Linux commands are text-based instructions entered in the terminal to interact with the operating system. They allow users to navigate the file system, manage files and processes, control system behavior, and automate tasks efficiently with precision and speed.

Linux commands are used for:

- **Fast system control:** Perform tasks faster than a GUI using simple commands.
- **Powerful automation:** Combine commands and scripts to automate repetitive tasks.
- **Efficient resource management:** Monitor and control processes, memory, and disk usage.
- **Remote system administration:** Manage servers securely over a network using terminal access.

1. File Operations Commands

File operations commands are used to create, view, copy, move, compare, rename, and delete files in a Linux system. They help users efficiently manage file data and perform day-to-day file handling tasks from the command line.

- access
- basename
- cat
- cksum
- cmp
- compress
- cp
- cpio
- csplit
- cut
- diff
- diff3
- echo
- expand
- file
- fold
- head
- join
- less
- ln
- locate
- look
- more
- mv
- od
- paste
- readlink
- rename
- rev
- rm
- shred
- sort
- split
- tac
- tail
- tar
- tee
- touch
- unexpand

- uniq
- wc

2. Directory Operations Commands

Directory operations commands are used to navigate, list, create, search, and remove directories in a Linux file system. They help users organize files and manage directory structures efficiently from the command line.

- cd
- dir
- dirname
- dirs
- du
- find
- lsblk
- mkdir
- mount
- pwd
- rmdir
- tree

3. File Permission and Ownership Commands

File permission and ownership commands are used to control access rights for files and directories by defining who can read, write, or execute them. They also allow administrators to change file ownership and group assignments to maintain system security.

- chmod
- chattr
- chown
- chgrp

4. User Management Commands

User management commands are used to create, modify, and delete user accounts in a Linux system. They help administrators manage user access, authentication, and account-related settings securely.

- chage
- chfn
- chsh
- chpasswd
- finger
- id
- passwd
- pinky
- username
- useradd
- userdel
- usermod
- users
- who
- whoami

5. Group Management Commands

Group management commands are used to create, modify, and delete user groups in a Linux system. They help administrators control group-based permissions and manage multiple users efficiently.

- groupadd
- groupdel
- groupmod
- groups
- gpasswd
- grpck
- grpconv

6. Process Management Commands

Process management commands are used to monitor, control, and manage running processes. They help track system performance, control resource usage, and terminate or prioritize processes.

- accton
- bg
- chrt
- fg
- kill
- mpstat
- pidof
- pmap
- ps
- top
- htop
- strace
- time
- watch
- vmstat
- uptime
- w

7. Networking Commands

Networking commands are used to configure, monitor, and troubleshoot network connections. They help manage IP addresses, test connectivity, transfer data, and analyze network performance.

- arp
- curl
- host
- hostid
- hostname
- hostnamectl
- ifconfig
- iftop
- ifup
- ip
- ipcrm
- ipcs
- iptables
- iptables-save
- iwconfig
- nc (netcat)
- netstat

- nmcli
- nslookup
- ping
- rcp
- route
- rsync
- scp
- ssh
- tracepath
- traceroute
- vnstat
- wget

8. Package Management Commands

Package management commands are used to install, update, upgrade, and remove software packages. They ensure proper dependency handling and keep the system up to date.

- apt
- apt-get
- aptitude

9. Job Scheduling Commands

Job scheduling commands are used to schedule tasks for future or recurring execution. They help automate routine system jobs like backups, updates, and maintenance.

- atd
- atrm
- atq
- batch
- cron
- crontab

10. Disk and File System Commands

Disk and file system commands are used to manage disks, partitions, and file systems. They help mount storage, check disk usage, and maintain data integrity.

- cfdisk
- df
- dosfsck
- dump
- dumpe2fs
- fdisk
- mount
- restore
- sync

11. Hardware and System Information Commands

These commands are used to display hardware details and system resource information. They help monitor CPU, memory, storage, and device-related data.

- acpi
- acpi_available

- acpid
- arch
- dmesg
- dmidecode
- dstat
- free
- hdparm
- hwclock
- iostat
- iotop
- lsusb
- lshw
- uname

12. Compression and Archiving Commands

Compression and archiving commands are used to compress, extract, and manage archived files. They help reduce storage usage and simplify file transfer.

- ar
- bzcmp
- bzdiff
- bzgrep
- bzip2
- bzless
- bzmore
- gunzip
- gzip
- gzexe
- zip
- zdiff
- zgrep

13. Text Processing and Formatting Commands

Text processing commands are used to search, filter, format, and manipulate text data. They are widely used for log analysis, scripting, and data processing.

- awk
- aspell
- banner
- bc
- col
- colcrt
- colrm
- column
- dc
- egrep
- fgrep
- fmt
- grep
- sdiff
- sed
- tr
- unix2dos

14. Kernel and Module Management Commands

Kernel and module management commands are used to load, remove, and manage kernel modules and system services. They help control low-level system functionality.

