

# Linux Command Line Cheat Sheet: All the Commands You Need

February 14, 2023 / By Nathan House

## LINUX CHEAT SHEET



 Listen to the article

You may need to open a compressed file, but you've forgotten the TAR commands. Or you're new to Linux and need to know the top ten terminal commands that open and modify files and folders. The sheer volume of Linux terminal commands can overwhelm beginners, not to mention server administrators, IT professionals, and hobbyists.

Therefore, we've prepared this essential Linux command line cheat sheet to help you get familiar with Linux security commands categorized by the scope of their actions. We're confident that this compilation can help you master Linux quickly.

Feel free to download a copy of this cheat sheet [here](#) and scroll down to find the commands you need today.

Search cheats here



## Essential Commands

We recommend that you memorize these commands. You'll need them the most when operating Linux.

# File Management

In the following commands: `X` may refer to a single file, a string containing a wildcard symbol referring to a set of multiple files, e.g., `file*.txt`, or the stream output of a piped command (in which case the syntax would be `X | command` instead of `command X`); `Y` is a single directory; `A` and `B` are path strings of files/directories.

COMMAND	DESCRIPTION
<code>*</code>	Wildcard symbol for variable length, e.g., <code>*.txt</code> refers to all files with the TXT extension
<code>?</code>	Wildcard symbol referring to a single character, e.g., <code>Doc?.docx</code> can refer to <code>Doc1.docx</code> , <code>DocA.docx</code> , etc.
<code>ls</code>	List the names of files and subfolders in the current directory. Options include <code>-l</code> , <code>-a</code> , <code>-t</code> which you may combine, e.g., <code>-alt</code> .
<code>ls -l</code>	Also show details of each item displayed, such as user permissions and the time/date when the item was last modified
<code>ls -a</code>	Also display hidden files/folders. May be combined with <code>ls -l</code> to form <code>ls -al</code> .
<code>ls -t</code>	Sort the files/folders according to the last modified time/date, starting with the most recently modified item
<code>ls X</code>	List the files
<code>cd Y</code>	Change directory to <code>Y</code> .  Special instances of <code>Y</code> : <code>.</code> — current directory <code>..</code> — parent directory
<code>cd</code>	To the <code>\$HOME</code> directory
<code>cd ..</code>	Up one level to enclosing folder or parent directory
<code>cd /etc</code>	To the <code>/etc</code> directory
<code>cmp A B</code>	Compare two files <code>A</code> and <code>B</code> for sameness. No output if <code>A</code> and <code>B</code> are identical, outputs character and line number otherwise.

COMMAND	DESCRIPTION
<code>diff A B</code>	Compare two files <code>A</code> and <code>B</code> for differences. Outputs the difference.
<code>pwd</code>	Display the path of the current working directory.
<code>mkdir X</code>	Make a new directory named <code>X</code> inside the current directory.
<code>mv A B</code>	<p>Move a file from path <code>A</code> to path <code>B</code>. Also used for renaming files.</p> <p>Examples:</p> <ul style="list-style-type: none"> <li>– Moving between directories <code>folder1</code> and <code>folder2</code>:  <code>mv ./folder1/file.txt ./folder2</code></li> </ul> <p>The file name will remain unchanged, and its new path will be <code>./folder2/file.txt</code>.</p> <li>– Renaming a file: <code>mv new_doc.txt expenses.txt</code></li> <p>The new file name is <code>expenses.txt</code>.</p>
<code>cp A B</code>	<p>Copy a file from path <code>A</code> to path <code>B</code>. Usage similar to <code>mv</code> both in moving to a new directory and simultaneously renaming the file in its new location.</p> <p>Example: <code>cp ./f1/file.txt ./f2/expenses.txt</code> simultaneously copies the file <code>file.txt</code> to the new location with a new name <code>expenses.txt</code>.</p>
<code>cp -r Y Z</code>	Recursively copy a directory <code>Y</code> and its contents to <code>Z</code> . If <code>Z</code> exists, copy source <code>Y</code> into it; otherwise, create <code>Z</code> and <code>Y</code> becomes its subdirectory with <code>Y</code> 's contents
<code>rm X</code>	Remove (delete) <code>X</code> permanently.
<code>rm -r Y</code>	Recursively delete a directory <code>Y</code> and its contents
<code>rm -f X</code>	Forcibly remove file <code>X</code> without prompts or confirmation
<code>rm -rf Y</code>	Forcibly remove directory <code>Y</code> and its contents recursively
<code>rmdir Y</code>	Remove a directory <code>Y</code> permanently, provided <code>Y</code> is empty.
<code>open X</code>	Open <code>X</code> in its default program.
<code>open -e X</code>	Opens <code>X</code> in the default text editor (macOS: TextEdit)
<code>touch X</code>	Create an empty file <code>X</code> or update the access and modification times of <code>X</code> .
<code>cat X</code>	View contents of <code>X</code> .

