...for a complete human being

Texas International College

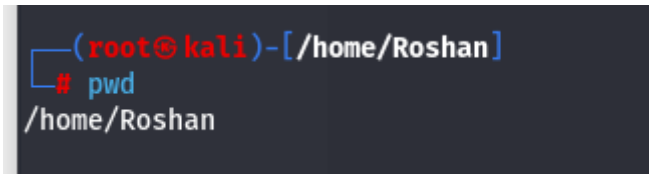
Lab No: 14 **Date:** 2082/

Title: Prepare a lab report for basic Linux command.

Linux commands are text-based instructions that allow users to communicate directly with the operating system through the terminal. They are not limited to just managing files or moving between directories but also extend to tasks like monitoring system performance, managing processes, setting permissions, and even networking operations. They are widely used in cloud computing and ethical hacking.

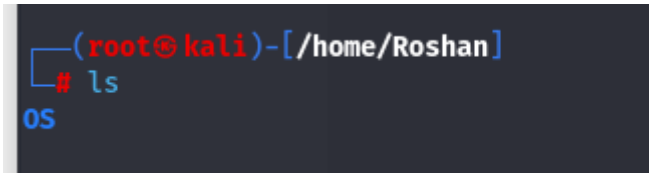
1. pwd

- Prints the current working directory.
- Helps you know where you are inside the file system.
- Useful when navigating deep directory structures.



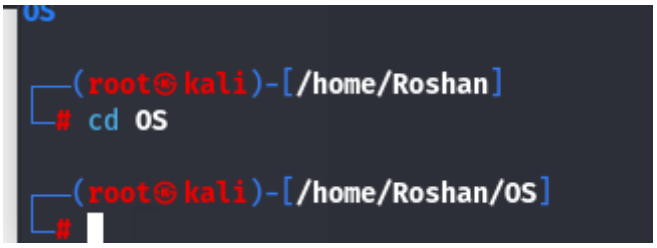
2. ls

- Lists files and directories in the current directory.
- Can be combined with flags like ls -l (long format) or ls -a (show hidden).
- Useful for exploring directory contents.



3. cd

- Changes the current working directory.
- Example: cd /home/user/Documents takes you into Documents.
- cd .. goes one directory up.

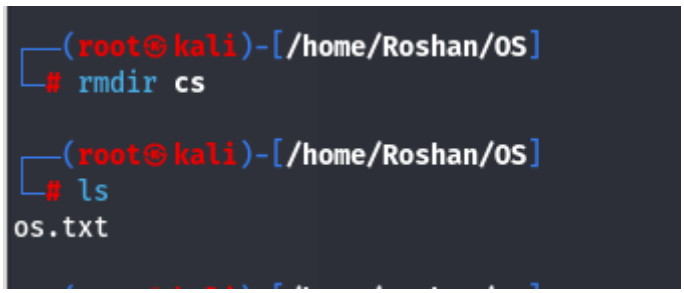


4. `mkdir`

- Creates a new directory.
- Example: mkdir myfolder makes a folder named *myfolder*.
- Often used before organizing or storing files.

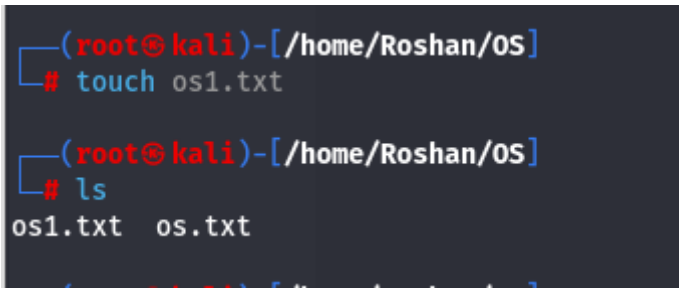
5. rmdir

- Removes an empty directory.
- Example: rmdir oldfolder deletes *oldfolder* only if it has no files.
- For non-empty folders, rm -r is used instead.



