Linux Commands

Basic Commands

1. **cd: Change Directory**
    - ***Description:*** **Changes the current working directory.**
    - ***Example:*** **cd Documents (moves to the "Documents" directory).**
2. **ls: List Directory Contents**
    - ***Description:*** **Lists the files and directories in the current location.**
    - ***Example:*** **ls (displays a list of files and directories).**
3. **pwd: Print Working Directory**
    - ***Description:*** **Shows the full path of the current working directory.**
    - ***Example:*** **pwd (displays the current working directory, e.g., "/home/user").**
4. **mkdir: Make Directory**
    - ***Description:*** **Creates a new directory.**
    - ***Example:*** **mkdir NewFolder (creates a directory named "NewFolder").**
5. **cp: Copy Files or Directories**
    - ***Description:*** **Copies files or directories from one location to another.**
    - ***Example:*** **cp file.txt /backup (copies "file.txt" to the "/backup" directory).**
6. **mv: Move or Rename Files/Directories**
    - ***Description:*** **Moves or renames files or directories.**
    - ***Example:*** **mv oldfile.txt newfile.txt (renames "oldfile.txt" to "newfile.txt").**
7. **rm: Remove Files or Directories**
    - ***Description:*** **Deletes files or directories.**
    - ***Example:*** **rm file.txt (deletes the "file.txt" file).**
8. **touch: Create an Empty File**
    - ***Description:*** **Creates a new empty file.**
    - ***Example:*** **touch newfile.txt (creates an empty file named "newfile.txt").**
9. **cat: Display or Concatenate Files**
    - ***Description:*** **Displays the content of a file or concatenates multiple files.**
    - ***Example:*** **cat file.txt (displays the content of "file.txt").**

Tasks or Assignment's on Basic Commands:

1. **Navigate and List:**
    - **Use cd to navigate to your home directory.**
    - **Use ls to list the contents of the directory.**
2. **Create and Navigate:**
    - **Use mkdir to create a new directory named "MyProjects."**
    - **Use cd to navigate into the "MyProjects" directory.**
3. **File Operations:**
    - **Use touch to create a file named "README.md" inside "MyProjects."**
    - **Use cp to make a copy of "README.md" and name it "COPY_README.md."**
4. **Move and Rename:**
    - **Use mv to rename "COPY_README.md" to "NEW_README.md."**
    - **Use mv to move "NEW_README.md" to the parent directory.**
5. **Edit a File:**
    - **Use nano or vi to open "README.md" and add a line of text.**
    - **Save and exit the text editor.**
6. **Remove and Confirm:**
    - **Use rm to delete "NEW_README.md."**
    - **Use ls to confirm that "NEW_README.md" is no longer in the directory.**

Level - 2 Assignment's For basic Commands

1. **Nested Directories:**
    - **Inside "MyProjects," create two directories: "ProjectA" and "ProjectB" using mkdir.**
    - **Navigate into "ProjectA" using cd.**
2. **File Operations in Nested Directories:**
    - **Use touch to create files named "file1.txt" and "file2.txt" inside "ProjectA."**
    - **Use ls to confirm the presence of these files.**
3. **Copying Across Directories:**
    - **Use cp to copy "file1.txt" from "ProjectA" to "ProjectB."**
    - **Check if the file is successfully copied.**
4. **Remove with Confirmation:**
    - **Navigate to "ProjectA" and use rm to remove "file2.txt" with confirmation prompts.**
    - **Ensure the file is deleted.**
5. **Recursive Removal:**
    - **Navigate to "MyProjects" and use rm -r ProjectA to delete the entire "ProjectA" directory and its contents.**
    - **Confirm the deletion.**
6. **Viewing File Content:**
    - **Use cat to display the contents of "file1.txt" in the "ProjectB" directory..**

File Permission Commands

1. **chmod (Change Mode):**
    a. **Usage: chmod [options] mode file**
    b. **Description: Changes the file mode (permissions).**
2. **chown (Change Owner):**
    a. **Usage: chown [options] owner:group file**
    b. **Description: Changes the owner and/or group of a file.**
3. **chgrp (Change Group):**
    a. **Usage: chgrp [options] group file**
    b. **Description: Changes the group ownership of a file.**
4. **ls (List):**
    a. **Usage: ls -l**
    b. **Description: Lists files with detailed information, including permissions.**
5. **umask (User Mask):**
    a. **Usage: umask [options]**
    b. **Description: Sets the default permissions for new files.**
6. **su (Switch User):**
    a. **Usage: su [username]**
    b. **Description: Switches to a different user account.**
7. **sudo (Superuser Do):**
    a. **Usage: sudo command**
    b. **Description: Executes a command with elevated privileges.**
8. **passwd (Password):**
    a. **Usage: passwd [username]**
    b. **Description: Changes a user's password.**
9. **getfacl (Get File ACL):**
    a. **Usage: getfacl file**
    b. **Description: Displays the Access Control List (ACL) of a file.**

