Guide to getting familiar with Linux commands, focusing on directory structure, navigation, file operations, editing files, piping/redirecting, and using aliases.

1. Directory Structure

Linux organizes files in a hierarchical directory structure, starting from the root directory (/). Key directories include:

/home: User home directories (e.g., /home/username)

/etc: System configuration files

/var: Variable data (logs, databases)

/tmp: Temporary files

/usr: User programs and libraries

/bin: Essential system binaries

**Commands to explore structure:**

pwd: Print working directory (shows current location, e.g., /home/username).

bash

pwd

ls: List directory contents.

ls -l: Detailed list (permissions, size, etc.).

ls -a: Show hidden files (starting with .).

bash

ls -la

tree: Display directory tree (if installed).

bash

tree -L 2 # Shows 2 levels deep

**2. Moving Between Directories**

Use the cd command to navigate:

cd path: Change to a specific directory.

Absolute path: cd /home/username/documents

Relative path: cd documents (from current directory)

**cd ..:** Move up one directory.

**cd ~:** Go to home directory.

**cd -:** Return to the previous directory.

**cd /:** Go to root directory.

**3. File Operations**

Manage files with these commands:

touch filename: Create an empty file.

touch notes.txt

cp source dest: Copy files or directories.

cp file.txt /tmp/: Copy file to /tmp.

cp -r dir1 dir2: Copy directory recursively.

mv source dest: Move or rename files/directories.

mv notes.txt notes\_bak.txt # Rename

mv notes\_bak.txt /tmp/ # Move to /tmp

rm file: Delete files.

rm -r dir: Delete directory recursively.

rm -f file: Force deletion without prompt.

bash

rm -rf temp\_dir # Remove directory and contents

mkdir dirname: Create a directory.

bash

mkdir projects

rmdir dirname: Remove empty directory.

**4. Editing Files**

Edit files using terminal-based editors like nano, vim, or vi. nano is beginner-friendly:

nano filename: Open file in nano.

Edit, then save with Ctrl+O, exit with Ctrl+X.

bash

nano notes.txt

vim filename: Open in vim.

Press i to insert, edit, then :wq to save and quit, or :q! to quit without saving.

cat filename: Display file contents.

bash

cat notes.txt

less filename: View file contents interactively (use q to quit).

echo "text" > file: Write text to a file (overwrites).

echo "Hello" > greeting.txt

echo "text" >> file: Append text to a file.

echo "World" >> greeting.txt

**5. Pipe and Redirect**

Pipes (|) and redirects (>, >>, <) control command output/input:

Pipe (|): Send output of one command to another.

ls -l | grep txt # List files, filter for those containing "txt"

Redirect output:

>: Write output to a file (overwrites).

ls > files.txt # Save ls output to files.txt

>>: Append output to a file.

echo "More text" >> files.txt

Redirect input (<): Use a file as input.

sort < files.txt # Sort contents of files.txt

Example combo:

cat notes.txt | sort > sorted\_notes.txt # Sort notes.txt and save to sorted\_notes.txt

**6. Using Alias**

Aliases are shortcuts for commands, defined in shell configuration files (e.g., ~/.bashrc or ~/.zshrc).

Create an alias:

alias ll='ls -l' # Now `ll` runs `ls -l`

Make aliases permanent:

Edit ~/.bashrc:

nano ~/.bashrc

Add alias:

bash

alias ll='ls -l'

Save and reload:

source ~/.bashrc

View all aliases:

alias

Remove alias (temporary, for current session):

bash

unalias ll