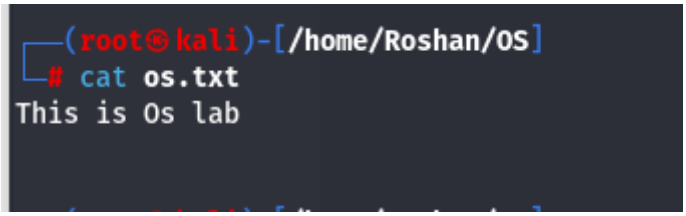
6. touch

- Creates an empty file.
- Example: touch abc.txt creates a blank text file.
- Also used to update file timestamps.



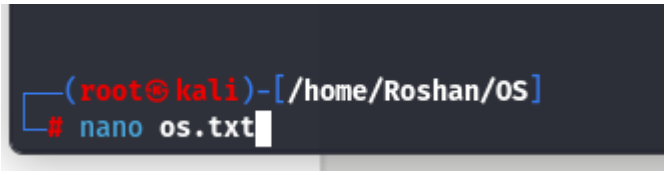
7. cat

- Displays file content in the terminal.
- Example: cat abc.txt prints the contents of abc.*txt*.
- Can also join multiple files into one.



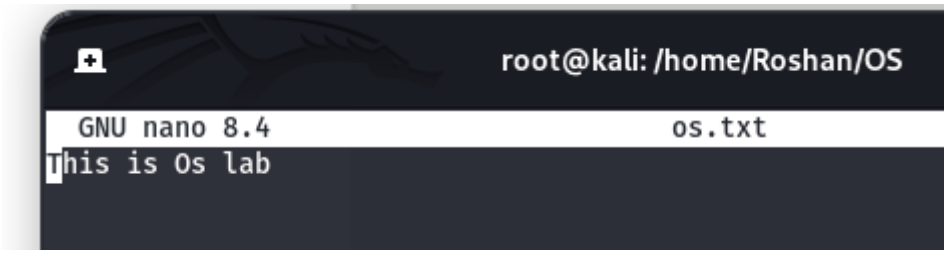
8. nano

- Opens the Nano text editor inside the terminal.
- Example: nano file.txt creates or edits *file.txt*.
- Simple and beginner-friendly editor for editing configs, scripts, or notes.



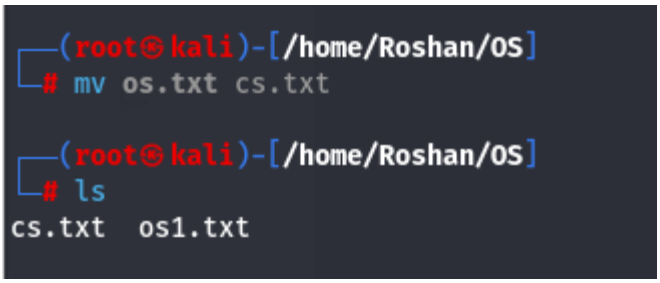
Nano Shortcuts:

- Ctrl + O → Save the file (then press Enter to confirm).
- Ctrl + X → Exit the editor.



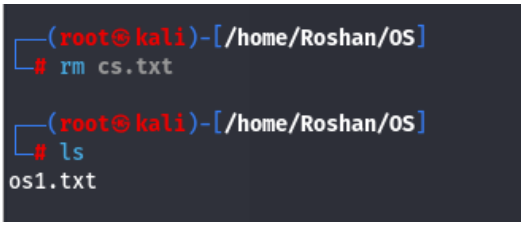
9. mv

- Moves or renames files and directories.
- Example: mv oldname.txt newname.txt renames a file.
- Can also move files between directories.