COMMAND	DESCRIPTION
<code>cat -b X</code>	Also display line numbers as well.
<code>wc X</code>	Display word count of <code>X</code> .
<code>head X</code>	Display the first 10 lines of <code>X</code> . If more than a single file is specified, each file is preceded by a header consisting of the string “ <code>==&gt; X &lt;==</code> ” where “ <code>X</code> ” is the name of the file.
<code>head -n 4 X</code>	Show the first 4 lines of <code>X</code> .
<code>ls *.c   head -n 5</code>	Display the first 5 items of a list of <code>*.c</code> files in the current directory.
<code>tail X</code>	Display the last (10, by default) lines of <code>X</code> . If more than a single file is specified, each file is preceded by a header consisting of the string “ <code>==&gt; X &lt;==</code> ” where “ <code>X</code> ” is the name of the file.
<code>tail -n +1 X</code>	Display entire contents of the file(s) <code>X</code> specified, with header of respective file names
<code>tail -f X</code>	Display the last 10 lines of the file(s) <code>X</code> specified, and track changes appended to them at the end. Overwriting <code>X</code> or modifying <code>X</code> with a text editor such as <code>vim</code> would mess up this command’s output.
<code>less</code>	Read a file with forward and backward navigation. Often used with pipe, e.g., <code>cat file.txt   less</code>
<code>ln -s A S</code>	Create symbolic link of path <code>A</code> to link name <code>S</code> .

## Input/Output Redirection

These are helpful for logging program output and error messages.

COMMAND	DESCRIPTION
<code>echo TEXT</code>	Display a line of <code>TEXT</code> or the contents of a variable.
<code>echo -e TEXT</code>	Also interprets escape characters in <code>TEXT</code> , e.g., <code>\n</code> → new line, <code>\b</code> → backslash, <code>\t</code> → tab.

COMMAND	DESCRIPTION
<code>echo -n TEXT</code>	Omits trailing newline of <code>TEXT</code> .
<code>cmd1   cmd2</code>	<code> </code> is the pipe character; feeds the output of the command <code>cmd1</code> and sends it to the command <code>cmd2</code> , e.g., <code>ps aux   grep python3</code> .
<code>cmd &gt; file</code>	Redirect output of a command <code>cmd</code> to a file <code>file</code> . Overwrites pre-existing content of <code>file</code> .
<code>cmd &gt;&amp; file</code>	Redirect output of <code>cmd</code> to <code>file</code> . Overwrites pre-existing content of <code>file</code> . Suppresses the output of <code>cmd</code> .
<code>cmd &gt; /dev/null</code>	Suppress the output of <code>cmd</code> .
<code>cmd &gt;&gt; file</code>	Append output of <code>cmd</code> to <code>file</code> .
<code>cmd &lt; file</code>	Read input of <code>cmd</code> from <code>file</code> .
<code>cmd &lt;&lt; delim</code>	<p>Read input of <code>cmd</code> from the standard input with the delimiter character <code>delim</code> to tell the system where to terminate the input. Example for counting the number of lines of ad-hoc input:</p> <pre>wc -l &lt;&lt; EOF &gt; I like &gt; apples &gt; and &gt; oranges. &gt; EOF 4</pre> <p>Hence there are only 4 lines in the standard input delimited by <code>EOF</code>.</p>
<code>cmd &lt;&lt;&lt; string</code>	Input a text <code>string</code> to <code>cmd</code> .
<code>cmd 2&gt; foo</code>	Redirect error messages of <code>cmd</code> to <code>foo</code> .
<code>cmd 2&gt;&gt; foo</code>	Append error messages of <code>cmd</code> to <code>foo</code> .
<code>cmd &amp;&gt; file</code>	Redirect output and error messages of <code>cmd</code> to <code>file</code> .

