Softwarica
College of IT & E-commerce

Linux

2024

Submitted to:

Rikesh Maharjan

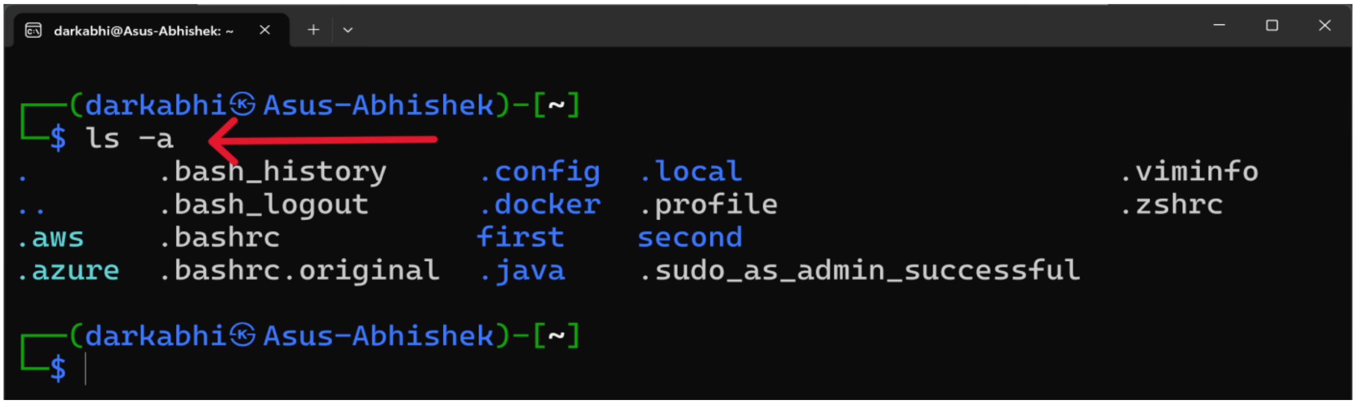
About

This project is completed as part of Task 1 for Linux Fundamentals, covering basic Linux commands and its architecture.

~ Abhishek Adhikari

Table of Contents

1. List all files and directories in your current directory, including hidden ones.

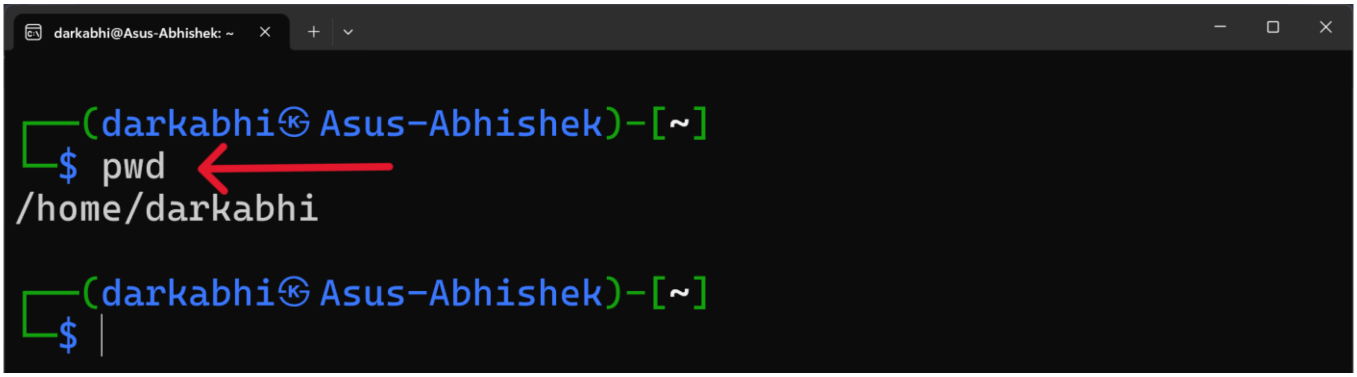


```
┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ ls -a
.        .bash_history     .config  .local                        .viminfo
..       .bash_logout      .docker  .profile                      .zshrc
.aws     .bashrc           first    second
.azure   .bashrc.original  .java    .sudo_as_admin_successful

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$
```

COMMAND: ls -a

The **ls -a** command lists all files and directories in the current directory, including hidden files (those starting with a dot remember not with double dot). The **-a** flag provides and includes hidden files.

