Linux Filesystem Hierarchy

Overview

Linux organizes all files and directories under a single root directory `/`, following the Filesystem Hierarchy Standard (FHS). This standard defines the structure and purpose of directories in Unix-like systems.

Root Directory `/`

The root directory is the starting point of the filesystem tree. Every file and directory branches out from here.

Common Directories and Their Roles

`/bin` — Essential System Binaries

- Stores critical command-line programs needed for system boot and repair.
- Accessible to all users.
- Examples: `ls`, `cp`, `mv`, `cat`, `grep`.
- Functions even if other filesystems are not mounted.

`/dev` — Device Files

- Contains files representing hardware devices and pseudo-devices.
- Devices are accessed as files in Linux.
- Examples:
  - `/dev/sda1`: First partition of the first SATA drive.
  - `/dev/null`: Discards all written data.
  - `/dev/random`: Supplies random data.

`/etc` — System Configuration

- Holds system-wide configuration files and scripts.
- Only administrators should modify these files.
- Organized by service/application.
- Key files:
  - `/etc/passwd`: User accounts.
  - `/etc/fstab`: Filesystem mounts.
  - `/etc/hosts`: Hostname-to-IP mappings.

`/etc/os-release` — OS Information

- Contains details about the Linux distribution and version.
- View with: `cat /etc/os-release`
- Example:

```
NAME="Ubuntu"
VERSION="20.04.3 LTS (Focal Fossa)"
```

```
ID=ubuntu
ID_LIKE=debian
```

`/home` — User Home Directories

- Personal directories for each user.
- Structure: `/home/[username]`
- Examples: `/home/user1`, `/home/alice`
- Stores user files, settings, and configurations.

`/lib` and `/lib64` — Shared Libraries

- `/lib`: Essential libraries for binaries in `/bin` and `/sbin`.
- `/lib64`: 64-bit libraries for 64-bit systems.
- Often symbolic links to `/usr/lib` and `/usr/lib64`.
- Libraries are shared code modules for multiple programs.

`/usr` — User System Resources

- Contains most user utilities and applications.
- Subdirectories:
  - `/usr/bin`: Non-essential user commands.
  - `/usr/lib`: Libraries for `/usr/bin` and `/usr/sbin`.
  - `/usr/local`: Locally installed software.
  - `/usr/share`: Architecture-independent data.

`/var` — Variable Data

- Stores files that change frequently, like logs and caches.

`/var/log` — Logs

- Central location for system and application logs.
- Important for monitoring and troubleshooting.
- Examples:
  - `/var/log/syslog`: System messages.
  - `/var/log/auth.log`: Authentication logs.
  - `/var/log/kern.log`: Kernel messages.
  - `/var/log/apache2/`: Web server logs.

Filesystem Navigation Tips

- `ls -la /`: List all root directories.
- `df -h`: Show mounted filesystems and usage.
- `tree /`: Display directory tree (if installed).
- Use absolute paths (starting with `/`) for system directories.

Best Practices

1. Only modify system directories if you understand the impact.
2. Regularly back up important directories like `/etc` and `/home`.
3. Monitor disk usage in `/var` to prevent log files from filling up storage.
4. Set proper permissions when working with system directories.

Using Filesystem Paths

Paths can be:

1. **Absolute**: Start from root, e.g., `/bin/bash`
2. **Relative**: Start from current directory, e.g., `folder1/file2.txt`

File Permissions Format

Understanding File Permissions Format

When you run `ls -l`, you'll see lines like:

```
d rwx --- ---
  -+- -+- -+-
   |   |   |
   |   |   +-- Permissions for others
   |   +------ Permissions for group
   +---------- Permissions for owner
```

- `d`: Item type (`d` for directory)
- `rwx`: `r` (read), `w` (write), `x` (execute/explore)