## Search and Filter

These commands help you find the files you want.

COMMAND	DESCRIPTION
<code>grep patt /path/to/src</code>	Search for a text pattern <code>patt</code> in <code>X</code> . Commonly used with pipe e.g., <code>ps aux   grep python3</code> filters out the processes containing <code>python3</code> from all running processes of all users.
<code>grep -r patt /path/to/src</code>	Search recursively (the target directory <code>/path/to/src</code> and its subdirectories) for a text pattern <code>patt</code> .
<code>grep -v patt X</code>	Return lines in <code>X</code> not matching the specified <code>patt</code> .
<code>grep -l patt X</code>	Write to standard output the names of files containing <code>patt</code> .
<code>grep -i patt X</code>	Perform case-insensitive matching on <code>X</code> . Ignore the case of <code>patt</code> .
<code>find</code>	Find files.
<code>find /path/to/src -name "*.sh"</code>	Find all files in <code>/path/to/src</code> matching the pattern “ <code>*.sh</code> ” in the file name.
<code>find /home -size +100M</code>	Find all files in the <code>/home</code> directory larger than 100MB.
<code>locate name</code>	Find files and directories by <code>name</code> .
<code>sort X</code>	Arrange lines of text in <code>X</code> alphabetically or numerically.

## Archives

These commands are for unpacking compressed files (`.zip`, `.gz`, `.bz2`, `.tar`, etc.) with large or complex contents, such as programs.

COMMAND	DESCRIPTION
<code>tar</code>	Manipulate archives with <code>.tar</code> extension.
<code>tar -v</code>	Get verbose output while manipulating TAR archives. May combine this

COMMAND	DESCRIPTION
	option with others, e.g., <code>tar -tvf</code> .
<code>tar -cf archive.tar Y</code>	Create a TAR archive named <code>archive.tar</code> containing <code>Y</code> .
<code>tar -xf archive.tar</code>	Extract the TAR archive named <code>archive.tar</code> .
<code>tar -tf archive.tar</code>	List contents of the TAR archive named <code>archive.tar</code> .
<code>tar -czf archive.tar.gz Y</code>	Create a gzip-compressed TAR archive named <code>archive.tar.gz</code> containing <code>Y</code> .
<code>tar -xzf archive.tar.gz</code>	Extract the gzip-compressed TAR archive named <code>archive.tar.gz</code> .
<code>tar -cjf archive.tar.bz2 Y</code>	Create a bzip2-compressed TAR archive named <code>archive.tar.bz2</code> containing <code>Y</code> .
<code>tar -xjf archive.tar.bz2</code>	Extract the bzip2-compressed TAR archive named <code>archive.tar.bz2</code> .
<code>gzip</code>	Manipulate archives with <code>.gz</code> extension.
<code>gzip Y</code>	Create a gzip archive named <code>Y.gz</code> containing <code>Y</code> .
<code>gzip -l Y.gz</code>	List contents of gzip archive <code>Y.gz</code> .
<code>gzip -d Y.gz</code> <code>gunzip Y.gz</code>	Extract <code>Y.gz</code> and recover the original file <code>Y</code> .
<code>bzip2</code>	Manipulate archives with <code>.bz2</code> extension.
<code>bzip2 Y</code>	Create a bzip2 archive named <code>Y.bz2</code> containing <code>Y</code> .
<code>bzip2 -d Y.gz</code> <code>bunzip2 Y.gz</code>	Extract <code>Y.bz2</code> and recover the original file <code>Y</code> .
<code>zip -r Z.zip Y</code>	Zip <code>Y</code> to the ZIP archive <code>Z.zip</code> .
<code>unzip Z.zip</code>	Unzip <code>Z.zip</code> to the current directory.
<code>unzip Z.zip</code>	List contents of <code>Z.zip</code> .

