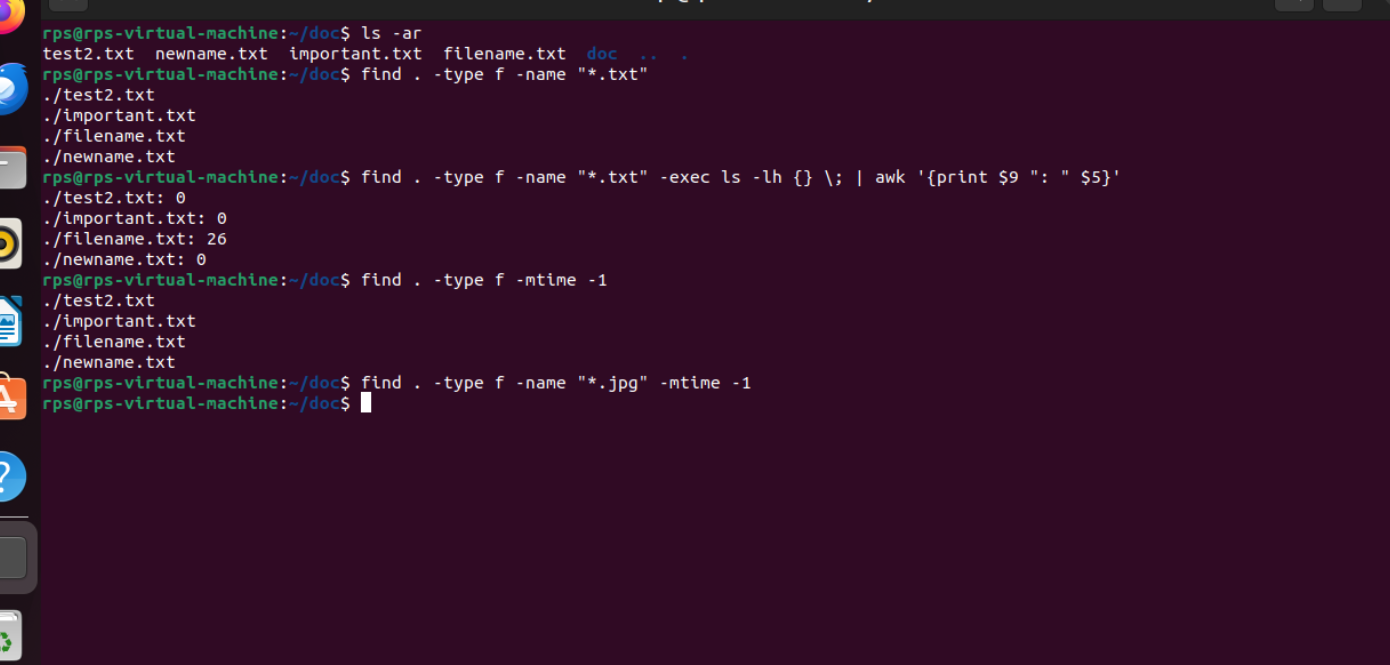
Write a command using ls to list all files (including hidden files) in the current directory and its subdirectories.

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.



Explanaton: 1)ls -aR: Lists all files and directories, including hidden ones, recursively.2)find . -type f -name "\*.txt": Finds all files (-type f) with a .txt extension starting from the current directory (.).3)find . -type f -name "\*.txt" -exec ls -lh {} \; | awk '{print $9 ": " $5}': Finds all .txt files and uses ls -lh to display file details, then awk extracts the file name and size.4)find . -type f -mtime -1: Finds all files modified in the last 24 hours (-mtime -1).5)find . -type f -name "\*.jpg" -mtime -1: Combines both conditions to find .jpg files modified in the last day.

**dir / ls (5):**

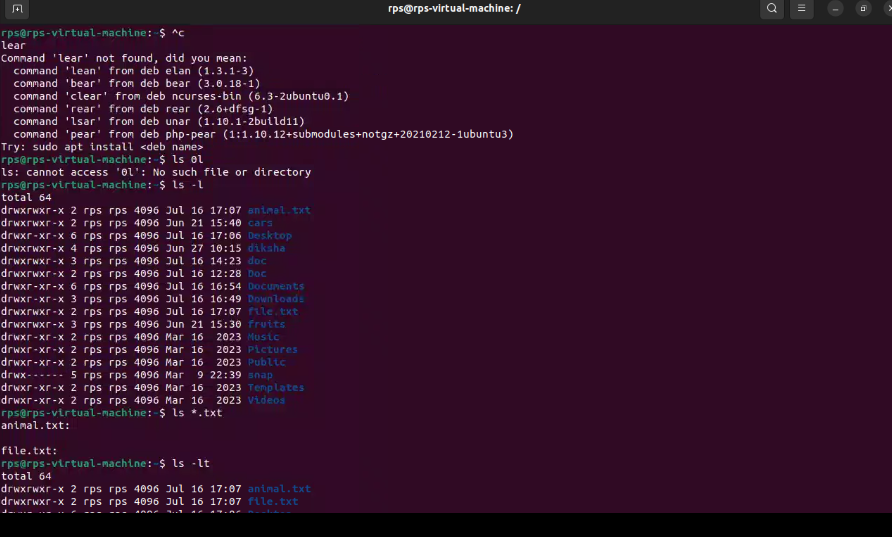
Use dir / ls to list all files and folders in your current directory. How many files are there? (Excluding hidden files if applicable)

