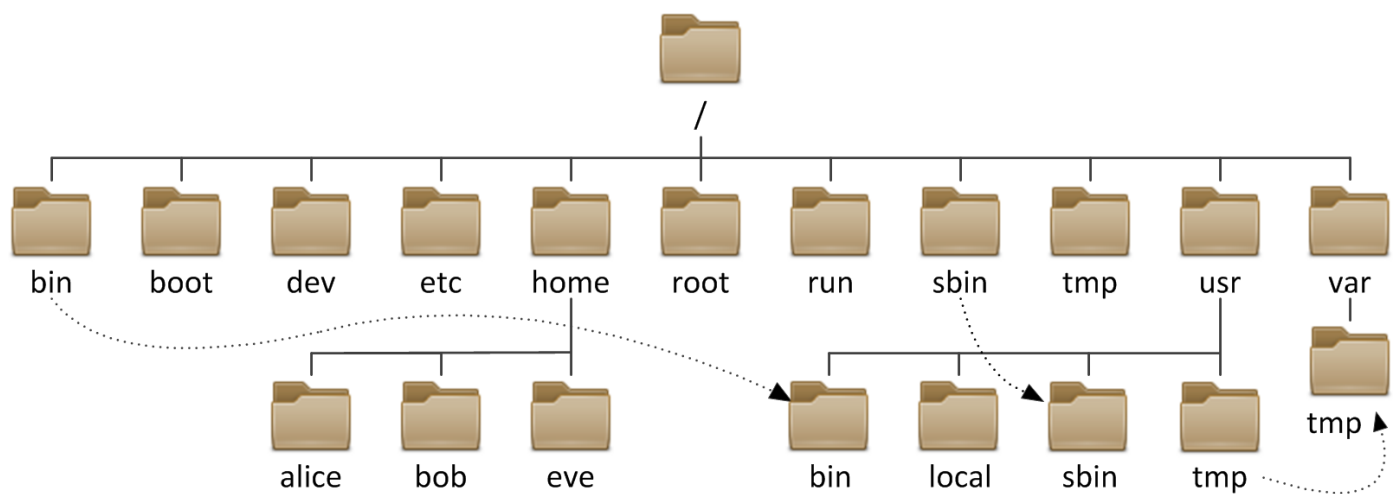


File system



System Directories

"A typical Linux system has tens of thousands of system directories. These directories contain operating system files, applications, documentation, and just about everything except personal user files (which typically live in /home).

Unless you're a system administrator, you'll rarely visit most system directories—but with a little knowledge you can understand or guess their purposes. Their names often contain three parts, as shown"

Categories for programs

1. **/bin (bin)**: Contains essential command binaries that are required for the system to run in single-user mode.
2. **/sbin (system bin)**: Contains binaries that are essential for system administration, maintenance, and repair. These binaries often require superuser privileges for execution.

In summary, while both `/bin` and `/sbin` contain executable binaries essential for the functioning of the system, `/bin` is for general-purpose commands accessible to all users, and `/sbin` is for system administration commands that require elevated privileges and are typically used by system administrators or the root user. The separation helps maintain a clear distinction between commands for regular users and those for system maintenance and administration.

"Programs (usually binary files) intended to be run by the superuser"

`/lib` "Libraries of code used by programs"

This directory contains essential system libraries needed for the basic operation of the system. These libraries are crucial for the system's core functionality and may include dynamic-link libraries (shared libraries) used by the kernel and other critical system components.