## File Transfer

These commands are for logging in to local and remote hosts, and for uploading and downloading files, transferring them between devices. Remember to omit the square brackets “ [ ” and “ ] ” when you input the optional parameters they enclose.

COMMAND	DESCRIPTION
<code>ssh user@access</code>	Connect to <code>access</code> as <code>user</code> .
<code>ssh access</code>	Connect to <code>access</code> as your local username.
<code>ssh -p port user@access</code>	Connect to <code>access</code> as <code>user</code> using <code>port</code> .
<code>scp [user1@]host1: [path1] [user2@]host2: [path2]</code>	<p>Login to <code>hostN</code> as <code>userN</code> via secure copy protocol for <code>N=1, 2</code>.</p> <p>Example usage:</p> <p><code>scp alice@pi:/home/source bob@arduino:/destination</code></p> <p><code>path1</code> and <code>path2</code> may be local or remote, but ensure they're absolute rather than relative paths, e.g., <code>/var/www/*.html</code>, <code>/usr/bin</code>.</p> <p>If <code>user1</code> and <code>user2</code> are not specified, <code>scp</code> will use your local username.</p>
<code>scp -P port [user1@]host1:[path1] [user2@]host2:[path2]</code>	Connect to <code>hostN</code> as <code>userN</code> using <code>port</code> for <code>N=1, 2</code> .
<code>scp -r [user1@]host1: [path1] [user2@]host2: [path2]</code>	Recursively copy all files and directories from <code>path1</code> to <code>path2</code> .
<code>sftp [user@]access</code>	Login to <code>access</code> as <code>user</code> via secure file transfer protocol. If <code>user</code> is not specified, your local username will be used.
<code>sftp access</code>	Connect to <code>access</code> as your local username.
<code>sftp -P port user@access</code>	Connect to <code>access</code> as <code>user</code> using <code>port</code> .
<code>rsync -a [path1] [path2]</code>	Synchronize <code>[path1]</code> to <code>[path2]</code> , preserving symbolic links, attributes, permissions, ownerships, and other settings.



COMMAND	DESCRIPTION
<code>rsync -avz host1: [path1] [path2]</code>	Synchronize <code>[path1]</code> on the remote host <code>host1</code> to the local path <code>[path2]</code> , preserving symbolic links, attributes, permissions, ownerships, and other settings. It also compresses the data involved during the transfer.

## File Permissions

Not all files are equally accessible. To prevent unwanted tampering, some files on your device may be read-only. For more information about file permissions on Linux, refer to our [Linux File Permissions Cheat Sheet](#).

COMMAND	DESCRIPTION
<code>chmod permission file</code>	Change permissions of a file or directory. Permissions may be of the form <code>[u/g/o/a] [+/-/=] [r/w/x]</code> (see examples below) or a three-digit octal number.
<code>chown user2 file</code>	Change the owner of a file to <code>user2</code> .
<code>chgrp group2 file</code>	Change the group of a file to <code>group2</code> .

Usage examples:

- `chmod +x testfile` → allow all users to execute the file
- `chmod u-w testfile` → forbid the current user from writing or changing the file
- `chmod u+wx,g-x,o=rx testfile` → simultaneously add write and execute permissions to user, remove execute permission from group, and set the permissions of other users to only read and write.

# Numeric Representation

The table below compares Linux file permissions in octal form and in the format [u/g/o/a] [+/-/=] [r/w/x].

OCTAL	PERMISSION(S)	EQUIVALENT TO APPLICATION OF
0	No permissions	=rwx
1	Execute permission only	=x
2	Write permission only	=w
3	Write and execute permissions only: $2 + 1 = 3$	=wx
4	Read permission only	=r
5	Read and execute permissions only: $4 + 1 = 5$	=rx
6	Read and write permissions only: $4 + 2 = 6$	=rw
7	All permissions: $4 + 2 + 1 = 7$	=rwx

## Examples

- `chmod 777 testfile` → allow all users to execute the file
- `chmod 177 testfile` → restrict current user (u) to execute-only, while the group (g) and other users (o) have read, write and execute permissions
- `chmod 365 testfile` → user (u) gets to write and execute only; group (g), read and write only; others (o), read and execute only.

