Oranium Tech
SOFTWARE TRAINING & PLACEMENTS

TOP LINUX COMMANDS

Prepared By

Oranium Tech Team

For More Update
www.oraniumtech.com
+91-80723 63772

1. **ls**: Lists files and directories in the current directory.
   Example: `ls` or `ls -l` (to display in long format)

2. **cd**: Changes the current directory.
   Example: `cd /path/to/directory`

3. **pwd**: Prints the current working directory.
   Example: `pwd`

4. **mkdir**: Creates a new directory.
   Example: `mkdir new_directory_name`

5. **rm**: Removes files or directories.
   Example: `rm file.txt` (removes a file)
   `rm -r directory_name` (removes a directory)

6. **cp**: Copies files or directories.
   Example: `cp file.txt /path/to/destination/` (copies a file)
   `cp -r directory_name /path/to/destination/` (copies a directory)

7. **mv**: Moves or renames files or directories.
   Example: `mv file.txt /path/to/destination/` (moves a file)
   `mv old_name.txt new_name.txt` (renames a file)

8. **cat**: Concatenates and displays file content.
   Example: `cat file.txt`

9. **grep**: Searches for a specific pattern in files.
   Example: `grep "keyword" file.txt`

10. **chmod**: Changes file permissions.
    Example: `chmod 755 file.sh` (gives read, write, and execute permissions to the owner, and read/execute permissions to others)

11. **chown**: Changes file ownership.
    Example: `chown user:group file.txt` (changes the owner and group of the file)

12. **ps**: Displays information about running processes.

    Example: `ps aux`

13. **kill**: Sends a signal to terminate a process.

    Example: `kill PID` (PID is the Process ID)

14. **df**: Shows disk space usage of file systems.

    Example: `df -h` (displays usage in human-readable format)

15. **free**: Displays the amount of free and used memory in the system.

    Example: `free -h` (displays usage in human-readable format)

16. **top**: Displays real-time information about system processes.

    Example: `top`

17. **wget**: Downloads files from the web.

    Example: `wget https://example.com/file.txt`

18. **curl**: Transfers data to or from a server (supports various protocols).

    Example: `curl -X GET https://api.example.com/data`

19. **tar**: Archives files and directories.

    Example: `tar -czvf archive.tar.gz file1 file2` (creates a gzip-compressed tar archive)

20. **ssh**: Connects securely to a remote server via SSH.

    Example: `ssh user@remote_server`

21. **ping**: Tests network connectivity by sending ICMP echo requests to a host.

    Example: `ping google.com`

22. **traceroute**: Displays the route that packets take to reach a destination host.

    Example: `traceroute google.com`

23. **netstat**: Shows network statistics and connections.

    Example: `netstat -tuln` (displays TCP/UDP connections)

24. **ifconfig / ip**: Configures and displays network interfaces.

    Example: `ifconfig` or `ip addr`

25. **route**: Manages network routing tables.

    Example: `route -n`

26. **ssh-keygen**: Generates SSH key pairs for secure authentication.

    Example: `ssh-keygen -t rsa`

27. **scp**: Securely copies files between local and remote systems via SSH.

    Example: `scp local_file.txt user@remote_host:/path/to/destination/`

28. **find**: Searches for files and directories in a directory hierarchy.

    Example: `find /path/to/search -name "file.txt"`

29. **du**: Shows disk usage of files and directories.

    Example: `du -h /path/to/directory` (displays usage in human-readable format)

30. **cron**: Schedules tasks to run at specific times or intervals.

    Example: `crontab -e` (edit the cron table)

31. **systemctl**: Manages system services (systemd-based distributions).

    Example: `systemctl start service_name` (starts a service)

32. **journalctl**: Views system log (journal) messages.

    Example: `journalctl -xe` (displays error messages)

33. **wget**: Downloads files from the web.

    Example: `wget https://example.com/file.txt`

34. **curl**: Transfers data to or from a server (supports various protocols).

    Example: `curl -X GET https://api.example.com/data`

35. **lsof**: Lists open files and processes.

    Example: `lsof -i :80` (lists processes using port 80)

36. **scp**: Securely copies files between local and remote systems via SSH.

    Example: `scp local_file.txt user@remote_host:/path/to/destination/`

37. **tail**: Displays the last few lines of a file (useful for log files).

Example: `tail -n 10 file.log` (displays the last 10 lines)

38. **head**: Displays the first few lines of a file.

Example: `head -n 5 file.txt` (displays the first 5 lines)

39. **sed**: Stream editor for text manipulation.

Example: `sed 's/foo/bar/g' file.txt` (replaces all occurrences of 'foo' with 'bar' in the file)

40. **awk**: Text processing tool for extracting and manipulating data.

Example: `awk '{print $1}' file.txt` (prints the first column of the file)

41. **grep**: Searches for a pattern in files.