2. Display the current working directory path.

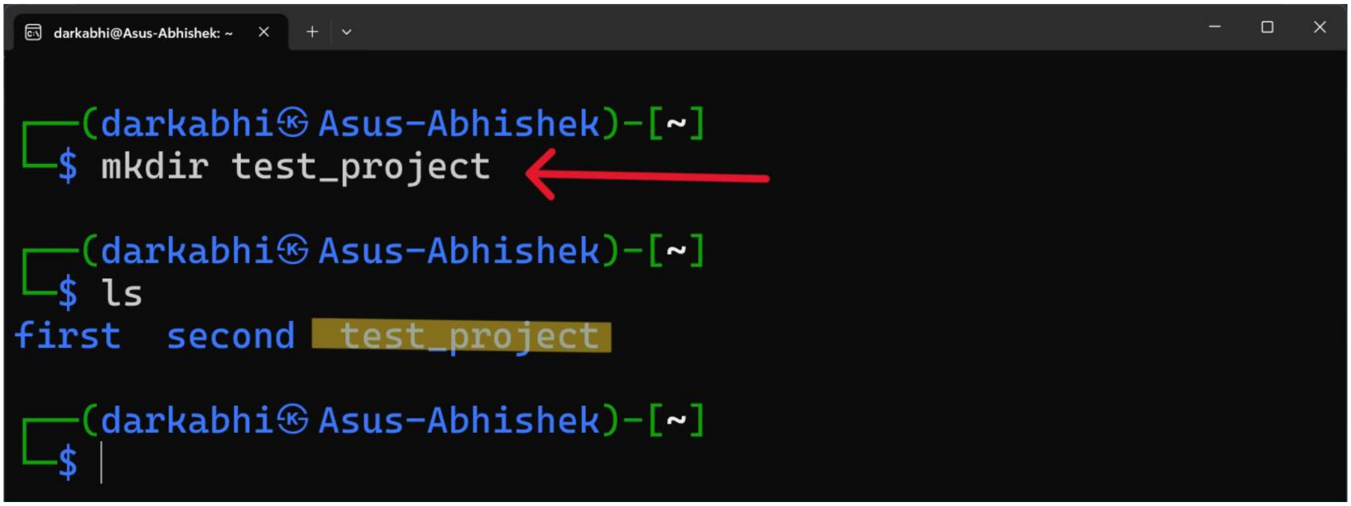


```
┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ pwd
/home/darkabhi

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$
```

COMMAND: pwd

The **pwd (print working directory)** command outputs the full path of the current directory you are working in. This helps confirm your current location in the file system.

3. Create a new directory named “test_project”.

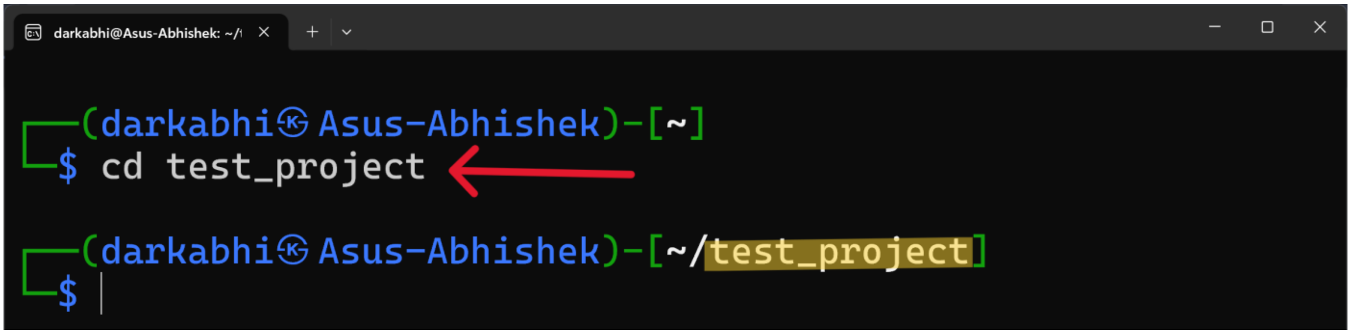


```
┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ mkdir test_project

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ ls
first   second   test_project

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ 
```

COMMAND: mkdir FOLDER_NAME

The **mkdir** command creates a new directory (i.e. folder). Here, it creates a directory named **“test_project”** in the current location. Also after creation **“test_project”** is highlighted with yellow color.

4. Navigate into the “test_project” directory you just created.

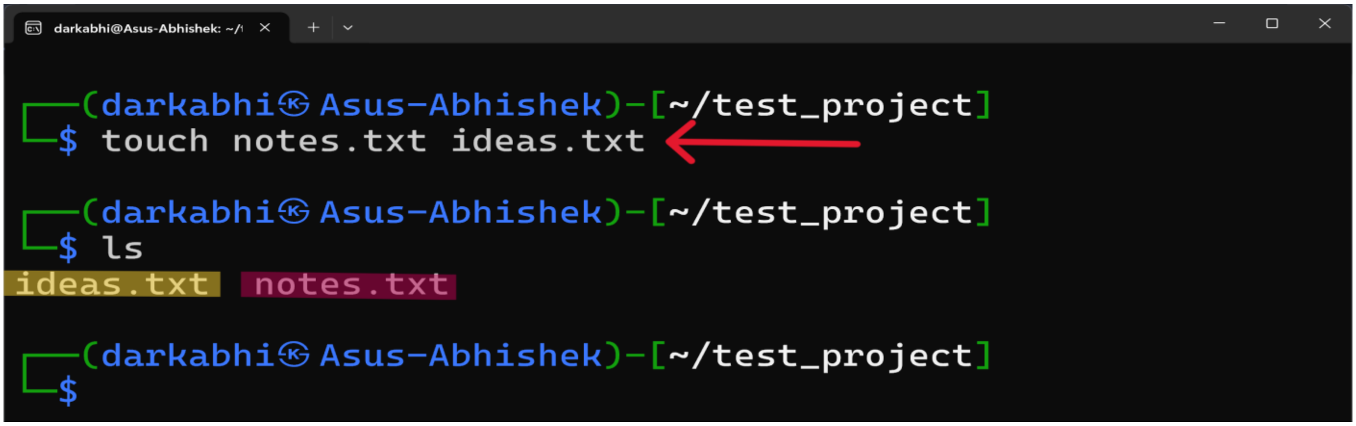


```
┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ cd test_project

┌──(darkabhi㉿Asus-Abhishek)-[~/test_project]
└─$ 
```

COMMAND: cd FOLDER_NAME

The **cd (change directory)** command is used to move into a specified directory. This command moves you into the newly created **“test_project”** directory.

