



Linux Commands Cheat Sheet {with Free Downloadable PDF}

November 2, 2023

COMMANDS LINUX

[Home](#) » [SysAdmin](#) » Linux Commands Cheat Sheet {with Free Downloadable PDF} Contents

Introduction

Linux commands may seem intimidating at first glance if you do not use the terminal often. There are many commands for performing operations and processes on [Linux systems](#).

Whether you are new to Linux or an experienced user, having a list of common commands close at hand is helpful.

In this tutorial, you will find commonly used Linux commands and a downloadable cheat sheet with syntax and examples.





Important: Depending on your system setup, some of the commands below may require invoking **sudo** to be executed.

Linux Commands PDF

If you prefer having all the commands on a one-page reference sheet, we created a helpful **Linux command line cheat sheet**. You can save the **list of Linux commands** in PDF format by clicking the **Download Linux Cheat Sheet** button below.

Download Linux Commands Cheat Sheet PDF



Linux Commands List

The commands from the downloadable cheat sheet are listed below. If you're looking for the must-know commands only and a shorter cheat sheet, check out [Linux Commands All Users Should Know](#).

Hardware Information Commands

Hardware information commands help provide insight into various hardware devices on the system. Use these commands to check [hardware](#) information and see hardware device statuses.

Command	Description
<code>lscpu</code>	See CPU information.
<code>lsblk</code>	See information about block devices .
<code>lspci -tv</code>	Show PCI devices (graphics card, network card, etc.) in a tree-like diagram.
<code>lsusb -tv</code>	Display USB devices in a tree-like diagram.

Command	Description
<code>lshw</code>	List hardware configuration information.
<code>cat /proc/cpuinfo</code>	Show detailed CPU information.
<code>cat /proc/meminfo</code>	View detailed system memory information.
<code>cat /proc/mounts</code>	See mounted file systems .
<code>free -h</code>	Display free and used memory .
<code>sudo dmidecode</code>	Show hardware information from the BIOS .
<code>hdparm -i /dev/[device_name]</code>	Display disk data information.
<code>hdparm -tT /dev/[device_name]</code>	Conduct a read speed test on the device/disk.
<code>badblocks -s /dev/[device_name]</code>	Test for unreadable blocks on the device/disk.
<code>fsck /dev/[device_name]</code>	Run a disk check on an unmounted disk or partition.

Searching Commands

Linux offers various commands to search for files, directories, and text. Use these commands to search for files and directories on the system and filter the search using various patterns.

Command	Description
<code>find [path] -name [search_pattern]</code>	Find files and directories that match the specified pattern in a specified location.
<code>find [path] -size [+100M]</code>	See files and directories larger than a specified size in a directory.
<code>grep [search_pattern] [file_name]</code>	Search for a specific pattern in a file with grep .
<code>grep -r [search_pattern] [directory_name]</code>	Recursively search for a pattern in a directory.
<code>locate [name]</code>	Locate all files and directories related to a particular name.
<code>which [command]</code>	Search the command path in the <code>\$PATH</code> environment variable.
<code>whereis [command]</code>	Use the whereis command to find the source, binary, and manual page for a command.
<code>awk '[search_pattern] {print \$0}' [file_name]</code>	Print all lines matching a pattern in a file. See also the gawk command , the GNU version of <code>awk</code> .
<code>sed 's/[old_text]/[new_text]/' [file_name]</code>	Find and replace text in a specified file.



Note: Some commands are not recommended to use. Learn about them in our list of [dangerous Linux commands](#).

File Commands

File commands help with file and directory management on the system. Create, delete, move, and modify files and directories from the terminal using the commands in the following table.

Command	Description
<code>mkdir [directory_name]</code>	Create a new directory .
<code>rm [file_name]</code>	Remove a file .
<code>rm -r [directory_name]</code>	Remove a directory recursively.
<code>rm -rf [directory_name]</code>	Recursively remove a directory without requiring confirmation.
<code>cp [source_file] [destination_file]</code>	Copy the contents of one file to another file using the cp command .
<code>cp -r [source_directory] [destination_directory]</code>	Recursively copy a directory to a second directory.
<code>mv [source_file] [destination_file]</code>	Move or rename files or directories.
<code>ln -s [path]/[file_name] [link_name]</code>	Create a symbolic link to a file.
<code>touch [file_name]</code>	Create a new file using touch .
<code>cat [file_name]</code>	Show the contents of a file .
<code>cat [source_file] >> [destination_file]</code>	Append file contents to another file.
<code>head [file_name]</code>	Show the first ten lines of a file.
<code>tail [file_name]</code>	Show the last ten lines of a file with the tail command .
<code>more [file_name]</code>	Display contents of a file page by page.
<code>less [file_name]</code>	Show the contents of a file with navigation using the less command .
<code>nano [file_name]</code>	Open or create a file using the nano text editor .
<code>vi [file_name]</code> <code>vim [file_name]</code>	Open or create a file using the Vi/Vim text editor .
<code>gpg -c [file_name]</code>	Encrypt a file.
<code>gpg [file_name].gpg</code>	Decrypt an encrypted .gpg file.

