

1. **ls (list)** - Lists the files and directories in the current directory.  
**Example:** `ls -l` (lists files and directories in long format)
2. **cd (change directory)** - Changes the current working directory.  
**Example:** `cd /home/user` (changes to the /home/user directory)
3. **mkdir (make directory)** - Creates a new directory.  
**Example:** `mkdir newdir` (creates a new directory named "newdir")
4. **rm (remove)** - Deletes files or directories.  
**Example:** `rm file.txt` (deletes the file named "file.txt")
5. **cp (copy)** - Copies files or directories.  
**Example:** `cp file.txt /home/user/Documents` (copies "file.txt" to the Documents directory)
6. **mv (move)** - Moves or renames files or directories.  
**Example:** `mv file.txt newname.txt` (renames "file.txt" to "newname.txt")
7. **cat (concatenate)** - Displays the contents of a file.  
**Example:** `cat file.txt` (displays the contents of "file.txt")
8. **grep (global regular expression print)** - Searches for a pattern in files.  
**Example:** `grep "hello" file.txt` (searches for the word "hello" in "file.txt")
9. **chmod (change mode)** - Changes the permissions of files or directories.  
**Example:** `chmod 755 file.sh` (gives read, write, and execute permissions to the owner, and read and execute permissions to group and others for "file.sh")
10. **sudo (super user do)** - Runs a command with superuser (root) privileges.  
**Example:** `sudo apt-get update` (updates the package lists as the root user)
11. **apt-get (Advanced Packaging Tool)** - Used for installing, updating, and removing packages in Debian-based Linux distributions.  
**Example:** `sudo apt-get install package_name` (installs the specified package)
12. **yum (Yellowdog Updater, Modified)** - Used for installing, updating, and removing packages in Red Hat-based Linux distributions.  
**Example:** `sudo yum install package_name` (installs the specified package)
13. **nano** - A command-line text editor.  
**Example:** `nano file.txt` (opens the "file.txt" file for editing in the nano editor)
14. **vim (Vi Improved)** - A powerful text editor with advanced features.  
**Example:** `vim file.txt` (opens the "file.txt" file for editing in the vim editor)
15. **pwd (print working directory)** - Prints the current working directory.  
**Example:** `pwd` (prints the full path of the current directory)

16. **clear** - Clears the terminal screen.  
**Example:** clear (clears the terminal screen)
17. **history** - Shows the list of recently executed commands.  
**Example:** history (displays the command history)
18. **man (manual)** - Displays the user manual for a command.  
**Example:** man ls (displays the manual pages for the "ls" command)
19. **echo** - Prints a line of text to the terminal.  
**Example:** echo "Hello, World!" (prints "Hello, World!" to the terminal)
20. **tar (tape archive)** - Used for archiving and extracting files.  
**Example:** tar -czf archive.tar.gz directory (creates a gzip compressed archive named "archive.tar.gz" from the specified directory)
21. **ssh (secure shell)** - Used for securely connecting to a remote system.  
**Example:** ssh user@remote\_host (connects to the remote host as the specified user)
22. **ping** - Used to test connectivity with a remote host.  
**Example:** ping google.com (sends ICMP echo requests to google.com)
23. **top** - Displays real-time information about running processes and system resources.  
**Example:** top (shows the top CPU-consuming processes)
24. **ps (process status)** - Lists the currently running processes.  
**Example:** ps aux (lists all running processes)
25. **kill** - Terminates a process by sending a signal.  
**Example:** kill 1234 (terminates the process with PID 1234)
26. **env (environment)** - Displays or sets environment variables.  
**Example:** env (displays all environment variables)
27. **cron** - Used for scheduling tasks to run at specific times or intervals.  
**Example:** crontab -e (opens the crontab editor to schedule tasks)
28. **sed (stream editor)** - A powerful text stream editor for editing files.  
**Example:** sed 's/old/new/g' file.txt (replaces all occurrences of "old" with "new" in "file.txt")
29. **awk (Aho Weinberger Kernighan)** - A programming language and text processing utility.  
**Example:** awk '/pattern/ {print \$1}' file.txt (prints the first field of lines matching the pattern in "file.txt")
30. **find** - Searches for files and directories based on specified criteria.

**Example:** find / -name "file.txt" (searches for a file named "file.txt" starting from the root directory)

31. **sort** - Sorts the lines of a file or input.

**Example:** sort file.txt (sorts the lines in "file.txt" in alphabetical order)

32. **uniq (unique)** - Removes or reports repeated lines in a file.

**Example:** uniq file.txt (displays the unique lines in "file.txt")

33. **diff** - Compares the contents of two files or directories.

**Example:** diff file1.txt file2.txt (compares the contents of "file1.txt" and "file2.txt")

34. **head** - Displays the first few lines of a file.

**Example:** head -n 5 file.txt (displays the first 5 lines of "file.txt")

35. **tail** - Displays the last few lines of a file.

**Example:** tail -n 10 file.log (displays the last 10 lines of "file.log")