LINUX & COMMANDS 101

What is Linux?

Linux is an open-source, Unix-like operating system (OS) based on the Linux kernel, first released by Linus Torvalds in 1991. Unlike Windows and macOS, Linux is free to use, modify, and distribute. It powers everything from smartphones (Android) and servers to supercomputers and IoT devices.

Why use Linux?

Linux is widely used in development, cybersecurity, networking, and even personal computing. Here's why people choose it:

1. Free & Open-Source
    - No licensing fees. Modify it as per your needs.
2. Security & Privacy
    - Less prone to viruses and malware compared to Windows.
    - No telemetry (spying) like Windows 11.
3. Customizability
    - Modify everything—appearance, commands, window managers, etc.
    - Choose from different distros (Ubuntu, Arch, Fedora, etc.).
4. Performance & Stability
    - Faster, especially on old/low-end hardware.
    - Doesn't slow down over time like Windows.
5. Development-Friendly
    - Comes with Bash, Python, Git, Docker, and more.
    - Preferred by developers for software and web development.
6. Command Line Power
    - Automate everything using shell scripting.
    - More control than GUI-based OS.
7. Massive Community Support
    - Tons of forums, documentation, and tutorials.
8. Ideal for Servers & Cloud Computing
    - Powers most web servers (Google, Facebook, Amazon).
    - Used in cloud platforms (AWS, Azure, Google Cloud).

Where is Linux used?

- **Servers** – Most web servers run Linux (Apache, Nginx).
- **Programming** – Developers prefer it for Python, Java, C++, etc.
- **Cybersecurity** – Kali Linux is a hacking/pentesting OS.
- **Embedded Systems** – Used in IoT devices, routers, and TVs.
- **Supercomputers** – 100% of the world's top 500 supercomputers run Linux.

Basic Commands

1. **`pwd`** – Print the current working directory.
2. **`ls`** – List files and directories in the current directory.
3. **`cd <directory>`** – Change directory. Example: `cd /home/user`
4. **`mkdir <dirname>`** – Create a new directory. Example: `mkdir projects`
5. **`rmdir <dirname>`** – Remove an empty directory.
6. **`rm <file>`** – Remove a file. Example: `rm file.txt`
7. **`rm -r <dirname>`** – Remove a directory and its contents.
8. **`cp <source> <destination>`** – Copy files or directories. Example: `cp file1.txt file2.txt`
9. **`mv <source> <destination>`** – Move or rename files. Example: `mv oldname.txt newname.txt`
10. **`touch <filename>` –** Create a new empty file. Example: `touch newfile.txt`

File Viewing & Searching

11. **`cat <filename>`** – Display file contents.
12. **`less <filename>`** – View file content page by page (press q to quit).
13. **`head <filename>`** – Show the first 10 lines of a file.
14. **`tail <filename>`** – Show the last 10 lines of a file.
15. **`tail -f <filename>`** – Continuously monitor a file (useful for logs).
16. **`grep '<pattern>' <filename>`** – Search for text inside a file. Example: `grep "error" log.txt`
17. **`find <directory> -name "<pattern>"`** – Find files by name. Example: `find /home -name "*.txt"`
18. **`locate <filename>`** – Find a file's location (must run `updatedb` first).
19. **`wc <filename>`** – Count the lines, words, and characters in a file.
20. **`diff <file1> <file2>`** – Compare two files line by line.

Permissions & Ownership

21. **`chmod <permissions> <filename>`** – Change file permissions. Example: `chmod 755 script.sh`
22. **`chown <user>:<group> <filename>`** – Change file owner and group.
23. **`ls -l`** – Show detailed file permissions and ownership.
24. **`umask <mask>`** – Set default permissions for new files.

Process Management

25. **`ps aux`** – Show running processes.
26. **`top`** – Display real-time system resource usage.
27. **`htop`** – Better version of `top` (requires installation).
28. **`kill <PID>`** – Terminate a process by PID. Example: `kill 1234`
29. **`killall <process_name>`** – Kill all instances of a process.
30. **`pkill <pattern>`** – Kill processes matching a pattern. Example: `pkill firefox`

Networking Commands

31. **`ping <host>`** – Check connectivity to a host. Example: `ping google.com`
32. **`curl <URL>`** – Fetch a webpage or API response.
33. **`wget <URL>`** – Download a file from a URL.
34. **`netstat -tulnp`** – Show active network connections.
35. **`ifconfig`** / **`ip a`** – Show network interfaces and IP addresses.
36. **`nslookup <domain>`** – Get DNS information about a domain.
37. **`traceroute <host>`** – Show the path packets take to a host.
38. **`scp <file> <user>@<host>:<destination>`** – Securely copy files over SSH.