10. **setfacl (Set File ACL):**
    a. **Usage: setfacl [options] file**
    b. **Description: Sets the Access Control List (ACL) of a file.**
11. **sudoers (sudo Configuration):**
    a. **Usage: visudo**
    b. **Description: Edits the sudoers file, controlling sudo permissions.**
12. **find (Search for Files):**
    a. **Usage: find /path/to/search -type f -exec chmod permissions {} \;**
    b. **Description: Searches for files and applies permissions.**
13. **grep (Search Text):**
    a. **Usage: grep -r "pattern" /path/to/search**
    b. **Description: Searches for a specific text pattern recursively in files.**
14. **ps (Process Status):**
    a. **Usage: ps aux**
    b. **Description: Lists running processes, useful for monitoring.**

Assignment's For the File Permissions

1. **Explore Permissions:**
    a. **Use ls -l to list files in your home directory.**
    b. **Use chmod to change the permissions of a file to allow read, write, and execute for the owner.**
2. **Create and Inspect ACL:**
    a. **Create a new file using touch named "data.txt."**
    b. **Use setfacl to grant read access to a specific user for "data.txt."**
    c. **Use getfacl to inspect the ACL of "data.txt."**
3. **Switch User and Modify File:**
    a. **Use su to switch to another user.**
    b. **Navigate to your home directory and edit a file using nano.**
    c. **Attempt to save; observe the permissions error.**
4. **Recursive Permission Change:**
    a. **Create a directory using mkdir named "projects."**
    b. **Inside "projects," create subdirectories.**
    c. **Use chmod -R to grant read and write permissions recursively to all files in "projects."**
5. **Process Monitoring:**
    a. **Run a sample process using sleep in the background.**
    b. **Use ps to monitor the running processes.**
    c. **Try using kill to stop the process.**

6. **Explore Permissions:**
    a. **Use ls -l to list files in your home directory.**
    b. **Use chmod to change the permissions of a file to allow read, write, and execute for the owner.**
7. **Create and Inspect ACL:**
    a. **Create a new file using touch named "data.txt."**
    b. **Use setfacl to grant read access to a specific user for "data.txt."**

c. Use getfacl to inspect the ACL of "data.txt."

8. Switch User and Modify File:
    a. Use su to switch to another user.
    b. Navigate to your home directory and edit a file using nano.
    c. Attempt to save; observe the permissions error.
9. Recursive Permission Change:
    a. Create a directory using mkdir named "projects."
    b. Inside "projects," create subdirectories.
    c. Use chmod -R to grant read and write permissions recursively to all files in "projects."
10. Process Monitoring:
    a. Run a sample process using sleep in the background.
    b. Use ps to monitor the running processes.
    c. Try using kill to stop the process.

11. Explore Permissions:
    a. Use ls -l to list files in your home directory.
    b. Use chmod to change the permissions of a file to allow read, write, and execute for the owner.
12. Create and Inspect ACL:
    a. Create a new file using touch named "data.txt."
    b. Use setfacl to grant read access to a specific user for "data.txt."
    c. Use getfacl to inspect the ACL of "data.txt."
13. Switch User and Modify File:
    a. Use su to switch to another user.
    b. Navigate to your home directory and edit a file using nano.
    c. Attempt to save; observe the permissions error.
14. Recursive Permission Change:
    a. Create a directory using mkdir named "projects."
    b. Inside "projects," create subdirectories.
    c. Use chmod -R to grant read and write permissions recursively to all files in "projects."
15. Process Monitoring:
    a. Run a sample process using sleep in the background.
    b. Use ps to monitor the running processes.
    c. Try using kill to stop the process.

Pipe and Grep Commands

Pipe (|) Command:

- **Usage: command1 | command2**
- **Description: Passes the output of command1 as the input to command2, allowing the combination of multiple commands.**

Grep Command:

- Usage: grep [options] pattern [file]
- Description: Searches for a specified pattern in a file or input stream and prints matching lines.

Common Grep Options:

1. -i (Ignore Case): Ignores case distinctions.
2. -r (Recursive): Searches subdirectories recursively.
3. -n (Line Numbers): Displays line numbers along with matching lines.
4. -v (Invert Match): Prints lines that do not match the specified pattern.
5. -w (Whole Word): Matches whole words, not substrings.
6. -c (Count): Prints only the count of matching lines.
7. -A (After Context): Prints lines after the matching line.
8. -B (Before Context): Prints lines before the matching line.
9. -E (Extended Regex): Enables extended regular expressions.

