cache`
State	`/var/lib`	`~/.local/state`
Runtime data	`/run`	none ¶

† or `/usr/sbin`, `/bin`, `/sbin`

‡ or `/usr/local/sbin`

* commonly used but nonstandard

§ or `/lib`, other directories (ex. `/lib64`) may also exist

¶ not set by default, systemd uses `/run/user/$UID`

3. /dev, /proc, and /sys

/dev

- Device files – hard drives, SSDs, CDs, flash drives, terminals, audio and video devices
- **Block device**: stores data, supports seeking
- **Character device**: like a stream or pipe, cannot seek
- Special devices:
  - `/dev/null`: reads immediately reach EOF, writes are ignored
  - `/dev/zero`: reads always yield `0x00`, writes are ignored
  - `/dev/full`: like `/dev/zero` but writes return ENOSPC (no space left on device)
  - `/dev/urandom`: reads yield random bytes generated by the kernel
  - `/dev/random`: similar, but blocks when high-quality randomness runs out

/dev block devices

- `sdX`: SCSI/SATA devices
  - `X` is a lowercase letter starting at `a`
  - `sda`, `sdb`, etc
- `hdX`: IDE devices
- `fdX`: floppy disks
- `vdX`: virtio devices
- `srX`: optical media
- `nvmeXnY`: NVME devices
  - `X` is the controller number (starts at 0)
  - `Y` is the namespace number (starts at 1??!?)
  - `nvme0n1`, `nvme1n3`, etc
- `mmcblkX`: MMC/eMMC devices (`X` is a number)
- `loopX`: loop device (`X` is a number)

/dev block devices

- Partitions: append the partition number (starting at 1)
  - Interpose a `p` if drive ends in a digit
  - Examples: `sda1`, `hdb3`, `nvme0n1p4`, `loop0p1`
- `disk`: view block devices in hierarchical structure
- `mapper`: Linux device mapper, often used with disk encryption

/dev character devices

- `tty#`: Linux virtual consoles (Ctrl+Alt+F#)
- `ttyS#`: serial ports
- `ttyUSB#`: serial over USB
- `pts/#`: pseudo-TTYs (used by terminal emulators)
- `pts/#`: pseudo-TTYs (used by terminal emulators)
- `stdin`, `stdout`, `stderr`: the current process's standard streams

/proc

- Information about the Linux kernel and processes
- `cmdline`: Linux kernel arguments
- `cpuinfo`: CPU information
- `kmsg`: kernel messages
- `meminfo`: memory usage and information
- `stat`: CPU usage statistics
- `uptime`: time since last boot
- `mounts`: mounted filesystems
- `version`: system version information

/proc process directories

- /proc/N/ is a directory containing information about process number N.
- /proc/self/ is a symlink to the current process's directory
- exe: a symlink to the process's executable file
- cmdline, environ: the command-line arguments and environment variables
- cwd: the current working directory
- fd/N: the Nth open file descriptor
- status: various information about the process

/sys

- Yet again, files representing information about the kernel and devices
- `/sys/devices`: devices categorized based on how they're connected to the system
- `/sys/class`: devices categorized by what type they are (eg. input devices, LEDs, backlight, networking)
- `/sys/module`: loaded kernel modules