Command	Description
<code>wc -w [file_name]</code>	Show the number of words, lines, and bytes in a file using wc .
<code>ls xargs wc</code>	List the number of lines/words/characters in each file in a directory with the xargs command .
<code>cut -d [delimiter] [file_name]</code>	Cut a section of a file and print the result to standard output.
<code>[data] cut -d [delimiter]</code>	Cut a section of piped data and print the result to standard output.
<code>shred -u [file_name]</code>	Overwrite a file to prevent its recovery, then delete it.
<code>diff [first_file] [second_file]</code>	Compare two files and display differences.
<code>source [file_name]</code>	Read and execute the file content in the current shell.
<code>[command] tee [file_name] >/dev/null</code>	Store the command output in a file and skip the terminal output.



Note: Want to read more about file creation? Check out our article about [how to create a file in Linux using the command line](#).

And if you want to find out how to determine the type of a file and its data, read the [Linux file command](#) article.

Directory Navigation Commands

Directory navigation commands provide shortcuts to navigate to the desired location quickly. Below are several crucial shortcuts to remember when navigating directories in Linux through the terminal.

Directory Navigation Commands	Description
<code>ls</code>	List files and directories in the current directory.
<code>ls -a</code>	List all files and directories in the current directory (shows hidden files).
<code>ls -l</code>	List files and directories in long format.
<code>pwd</code>	Show the directory you are currently working in.
<code>cd</code> <code>cd ~</code>	Change directory to <code>\$HOME</code> .
<code>cd ..</code>	Move up one directory level.
<code>cd -</code>	Change to the previous directory.
<code>cd [directory_path]</code>	Change location to a specified directory.

Directory Navigation Commands	Description
<code>dirs</code>	Show current directory stack.

File Compression Commands

Archive and compress files to save space on your system and organize your data. There are various compression formats available on Linux, and the table below shows the most commonly used file compression commands.

Command	Description
<code>tar cf [archive.tar] [file/directory]</code>	Archive an existing file or directory.
<code>tar xf [archive.tar]</code>	Extract an archived file.
<code>tar czf [archive.tar.gz]</code>	Create a <code>.gz</code> compressed tar archive.
<code>gzip [file_name]</code> <code>gunzip [file_name.gz]</code>	Compress or decompress <code>.gz</code> files.
<code>bzip2 [file_name]</code> <code>bunzip2 [file_name.bz2]</code>	Compress or decompress <code>.bz2</code> files.



Note: For a more comprehensive overview of how to use `tar` refer to our guide [tar Command in Linux With Examples](#). See also how to [unzip tar.bz2 files](#).

File Transfer Commands

Depending on the transfer method and protocol, Linux has various file transfer commands. Use these commands to quickly transfer files between two remote machines or to download files from web servers.

Command	Description
<code>scp [source_file] [user]@[remote_host]:[destination_path]</code>	Copy a file to a server directory securely using the Linux scp command .
<code>rsync -a [source_directory] [user]@[remote_host]:[destination_directory]</code>	Synchronize the contents of a directory with a backup directory using the rsync command .
<code>wget [link]</code>	Download files from FTP or web servers via the wget command .
<code>curl -O [link]</code>	Transfer data to or from a server with various protocols using the curl command .
<code>ftp [remote_host]</code>	Transfer files between local and remote systems interactively using FTP.
<code>sftp [user]@[remote_host]</code>	Securely transfer between local and remote hosts

Command	Description
	using SFTP .

User and Group Commands

Linux is a multiuser system, and there are various commands to add, modify, remove, and manage users and groups. The table below outlines the critical user and group commands.