## Other Links You Might Like:

- [Linux File Permissions Cheat Sheet](#)
- [Linux Command Line Cheat Sheet](#)
- [Unix Commands Cheat Sheet](#)
- [Tmux Cheat Sheet](#)
- [Tcpdump Cheat Sheet](#)
- [Bash Cheat Sheet](#)
- [Our Linux Courses Collection](#)

# System Information

These commands come in handy when you’re developing new applications for Linux or troubleshooting your Linux machine.

## General

These provide information about your Linux machine and perform administrative tasks.

COMMAND	DESCRIPTION
uname	Show the Linux system information.
uname -a	Detailed Linux system information
uname -r	Kernel release information, such as kernel version
uptime	Show how long the system is running and load information.
su sudo	Superuser; use this before a command that requires root access e.g., su shutdown
cal	Show calendar where the current date is highlighted.
date	Show the current date and time of the machine.
halt	Stop the system immediately.
shutdown	Shut down the system.

00:00

00:001⚡


✕

man COMMAND	Shows the manual for a given COMMAND. To exit the manual, press “q”
hostname	Show system host name
hostname -I	Display IP address of host
cat /etc/*-release	Show the version of the Linux distribution installed. For example, if you’re using Red Hat Linux, you may replace * with redhat.

# Hardware

These commands provide details about the hardware supporting your Linux machine.

COMMAND	DESCRIPTION
<code>dmesg</code>	Display messages in kernel ring buffer (data structure that records messages related to the operation of the program running the operating system)
<code>cat /proc/cpuinfo</code>	Display information about the central processing unit (CPU)
<code>cat /proc/meminfo</code>	Display memory information
<code>lspci -tv</code>	Displays information about each Peripheral Component Interconnect (PCI) device on your system. The option <code>-t</code> outputs the information as a tree diagram, and <code>-v</code> is for verbose output.
<code>lsusb -tv</code>	Display information about Universal Serial Bus (USB) devices and the devices connected to them. The option <code>-t</code> outputs the information as a tree diagram, and <code>-v</code> is for verbose output.
<code>dmidecode</code>	Display system hardware components, serial numbers, and BIOS version
<code>hdparm -i /dev/sda</code>	Display information about the disk <code>sda</code>



00:00

00:00

1 ⚡

✕

<code>badblocks -s /dev/sda</code>	Test for unreadable blocks on the disk <code>sda</code>
------------------------------------	---

# Disk Usage


These commands provide storage details regarding your Linux machine.

COMMAND	DESCRIPTION
df	Display free disk space.
du	Show file/folder sizes on disk.
du -ah	Disk usage in human readable format (KB, MB etc.)
du -sh	Total disk usage of the current directory
du -h	Free and used space on mounted filesystems
du -i	Free and used inodes on mounted filesystems
fdisk -l	List disk partitions, sizes, and types
free -h	Display free and used memory in human readable units.
free -m	Display free and used memory in MB.
free -g	Display free and used memory in GB.

## Process Management and Performance Monitoring

The following is redolent of functions in Windows Task Manager, but on the command line.

COMMAND	DESCRIPTION
&	Add this character to the end of a command/process to run it in the background



00:00


00:00

1⚡

✕

ps	Meaning of any: a = show processes for all users u = show user or owner column in output x = show processes not attached to a terminal
ps -e ps -A	Either of these two commands prints all running processes in the system

COMMAND	DESCRIPTION
<code>ps -ef</code>	Print detailed overview
<code>ps -U root -u root</code>	Display all processes running under the account <code>root</code> .
<code>ps -eo pid,user,command</code>	Display only the columns <code>pid</code> , <code>user</code> and <code>command</code> in <code>ps</code> output
<code>top</code>	Display sorted information about processes
<code>htop</code>	Display sorted information about processes with visual highlights. It allows you to scroll vertically and horizontally, so you can see every process running on your system and entire commands.
<code>atop</code>	Display detailed information about processes and hardware
<code>kill PID</code>	Kill a process specified by its process ID <code>PID</code> , which you obtain using the <code>ps</code> command
<code>killall proc1</code>	Kill all processes containing <code>proc1</code> in their names
<code>lsof</code>	List all open files on the system. (This command helps you pinpoint what files and processes are preventing you from successfully ejecting an external drive.)
<code>lsof -u root</code>	List all files on the system opened by the <code>root</code> user. As the output can be long, you may use <code>lsof -u root   less</code> to keep this list from taking up space in the terminal output.