Example: `grep "pattern" file.txt` (searches for "pattern" in file.txt)

42. **awk**: A versatile text processing tool for data extraction and manipulation.

Example: `awk '{print $2}' file.txt` (prints the second column of the file)

43. **sort**: Sorts lines of text.

Example: `sort file.txt` (sorts the lines in file.txt alphabetically)

44. **uniq**: Filters out duplicate lines from a sorted file.

Example: `sort file.txt | uniq` (removes duplicate lines from file.txt)

45. **diff**: Compares files and shows the differences.

Example: `diff file1.txt file2.txt` (compares file1.txt and file2.txt)

46. **tar**: Archives files or directories.

Example: `tar -czvf archive.tar.gz file1 file2` (creates a gzip-compressed tar archive)

47. **gzip**: Compresses files.

Example: `gzip file.txt` (creates file.txt.gz)

48. **gunzip**: Decompresses files compressed with gzip.

Example: `gunzip file.txt.gz` (restores file.txt)

49. **crontab**: Schedules tasks to run at specific times or intervals.

Example: `crontab -e` (edit the cron table)

50. **rsync**: Synchronizes files and directories between local and remote systems.

Example: `rsync -avz /local/path/ user@remote:/remote/path/`

51. **curl**: Transfers data to or from a server (supports various protocols).

Example: `curl -X POST -d "data" https://api.example.com/endpoint` (sends a POST request with data)

52. **dig**: DNS lookup utility to query DNS servers.

Example: `dig example.com` (performs a DNS lookup for example.com)

53. **hostname**: Displays or sets the system's hostname.

Example: `hostname` (shows the current hostname)

54. **history**: Displays the command history of the current user.

Example: `history`

55. **who**: Shows the users currently logged in.

Example: `who`

56. **uptime**: Displays the system's uptime and load average.

Example: `uptime`

57. **nc**: Netcat, used for network connections and troubleshooting.

Example: `nc -vz example.com 80` (checks if port 80 is open on example.com)

58. **jq**: A command-line JSON processor for parsing and manipulating JSON data.

Example: `cat data.json | jq '.key'` (extracts the value of 'key' from data.json)

59. **htop**: An interactive process viewer and system monitor.

Example: `htop`

60. **find**: Searches for files and directories in a directory hierarchy.

Example: `find /path/to/search -name "file.txt"`

61. **screen**: Creates multiple virtual terminal sessions within a single terminal window.

Example: `screen` (starts a new screen session)

Press `Ctrl+A` and then `Ctrl+D` (detaches from the current screen session)

62. **tmux**: A terminal multiplexer similar to screen, allowing multiple terminal sessions.

Example: `tmux new-session` (starts a new tmux session)

Press `Ctrl+B` and then `D` (detaches from the current tmux session)

63. **ncdu**: A disk usage analyzer with a nice text-based interface.

Example: `ncdu /path/to/directory` (analyzes disk usage in the specified directory)

64. **iptables**: A powerful firewall management tool.

Example: `iptables -A INPUT -p tcp --dport 22 -j ACCEPT` (allows SSH connections)

65. **ss**: Displays socket statistics, showing details about active connections.

Example: `ss -tuln` (displays TCP/UDP socket statistics)

66. **route**: Manages network routing tables.

Example: `route -n` (displays the routing table)

67. **scp**: Securely copies files between local and remote systems via SSH.

Example: `scp local_file.txt user@remote_host:/path/to/destination/`

68. **lftp**: A sophisticated file transfer program supporting various protocols.

Example: `lftp sftp://user@hostname` (connects to a remote host using SFTP)

69. **lshw**: Lists hardware details of the system.

Example: `lshw` (displays detailed hardware information)

70. **lscpu**: Shows information about the CPU architecture.

Example: `lscpu`

71. **blkid**: Displays information about block devices (partitions, filesystems).

Example: `blkid`

72. **fdisk**: Manages disk partitions on the system.

Example: `fdisk -l` (lists disk partitions)

73. **du**: Shows disk usage of files and directories.

Example: `du -h /path/to/directory` (displays usage in human-readable format)

74. **dig**: DNS lookup utility to query DNS servers.

Example: `dig example.com` (performs a DNS lookup for example.com)

75. **nc**: Netcat, used for network connections and troubleshooting.

Example: `nc -vz example.com 80` (checks if port 80 is open on example.com)

76. **jq**: A command-line JSON processor for parsing and manipulating JSON data.

Example: `cat data.json | jq '.key'` (extracts the value of 'key' from data.json)

77. **htop**: An interactive process viewer and system monitor.

Example: `htop`

78. **find**: Searches for files and directories in a directory hierarchy.

Example: `find /path/to/search -name "file.txt"`

79. **watch**: Repeatedly executes a command and displays the output.

Example: `watch df -h` (displays disk usage in real-time)

80. **dig**: DNS lookup utility to query DNS servers.

Example: `dig example.com` (performs a DNS lookup for example.com)