Command	Description
<code>id</code>	See details about the active users.
<code>last</code>	Show the last system logins.
<code>who</code>	Display who is currently logged into the system with the who command .
<code>w</code>	Show which users are logged in and their activity with the w command .
<code>finger [user_name]</code>	Show user information.
<code>sudo useradd [user_name]</code>	Create a new user account.
<code>sudo adduser [user_name]</code>	Create a new user account through the adduser command interface.
<code>sudo userdel [user_name]</code>	Delete a user account.
<code>sudo usermod -aG [group_name] [user_name]</code>	Modify user information (add a user to a group) .
<code>passwd</code> <code>sudo passwd [user_name]</code>	Change the current user's password or another user's password with the passwd command .
<code>sudo groupadd [group_name]</code>	Add a new group.
<code>sudo groupdel [group_name]</code>	Delete a group.
<code>sudo groupmod -n [new_name] [old_name]</code>	Modify a user group (change group name).
<code>sudo [command]</code>	Temporarily elevate user privileges to superuser or root using the sudo command .
<code>su - [user_name]</code>	Switch the user account or become a superuser.
<code>chgrp [group_name] [file/directory]</code>	Change file or directory group .

Package Installation Commands

Specific package manager commands vary between different [Linux distributions](#). Choose the commands that match your specific distribution.

Debian and Ubuntu-Based Distributions

Debian and Ubuntu-based distributions use the `apt-get`, `apt`, and `dpkg` commands to manage packages. Below are crucial commands necessary to install packages on these systems.

Command	Description
<code>sudo apt-get install [package_name]</code>	Install an APT package using the apt-get package utility .
<code>sudo apt install [package_name]</code>	Install an APT package using a newer APT package manager .
<code>apt search [keyword]</code>	Search for a package in the APT repositories.
<code>apt list</code>	List packages installed with APT.
<code>apt show [package_name]</code>	Show information about a package.
<code>sudo dpkg -i [package_name.deb]</code>	Install a <i>.deb</i> package with the Debian package manager (dpkg command).
<code>sudo dpkg -l</code>	List packages installed with dpkg.

Red Hat-Based Distributions (CentOS, Fedora)

Red Hat-based distributions, including CentOS and Fedora, use the **yum**, **dnf**, and **rpm** commands for package installation. The table below provides the syntax for each option.

Command	Description
<code>sudo yum install [package_name]</code>	Install a package using the YUM package manager.
<code>yum search [keyword]</code>	Find a package in the YUM repositories based on the provided keyword.
<code>yum list installed</code>	List all packages installed with YUM.
<code>yum info [package_name]</code>	Show package information for a package.
<code>sudo dnf install [package_name]</code>	Install a package using the DNF package manager.
<code>sudo rpm -i [package_name.rpm]</code>	Install a <i>.rpm package</i> from a local file.



Note: See our detailed comparison between [RPM vs. YUM](#).

Universal Linux Packages

Universal Linux packages are distribution-independent package collections. The two most popular options that come preinstalled on many systems are **snap** and **flatpak** commands. Alternatively, most [programs](#) offer the option to install from source. The table below shows the syntax for each listed option.

Command	Description
<code>tar zxvf [file_name.tar.gz]</code> <code>cd [extracted_directory]</code>	Install software from source code .

Command	Description
<code>./configure make</code> <code>make install</code>	
<code>sudo snap install [package_name]</code>	Install a Snap package.
<code>sudo snap find [keyword]</code>	Search for a package in the Snap store.
<code>sudo snap list</code>	List installed Snap packages.
<code>flatpak install [package_name]</code>	Install a Flatpak package.
<code>flatpak search [keyword]</code>	Search for a Flatpak application in repositories.
<code>flatpak list</code>	List installed Flatpack packages.



Note: Check out our comparison between three popular distribution-independent package formats: [Flatpak vs. Snap vs. Appimage](#).

Process Related Commands

Every application and command in Linux creates a process. The table below shows various process-related commands to help manage Linux processes.

Command	Description
<code>ps</code>	List active processes .
<code>pstree</code>	Show processes in a tree-like diagram.
<code>pmap</code>	Display a memory usage map of processes.
<code>top</code>	See all running processes .
<code>htop</code>	Interactive and colorful process viewer.
<code>kill [process_id]</code>	Terminate a Linux process under a given ID.
<code>pkill [process_name]</code>	Terminate a process under a specific name.
<code>killall [label]</code>	Terminate all processes with a given label.
<code>prgrep [keyword]</code>	List processes based on the provided keyword.
<code>pidof [process_name]</code>	Show the PID of a process.
<code>bg</code>	List and resume stopped jobs in the background.
<code>fg</code>	Bring the most recently suspended job to the foreground.
<code>fg [job]</code>	Bring a particular job to the foreground.