User Management

39. **`whoami`** – Show the current user.
40. **`who`** – Show logged-in users.
41. **`id <username>`** – Show user ID and group ID.
42. **`passwd`** – Change the current user's password.
43. **`useradd <username>`** – Create a new user.
44. **`usermod -aG <group> <username>`** – Add a user to a group.
45. **`groupadd <groupname>`** – Create a new user group.
46. **`deluser <username>`** – Remove a user.

Disk & System Info

47. **`df -h`** – Show disk usage in a human-readable format.
48. **`du -sh <directory>`** – Show the size of a directory.
49. **`free -m`** – Show memory usage in MB.
50. **`uname -a`** – Show Linux kernel and system information.

File Management & Manipulation

51. **`basename <path>`** – Extracts filename from a path. Example: `basename /home/user/file.txt` → `file.txt`
52. **`dirname <path>`** – Extracts directory from a path. Example: `dirname /home/user/file.txt` → `/home/user`
53. **`file <filename>`** – Identifies file type.
54. **`stat <filename>`** – Displays detailed file info (size, permissions, timestamps).
55. **`lsattr`** – Lists file attributes (like immutability).
56. **`chattr +i <file>`** – Makes a file immutable (can't be deleted/modified).
57. **`chattr -i <file>`** – Removes immutability.
58. **`split -b 10M <file> part_`** – Splits a large file into 10MB chunks.
59. **`xxd <file>`** – Displays file content in hexadecimal (useful for binary files).
60. **`shred -u <file>`** – Securely delete a file (overwrites it multiple times).

Text Processing Commands

61. **`cut -d':' -f1 /etc/passwd`** – Extracts the first column from a file (delimiter `:`).
62. **`awk '{print $1, $3}' file.txt`** – Prints the 1st and 3rd column of a text file.
63. **`sed 's/old/new/g' file.txt`** – Replaces all occurrences of "old" with "new".
64. **`tr 'a-z' 'A-Z'`** – Converts lowercase to uppercase.
65. **`sort file.txt`** – Sorts lines in a file.
66. **`uniq -c file.txt`** – Removes duplicate lines and shows counts.
67. **`tee file.txt`** – Writes input to both stdout & a file.
68. **`nl file.txt`** – Adds line numbers to a file.
69. **`rev file.txt`** – Reverses the text in a file.
70. **`wc -l file.txt`** – Counts the number of lines in a file.

Compression & Archiving

71. **`tar -cvf archive.tar folder/`** – Create a tar archive of a folder.
72. **`tar -xvf archive.tar`** – Extract a tar archive.
73. **`tar -czvf archive.tar.gz folder/`** – Create a gzip-compressed tar archive.
74. **`tar -xzvf archive.tar.gz`** – Extract a gzip-compressed tar archive.
75. **`zip -r archive.zip folder/`** – Create a zip file.
76. **`unzip archive.zip`** – Extract a zip file.
77. **`gzip file.txt`** – Compress a file using gzip.
78. **`gunzip file.txt.gz`** – Decompress a gzip file.
79. **`7z a archive.7z folder/`** – Compress using 7z (requires `p7zip`).
80. **`7z x archive.7z`** – Extract a 7z file.

System Monitoring & Performance

81. **`uptime`** – Shows system uptime and load averages.
82. **`vmstat`** – Displays memory, CPU, and process info.
83. **`iostat`** – Shows CPU and disk usage statistics.
84. **`mpstat`** – Displays per-CPU usage.
85. **`free -h`** – Displays memory usage in a human-readable format.
86. **`df -hT`** – Shows disk usage with filesystem type.
87. **`du -ah | sort -rh | head -10`** – Finds the top 10 largest files/directories.
88. **`watch -n 1 df -h`** – Updates disk usage stats every second.
89. **`iotop`** – Shows real-time disk I/O usage per process.
90. **`sar -u 5`** – Displays CPU usage every 5 seconds (requires `sysstat`).

Networking & Security

91. **`hostname -I`** – Shows the system's IP address.
92. **`dig google.com`** – Fetches DNS records for a domain.
93. **`whois google.com`** – Fetches domain registration details.
94. **`telnet <host> <port>`** – Checks if a port is open.
95. **`nc -zv <host> <port>`** – Another way to check open ports.
96. **`iptables -L`** – Lists firewall rules (for servers).
97. **`ufw status`** – Shows firewall status (`ufw` is for Ubuntu).
98. **`lsof -i :80`** – Finds processes using port 80.
99. **`tcpdump -i eth0`** – Captures network packets on eth0 (needs root).
100. **`nmap -A <IP>`** – Scans an IP for open ports & services (requires `nmap`).