Day 1 - Linux Learning Summary

1. Installing WSL on Windows

- **WSL (Windows Subsystem for Linux)** allows running a Linux environment on Windows without a virtual machine.
- Installed WSL and set up the Linux distribution.

2. Basic Linux Commands

a) Navigation & Directory Operations

- `pwd` → Prints the current working directory.
- `ls` → Lists files and directories in the current location.
- `mkdir <dir_name>` → Creates a new directory.
- `mkdir -p <dir1/dir2/dir3>` → Creates nested directories (parent-child structure).
- `cd <path>` → Changes directory.
- `cd ..` → Moves one level up in the directory hierarchy.

b) Exploring File System

- `cd /mnt/c/` → Navigates to the C drive in Windows from Linux.
- `ls -lrt` → Lists files in **long format (-l), sorted by modification time (-t), in reverse order (-r)**.

3. Hands-on with File & Directory Operations

- Created a new directory: `mkdir LinuxPractise`
- Navigated into it: `cd LinuxPractise/`
- Created a new file using `vi`: `vi a.txt` (Opened in Vim editor)
- Created a deep directory structure:
- `mkdir -p a/b/c/d/e/f/g/h/i/j/k/l/m/temp.txt`
- Used `touch` to create files:
- `touch c406.txt`
- `touch {1..5}.txt  # Creates files: 1.txt, 2.txt, 3.txt, 4.txt, 5.txt`
- `touch {a..z}.txt  # Creates files: a.txt, b.txt, ..., z.txt`
- Deleted all files in the directory: `rm -rf *`
- Copied files and directories:
- `cp -rf b.txt /mnt/c/Users/srs33/  # Copies b.txt to a Windows directory`
- `cp -rf a /mnt/c/Users/srs33/      # Copies folder "a" to Windows directory`
- `rm -rf /mnt/c/Users/srs33/a       # Deletes the copied folder`

4. Understanding File Permissions in Linux

In Linux, every file and directory has permissions associated with three categories:

1. **Owner** – The user who owns the file.
2. **Group** – A group of users who share the same permissions.
3. **Others** – All other users.

Each category has three types of permissions:

- **Read (4)** → Allows viewing the file's content.
- **Write (2)** → Allows modifying or deleting the file.
- **Execute (1)** → Allows running the file as a program or script.

Permission Representation in Binary & Numeric Form

Read	Write	Execute	Permission Description
1	1	1	Read, Write, Execute (Full permissions)
1	1	0	Read and Write only
1	0	1	Read and Execute only
0	1	1	Write and Execute only
0	0	1	Execute only
0	1	0	Write only
1	0	0	Read only
0	0	0	No permission given

Example: Changing File Permissions

- `chmod 755 filename` → Sets permissions to:
  - **Owner:** Read, Write, Execute (7 → 4+2+1)
  - **Group:** Read, Execute (5 → 4+0+1)
  - **Others:** Read, Execute (5 → 4+0+1)
- `ls -l filename` → Displays the file's permissions in `rwx` format.

5. Using `grep` Command

- `grep` is used for searching text patterns in files.
- Example:
- `man grep  # Opens manual for grep command`
  - **Common Grep Usage**:
    - `grep "pattern" filename` → Searches for "pattern" inside `filename`.
    - `grep -i "pattern" filename` → Case-insensitive search.
    - `grep -r "pattern" /path/` → Recursively searches in directories.