Session - 10

Inode Number, Soft Link, and Hard Link in Linux – Explained

🧩 1. What is an Inode Number?

An **inode** (Index Node) is a data structure used by the Linux file system to **store information about a file**, except its name.

📌 Inode stores:

- File type (regular, directory, etc.)
- File size
- Ownership (user/group)
- Permissions (rwx)
- Time stamps (created, modified, accessed)
- Pointers to data blocks on disk

Each file has a **unique inode number** within its file system.

Example:

bash

ls -i myfile.txt

Output:

123456 myfile.txt

123456 is the inode number.

🔗 2. Hard Link

A **hard link** is another name for the same file — it **points to the same inode**.

✅ Features:

- Shares the **same inode number** as the original file.
- Changes to content affect all links.
- Removing one **does not delete** the file as long as one link exists.
- Cannot link to directories (for safety).
- Cannot link across file systems.

Example:

bash

ln file1.txt file2.txt

ls -li file1.txt file2.txt

Output:

css

123456 -rw-r--r-- 2 user file1.txt

123456 -rw-r--r-- 2 user file2.txt

Both point to **inode 123456** and share data.

🔗 3. Soft Link (Symbolic Link)

A **soft link** is like a shortcut — it **points to a file name**, not the inode.

✅ Features:

- Has a **different inode**.
- If original file is deleted, link becomes **broken** ("dangling link").

- Can link to **directories**.
- Can **span across file systems**.

Example:

bash

ln -s original.txt link.txt

ls -li original.txt link.txt

Output:

sql

123456 -rw-r--r-- 1 user original.txt

654321 lrwxrwxrwx 1 user link.txt -> original.txt

🧠 link.txt is a soft link pointing to the name original.txt.

Summary Table

Feature	**Hard Link**	**Soft Link (Symbolic)**
Points to	Inode (file content)	File name (path)
Inode number	Same as original	Different
File system	Same only	Can span file systems
Directory link	❌ Not allowed	✅ Allowed
Broken if source deleted	❌ No	✅ Yes