5. Create two empty files inside “test_project” named “notes.txt” and “ideas.txt”.

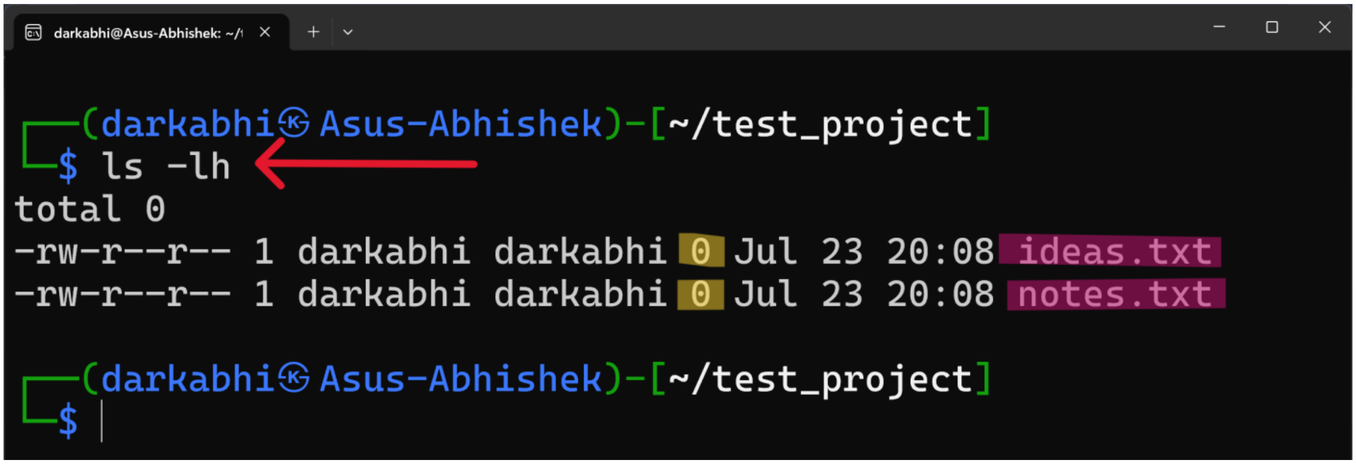


```
┌──(darkabhi㉿Asus-Abhishek)-[~/test_project]
└─$ touch notes.txt ideas.txt

┌──(darkabhi㉿Asus-Abhishek)-[~/test_project]
└─$ ls
ideas.txt  notes.txt

┌──(darkabhi㉿Asus-Abhishek)-[~/test_project]
└─$
```

COMMAND: touch FILE_NAME

The **touch** command creates empty files or updates the timestamps of existing files. Here, it creates two new files named **“notes.txt”** and **“ideas.txt”** inside the **“test_project”** directory.

6. List files in “test_project”, displaying file sizes in human-readable format (e.g., KB, MB).

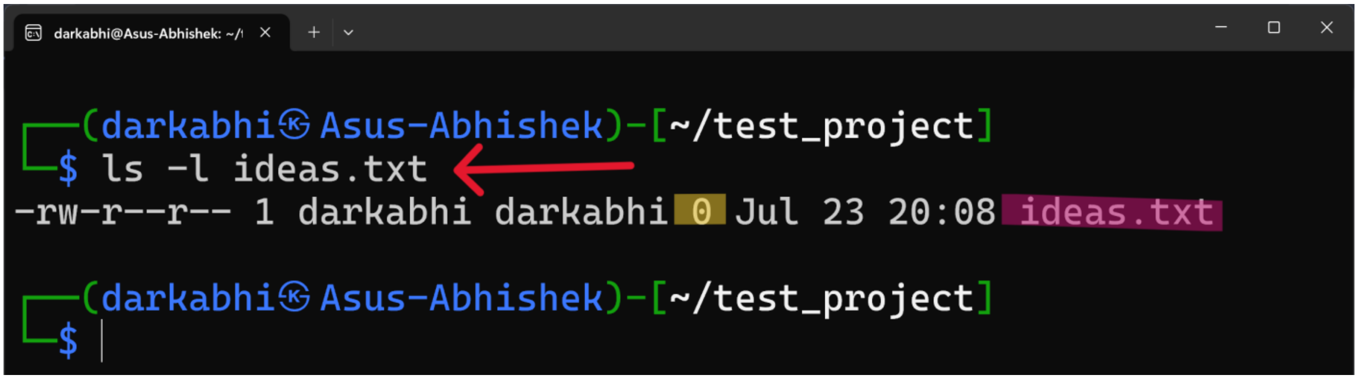


```
┌──(darkabhi㉿Asus-Abhishek)-[~/test_project]
└─$ ls -lh
total 0
-rw-r--r-- 1 darkabhi darkabhi 0 Jul 23 20:08 ideas.txt
-rw-r--r-- 1 darkabhi darkabhi 0 Jul 23 20:08 notes.txt

┌──(darkabhi㉿Asus-Abhishek)-[~/test_project]
└─$
```

COMMAND: ls -lh

The **ls -lh** command lists files and directories with detailed information, displaying file sizes in a human-readable format (e.g., KB, MB). The -h flag makes the file sizes easier to understand.