Command	Description
<code>lsof</code>	List files opened by running processes with lsof command .
<code>trap "[commands]" [signal]</code>	Catch a system error signal in a shell script. Executes provided commands when the signal is caught.
<code>wait</code>	Pause the terminal or a Bash script until a running process is completed.
<code>nohup [command] &</code>	Run a Linux process in the background.



Note: If you want to learn more about shell jobs, how to terminate jobs or keep them running after you log off, check out our article on [how to use disown command](#).

System Management and Information Commands

Use the terminal to manage the system directly. The commands show how to view basic system information, change options, and reboot or restart the system.

Command	Description
<code>uname -r</code>	Show system information via uname command .
<code>uname -a</code>	See kernel release information .
<code>uptime</code>	Display system uptime , including the load average.
<code>hostname</code>	View system hostname.
<code>hostname -i</code>	Show the IP address of the system.
<code>last reboot</code>	List system reboot history.
<code>date</code>	See current time and date .
<code>timedatectl</code>	Query and change the system clock .
<code>cal</code>	Show current calendar (month and day).
<code>w</code>	List logged-in users .
<code>whoami</code>	See which user you are using .
<code>finger [user_name]</code>	Show information about a particular user.
<code>ulimit [flags] [limit]</code>	View or limit system resource amounts.
<code>shutdown [hh:mm]</code>	Schedule a system shutdown .
<code>shutdown now</code>	Shut down the system immediately.

Command	Description
<code>modprobe [module_name]</code>	Add a new kernel module.
<code>dmesg</code>	Show bootup messages.

Disk Usage Commands

Disk usage commands provide insight into disk space status. You can use the `df` and `du` commands to [check disk space in Linux](#). Common disk usage commands are outlined in the table below.

Command	Description
<code>df -h</code>	Check free and used space on mounted systems.
<code>df -i</code>	Show free inodes on mounted file systems.
<code>fdisk -l</code>	Display disk partitions, sizes, and types with the command.
<code>du -ah</code>	See disk usage for all files and directories .
<code>du -sh</code>	Show disk usage of the current directory.
<code>mount</code>	Show currently mounted file systems.
<code>findmnt</code>	Display target mount point for all file systems.
<code>mount [device_path] [mount_point]</code>	Mount a device.

SSH Login Commands

[SSH](#) commands enable connecting to a remote host using the SSH protocol. Other commands use this protocol for copying and transferring files between two systems.

See the table below for common SSH commands. For a detailed explanation of SSH Linux Commands, refer to our [SSH Commands in Linux](#) tutorial.

Command	Description
<code>ssh [user_name]@[remote_host]</code>	Connect to a remote host as a user via SSH.
<code>ssh [host]</code>	Securely connect to a host via SSH default port 22 .
<code>ssh -p [port] [user_name]@[remote_host]</code>	Connect to the host using a particular port.
<code>ssh-keygen</code>	Generate SSH key pairs.
<code>sudo service sshd start</code>	Start SSH server daemon .
<code>scp [file_name] [user_name]@[remote_host]:[remote_path]</code>	Securely copy files between local and remote systems via SSH.
<code>sftp [user_name]@[remote_host]</code>	Interactive file transfer over encrypted SSH session using SFTP protocol.

Command	Description
<code>telnet [host]</code>	Connect to the host via Telnet default port 23.

File Permission Commands

File permission commands control user permissions over files and directories. Use these commands to change and manage the owner, group, and user permissions on the system.

Command	Description
<code>chmod 777 [file_name]</code>	Assign read, write, and execute file permission to everyone (<code>rw-rw-rw-</code>).
<code>chmod 755 [file_name]</code>	Give read, write, and execute permission to owner, and read and execute permission to group and others (<code>rw-r-xr-x</code>).
<code>chmod 766 [file_name]</code>	Assign full permission to the owner, and read and write permission to the group and others (<code>rw-rw-rw-</code>).
<code>chown [user_name] [file_name]</code>	Change the ownership of a file with chown command .
<code>chown [user_name]:[group_name] [file_name]</code>	Change the owner and group ownership of a file.



Note: To recursively change file permissions inside multiple directories, see our [chmod recursive](#) guide.

Network Commands

The table below shows a brief list of common Linux network commands. For a more comprehensive list, check out our [Linux Network Commands Cheat Sheet](#).

Description	Command
<code>ip addr show</code>	List IP addresses and network interfaces.
<code>ip address add [IP_address]</code>	Assign an IP address to interface <code>eth0</code> .
<code>ifconfig</code>	Display IP addresses of all network interfaces.
<code>ping [remote_host]</code>	Ping remote host.
<code>netstat -pnltu</code>	See active (listening) ports with the netstat command .
<code>netstat -tuln</code>	Show TCP and UDP ports and their programs.
<code>whois [domain_name]</code>	Display more information about a domain .