- depmod
- insmod
- lsmod
- modinfo
- rmmod
- systemctl

15. System Control and Power Commands

System control commands are used to safely shut down, reboot, or power off the system. They help ensure proper system termination and data safety.

- halt
- poweroff
- reboot
- shutdown

16. Logging and Monitoring Commands

Logging and monitoring commands are used to view system logs and track system activity. They help diagnose issues, analyze usage, and audit system behavior.

- journalctl
- last
- history
- sar
- script
- scriptreplay

17. Checksum and File Integrity Commands

These commands are used to verify file integrity using hash values. They help detect file corruption or unauthorized changes.

- md5sum
- cksum
- sum

18. Date and Time Commands

Date and time commands are used to display and manage system date, time, and uptime. They help with time synchronization and system monitoring.

- cal
- date
- uptime

19. Mail and User Communication Commands

Mail and communication commands are used for user messaging and system notifications. They help administrators communicate with users and manage mail queues.

- biff
- mailq
- write
- wall

20. Printing and Media Commands

Printing and media commands are used to manage audio, printing services, and media devices. They help control sound, printers, and removable media.

- amixer
- aplay
- aplaymidi
- cupsd
- eject
- import

21. Shell Built-in and Scripting Commands

Shell built-in commands are used for scripting, automation, and flow control within the shell. They help write efficient scripts and control command execution.

- alias
- bind
- break
- builtin
- case
- continue
- declare
- enable
- env
- eval
- exec
- exit
- expect
- export
- expr
- factor
- fc
- function
- for
- if
- let
- printf
- read
- return
- select
- seq
- setsid
- shift
- source
- type
- until
- while

- yes
- sudo
- sleep

Bash Shortcuts Commands:

Bash shortcut commands are keyboard combinations used in the Linux terminal to quickly perform common actions without typing full commands. They help users work faster and more efficiently by improving navigation, editing, and command execution.

1. Navigation Shortcuts

Used to move the cursor quickly within the command line.

- **Ctrl + A :** Move to the beginning of the line
- **Ctrl + E :** Move to the end of the line
- **Ctrl + B :** Move back one character
- **Ctrl + F :** Move forward one character
- **Alt + B :** Move back one word
- **Alt + F :** Move forward one word

2. Editing Shortcuts

Used to edit or modify the command line efficiently.

- **Ctrl + U :** Cut/delete text from the cursor to the beginning of the line
- **Ctrl + K :** Cut/delete text from the cursor to the end of the line
- **Ctrl + W :** Cut/delete the word before the cursor
- **Ctrl + Y :** Paste the last cut text
- **Ctrl + L :** Clear the terminal screen
- **Ctrl + C :** Terminate the currently running command

3. History Shortcuts

Used to search and navigate through previously executed commands.

- **Ctrl + R :** Search command history (reverse search)
- **Ctrl + G :** Exit history search mode
- **Ctrl + P :** Go to the previous command in history
- **Ctrl + N :** Go to the next command in history

22. Development and Build Automation Commands

Development commands are used to compile, build, debug, and analyze programs. They support software development and build automation processes.

- aclocal
- addr2line
- autoconf
- autoheader
- automake
- autoreconf
- autoupdate
- bison
- cc
- cpp
- ctags

- g++
- gcc
- gdb
- ranlib
- readelf

23. Terminal and Session Management Commands

Terminal and session management commands are used to manage terminal sessions and input/output behavior. They help control multiple sessions and terminal settings.

- agetty
- chvt
- reset
- screen
- showkey
- stty
- tty
- xdg-open

24. Help and Documentation Commands

Help and documentation commands are used to view manuals, usage guides, and command descriptions. They assist users in learning and understanding Linux commands.

- apropos
- help
- info
- man
- whatis
- which

25. Text Editors in Linux

Text editors are used to create and modify files from the terminal.

1. nano

`nano` is a simple and beginner-friendly text editor used in the terminal. It provides on-screen shortcuts, making it easy to edit files without prior experience.

2. vi

`vi` is a powerful and lightweight text editor available on almost all Linux systems. It works in different modes, which allows efficient text editing using keyboard commands.

3. vim

`vim` (Vi Improved) is an advanced version of vi with enhanced features like syntax highlighting and plugins. It is widely used by developers for fast and efficient coding.

4. ed

`ed` is a line-based text editor and one of the oldest editors in Linux. It is mainly used for scripting and low-level text processing.

5. emacs

`emacs` is a highly customizable and extensible text editor. It supports programming, scripting, email, and many other tasks beyond basic text editing.

Shortcuts Commands & Keys of Text Editors

There are many shortcuts commands in Linux that can help you be more productive. Here are a few of the most common ones:

1. Nano Shortcuts Commands:

File Operations

Used to open, save, and exit files.

- **Ctrl + O –** Save (write) the current file
- **Ctrl + X –** Exit Nano (prompts to save if the file is modified)
- **Ctrl + R –** Read and insert another file into the current buffer

Navigation

Used to move through the file quickly.

