**Date:16/07/2024**

**Day\_1\_Linux\_Assignments**

Top 20 Basic Commands with Use Cases and Exercises

While there are many commands across different operating systems and applications, here are 20 basic commands commonly found on computers:

1. dir (Windows) / ls (Linux/macOS): Lists the contents of a directory.

Use Case: You want to see all the files and folders in your current location.

Exercise: Open a terminal window (Command Prompt on Windows, Terminal on macOS/Linux) and type dir (Windows) or ls (Linux/macOS). Press Enter.

2. cd (all): Changes the current directory.

Use Case: You want to navigate to a different folder on your computer.

Exercise: Try cd Desktop (Windows/Linux/macOS) to navigate to your Desktop folder. Then use dir (Windows) or ls (Linux/macOS) to see the contents.

3. mkdir (all): Creates a new directory.

Use Case: You want to organize your files by creating a new folder.

Exercise: Use mkdir Documents (Windows/Linux/macOS) to create a new folder named "Documents". Then use dir (Windows) or ls (Linux/macOS) to see if it's there.

4. rm (Linux/macOS) / del (Windows): Deletes a file or directory (use with caution!).

Use Case: You want to remove an unwanted file or folder.

Exercise: Important: Never delete anything critical! In a safe space (like a temporary folder), create a text file named "test.txt" and then use rm test.txt (Linux/macOS) or del test.txt (Windows) to delete it.

5. copy (Windows) / cp (Linux/macOS): Copies a file.

Use Case: You want to duplicate a file to another location.

Exercise: Create another text file named "test2.txt". Use copy test.txt test2.txt (Windows) or cp test.txt test2.txt (Linux/macOS) to copy "test.txt" as "test2.txt".

6. move (Windows) / mv (Linux/macOS): Moves a file from one location to another.

Use Case: You want to organize your files by moving them to a different folder.

Exercise: Use move test2.txt Documents (Windows) or mv test2.txt Documents (Linux/macOS) to move "test2.txt" to the "Documents" folder (assuming it exists).

7. rename (Windows) / mv (Linux/macOS): Renames a file.

Use Case: You want to give a file a different name.

Exercise: Use rename test.txt newname.txt (Windows) or mv test.txt newname.txt (Linux/macOS) to rename "test.txt" to "newname.txt".

8. ping (all): Checks if another computer is reachable on a network.

Use Case: You want to see if you can connect to a website or another device.

Exercise: Use ping google.com (all) to see if you can reach Google's servers.

9. ipconfig (Windows) / ifconfig (Linux/macOS): Shows network configuration information.

Use Case: You want to troubleshoot network connectivity issues.

Exercise: Use ipconfig (Windows) or ifconfig (Linux/macOS) to see your IP address and other network details.

10. help (all): Provides help information for other commands.

Use Case: You're unsure about how to use a specific command.

Exercise: If you're stuck on command like mv, type help mv (all) to see a manual page with usage information.

11. clear (all): Clears the screen (text) in the terminal window.

Use Case: Your terminal window is cluttered with previous commands, and you want a clean slate.

Exercise: Type clear (all) to clear the screen.

12. date (all): Shows the current date and time.

Use Case: You need to know the current date and time.

Exercise: Type date (all) to see the current date and time.

[12:17 PM, 7/16/2024] +91 79820 39479: 14. time (all): (continued) You want to see how long a command takes to execute.

Exercise: Try time ls (all) to see how long it takes to list the directory contents.

15. mkdir -p (Linux/macOS): Creates a directory and any missing parent directories.

Use Case: You want to create a new folder within a nested structure that might not exist yet.

Exercise: Use mkdir -p Documents/Subfolder1/Subfolder2 (Linux/macOS) to create "Subfolder2" within "Subfolder1" inside the "Documents" folder (assuming "Documents" exists).

16. cat (Linux/macOS): Displays the contents of a text file.

Use Case: You want to read the contents of a text file without opening it in a separate program.

Exercise: Create a text file with some content and use cat filename.txt (Linux/macOS) to see its contents.

17. echo (all): Prints text to the terminal window.

Use Case: You want to display a message or variable in the terminal.

Exercise: Use echo Hello, world! (all) to print the message to the screen.

18. sudo (Linux/macOS): Grants temporary superuser privileges to execute a command (use with caution!).

Use Case: You need to perform an action that requires administrative rights.

Exercise: Important: Never use sudo for untrusted commands! In a safe scenario (like creating a test file), use sudo touch important.txt to create a file that might require admin access (assuming you have the password).

19. shutdown (Linux/macOS) / shutdown /s /t (Windows): Initiates a system shutdown or restart.

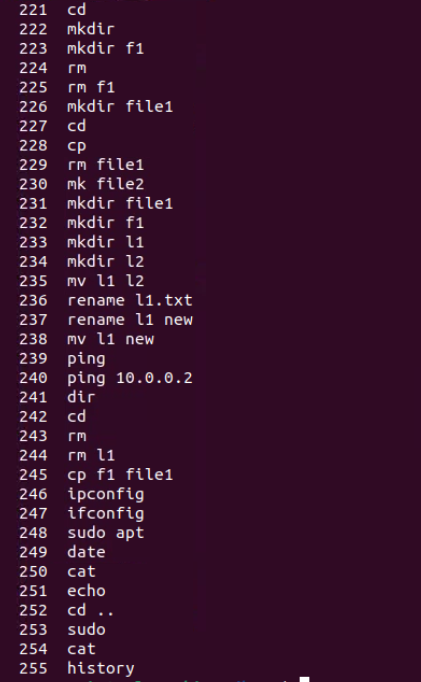
Use Case: You want to turn off or restart your computer.