Categories for documentation

doc

a "doc" folder, short for "documentation," is a common location where documentation or manuals for installed software packages are stored. This folder is often found within the directory structure of software installations, particularly within the `"/usr/share/doc"` directory

- **Location:** The "doc" folder is commonly found within the `"/usr/share/doc"` directory.
- **Format:** Documentation in the "doc" folder can be in various formats, including plain text files, README files, HTML pages, examples, and manuals.
- **Content:** The "doc" folder typically contains detailed documentation related to the software package it corresponds to. This documentation provides information on installation, configuration, and usage of the software.

info

In summary, "man" pages are quick-reference manuals displayed in the terminal, "info" pages provide more detailed and structured documentation with a hypertext interface, and the "doc" folder contains additional documentation files in various formats related to specific software packages. Each serves a different purpose and caters to different user preferences and needs

- **Location:** The "info" pages are typically stored in the `"/usr/share/info"` directory.
- **Format:** Info pages use the Info format, providing a hypertext documentation system. They are viewed in the terminal using the `info` command.
- **Content:** Info pages offer more detailed and structured documentation with navigation capabilities. They cover a wide range of topics, including commands, utilities, and other aspects of the system.

Man (Manual Pages):

- **Location:** The "man" pages are stored in directories commonly known as "man directories," such as `"/usr/share/man"` and `"/usr/local/share/man."`
- **Format:** Manual pages, or "man pages," are typically plain text documents displayed in a terminal using the `man` command.
 - **Content:** Man pages provide concise and quick-reference information about commands, utilities, system calls, and library functions. They are organized into sections based on the type of information they contain (e.g., user commands, system calls, library functions).

Categories for configuration

etc

“Configuration files for the system (and other miscellaneous stuff)”

The `/etc` directory plays a crucial role in maintaining system configuration consistency. It houses settings that affect various aspects of the system, including user accounts, network configurations, services, and more.

init.d

“Configuration files for booting Linux”

The `/etc/init.d` directory in Linux is a directory commonly used for storing scripts that are used to control the initialization and termination of system services. The name “init.d” is a historical reference to the “init” system, which was traditionally responsible for initializing the system during the boot process

The `/etc/init.d` directory typically contains scripts written in Bash or another scripting language. These scripts are used to start, stop, restart, and manage system services.

Categories for hardware

dev

Device files for interfacing with disks and other hardware

The `/dev` directory in Linux is a crucial part of the filesystem hierarchy, and it stands for “devices.” This directory contains special files that represent and provide access to various devices in the system. In Linux, devices are treated as files, and interacting with devices often involves reading from or writing to these special files in the `/dev` directory.

- **Block Devices:** Represented as files like `/dev/sda` for hard drives or `/dev/sdb1` for partitions.
- **Character Devices:** Represented as files like `/dev/tty` for terminals or `/dev/random` for random number generators.
- **Other Devices:** Devices such as printers, sound cards, and more can also have corresponding device nodes in the `/dev` directory.
- For example, reading from `/dev/sda` might retrieve data from a hard drive, and writing to `/dev/ttyS0` might send data to a serial port.

media

“Mount points: directories that provide access to disks”

The `/media` and `/mnt` directories in Linux both serve as mount points for filesystems and devices, but they are typically used for slightly different purposes and have different conventions.

1. `/media` Directory:

- **Automatic Mounting:** `/media` is often used for the automatic mounting of removable media, such as USB drives, external hard drives, and optical discs. Many Linux desktop environments automatically handle the mounting and unmounting of these devices under `/media` when they are connected or inserted.
- **User-Friendly Access:** The `/media` directory provides a user-friendly way for users to access the contents of connected removable media. Graphical file managers often display these mounts in a user interface for easy navigation.
- **Dynamic Creation:** The mount points under `/media` are often dynamically created by the system, and the naming convention might include the device name or label.

2. `/mnt` Directory:

- **Manual Mounting:** `/mnt` is generally used as a manual and user-defined mount point for temporarily mounting filesystems or devices. Users or administrators can create subdirectories under `/mnt` to mount specific filesystems or devices when needed.
- **System Independence:** Unlike `/media`, which is often automatically managed by the system, `/mnt` is more generic and user-controlled. Users have greater flexibility in defining mount points based on their needs.
- **Temporary Mounting:** Mount points under `/mnt` are often used for temporary mounting, meaning that they may not persist across reboots. For permanent mounts, other directories like `/etc/fstab` or `/media` might be more appropriate.