7. Display detailed information (permissions, owner, size, etc.) about the file “ideas.txt”.

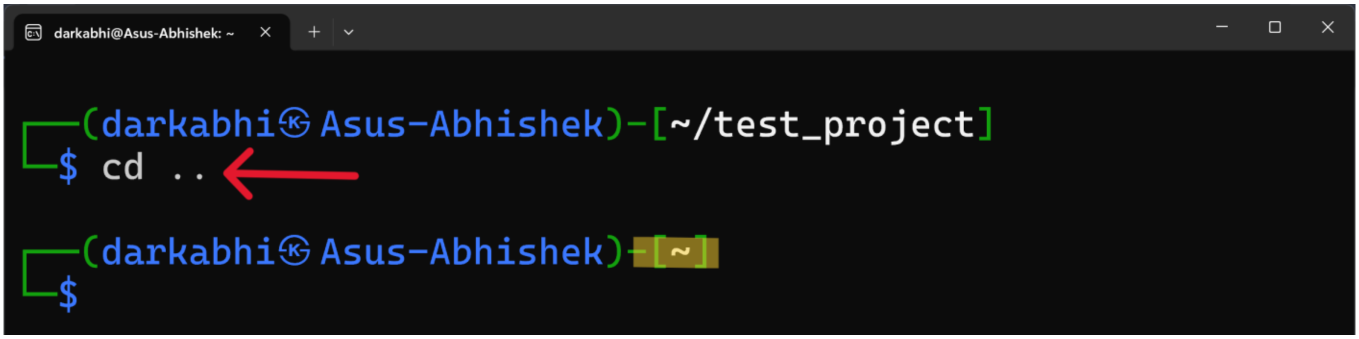


```
┌──(darkabhi㉿Asus-Abhishek)-[~/test_project]
└─$ ls -l ideas.txt
-rw-r--r-- 1 darkabhi darkabhi 0 Jul 23 20:08 ideas.txt

┌──(darkabhi㉿Asus-Abhishek)-[~/test_project]
└─$ 
```

COMMAND: ls -l FILE_NAME

The ls -l command provides detailed information about files and directories. When used with a specific file, like **“ideas.txt”,** it shows details such as permissions, owner, **size**, and modification date.

8. Go back one directory level (back to your home directory).

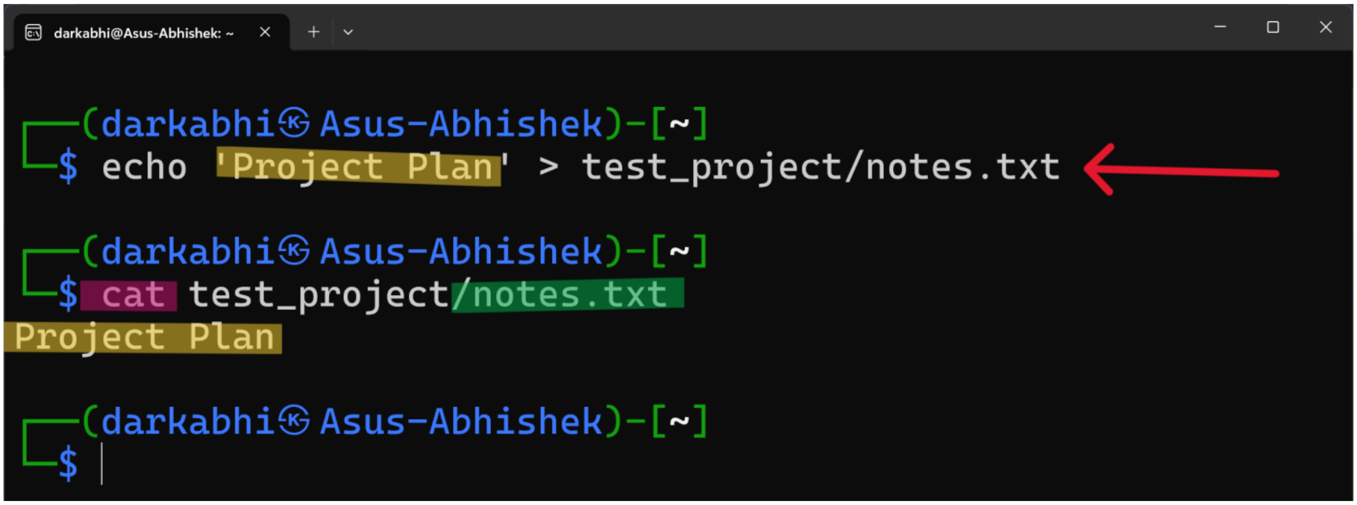


```
┌──(darkabhi㉿Asus-Abhishek)-[~/test_project]
└─$ cd ..

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$
```

COMMAND: cd PATH

The **cd ..** command navigates up one level in the directory structure, taking you back to the parent directory.

