Chapter 3: Linux Directory Structure & Filesystem

1. Linux Directory Structure (Filesystem Hierarchy)

Definition (Interview-Oriented)

Linux follows a **hierarchical directory structure** where everything starts from a single root directory `/`.

Unlike Windows (C:, D:), Linux has **one unified filesystem tree**.

Why Linux Uses This Structure

- Logical organization of system files
- Separation of configuration, binaries, logs, and user data
- Better security and manageability
- Predictable locations (important for automation and DevOps)

Interview insight:

In Linux, everything is treated as a file, including devices and processes.

2. Important Linux Directories (VERY IMPORTANT)

`1. /` (Root)

- Top-level directory
- All other directories branch from here
- System cannot boot without it

---

2. /bin

- Essential user binaries (commands)
- Required for basic system operation

Examples:

```
Ls, cp, mv, cat
```

---

3. /sbin

- System administration binaries
- Typically used by root user

Examples:

```
Iptables, reboot, fsck
```

Interview note:

/bin is for users, /sbin is for system administration.

---

4. /etc

- System-wide configuration files
- No binaries, only configs

Examples:

- `/etc/passwd`
- `/etc/shadow`
- `/etc/ssh/sshd_config`

Interview question:

Where are configuration files stored in Linux?

Answer:

Mostly in /etc.

---

5. `/var`

- Variable data that changes frequently

Contains:

- Logs
- Spool files
- Cache

Important subdirectories:

- `/var/log` – system and application logs
- `/var/spool` – mail, cron jobs

Real-life example:

When disk becomes full, `/var/log` is often the cause.

---

6. `/usr`

- User system resources
- Installed software and libraries

Contains:

- `/usr/bin` – user commands
- `/usr/lib` – libraries
- `/usr/local` – manually installed software

Interview insight:

/usr is read-only in many production systems.

---

7. `/tmp`

- Temporary files
- Cleared on reboot (usually)

Interview caution:

Never store important data in /tmp.

---

8. `/home`

- User home directories

Example:

`/home/ravi`

Contains:

- User files
- User-specific configuration

---

9. `/dev`

- Device files
- Represents hardware as files

Examples:

- `/dev/sda`
- `/dev/null`
- `/dev/random`

Interview line:

In Linux, devices are accessed as files under /dev.

---

10. `/proc` (Very Important)

Definition

`/proc` is a **virtual filesystem** that provides real-time system and process information.

It does not exist on disk.

Why /proc Exists

- Exposes kernel data to users
- Used for monitoring and debugging

Examples:

```
cat /proc/cpuinfo
cat /proc/meminfo
```

Interview insight:

/proc is dynamically generated by the kernel.

---

3. Filesystem Types

ext4 vs xfs

Feature	ext4	xfs
Stability	Very high	High
Performance	Balanced	Better for large files
Resizing	Offline	Online
Use case	General purpose	Large-scale systems

Interview answer:

ext4 is common, xfs is preferred for large filesystems.

---

4. What Is an Inode?

Definition

An inode is a data structure that stores metadata about a file.

What an Inode Contains

- File size
- Owner
- Permissions
- Timestamps
- Pointer to data blocks

What an Inode Does NOT Contain

- File name

Interview trap question:

Does inode store filename?

Correct answer:

No, filename is stored in the directory.

---

5. Hard Link vs Soft Link

Hard Link

- Points to the same inode
- Cannot cross filesystems
- File exists as long as at least one hard link exists

Soft Link (Symbolic Link)

- Points to file path
- Can cross filesystems
- Breaks if target file is deleted

Comparison Table

Feature	Hard Link	Soft Link
Inode	Same	Different
Cross filesystem	No	Yes
Breaks on delete	No	Yes

---

6. Mount and Unmount

Definition

Mounting attaches a filesystem to a directory.

Example

```
mount /dev/sdb1 /data
umount /data
```

Interview explanation:

Mounting makes storage accessible through the directory tree.

---

7. Block Devices vs Character Devices

Block Devices

- Transfer data in blocks
- Support buffering

Examples:

- Hard disks
- SSDs

Character Devices

- Transfer data character by character
- No buffering

Examples:

- Keyboard
- Mouse

Interview line:

Disks are block devices; input devices are character devices.

---

8. What Happens When Disk Is 100% Full?

Real-Life Impact

- Services fail
- Logs cannot be written
- System may become unstable

Debugging Steps

1. Check disk usage

```
df -h
```

2. Find large directories

```
du -sh /*
```

3. Identify log files
4. Clean or archive data
5. Rotate logs

Interview insight:

Disk full issues are common production problems.

---

9. Finding Large Files

Commands

```
du -ah / | sort -rh | head -20
```

Use case:

- Disk cleanup
- Performance troubleshooting

---

10. RAID (Basics)

Definition

RAID combines multiple disks for performance or redundancy.

Common RAID Levels

- RAID 0 – Performance, no redundancy
- RAID 1 – Mirroring
- RAID 5 – Parity-based redundancy
- RAID 10 – Performance + redundancy

Interview expectation:

Know use cases, not deep math.

---

11. LVM (Logical Volume Manager)

Definition

LVM allows flexible disk management by abstracting physical storage.

Why LVM Is Used

- Resize disks without downtime
- Combine multiple disks
- Easier storage management

LVM Components

- Physical Volume (PV)
- Volume Group (VG)
- Logical Volume (LV)

Interview line:

LVM provides flexibility compared to traditional partitioning.

Chapter 3: Interview Takeaways

After this chapter, you should confidently explain:

- Linux directory hierarchy
- Purpose of key directories
- Inodes and links
- Filesystem types
- Disk full troubleshooting
- Mounting, RAID, and LVM basics