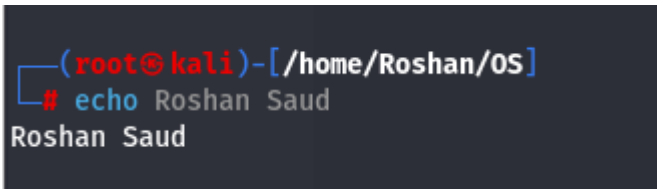
10. rm

- Removes (deletes) files or directories.
- Example: rm file.txt deletes *file.txt*.
- Use with caution; rm -rf is very powerful.



11.echo

- Prints text or variables to the terminal.
- Example: echo Hello prints *Hello*.
- Often used in scripts to display messages.

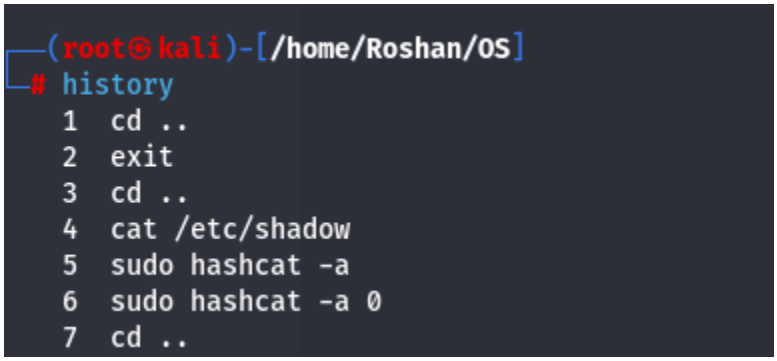


12.grep

- Stands for Global Regular Expression Print.
- Used to search for text patterns inside files or command output.
- Very powerful for filtering logs, configs, or outputs.

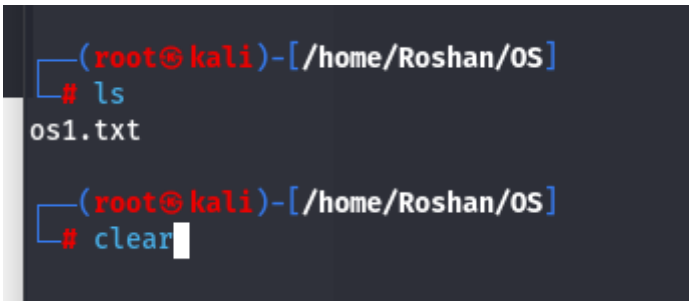
13. history

- Displays previously executed commands.
- Each command is numbered for easy reference.
- Example: !15 re-runs the 15th command from history.

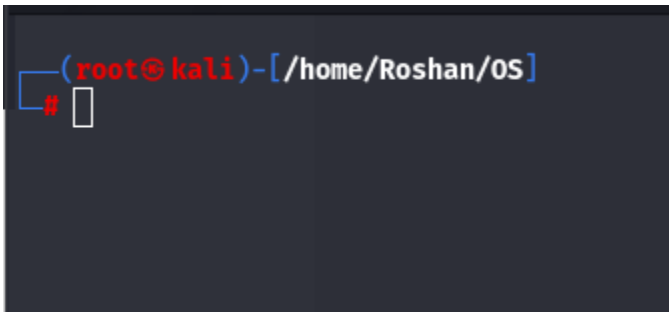


14. clear

- Clears the terminal screen.
- Does not delete data, just removes clutter.
- Keeps your workspace clean while working.
- Before

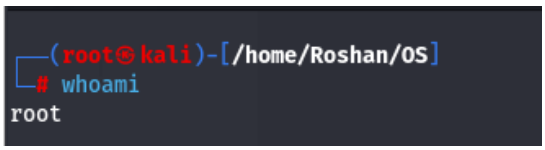


- After



15. whoami

- Prints the current logged-in username.
- Useful when working with multiple accounts.
- Confirms your user identity in the system.



Conclusion

Linux offers a wide range of commands for effective process management. Processes can be started with simple utilities such as sleep, observed using commands like ps or pgrep, and ended with tools such as kill. Gaining familiarity with these fundamental operations allows users to handle active programs smoothly and make better use of system resources.