00:00
00:00
1 ⚡
✕


<code>vmstat 1</code>	Display virtual memory statistics (information about memory, system processes, paging, interrupts, block I/O, disk, and CPU scheduling), updated every (1) second
<code>iostat 1</code>	Display system input/output statistics for devices and partitions. virtual memory statistics, updated every (1) second

COMMAND	DESCRIPTION
<code>tail -n 100 /var/log/messages</code>	Display the last 100 lines in the system logs. Replace <code>/var/log/messages</code> with <code>/var/log/syslog</code> for Debian-based systems.
<code>tcpdump -i eth0</code>	Capture and display all packets on interface <code>eth0</code>
<code>tcpdump -i eth0 port 80</code>	Monitor all traffic on interface <code>eth0</code> port 80 (HTTP)
<code>watch df -h</code>	Execute <code>df -h</code> and show periodic updates. To exit, press <b>Ctrl+C</b> .

## User Management

These commands give information on the system’s users and allows superuser administrators to change user settings.

COMMAND	DESCRIPTION
<code>who</code>	Display who is logged in
<code>w</code>	Display what users are online and what they are doing
<code>users</code>	List current users
<code>whoami</code>	Display what user you are logged in as
<code>id</code>	Display the user ID and group IDs of your current user

 00:00 00:00 1⚡ X

<code>groupadd gp1</code>	Create a group named <code>gp1</code>
<code>useradd -c "Alice Bob" -m ab1</code>	Create an account named <code>ab1</code> , with a comment of “Alice Bob” and create the new user’s home directory
<code>userdel ab1</code>	Delete the account named <code>ab1</code>
<code>usermod -aG gp1 ab1</code>	Add the account <code>ab1</code> to the group <code>gp1</code>

# Networking

These commands regulate how your Linux machine communicates with other computers, such as the local area network (LAN) router or external websites.

COMMAND	DESCRIPTION
<code>ifconfig</code>	Display all network interfaces with IP addresses
<code>ifconfig -a</code>	Display all network interfaces, even if any of them is down, with IP addresses
<code>ifconfig eth0</code>	Display IP addresses and details of the <code>eth0</code> interface
<code>ip a</code>	Another way to display all network interfaces with IP addresses
<code>ethtool eth0</code>	Query or control network driver and hardware settings of the interface <code>eth0</code>
<code>netstat</code>	Print open sockets of network connections, routing tables, interface statistics, masquerade connections, and multicast memberships. Pipe with the <code>less</code> command: e.g., <code>netstat -a   less</code>
<code>netstat -a</code>	Show both listening and non-listening sockets
<code>netstat -l</code>	Show only listening sockets
<code>netstat -nutlp</code>	Show listening TCP and UDP ports and corresponding programs

00:00

00:001⚡

✕

<code>whois domain</code>	Display whois information for <code>domain</code>
<code>dig domain</code>	Display DNS information for <code>domain</code>
<code>dig -x addr</code>	Do a reverse lookup on an IPv4 or IPv6 address <code>addr</code>
<code>host domain</code>	Display DNS IP address for <code>domain</code>



COMMAND	DESCRIPTION
wget LINK	Download from location LINK
curl LINK	Display the HTML source of LINK. Check out our <a href="#">curl Cheat Sheet</a> for details.

## Installing New Programs

A package file is an archive containing compiled software and other resources it uses. The package file extension and the package installer (a utility for installing new programs) depend on the Linux distribution.

Know your system's Linux distribution to understand the correct installation commands tailored to it. If the package comes with a README component, it should contain application-specific installation instructions such as extracting TAR files (refer to this article's [Archives](#) section), `./setup.sh`, and `make install`.

## Package Management Overview

The following table is on package management in popular Linux distributions.