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).



1. **List all files and folders in the current directory**

ls -al

`-a`: Shows all files, including hidden ones (files starting with `.`

-l: Displays detailed information including permissions, size, owner, modification date, etc. dir /a

/a: Shows all files, including hidden ones.

1. **Count the number of files (excluding hidden files)**

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

ls -l: Long listing format to include detailed information.

grep "^-"`: Filters out non-file entries (directories, symbolic links, etc.).

‘wc -l’: Counts the number of lines (which corresponds to the number of files). dir /a /-p /o:n /s | find /c /v ""

‘/a`: Shows all files.

‘/-p`: Skips displaying the header.

`/o:n`: Sorts the listing by name.

`/s`: Displays files in the current directory and all subdirectories.

`find /c /v ""`: Counts non-empty lines (each line corresponds to a file).

1. **Display files with a specific extension (e.g., .txt) and count them**

ls \*.txt

* + Lists all files with `.txt` extension. dir \*.txt
  + Lists all files with `.txt` extension.

1. **Check for recently modified files in the Downloads folder** Assuming you've navigated to the Downloads folder (`cd Downloads`): ls -lt | head -n 5

`-lt`: Sorts files by modification time, newest first (`-l` for long format, `-t` for sort by time).

`head -n 5`: Displays the top 5 files (most recent). dir /od /p

`/od`: Sorts files by date/time, oldest first.

`/p`: Pauses after each screen of information.

1. **Display filename and size for each file in the current directory and identify the largest file**

ls -lhS

`-h`: Makes file sizes human-readable (e.g., KB, MB).

-'S`: Sorts files by size, largest first. dir /o:-s

* `/o:-s`: Sorts files by size, largest first.

1. **Use wildcards to list files with a specific extension pattern**

ls \*.docx

Lists all files with `.docx` extension. dir \*.docx

* Lists all files with `.docx` extension.

**cd (5):**

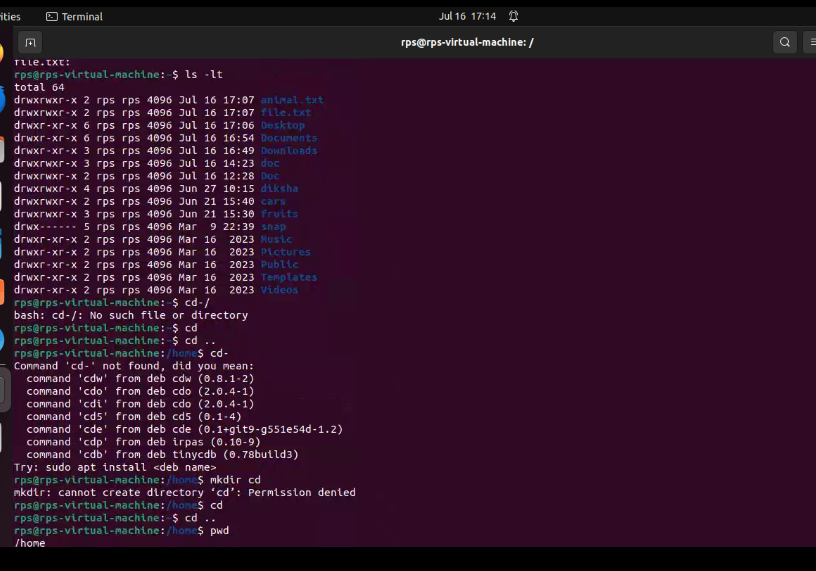
Use cd to navigate to your Documents folder. What is the full path of your Documents folder displayed by the prompt?

