

## Day 2

### Introduction to Basic Linux Commands:

When working with Linux, mastering basic commands is crucial for efficient system navigation and management. Below are some commonly used Linux commands for file and directory operations:

#### 1. **ls Command:**

- **Description:** Lists files and directories in the current location.
- **Syntax:** `ls`
- **Example:** `ls -l` (to display detailed information such as permissions and file sizes)

#### 2. **pwd Command:**

- **Description:** Prints the current working directory path.
- **Syntax:** `pwd`
- **Example:** `pwd`

#### 3. **cd Command:**

- **Description:** Changes the current directory.
- **Syntax:** `cd [directory]`
- **Example:** `cd Documents` (to navigate to the Documents directory)

#### 4. **mkdir Command:**

- **Description:** Creates a new directory.
- **Syntax:** `mkdir [directory_name]`
- **Example:** `mkdir NewFolder` (to create a directory named NewFolder)

#### 5. **rmdir Command:**

- **Description:** Removes an empty directory.
- **Syntax:** `rmdir [directory_name]`
- **Example:** `rmdir OldFolder` (to delete a directory named OldFolder)

#### 6. **cp Command:**

- **Description:** Copies files and directories.
- **Syntax:** `cp [source] [destination]`
- **Example:** `cp file1.txt /backup` (to copy file1.txt to the backup directory)

#### 7. **mv Command:**

- **Description:** Moves or renames files and directories.
- **Syntax:** `mv [source] [destination]`

- **Example:** mv file1.txt newfile.txt (to rename file1.txt to newfile.txt)

#### 8. **rm Command:**

- **Description:** Deletes files.
- **Syntax:** rm [file]
- **Example:** rm file1.txt (to remove file1.txt)

#### 9. **cat Command:**

- **Description:** Displays the content of a file.
- **Syntax:** cat [file]
- **Example:** cat file1.txt (to show the contents of file1.txt)

#### 10. **grep Command:**

- **Description:** Searches for specific text in files.
- **Syntax:** grep [pattern] [file]
- **Example:** grep "keyword" file.txt (to find occurrences of "keyword" in file.txt)

These essential commands form the foundation for navigating, managing files, and performing basic operations in a Linux environment. By becoming proficient in these commands, users can streamline their interactions with the Linux command line interface.

### **Introduction to Permission Table in Linux:**

In Linux systems, file permissions are fundamental for regulating access to files and directories. The permission table consists of a matrix that specifies the level of access granted to the owner of the file, the group associated with the file, and other users. Here is an in-depth look at the permission table:

#### 1. **File Permissions Overview:**

- **Read (r):** Enables users to view the contents of a file or list the files within a directory.
- **Write (w):** Allows users to modify the file contents, delete the file, or create new files within a directory.
- **Execute (x):** Permits the execution of a file if it is a program or access to the contents of a directory.

#### 2. **User Categories:**

- **Owner (u):** Refers to the user who owns the file and has the authority to modify permissions.
- **Group (g):** Represents users who are part of the group associated with the file.
- **Others (o):** Denotes all users who are neither the owner nor part of the file's group.

#### 3. **Format of the Permission Table:**

- The permission table is typically displayed as a 10-character string, such as -rwxrwxrwx.
- The first character indicates the file type (e.g., - for a regular file, d for a directory).
- The subsequent characters are divided into three sets, each representing the permissions for the owner, group, and others.

#### 4. Assigning and Modifying Permissions:

- Permissions can be assigned or altered using the chmod command, which accepts both numeric and symbolic representations.
- **Numeric Representation:** Each permission (read, write, execute) is assigned a numeric value (4, 2, 1), and these values are added to determine the permission code (e.g., 755).
- **Symbolic Representation:** Involves using symbols like + to add permissions, - to remove permissions, and = to set specific permissions for the user categories.

Understanding the intricacies of the permission table empowers users to effectively manage access control and security within a Linux environment. By configuring permissions appropriately, users can safeguard sensitive data and ensure that the right individuals have the necessary access to files and directories.