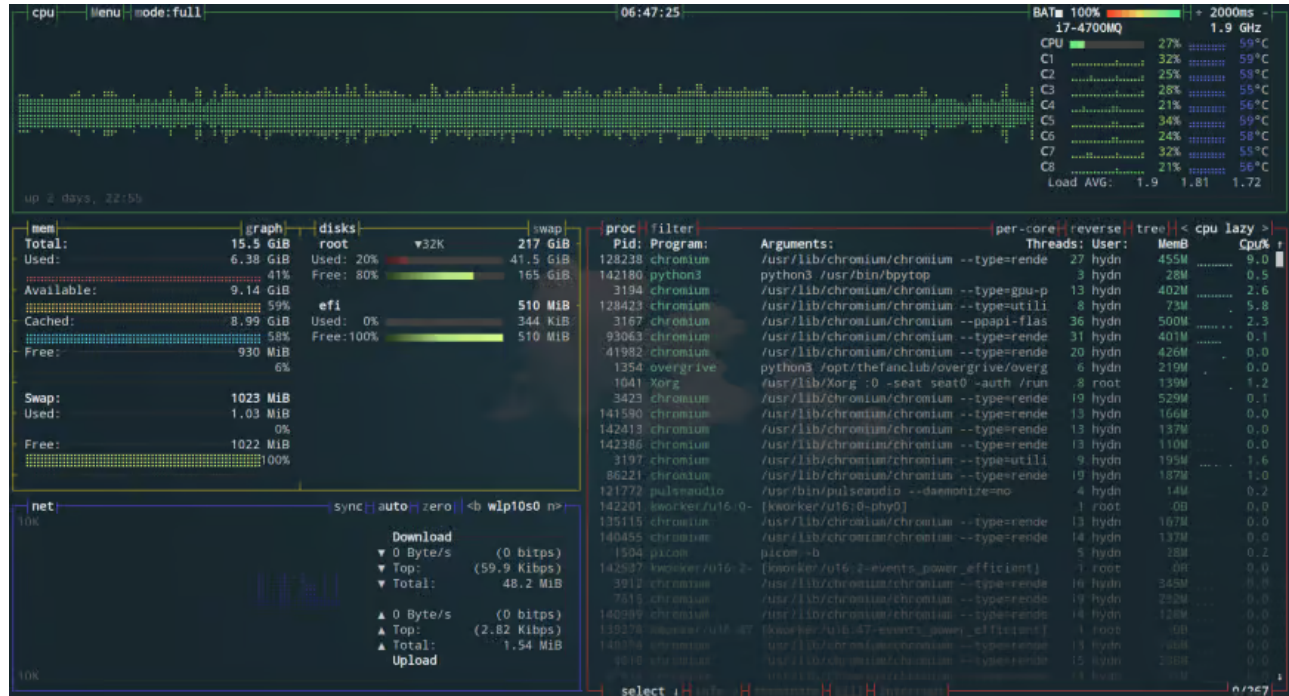


# 100+ Linux commands - [LinuxBlog.io](https://linuxblog.io) by Hayden James

Here I've **listed over 100 Linux commands**. There are even more commands if we include downloadable software and scripts, such as [btop](#) (pictured below). However, we will only cover Unix commands used on Linux by sysadmins and power users.



Pictured: [btop](#) - Also see: [network command-line tools available](#).

1. [ip](#) - from Iproute2, a collection of utilities for controlling TCP/IP networking and traffic control in Linux.
2. [ls](#) - list directory contents.
3. [df](#) - display disk space usage.
4. [du](#) - estimate file space usage.
5. [free](#) - display memory usage.
6. [scp](#) - securely Copy Files Using SCP, with examples.
7. [find](#) - locates files based on some user-specified criteria.
8. [ncdu](#) - a disk utility for Unix systems.
9. [pstree](#) - display a tree of processes.
10. [last](#) - show a listing of last logged-in users.
11. [w](#) - show a list of currently logged-in user sessions.
12. [grep](#) - Search a file for a pattern of characters, then display all matching lines.
13. [uptime](#) - shows system uptime and load average.
14. [top](#) - shows an overall system view.