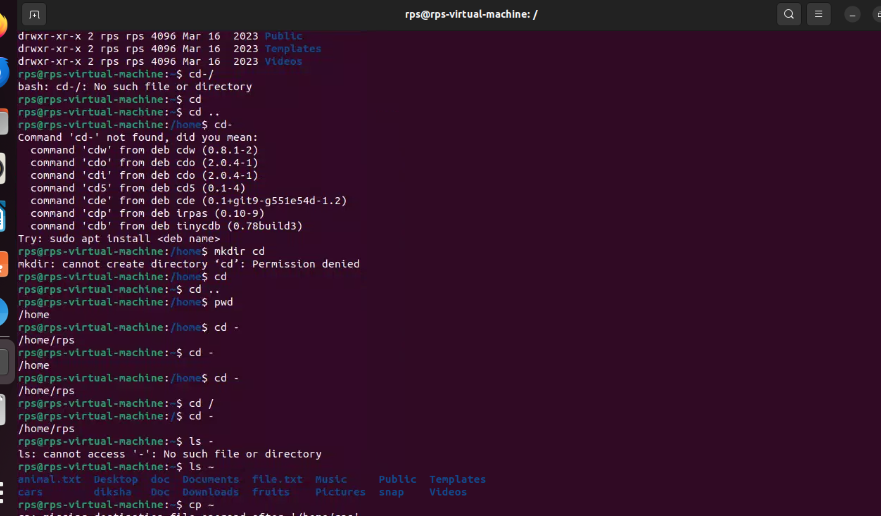
Practice using cd .. to move back one directory level from your current location.

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.



**1.Navigate to the Documents folder:**

Assuming your Documents folder is located within your home directory (`~/Documents`), you can use the following commands:

Cd This brings you to your home directory (~)

cd Documents Navigate into the Documents folder

After executing these commands, your prompt should display the full path to the Documents folder.

For instance, if your username is `user`, the prompt might display something like:

/home/user/Documents $

1. **Use `cd ..` to move back one directory level:**

If you are currently inside the Documents folder and want to move back to your home directory: cd .. Move back one directory level

This will take you back to your home directory. Your prompt will then display:

/home/user $

1. **Utilize `pwd` to display the current directory:**

`pwd` stands for "print working directory" and will display the full path of the current directory: pwd Print the current directory path

For example, if you are in your home directory, it will display:

/home/user

1. **Explore using directory shortcuts with `cd`:**
   * `cd ~` or `cd` (without any arguments) takes you to your home directory.
   * `cd /` takes you to the root directory.
   * `cd -` switches between the current directory and the previous directory you were in.
2. **Combine `cd` with `ls` to navigate to a folder and list its contents:**

Suppose you want to navigate to the Documents folder and then list its contents: cd ~/Documents Navigate to the Documents folder

ls List the contents of the Documents folder

**cp / mv (5):**

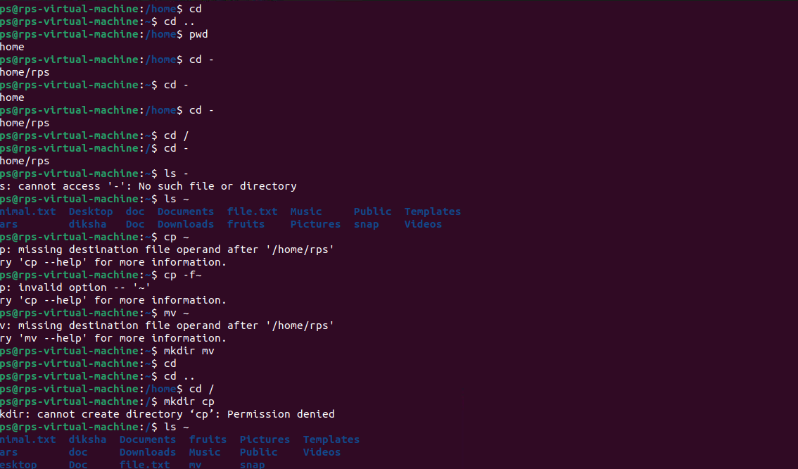
Identify a file on your Desktop. Use cp to copy that file to your Documents folder. Verify the copy exists in Documents.

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)



**Part 1: Using `cp` to copy a file**

1. Identify a file on your Desktop : Let's say the file we want to copy is `example.txt` located on the Desktop.
2. Copy the file to your Documents folder : Open a terminal and use the `cp` command:

cp ~/Desktop/example.txt ~/Documents/

This command copies `example.txt` from the Desktop to the Documents folder.

1. Verify the copy : To verify that the file exists in the Documents folder, you can list the contents of the Documents folder:

ls ~/Documents/

You should see `example.txt` listed among the files in your Documents folder.

**Part 2: Using `mv` to rename a file**

1. Rename a file on your Desktop : Let's rename `example.txt` to `newname.txt` using the `mv` command:

mv ~/Desktop/example.txt ~/Desktop/newname.txt

This command renames `example.txt` to `newname.txt` on the Desktop.

1. Confirm the change : To confirm the file has been renamed, list the files on the Desktop: ls ~/Desktop/

You should see `newname.txt` listed instead of `example.txt`.