Exercise: Important: Don't accidentally shut down your computer! This is for learning purposes only. Look up the specific options for your system to safely test a shutdown with a delay (e.g., shutdown /s /t 60 for Windows to shutdown in 60 seconds).

20. history (all): Shows a list of previously entered commands.

Use Case: You want to see what commands you've used recently, in case you need to refer back to one.

Exercise: Type history (all) to see a list of your recent commands.



1.Write a command using ls to list all files (including hidden files) in the current directory and its subdirectories.

Modify the previous command to display only files with a specific extension (e.g., .txt).

Enhance the report by including the file size for each listed file.

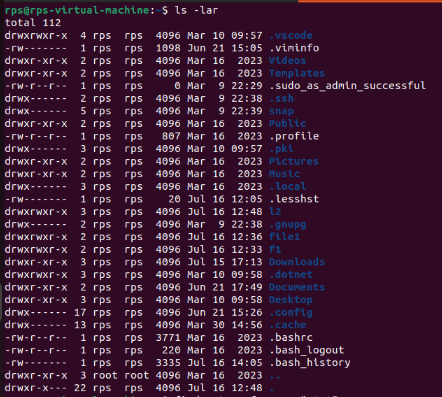
Further refine the output to display only files modified within the last 24 hours.

Combine the functionalities from points 2 and 4 to list only files with a specific extension (e.g., .jpg) modified in the last day

ls -laR

Explanation:

* ls: The command to list directory contents.
* -l: Use a long listing format.
* -a: Include hidden files (those starting with a dot .).
* -R: List subdirectories recursively.



2. find . -type f -name "\*.txt"

find .: Starts searching from the current directory (.).

-type f: Filters only regular files (ignores directories).

-name "\*.txt": Matches files with the extension .txt.

3. find . -type f -name "\*.txt" -exec ls -lh {} \;

-exec ls -lh {} \;: Executes ls -lh on each file found.

-lh: Options for ls to list files in long format with human-readable file sizes.

4. find . -type f -mtime -1

 -mtime -1: Filters files modified less than 24 hours ago.

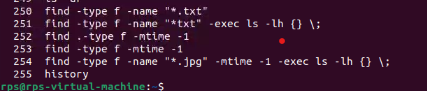
 Here, -1 specifies within the last 24 hours

5. find . -type f -name "\*.jpg" -mtime -1 -exec ls -lh {} \;

name "\*.jpg": Matches files with the extension .jpg.

-mtime -1: Filters files modified within the last 24 hours.

-exec ls -lh {} \;: Executes ls -lh on each matched .jpg file.



2. dir / ls (5): Use dir / ls to list all files and folders in your current directory. How many files are there? (Excluding hidden files if applicable) Utilize dir / ls with appropriate flags to display only files with a specific extension (e.g., .txt). How many files of that type exist? Navigate to your Downloads folder using cd. Then, use dir / ls to list the contents. Are there any recently downloaded files (modified today)? Use dir / ls with flags to display both the filename and its size for each file in your current directory. Identify the largest file. Practice using dir / ls with wildcards (e.g., dir \*.docx) to list all files with a specific extension pattern (e.g., all Word documents). cd (5):

ls -l | grep "^-" | wc -l

* **Explanation**:
  + ls -l: Lists detailed information about files and directories.
  + grep "^-": Filters lines starting with - (indicating regular files).
  + wc -l: Counts the number of lines (files).

ls \*.txt | wc -l

* **Explanation**:
  + ls \*.txt: Lists all files ending with .txt.
  + wc -l: Counts the number of lines (files).

cd Downloads

ls -lt

* **Explanation**:
  + cd Downloads: Changes directory to the Downloads folder.
  + ls -lt: Lists files sorted by modification time (-t flag) in long format (-l flag).

ls -l | sort -k 5 -rn | head -n 1

* **Explanation**:
  + ls -l: Lists detailed information about files.
  + sort -k 5 -rn: Sorts by the fifth column (file size) in reverse numeric order.
  + head -n 1: Shows the largest file.



3. Use cd to navigate to your Documents folder. What is the full path of your Documents folder displayed by the prompt?

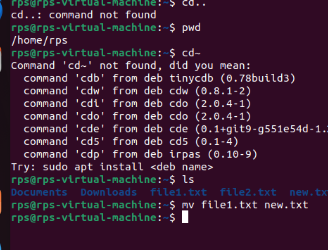
Practice using cd .. to move back one directory level from your current location.

Utilize pwd to display the full path of the current directory after navigating with cd.

Explore using directory shortcuts (e.g., ~ for home directory) with cd to quickly reach specific locations.

Combine cd with dir / ls to navigate to a specific folder and then list its contents.

cp / mv (5):



4. Identify a file on your Desktop. Use cp to copy that file to your Documents folder. Verify the copy exists in Documents.

Practice renaming a file on your Desktop using mv. Give it a new name and confirm the change using dir / ls.

Locate a folder containing images. Use cp to copy a specific image file from that folder to another folder.

Explore using mv to move a folder containing documents to a different location within your file system.

Try copying a file that already exists in the destination folder. What happens? (Experiment with different flags for cp if applicable on your system)

mkdir / rmdir (5