LINUX DISTRIBUTION	PACKAGE FILE EXTENSION	PACKAGE INSTALLER(S)
Debian / Ubuntu	.deb	apt, dpkg
Rocky / Fedora / Red Hat Enterprise Linux	.rpm	yum, dnf
Arch Linux / Manjaro / Garuda /	.pkg, .pacman	

00:0000:001⚡✕

## Package Management Commands

Here are the commands for package management in popular Linux distributions.

LINUX DISTRIBUTION	DEBIAN / UBUNTU	ROCKY / FEDORA / RED HAT ENTERPRISE LINUX	ARCH LINUX / MANJARO / GARUDA / CHAKRA
Update list of packages available from remote	<code>sudo apt update</code>	<code>dnf check-update</code>	The command <code>pacman -Syy</code> achieves this purpose but

LINUX DISTRIBUTION	DEBIAN / UBUNTU	ROCKY / FEDORA / RED HAT ENTERPRISE LINUX	ARCH LINUX / MANJARO / GARUDA / CHAKRA
repositories			may damage your system. Use <code>pacman -Syu</code> instead.
Upgrade installed packages	<code>sudo apt upgrade</code>	<code>sudo dnf upgrade</code>	<code>pacman -Syu</code>
Find a package with keyword in the name	<code>apt search keyword</code>	<code>dnf search keyword</code>	<code>pacman -Ss keyword</code>
View description and summary information about a package	<code>apt show package</code>	<code>dnf info package</code>	<code>pacman -Si package</code>
Install a package (with appropriate file extension) on the local file system	<code>sudo dpkg -i package.deb</code>	<code>sudo dnf install package.rpm</code>	<code>pacman -S package</code>
Remove / uninstall a package	<code>sudo apt remove package</code>	<code>sudo dnf erase package</code>	<code>pacman -R package</code>

## Conclusion

Learning basic Linux commands is a great way to kickstart your education in IT and cyber security. You’re now ready to practice them on any Linux distribution you fancy. If you want to focus on cyber security, go for [Kali Linux](#) and [Parrot OS](#). Also, don’t forget to check out our [Unix Commands Cheat Sheet here](#).

00:00

00:00
1
⚡

## Frequently Asked Questions

What is a basic Linux command?

A basic Linux command is a keyword at most ten characters long with only a few additional arguments. For example, **ls** (list directory contents) and **cd** (change directory) are basic Linux commands.

- ⊕ How do I get a list of commands in the terminal?
- ⊕ How do I learn Linux commands?
- ⊕ What is the most used Linux command?
- ⊕ Are Linux commands easy to learn?
- ⊕ What are Linux commands?
- ⊕ What are 5 Linux commands?

## Grow your Cyber Security Skills



00:00

00:00

14



CATEGORIES

CHEAT SHEETS

LINUX



## Nathan House

Nathan House is the founder and CEO of StationX. He has over 25 years of experience in cyber security, where he has advised some of the largest companies in the world. Nathan is the author of the popular "The Complete Cyber Security Course", which has been taken by over half a million students in 195 countries. He is the winner of the AI "Cyber Security Educator of the Year 2020" award and finalist for Influencer of the year 2022.

## Related Articles

NMAP  
CHEAT SHEET



WIRESHARK  
CHEAT SHEET

WIRESHARK



00:00

00:00

1 ⚡



[Read More »](#)

[Read More »](#)

## COMMON PORTS CHEAT SHEET



### Common Ports Cheat Sheet: The Ultimate Ports & Protocols List

[Read More »](#)

## IPV4 SUBNETTING CHEAT SHEET



### The Only IPv4 Subnetting Cheat Sheet You'll Ever Need

[Read More »](#)

#### INFO

[Affiliates](#)

[Legal Notices](#)

[Privacy Policy](#)

[Site Map](#)

#### SECURITY ASSESSME NT

[Penetration](#)

[Testing](#)

[Vulnerability](#)

#### CONSULTI NG

[Audit &](#)

[Compliance](#)

[Incident](#)

[Response](#)



00:00

00:00

1 ⚡



[Review](#)

[Social](#)

[Engineering](#)

[Security Training](#)



00:00

00:00

1 ⚡