Description	Command
<code>dig [domain_name]</code>	Show DNS information about a domain using the dig command .
<code>dig -x [domain_name]</code>	Do a reverse DNS lookup on the domain.
<code>dig -x [IP_address]</code>	Do a reverse DNS lookup of an IP address.
<code>host [domain_name]</code>	Perform an IP lookup for a domain.
<code>hostname -I</code>	Show the local IP address.
<code>nslookup [domain_name]</code>	Receive information about an internet domain .

Variable Commands

Shell variables are a convenient way to store critical data in a reusable format. Below are commands to create, view, and remove variables in the shell.

Command	Description
<code>let "[variable_name]=[value]"</code>	Assign an integer value to a variable.
<code>export [variable_name]</code>	Export a Bash variable .
<code>declare [variable-name]= "[value]"</code>	Declare a Bash variable .
<code>set</code>	List the names of all the shell variables and functions .
<code>unset [variable_name]</code>	Remove an environment variable.
<code>echo \${variable-name}</code>	Display the value of a variable.

Shell Command Management

Several Linux commands enable managing command execution flow. Schedule commands or create shorter names for lengthy commands.

Command	Description
<code>alias [alias-name]='[command]'</code>	Create an alias for a command.
<code>watch -n [interval-in-seconds] [command]</code>	Set a custom interval to run a user-defined command.
<code>sleep [time-interval] && [command]</code>	Postpone the execution of a command.
<code>at [hh:mm]</code>	Create a job to be executed at a certain time (Ctrl+D to exit prompt after you type in the command).
<code>man [command]</code>	Display a built-in manual for a command.

Command	Description
<code>history</code>	Print the command history used in the terminal.



Note: Also check out our [Bash commands cheat sheet](#).

Linux Shell Keyboard Shortcuts

The Linux shell contains several keyboard shortcuts that are important to know. The commands help control the terminal behavior in specific situations. The table below outlines the most important shortcuts.

Shortcut	Description
Ctrl + C	Kill process running in the terminal.
Ctrl + Z	Stop the current process. The process can be resumed in the foreground with fg or in the background with bg .
Ctrl + W	Cut one word before the cursor and add it to the clipboard.
Ctrl + U	Cut part of the line before the cursor and add it to the clipboard.
Ctrl + K	Cut part of the line after the cursor and add it to the clipboard.
Ctrl + Y	Paste from clipboard.
Ctrl + R	Recall the last command that matches the provided characters.
Ctrl + O	Run the previously recalled command.
Ctrl + G	Exit command history without running a command.
<code>clear</code>	Clear the terminal screen .
<code>!!</code>	Run the last command again.
<code>exit</code>	Log out of the current session.

Conclusion

The more you use Linux commands, the better you will get at remembering them. Do not stress about memorizing their syntax; use our cheat sheet.

Whenever in doubt, refer to this helpful guide for the most common Linux commands.

Was this article helpful?

Yes

No

Milica Dancuk

Milica Dancuk is a technical writer at phoenixNAP with a passion for programming. With a background in Electrical Engineering and Computing, coupled with her teaching experience, she excels at simplifying complex technical concepts in her writing.

Next you should read

[SysAdmin, Web Servers](#)

How to Start, Stop, and Restart Services in Linux

March 27, 2024

In most modern Linux operating systems, managing a service is quite simple....

READ MORE

[SysAdmin, Web Servers](#)

How to Check Memory Usage in Linux

March 28, 2024

In this tutorial, learn the five most commonly used commands to check...

READ MORE

[SysAdmin, Web Servers](#)

How to Remove (Delete) a File or

Directory in Linux

November 2, 2023

This article lists the most commonly used commands and tools to remove...

READ MORE

Security, SysAdmin

How to Use the su Command in Linux with Examples

April 16, 2024


Learn how to use the su command with practical examples and explanations. Change users...

READ MORE

 Live Chat

 Get a Quote

 Support | 1-855-330-1509

 Sales | 1-877-588-5918

[Privacy Center](#) [Do not sell or share my personal information](#)

[Contact Us](#)

[Legal](#)

[Privacy Policy](#)

[Terms of Use](#)

[DMCA](#)

[GDPR](#)

[Sitemap](#)

©2024 Copyright phoenixNAP | Global IT Services. All Rights Reserved.