In summary, while both `/media` and `/mnt` can be used as mount points for filesystems and devices, `/media` is often associated with automatic and user-friendly mounting of removable media, while `/mnt` is more commonly used for manual and user-defined mounting of filesystems or devices for temporary purposes. The specific usage and conventions may vary between Linux distributions and individual system configurations.

Categories for runtime files

- `/var` is used for storing data that varies over time and is not essential for system functionality during the initial boot process.
- Examples of data stored in `/var` include log files (`/var/log`), mail spool directories (`/var/spool/mail`), printer spool directories (`/var/spool/lpd`), and more.

1. `/var/lock`:

- **Purpose:** This directory is used for storing lock files. Lock files are small files that applications create to indicate that they have exclusive access to a particular resource or file. This prevents multiple instances of the same application from interfering with each other.
- **Example:** `/var/lock/subsys` might contain lock files indicating which subsystems are currently running.

2. `/var/log`:

- **Purpose:** The `/var/log` directory contains log files generated by the system and various applications. Log files are crucial for system administrators to diagnose issues, monitor system health, and track the activities of different services.
- **Example:** `/var/log/syslog` might contain system-wide messages, while `/var/log/apache2/access.log` might contain access logs for the Apache web server.

3. `/var/mail`:

- **Purpose:** This directory is traditionally used for storing user mailboxes. Each user may have a file in this directory containing their received mail. In modern setups, mail delivery may use other mechanisms, and the use of this directory may vary.
- **Example:** `/var/mail/username` might be the mailbox for a user named "username."

4. `/var/run`:

- **Purpose:** The `/var/run` directory contains system information and runtime data, particularly information that should persist between reboots. It often includes process-related information such as process IDs (PIDs) and other runtime data.
- **Example:** `/var/run/utmp` might contain information about currently logged-in users.

5. `/var/spool`:

- **Purpose:** The `/var/spool` directory is used for storing data that is queued for processing. It often includes subdirectories for different spooling purposes, such as mail spooling (`/var/spool/mail`), print spooling (`/var/spool/lpd`), etc.
- **Example:** `/var/spool/cron` might contain queued cron jobs.

6. `/var/tmp`:

- **Purpose:** Similar to the `/tmp` directory, `/var/tmp` is used for storing temporary files. The difference is that files in `/var/tmp` are expected to persist across reboots.
- **Example:** `/var/tmp/example_temp_file` might be a temporary file created by an application.

7. `/proc`:

- **Purpose:** The `/proc` directory is a virtual filesystem that provides information about processes and the kernel. It doesn't contain real files but rather exposes information as files that can be read or written. It's a crucial interface for interacting with the kernel and obtaining information about running processes.

- **Example:** `/proc/cpuinfo` provides information about the CPU, and `/proc/meminfo` provides information about memory usage.

Each of these directories plays a specific role in the Linux filesystem hierarchy, contributing to the organization and functioning of the operating system.

`/boot`

The `/boot` directory in Linux is a critical part of the filesystem where essential files for the boot process are stored. It contains files necessary for the initial stages of the boot process, including the kernel, bootloader configuration, and other related files.

- The Linux kernel images, which are the core of the operating system, are stored in the `/boot` directory. These files are usually named `vmlinuz` or `vmlinuz-{version}`.
- Example: `/boot/vmlinuz-5.4.0-77-generic`
- `/lost+found`
- The `/lost+found` directory in Linux is a special directory created at the root level of a filesystem. Its purpose is to store files and directories that are recovered during a filesystem check (fsck) after a system crash or unexpected shutdown.
- System administrators may manually inspect the contents of `/lost+found` after a filesystem check to determine whether any important files or directories need to be restored
- `/proc/ioprocs` A list of your computer's input/output hardware.
- `/proc/cpuinfo` Information about your computer's processors.
- `/proc/version` The operating system version. The `uname` command prints the same "
- `/proc/uptime` System uptime (i.e., seconds elapsed since the system was last booted). Run the `uptime` command for a more human-readable result."