## **Key Concepts**

* Command Shell: Program for typing commands.
* Command-Line Interface (CLI): Text-based method to interact with the computer.
* Scripting: Saving commands in a file to run them continuously.

## **Important Commands**

### **Basic Navigation**

* pwd: Print working directory (shows current directory).
* ls: List files in the current directory.
  + Options:
    - -F: Classifies files.
    - -a: Shows hidden files.

### **File Management**

* cd [directory]: Change directory.
* mkdir [directory]: Create a new directory.
* rm [file]: Remove a file.
  + rm -r [directory]: Remove a directory and its contents.

### **Viewing & Editing Files**

* cat [file]: Display the contents of a file.
* nano [file]: Open a text editor to edit files.

## **Working with Commands**

* Pipes (|): Connect the output of one command to the input of another.
* Redirection (>, >>): Send command output to a file.
  + >: Overwrite file.
  + >>: Append to file.

## **Loops and Scripting**

* For Loop: Repeats commands for each item in a list.

shell

Run

Copy code

for item in list; do

command $item

done

* Script Execution:
  + Run a script with bash script.sh.

## **Searching**

* grep [pattern] [file]: Search for lines matching a pattern in a file.
  + Options:
    - -i: Case-insensitive.
    - -v: Inverts the match.
* find [directory] -name [pattern]: Find files based on name patterns.

## **Using Variables**

* "$1", "$2": Access command-line arguments in scripts.
* "$@": Refers to all command-line arguments.

## **Comments**

* Add comments in scripts with # for clarity.

## **Practical Tips**

* Use quotes around file names with spaces.
* Use history to recall recent commands.
* Use man [command] to see command manuals.

## **Example Script**

bash

Run

Copy code

#!/bin/bash

# Example script to count lines in .txt files

wc -l \*.txt > line\_count.txt

* This saves the line count of all .txt files in line\_count.txt.

### **1. File Management**

* ls: Lists files in the current directory.
  + Example: ls -la (shows hidden files and detailed info)
* cd: Changes the current directory.
  + Example: cd /path/to/directory
* mkdir: Creates a new directory.
  + Example: mkdir new\_folder
* rm: Removes files or directories.
  + Example: rm filename (for files), rm -r foldername (for directories)
* cp: Copies files or directories.
  + Example: cp source\_file destination\_file
* mv: Moves or renames files or directories.
  + Example: mv old\_name new\_name

### **2. File Viewing and Editing**

* cat: Displays the contents of a file.
  + Example: cat file.txt
* less: Views file contents, allowing navigation.
  + Example: less file.txt
* nano/vim: Text editors for creating and editing files.
  + Example: nano file.txt

### **3. System Information**

* top: Displays running processes and system usage.
* df: Shows disk space usage.
  + Example: df -h (human-readable format)
* free: Displays memory usage.
  + Example: free -m (in MB)

### **4. Networking**

* ping: Tests connectivity to another host.
  + Example: ping google.com
* curl: Transfers data from or to a server.
  + Example: curl http://example.com

### **5. Permissions**

* chmod: Changes file permissions.
  + Example: chmod 755 filename
* chown: Changes file ownership.
  + Example: chown user:group filename

## **Scripting and Automation**

* Shell Scripts: Text files containing a series of commands executed by the shell.
  + Basic structure:
  + bash
  + Run
  + Copy code

#!/bin/bash

* + echo "Hello, World!"
* Executing a Script:
  + Make it executable: chmod +x script.sh
  + Run it: ./script.sh

## 