9. Write the phrase “Project Plan” into the file “notes.txt” (located inside “test_project”).

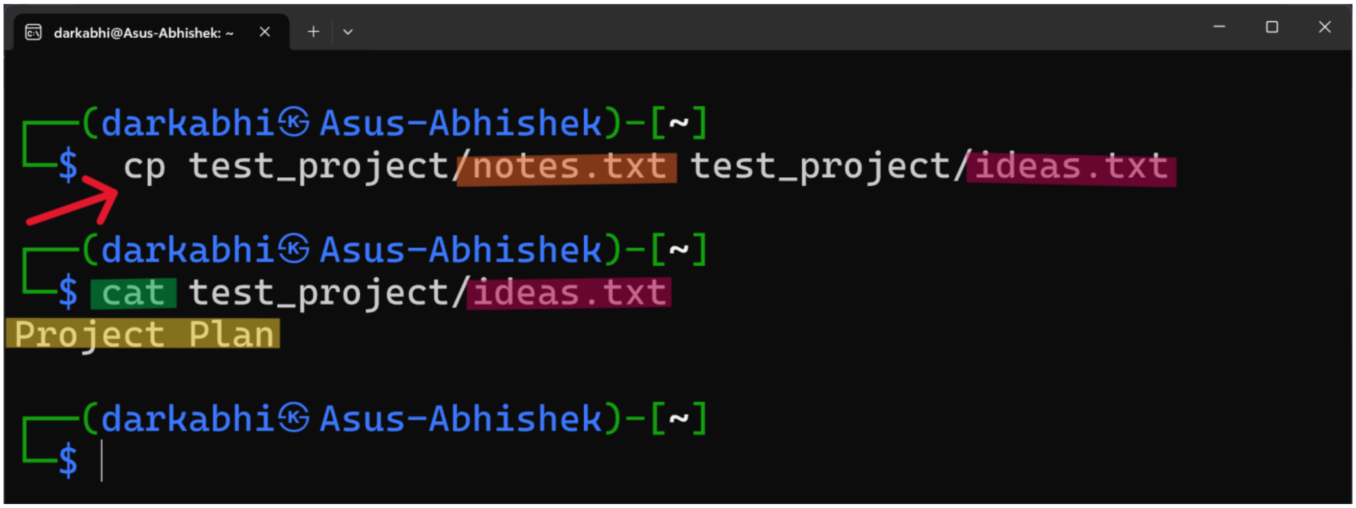


```
┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ echo 'Project Plan' > test_project/notes.txt

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ cat test_project/notes.txt
Project Plan

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$
```

COMMAND: echo CONTENT > FILE_NAME

The echo command outputs the specified text. The > symbol redirects this output into the file **“notes.txt”**, overwriting its contents with **“Project Plan”**.

10. Copy the contents of “notes.txt” into “ideas.txt” (both files still inside “test_project”).

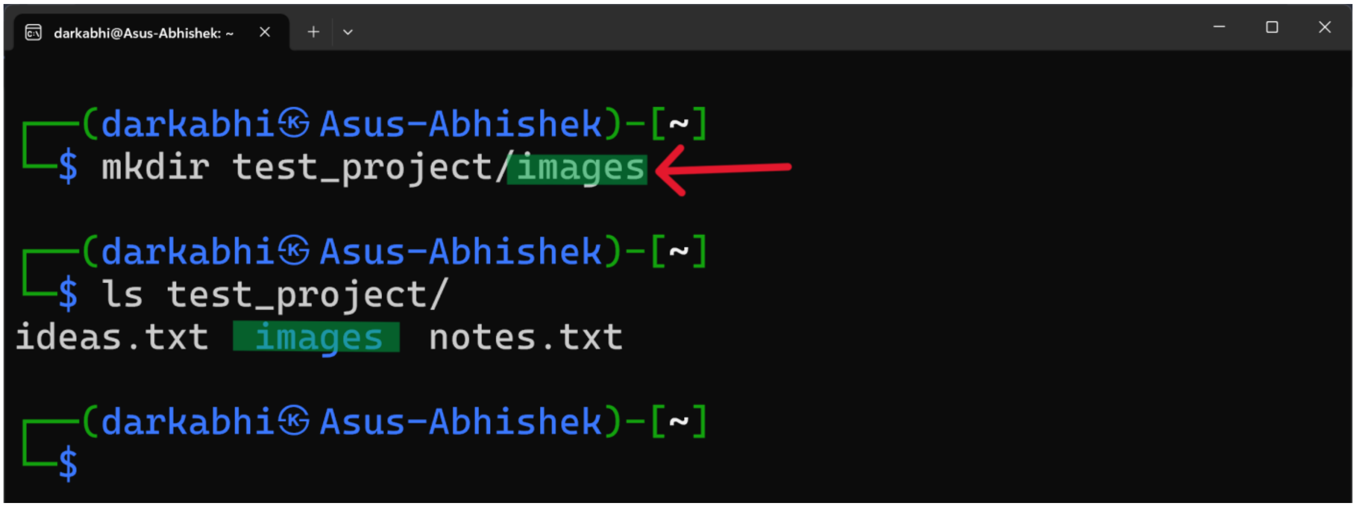


```
┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ cp test_project/notes.txt test_project/ideas.txt

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ cat test_project/ideas.txt
Project Plan

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$
```

COMMAND: cp FILE_TO_COPY WHERE_TO_COPY

The cp command copies files or directories. Here, it copies the contents of **“notes.txt”** into **“ideas.txt”**, overwriting any existing content in **“ideas.txt”**.