- **Ctrl + Y –** Scroll up one page
- **Ctrl + V –** Scroll down one page
- **Alt + \ –** Go to a specific line number
- **Alt + , –** Move to the beginning of the current line
- **Alt + . –** Move to the end of the current line

Editing

Used to modify text efficiently.

- **Ctrl + K –** Cut/delete text from the cursor to the end of the line
- **Ctrl + U –** Uncut (paste) the last cut text
- **Ctrl + 6 –** Mark a block of text for copying or cutting
- **Alt + 6 –** Copy the marked block of text
- **Ctrl + K –** Cut/delete the marked block of text
- **Ctrl + J –** Justify (format) the current paragraph

Search and Replace

Used to find and replace text in a file.

- **Ctrl + W –** Search for a string in the text
- **Alt + W –** Search and replace a string
- **Alt + R –** Repeat the last search

2. VI/VIM Shortcuts Commands:

Insert & Replace Mode Commands

Used to enter insert mode or replace existing text.

- **i –** Switch to insert mode before the cursor
- **a –** Switch to insert mode after the cursor

- **A –** Switch to insert mode at the end of the current line
- **o –** Insert a new line below the current line and switch to insert mode
- **R –** Enter replace mode and overwrite characters until Esc is pressed
- **r –** Replace the character under the cursor with a single new character
- **s –** Substitute the character under the cursor and switch to insert mode
- **S –** Delete the current line and switch to insert mode
- **C –** Delete from the cursor to the end of the line and switch to insert mode

Delete & Change Commands

Used to remove or modify text efficiently.

- **x –** Delete the character under the cursor
- **dd –** Delete the current line
- **3dd –** Delete the current line and the next two lines
- **D –** Delete from the cursor to the end of the line
- **dw –** Delete from the cursor to the beginning of the next word
- **4dw –** Delete the next four words from the cursor position
- **cw –** Change the current word and switch to insert mode

Undo & Restore Commands

Used to revert changes.

- **u –** Undo the last change
- **U –** Restore the current line to its original state

Case & Miscellaneous Commands

Used for quick character-level edits.

- **~ –** Toggle the case of the character under the cursor

Mode Control

Used to switch between editing modes.

- **Esc –** Exit insert or command-line mode and return to command mode

3. Vim Modes and Commands

Normal Mode

Used for navigation, deletion, copying, and undo/redo operations.

- **i –** Enter insert mode at the current cursor position
- **x –** Delete the character under the cursor
- **dd –** Delete the current line
- **yy –** Copy (yank) the current line
- **p –** Paste the copied or deleted text below the current line
- **u –** Undo the last change
- **Ctrl + R –** Redo the last undone change

Command Mode (Last Line Mode)

Used for saving files, quitting Vim, and performing advanced operations.

- **:w –** Save the file

- **:q –** Quit Vim
- **:q! –** Quit Vim without saving changes
- **:wq or :x –** Save and quit Vim
- **:set nu or :set number –** Display line numbers
- **:s/old/new/g –** Replace all occurrences of old with new in the file

Visual Mode

Used for selecting text to copy, delete, or modify.

- **v –** Enter visual mode to select text
- **y –** Copy (yank) the selected text
- **d –** Delete the selected text
- **p –** Paste the copied or deleted text

26. IO Redirection Commands

IO (Input/Output) redirection commands are used to redirect the standard input, output, and error streams of commands and processes. Here are some commonly used IO redirection commands:

1. **cmd < file :** Redirects the input of cmd to be read from file instead of the keyboard.
2. **cmd > file :** Redirects the standard output (stdout) of cmd to file, overwriting existing content.
3. **cmd >> file :** Appends the standard output (stdout) of cmd to the end of file.
4. **cmd 2> file :** Redirects the error output (stderr) of cmd to file.
5. **cmd 2>&1 :** Redirects stderr to the same destination as stdout.
6. **cmd &> file :** Redirects both stdout and stderr to file.
7. **cmd 1>&2 :** Redirects stdout to the same destination as stderr.
8. **cmd > /dev/null :** Discards the standard output by sending it to the null device.
9. **cmd1 <(cmd2) :** Uses the output of cmd2 as an input file for cmd1 (process substitution).

27. Environment Variable Commands

Environment variables are used to store configuration settings, system information, and other variables that can be accessed by processes and shell scripts. Here are some commonly used environment variable commands:

- **export VARIABLE_NAME=value :** Sets and exports an environment variable so it is available to child processes.
- **echo $VARIABLE_NAME :** Displays the value of a specific environment variable.
- **env :** Lists all environment variables currently set in the system.
- **unset VARIABLE_NAME :** Removes or unsets an existing environment variable.
- **export -p :** Shows a list of all currently exported environment variables.
- **env VAR1=value COMMAND :** Sets an environment variable temporarily for a specific command execution.
- **printenv :** Displays the values of all environment variables or a specific one if provided.