

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

Everything in Linux is a file and they do break into:

- Regular files (-):  
These contain data, text, or program binaries.  
Example: myfile.txt, /bin/ls.
  - Directory files (d):  
As folders in Windows, they are files that store lists of other files  
Example: /home/user, /etc.
  - Symbolic link files (l):  
As shortcuts in Windows, they do point to another file or directory.  
Example: ln -s /etc/passwd passwd\_link.
  - Device files:  
Represent hardware devices. Found in /dev.
    - Character device (c)  
devices that transfer data character by character (e.g., keyboard, /dev/tty).
    - Block device (b)  
devices that transfer data in blocks (e.g., hard disks, /dev/sda).
  - Socket files (s):  
It's like a window or a door between the devices in the same network, allows them to inter-process communication (IPC). Example: /var/run/docker.sock.
  - Named pipes (FIFO, p):  
It makes the output of some process be the input of the other or vice versa so it allows one process to pass data to another. Example: mkfifo mypipe.
- Commands to check file type:
    - ls -l → shows file type in the first character (-, d, l, c, b, s, p).
    - file filename → prints a description of the file's type.
    - stat filename → detailed info about the file.

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

- Hard Link
  - Points directly to the file's data on disk and it has the same inode number and number of links as in the original file.
  - Deleting the original file doesn't affect the hard link
  - Cannot span across different file systems or partitions.

Example usage: Backup scenarios, when you want multiple filenames for the same data (e.g., `ln file1.txt file2.txt`).

- Symbolic Link (Soft Link)
  - Points to another file by name (like a shortcut).
  - Different inode number and number of links don't change.
  - Can point across file systems and directories.
  - If the target is deleted, the symlink becomes broken, so it's still there but can't be opened.

Example usage: The file descriptors opened by a process like `/proc/<PID>/fd/` are symbolic links, useful for creating shortcuts, version switching (e.g., `/usr/bin/python` → `/usr/bin/python3.12`).

Command examples:

- Hard link: `ln file1.txt file2.txt`
- Symbolic link: `ln -s file1.txt link1.txt`

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

- `rmdir`:
  - Only removes empty directories.
  - Example: `rmdir myfolder` → works only if `myfolder` has no files/subfolders.
- `rm -r` (recursive remove):
  - Deletes a directory and everything inside it (files, subdirectories).
  - Example: `rm -r myfolder` → removes `myfolder` and all its contents.