11. Inside the “test_project” directory, create a subdirectory called “images”.

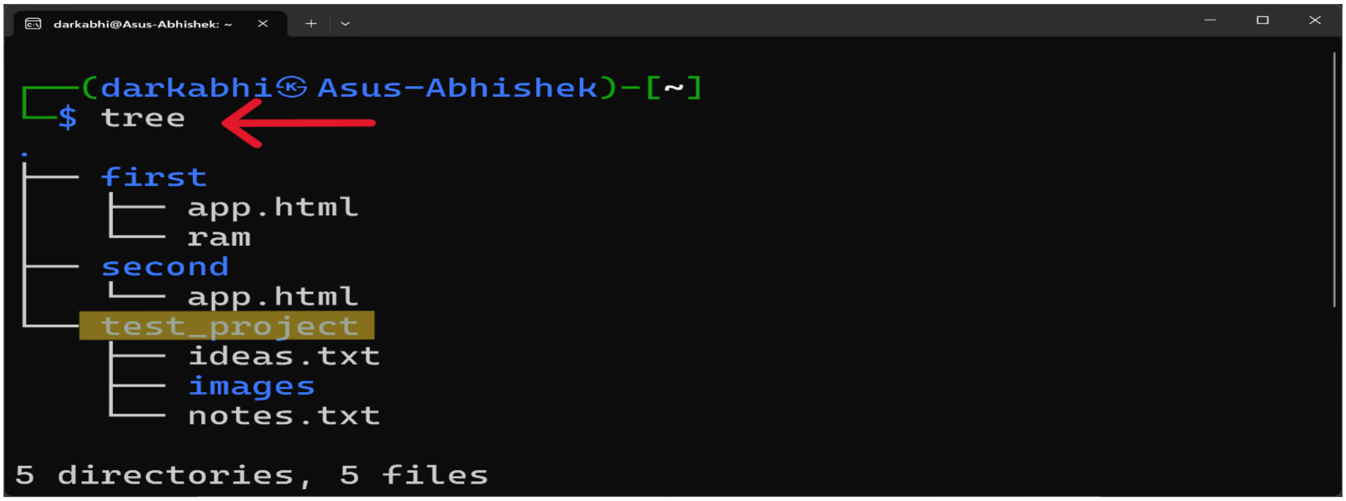


```
┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ mkdir test_project/images

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ ls test_project/
ideas.txt  images  notes.txt

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$
```

COMMAND: mkdir FOLDER_NAME

The **mkdir** command creates a new directory. Here, it creates a subdirectory named **“images”** inside the **“test_project”** directory.

12. List all files within the “test_project” directory and its subdirectories, showing full paths for each file.

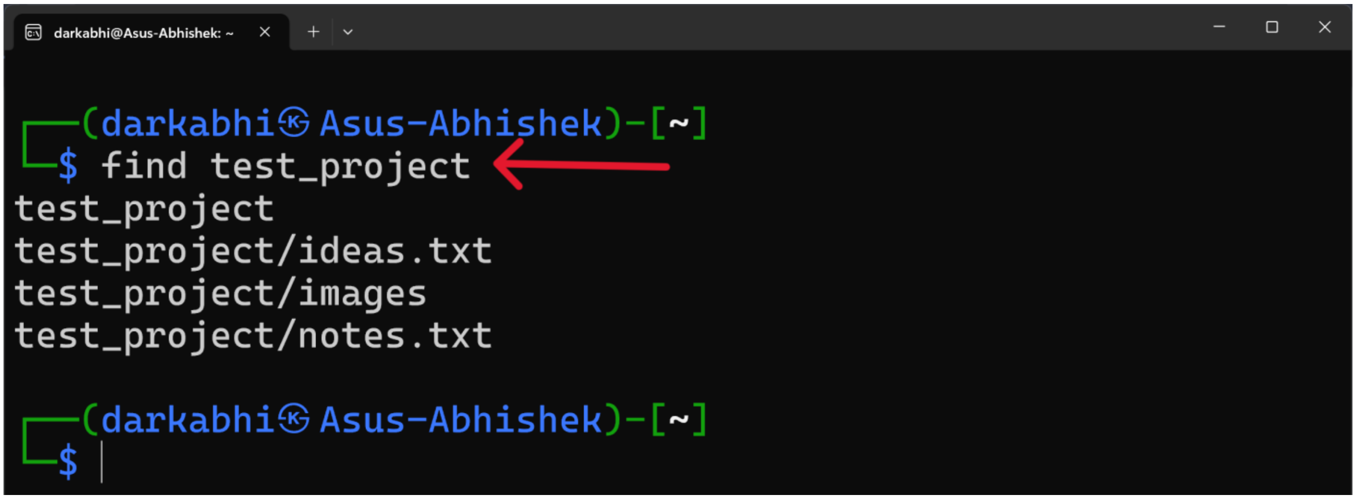


```
┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ tree
.
├── first
│   ├── app.html
│   └── ram
├── second
│   └── app.html
└── test_project
    ├── ideas.txt
    ├── images
    └── notes.txt

5 directories, 5 files
```

COMMAND: tree

The **tree** command displays the directory structure of a path or of the disk in a **tree-like format**. It lists all files and directories, showing their full paths.