1. **Basic Pipe Practice:**
    - **Create a file named "sample.txt" with some text.**
    - **Use cat to display the content of "sample.txt" and grep to filter lines containing a specific word.**
2. **Recursive Grep Challenge:**
    - **Create a directory structure with nested text files.**
    - **Use grep -r to search for a pattern recursively in all text files within the directories.**
3. **Line Number Mystery:**
    - **Generate a text file with multiple lines.**
    - **Use grep with the -n option to find a specific word and display line numbers.**
4. **Inverted Match Quest:**
    - **Create a file with various lines of text.**
    - **Use grep -v to display lines that do not contain a specific word.**
5. **Whole Word Hunt:**
    - **Write a paragraph with repeated words and substrings.**
    - **Use grep -w to find occurrences of a specific whole word.**
6. **Counting Matches Task:**
    - **Generate a file with multiple lines of text.**
    - **Use grep -c to count the occurrences of a specific word in the file.**
7. **Contextual Display Challenge:**
    - **Write a passage with repeating words.**
    - **Use grep -A and grep -B to display lines after and before a specific word.**
8. **Extended Regex Exploration:**
    - **Create a file with various patterns (e.g., emails, phone numbers).**
    - **Use grep -E with a regular expression to match specific patterns**

User and Group Management Commands

1. **User Accounts:**
    - **Creating user accounts (useradd command).**
    - **Modifying user account properties (usermod command).**
    - **Removing user accounts (userdel command).**
    - **Understanding the /etc/passwd file.**
2. **Group Management:**
    - **Creating groups (groupadd command).**
    - **Adding users to groups (usermod or gpasswd commands).**
    - **Removing users from groups (gpasswd command).**
    - **Understanding the /etc/group file.**
3. **User and Group Permissions:**
    - **Understanding file permissions (chmod).**
    - **Associating users and groups with file permissions.**
    - **Viewing and interpreting file permissions.**
4. **Special Users and Groups:**
    - **Learning about special user accounts like root.**
    - **Understanding privileged access and the sudo command.**
5. **Password Management:**
    - **Changing user passwords (passwd command).**
    - **Managing password policies (chage command).**
6. **File Ownership:**
    - **Changing file ownership (chown command).**
    - **Understanding the importance of file ownership.**
7. **User Environment:**
    - **Configuring user profiles and environment variables.**
    - **Understanding shell configuration files (.bashrc, .bash_profile).**
8. **Authentication and Authorization:**
    - **Understanding PAM (Pluggable Authentication Modules).**
    - **Configuring and securing login settings**

Tasks on user management

1. **User Account Creation and Modification:**
    a. **Task: Create a script that automates the process of adding multiple user accounts with customized properties such as home directory and shell.**
2. **Group Collaboration:**
    a. **Task: Simulate a collaborative project scenario. Create user accounts for team members, assign them to different groups, and set up a shared directory with appropriate permissions for group collaboration.**
3. **Permission Puzzles:**
    a. **Task: Create a set of files with varying permissions. Challenge students to decipher the permissions, explain the implications, and propose changes to enhance security.**
4. **User Profiles and Environment:**
    a. **Task: Customize your own Linux environment. Change the shell prompt, set environment variables, and configure aliases. Share your personalized environment settings with the class and explain the choices made.**
5. **Security Audit:**
    a. **Task: Conduct a security audit on a Linux system. Identify potential vulnerabilities related to user accounts, group memberships, and file permissions. Propose and implement improvements for better security.**
6. **Role-Playing Root:**
    a. **Task: Simulate scenarios where students take on the role of the root user. Execute tasks that require elevated privileges using the sudo command. Discuss the importance of responsible use of superuser privileges.**
7. **Password Policy Analysis:**
    a. **Task: Investigate and compare password policies on different Linux distributions. Discuss the rationale behind password policies, and propose improvements for enhancing security.**
8. **Group Presentation - Linux User Management:**
    a. **Task: Divide the class into groups, and assign each group a specific aspect of Linux user management to research and present. Topics could include user authentication, password encryption, or the role of PAM.**

9. **Create Users:**
    a. **Create three user accounts named Student1, Student2, and Student3.**
10. **Create Groups:**
    a. **Establish a group called "Classmates."**
11. **Add Users to Groups:**
    a. **Add Student1 and Student2 to the "Classmates" group.**
12. **View Users and Groups:**
    a. **Display the list of all users and groups on the system.**
13. **Modify User Attributes:**
    a. **Change the password for Student3.**
14. **Remove Users from Groups:**
    a. **Remove Student2 from the "Classmates" group.**
15. **Delete Users:**
    a. **Delete the Student3 account.**
16. **Add Multiple Users to Groups:**
    a. **Create two new users named NewStudent1 and NewStudent2, then add them to the "Classmates" group.**
17. **Check Group Memberships:**
    a. **Verify the members of the "Classmates" group.**
18. **Advanced Task - Permissions:**
19. **Assign read-only permissions for the "Classmates" group to a specific file or directory.**

Zip and Un zip commands

Compression Commands:

1. **tar:**
    - **Description: Creates or manipulates archive files. Can be used with gzip or bzip2 to compress.**
2. **gzip:**
    - **Description: Compresses files using the gzip compression algorithm. Creates files with a ".gz" extension.**
3. **bzip2:**
    - **Description: Compresses files using the bzip2 compression algorithm. Creates files with a ".bz2" extension.**

Decompression Commands:

1. **tar:**
    - **Description: Extracts files from an archive. Can be used with gzip or bzip2 for decompression.**
2. **gzip:**
    - **Description: Decompresses files compressed with gzip. Removes the ".gz" extension.**
3. **bzip2:**
    - **Description: Decompresses files compressed with bzip2. Removes the ".bz2" extension.**