15. [vmstat](#) – shows system memory, processes, interrupts, paging, block I/O, and CPU info.
16. [htop](#) – interactive process viewer and manager.
17. [dstat](#) – view processes, memory, paging, I/O, CPU, etc., in real-time. All-in-one for vmstat, iostat, netstat, and ifstat.
18. [iftop](#) – network traffic viewer.
19. [nethogs](#) – network traffic analyzer.
19. [nethogs](#) – network traffic analyzer.
20. [iotop](#) – interactive I/O viewer. Get an overview of storage r/w activity.
21. [iostat](#) – for storage I/O statistics.
22. [netstat](#) – for network statistics.
23. [ss](#) – utility to investigate sockets.
24. [atop](#) – For Linux server performance analysis.
25. [Glances](#) and [nmon](#) – htop and top Alternatives:
26. [ssh](#) – secure command-line access to remote Linux systems.
27. [sudo](#) – execute commands with administrative privilege.
28. [cd](#) – directory navigation.
29. [pwd](#) – shows your current directory location.
30. `[cp]` (<https://linuxblog.io/cp-command-linux/>) – copying files and folders.
31. [mv](#) – moving files and folders.
32. [rm](#) – removing files and folders.
33. [mkdir](#) – create or make new directories.
34. [touch](#) – used to update the access date and modification date of a computer file or directory.
35. [man](#) – for reading system reference manuals.
36. [apropos](#) – Search man page names and descriptions.
37. [rsync](#) – remote file transfers and syncing.
38. [tar](#) – an archiving utility.
39. [gzip](#) – file compression and decompression.
40. [bzip2](#) – similar to gzip. It uses a different compression algorithm.
41. [zip](#) – for packaging and compressing (to archive) files.
42. [locate](#) – search files in Linux.
43. [ps](#) – information about the currently running processes.
44. Making use of Bash scripts. Example: `./bashscript.sh`
45. [cron](#) – set up scheduled tasks to run.
46. [nmcli](#) – command-line tool for controlling [NetworkManager](#).
47. [ping](#) – send ICMP ECHO\_REQUEST to network hosts.

48. `traceroute` – check the route packets take to a specified host.
49. `mtr` – network diagnostic tool.
50. `nslookup` – query Internet name servers (NS) interactively.
51. `host` – perform DNS lookups in Linux.
52. `dig` – DNS lookup utility.
53. `wget` – retrieve files over HTTP, HTTPS, FTP, and FTPS.
54. `curl` – transferring data using various network protocols. (supports more protocols than wget)
55. `[dd]` (<https://linuxblog.io/web-host-doesnt-want-read-benchmark-vps/>) – convert and copy files.
56. `fdisk` – manipulate the disk partition table.
57. `parted` – for creating and manipulating partition tables.
58. `blkid` – command-line utility to locate/print block device attributes.
59. `mkfs` – build a Linux file system.
60. `fsck` – tool for checking the consistency of a file system.
61. `whois` – client for the whois directory service.
62. `nc` – command-line networking utility. (Also, see [60 Linux Networking commands and scripts.](#))
63. `umask` – set file mode creation mask.
64. `chmod` – change the access permissions of file system objects.
65. `chown` – change file owner and group.
66. `chroot` – run command or interactive shell with a special root directory.
67. `useradd` – create a new user or update default new user information.
68. `userdel` – used to delete a user account and all related files.
69. `usermod` – used to modify or change any attributes of an existing user account.
70. `vi` – text editor.
71. `cat` – display file contents.
72. `tac` – output file contents, in reverse.
73. `more` – display file contents one screen/page at a time.
74. `less` – similar to the more command with additional features.
75. `tail` – used to display the tail end of a text file or piped data.
76. `dmesg` – prints the message buffer of the kernel ring.
77. `journalctl` – query the systemd journal.
78. `kill` – terminate a process.
79. `killall` – Sends a kill signal to all instances of a process by name.
80. `sleep` – suspends program execution for a specified time.

81. `wait` – Suspend script execution until all jobs running in the background have been terminated.
82. `nohup` – Run Commands in the Background.
83. `screen` – hold a session open on a remote server. (also a full-screen window manager)
84. `tmux` – a terminal multiplexer.
85. `passwd` – change a user's password.
86. `chpassword` – allows users to change the password for various user accounts.
87. `mount` / `umount` – provides access to an entire filesystem in one directory.
88. `systemctl` – central management tool for controlling the init system.
89. `clear` – clears the screen of the terminal.
90. `env` -Run a command in a modified environment.
91. [cheat](#) – allows you to create and view interactive cheatsheets on the command line.”
92. [tldr](#) – Collaborative cheatsheets for console commands.
93. [bashtop](#) – the ‘cool’ top alternative.
94. [nload](#) – a super simple, command-line network interface monitoring tool.
95. `history` – used to view the previously executed commands.
96. [sar](#) – collects, reports, and saves system activity information, including CPU, memory, disk, and network usage.
97. `lsdf` : List open files and the corresponding processes. This command is invaluable for troubleshooting and understanding what files are used by processes.
98. `awk` : A powerful pattern scanning and processing language. It's used to manipulate data and generate reports.
99. `sed` : A stream editor used to perform basic text transformations on an input stream (a file or input from a pipeline).
100. `watch` : Execute a program periodically, showing output fullscreen. It's useful for monitoring commands or scripts that update regularly.
101. `crontab` : Schedule commands to run periodically at fixed times, dates, or intervals with the cron daemon.
102. `alias` : Create an alias for Linux commands. It's a shorthand way to customize and streamline your command-line experience.
103. `lspci` : List all PCI devices. This is particularly useful for diagnosing hardware and system problems related to PCI devices.
104. `lsusb` : List USB devices. Similar to `lspci` , but for USB hardware connected to the system.

105. `ln` : Make links between files. By creating a link, you can access the linked file by more than one path.
106. `diff` : Compare files line by line. It's particularly useful for comparing text files such as scripts or configuration files to find differences.

PLUS: [60 Linux Networking commands and scripts](#). | Source: [LinuxBlog.io](#)

[Browse our 200+ articles>](#)