OR

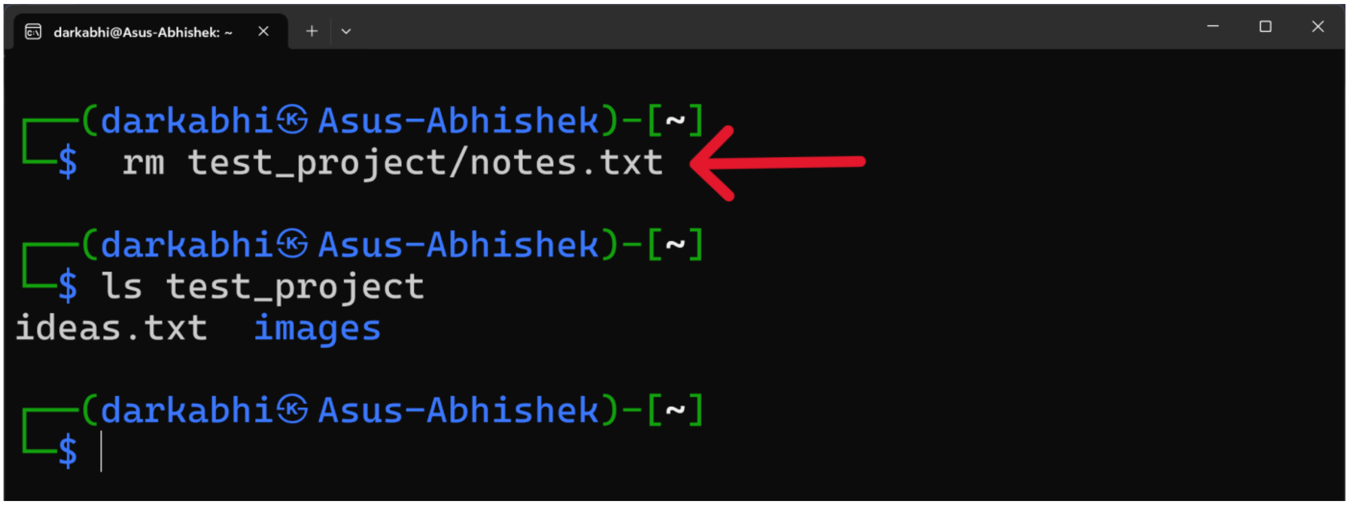


```
┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ find test_project
test_project
test_project/ideas.txt
test_project/images
test_project/notes.txt

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ 
```

COMMAND: find FILE_NAME

The **find** command **searches for files** and directories within a specified directory. Here, find **test_project** lists all files and directories within the **"test_project"** directory, showing their full paths.

13. Remove the file “notes.txt” from inside the “test_project” directory.

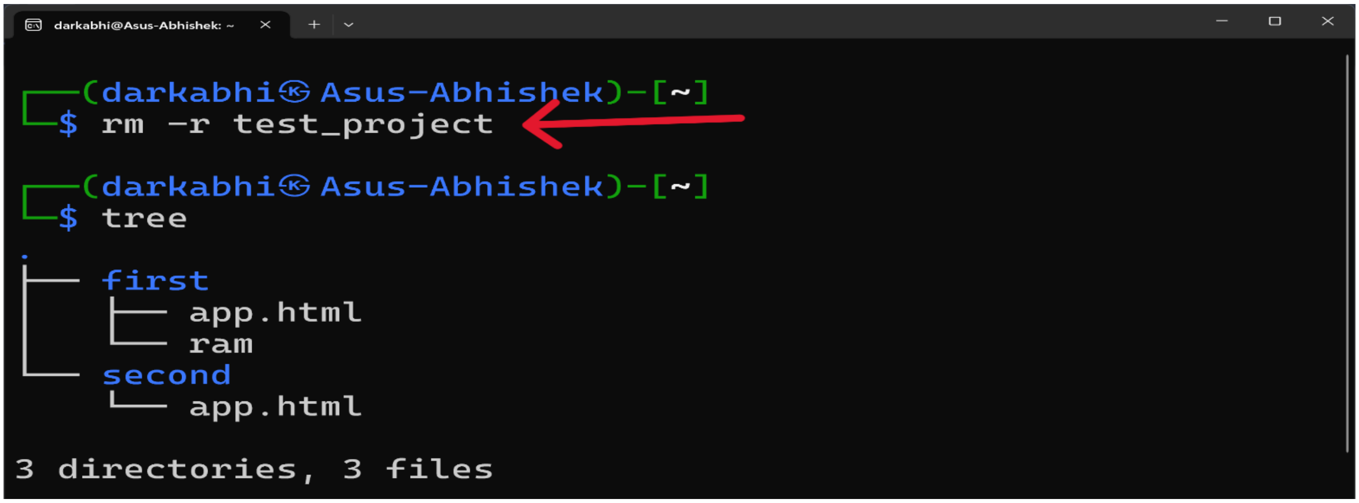


```
┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$  rm test_project/notes.txt

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ ls test_project
ideas.txt  images

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ 
```

COMMAND: rm FILE_NAME

The **rm (remove)** command deletes files. Here, it removes the file **“notes.txt”** from the **“test_project”** directory.