**Part 3: Using `cp` to copy an image file**

1. Locate a folder containing images : Assume there's a folder `~/Pictures/` that contains image files.
2. Copy a specific image file : Let's copy `photo.jpg` from `~/Pictures/` to `~/Documents/`: cp ~/Pictures/photo.jpg ~/Documents/

This copies `photo.jpg` from the Pictures folder to the Documents folder.

**Part 4: Using `mv` to move a folder**

1. Move a folder containing documents: Assume there's a folder `~/Documents/Reports/` that contains documents.
2. Move the folder to a different location: Let's move `Reports` from `~/Documents/` to `~/Desktop/ mv ~/Documents/Reports ~/Desktop/

This moves the entire `Reports` folder from Documents to the Desktop.

**Part 5: Handling existing files with `cp'**

1. Copying a file that already exists in the destination folder **:** If `example.txt` already exists in

`~/Documents/` and you try to copy it again: cp ~/Desktop/example.txt ~/Documents/

By default, `cp` will not overwrite the existing `example.txt`. It will give an error message like `cp: overwrite '~/Documents/example.txt'?`.

1. Using flags for `cp`: To force overwrite existing files without prompting, you can use the `-f` (force) flag:

cp -f ~/Desktop/example.txt ~/Documents/

This will overwrite `example.txt` in `~/Documents/` without asking for confirmation.

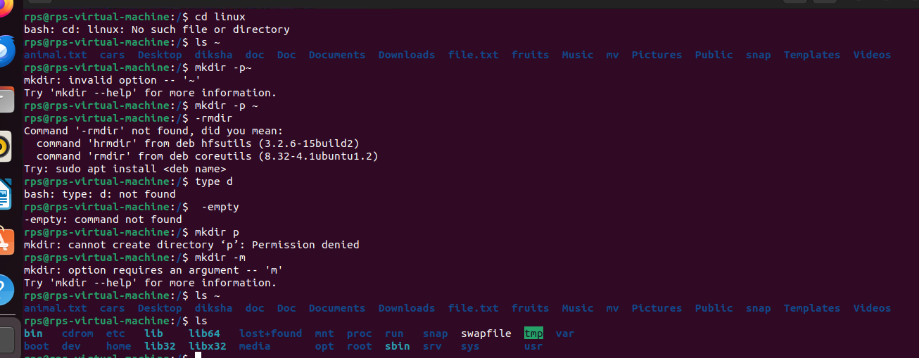
**mkdir / rmdir (5):**

Create a new folder named "Project Reports" inside your Documents folder using mkdir. Verify its existence using dir / ls.

Practice using mkdir with multiple arguments to create a nested folder structure (e.g., mkdir Documents/ProjectX/Reports).

Locate an empty folder you created earlier. Use rmdir to delete it. Confirm its removal with dir / ls.

Explore using dir / ls to identify empty folders within a specific directory



1. **Create a new folder named "Project Reports" inside your Documents folder using mkdir. Verify its existence using dir / ls.**

Assuming you're on a Unix-like system (Linux or macOS), here's how you would do it: mkdir ~/Documents/Project\ Reports

* `mkdir`: Command to create a directory.
* `~/Documents/Project\ Reports`: Path to the new directory, where `~` represents your home directory. To verify its existence, you can use `ls` (on Unix-like systems) or `dir` (on Windows):

ls ~/Documents

dir C:\Users\YourUsername\Documents

Look for the "Project Reports" directory in the output to confirm it exists.

1. **Practice using mkdir with multiple arguments to create a nested folder structure (e.g., mkdir Documents/ProjectX/Reports).**

To create nested folders:

mkdir -p ~/Documents/ProjectX/Reports

* + `-p` option: Allows mkdir to create parent directories as needed.

This command creates the `ProjectX` directory inside `Documents`, and then the `Reports` directory inside `ProjectX`.

1. **Locate an empty folder you created earlier. Use rmdir to delete it. Confirm its removal with dir / ls.**

Find the folder you want to delete. Let's assume it's `~/Documents/EmptyFolder`. rmdir ~/Documents/EmptyFolder

* + `rmdir`: Command to remove a directory (it only works if the directory is empty). After running `rmdir`, confirm its removal:

ls ~/Documents

dir C:\Users\YourUsername\Documents Ensure that `EmptyFolder` is no longer listed.

1. **Explore using dir / ls to identify empty folders within a specific directory.**

To find empty folders within a directory (`~/Documents` for example): find ~/Documents -type d -empty

* + `find`: Command to search for files and directories.
  + `~/Documents`: The directory you want to search within.
  + `-type d`: Only search for directories.
  + `-empty`: Only find directories that are empty.