

1. Explain the different types of files in Linux (regular, directory, symbolic link, device, etc.) and how to check them with commands.

Linux has 7 main file types:

- i. Regular File (-)
 - Contains data: text, binary, scripts, executables.
 - Example: file.txt, a.out.
- ii. Directory (d)
 - Special file that stores lists of filenames and their inodes.
 - Think of it as a "map" to files.
 - Example: /home, /etc.
- iii. Symbolic Link (l)
 - Shortcut (pointer) to another file/directory.
 - Can point across partitions.
 - Example: myfile -> /usr/local/bin/somefile.
- iv. Character Device (c)
 1. Provides unbuffered (character-by-character) access to hardware devices.
 2. Example: /dev/tty, /dev/ttyUSB0.
- v. Block Device (b)
 - Provides buffered access (blocks of data) to hardware devices.
 - Example: /dev/sda (hard drive), /dev/mmcblk0.
- vi. FIFO / Named Pipe (p)
 - Used for inter-process communication (IPC).
 - Acts like a queue: one process writes, another reads.
 - Example: mkfifo mypipe.
- vii. Socket (s)
 - Used for network communication or IPC.
 - Example: /tmp/mysql.sock.

- To check File Types:
 - a. `ls -l` (long listing): First character indicates file type: - → regular file, d → directory, l → symbolic link, c → character device, b → block device, p → FIFO/pipe, s → socket
 - b. `file <filename>`
 - Detects the content type (text, binary, script, etc.).
 - c. `stat <filename>`
 - Shows detailed metadata (inode, permissions, timestamps, etc.)

2. What's the difference between a hard link and a symbolic link? Give real examples of when to use each.

Hard Link

1. A direct pointer to the same inode (the actual data on disk).
2. Multiple filenames point to the same physical file.
3. Deleting one name does not delete the data as long as another link exists.
4. Cannot span across different filesystems or partitions.
5. Cannot link to directories (to avoid loops).

Example:

```
echo "hello" > file1.txt
```

```
ln file1.txt file2.txt
```

```
ls -li
```

Use Case:

- Backup within the same partition (saves space since no duplicate data).
- Keeping multiple names for the same file (like versioning).

Symbolic Link (Soft Link)

1. A special file that contains a path to another file.
2. Works like a shortcut (Windows style).
3. If the original file is deleted, the symlink becomes a broken link (dangling).
4. Can span across partitions.
5. Can point to directories too.

Example:

`ln -s file1.txt link_to_file1`

`ls -l`

Use Case:

- Creating shortcuts (like `/bin/sh -> /usr/bin/bash`).
- Linking libraries or configs (e.g., `/etc/nginx/sites-enabled/` uses symlinks).
- Sharing a single file across multiple locations.

3. Is `rmdir` the same as `rm -r` when deleting directories? Explain.

`rmdir`

- Stands for remove directory.
- Can only delete empty directories.

- If the directory has files/subdirectories, it will fail.

rm -r

- rm = remove.
- -r (recursive) means:
 1. Delete the directory
 2. Delete all files and subdirectories inside it, recursively.
- Much more powerful, this command recursively deletes a directory along with all its contents (files and subdirectories). It is more powerful and also more dangerous because it removes everything without requiring the directory to be empty.