14. Using a single command, delete the entire “test_project” directory and everything inside it.

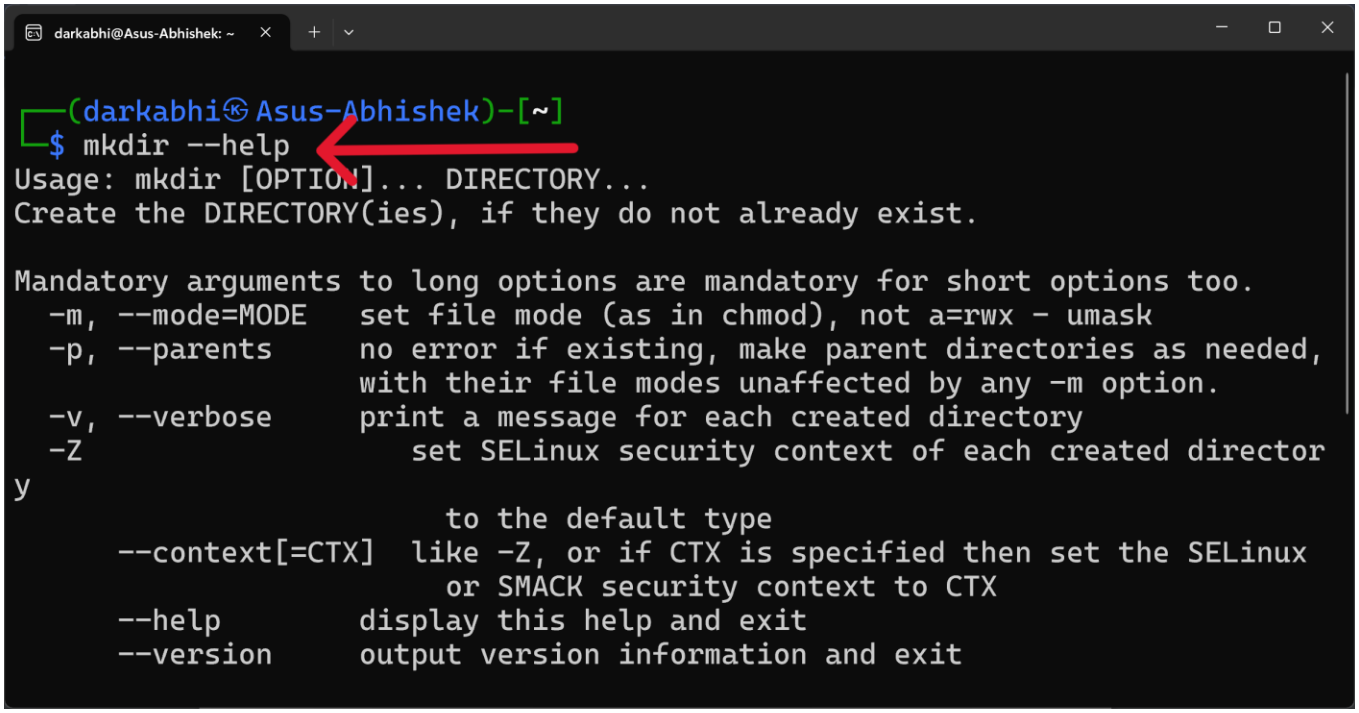


```
┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ rm -r test_project

┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ tree
.
├── first
│   ├── app.html
│   └── ram
└── second
    └── app.html

3 directories, 3 files
```

COMMAND: rm -r FOLDER_NAME

The **rm -r** command recursively deletes a directory and all its contents. Here, it removes the **“test_project”** directory and everything within it.

15. Get comprehensive help on the mkdir command.



```
┌──(darkabhi㉿Asus-Abhishek)-[~]
└─$ mkdir --help
Usage: mkdir [OPTION]... DIRECTORY...
Create the DIRECTORY(ies), if they do not already exist.

Mandatory arguments to long options are mandatory for short options too.
  -m, --mode=MODE   set file mode (as in chmod), not a=rwx - umask
  -p, --parents     no error if existing, make parent directories as needed,
                    with their file modes unaffected by any -m option.
  -v, --verbose     print a message for each created directory
  -Z                   set SELinux security context of each created director
y
                         to the default type
      --context[=CTX]  like -Z, or if CTX is specified then set the SELinux
                         or SMACK security context to CTX
      --help        display this help and exit
      --version     output version information and exit
```

COMMAND: mkdir --help

The **mkdir --help** command provides a summary of the usage, options, and arguments for the mkdir command. It is a quick reference for understanding how to create directories and use various options.

BIBILOGRAPY

1. ***GNU Core Utilities Documentation***
   "GNU Core Utilities: Basic File, Shell, and Text Manipulation Utilities." GNU Project. Accessed July 2024.
   https://www.gnu.org/software/coreutils/manual/coreutils.html
2. ***The Linux Command Line***
   William Shotts. "The Linux Command Line: A Complete Introduction." No Starch Press, 2012.
3. ***Ubuntu Documentation***
   "Ubuntu Man page: mkdir - Make directories." Canonical Ltd. Accessed July 2024.
   https://manpages.ubuntu.com/manpages/focal/man1/mkdir.1.html
4. ***Linux Handson Guide***
   Machtelt Garrels. "Introduction to Linux and File Handling" 1.27 Edition
   https://tldp.org/LDP/intro-linux/html/sect_03_03.html