

Linux Commands Cheat Sheet

Last Updated : 02 Aug, 2024



Linux, often associated with being a complex operating system primarily used by developers, may not necessarily fit that description entirely. While it can initially appear challenging for beginners, once you immerse yourself in the Linux world, you may find it difficult to return to your previous Windows systems. The power of Linux commands in controlling your PC, coupled with their clean user interface, can make it hard to switch back to older operating systems. If you're a developer, you can likely relate to the advantages and appeal of Linux.

To support developers and beginners alike, we have created a comprehensive **Linux/Unix command line cheat sheet**. This cheat sheet covers all the basic and advanced commands, including file and directory commands, file permission commands, file compression and archiving, process management, system information, networking, and more with proper examples and descriptions. In addition to that we provide all the most used Linux Shortcut which includes Bash shortcuts, Nano shortcuts, VI & Vim Shortcuts Commands. It provides a solid foundation on Linux OS commands, as well as insights into practical applications.

By the end of this cheat sheet, you will have a basic understanding of Linux/Unix Commands and how it makes development easy for developers.

Linux Cheat Sheet



What is Linux?

Linux is an open-source UNIX-like operating system (OS). An operating system is a software that directly manages a system's hardware and resources, like CPU, memory, and storage. OS acts as a GUI through which user can communicate with the computer. The OS sits between applications and hardware and makes the connections between all of your software and the physical resources that do the work.

Linux Commands List - Table of Content

- [File and Directory Operations Commands](#)
- [File Permission Commands](#)
- [File Compression and Archiving Commands](#)
- [Process Management Commands](#)
- [System Information Commands](#)
- [Networking Commands](#)
- [IO Redirection Commands](#)
- [Environment Variable Commands](#)
- [User Management Commands](#)
- [Shortcuts Commands List](#)
 - [Bash Shortcuts Commands](#)
 - [Nano Shortcuts Commands](#)
 - [VI Shortcuts Commands](#)
 - [Vim Shortcuts Commands](#)
- [FAQs on Linux Commands Cheat Sheet](#)

Basic Linux Commands with Examples

In this Linux cheat sheet, we will cover all the most important Linux commands, from the basics to the advanced. We will also provide some tips on how to practice and learn Linux commands. This cheat sheet is useful for Beginners and Experience professionals.

1. File and Directory Operations Commands

File and directory operations are fundamental in working with the Linux operating system. Here are some commonly used File and Directory Operations commands:

Command	Description	Options	Examples
ls	List files and directories.	<ul style="list-style-type: none"> • -l: Long format listing. • -a: Include hidden files hidden ones • -h: Human-readable file sizes. 	<ul style="list-style-type: none"> • ls -l displays files and directories with detailed information. • ls -a shows all files and directories, including • ls -lh displays file sizes in a human-readable format.
cd	Change directory.		<ul style="list-style-type: none"> • cd /path/to/directory changes the current directory to the specified path.
pwd	Print current working directory.		<ul style="list-style-type: none"> • pwd displays the current working directory.
mkdir	Create a new directory.		<ul style="list-style-type: none"> • mkdir my_directory creates a new directory named "my_directory".
rm	Remove files and directories.	<ul style="list-style-type: none"> • -r: Remove directories recursively. • -f: Force removal without confirmation. 	<ul style="list-style-type: none"> • rm file.txt deletes the file named "file.txt". • rm -r my_directory deletes the directory "my_directory" and its contents. • rm -f file.txt forcefully deletes the file "file.txt" without confirmation.

Command	Description	Options	Examples
cp	Copy files and directories.	<ul style="list-style-type: none"> • -r: Copy directories recursively. 	<ul style="list-style-type: none"> • cp -r directory destination copies the directory "directory" and its contents to the specified destination. • cp file.txt destination copies the file "file.txt" to the specified destination.
mv	Move/rename files and directories.		<ul style="list-style-type: none"> • mv file.txt new_name.txt renames the file "file.txt" to "new_name.txt". • mv file.txt directory moves the file "file.txt" to the specified directory.
touch	Create an empty file or update file timestamps.		<ul style="list-style-type: none"> • touch file.txt creates an empty file named "file.txt".
cat	View the contents of a file.		<ul style="list-style-type: none"> • cat file.txt displays the contents of the file "file.txt".
head	Display the first few lines of a file.	<ul style="list-style-type: none"> • -n: Specify the number of lines to display. 	<ul style="list-style-type: none"> • head file.txt shows the first 10 lines of the file "file.txt". • head -n 5 file.txt displays the first 5 lines of the file "file.txt".

Command	Description	Options	Examples
tail	Display the last few lines of a file.	<ul style="list-style-type: none"> • -n: Specify the number of lines to display. 	<ul style="list-style-type: none"> • tail file.txt shows the last 10 lines of the file "file.txt". • tail -n 5 file.txt displays the last 5 lines of the file "file.txt".
ln	Create links between files.	<ul style="list-style-type: none"> • -s: Create symbolic (soft) links. 	<ul style="list-style-type: none"> • ln -s source_file link_name creates a symbolic link named "link_name" pointing to "source_file".
find	Search for files and directories.	<ul style="list-style-type: none"> • -name: Search by filename. • -type: Search by file type. 	<ul style="list-style-type: none"> • find /path/to/search -name "*.txt" searches for all files with the extension ".txt" in the specified directory.

2. File Permission Commands

File permissions on Linux and Unix systems control access to files and directories. There are three basic permissions: read, write, and execute. Each permission can be granted or denied to three different categories of users: the owner of the file, the members of the file's group, and everyone else.

Here are some file permission commands:

Command	Description	Options	Examples
chmod	Change file permissions.	<ul style="list-style-type: none"> • u: User/owner permissions. • g: Group permissions. 	<ul style="list-style-type: none"> • chmod u+rw file.txt grants read, write, and execute permissions to the owner of the file.

Command	Description	Options	Examples
		<ul style="list-style-type: none"> • o: Other permissions. • +: Add permissions. • -: Remove permissions. • =: Set permissions explicitly. 	
chown	Change file ownership.		<ul style="list-style-type: none"> • chown user file.txt changes the owner of "file.txt" to the specified user.
chgrp	Change group ownership.		<ul style="list-style-type: none"> • chgrp group file.txt changes the group ownership of "file.txt" to the specified group.
umask	Set default file permissions.		<ul style="list-style-type: none"> • umask 022 sets the default file permissions to read and write for the owner, and read-only for group and others.

3. File Compression and Archiving Commands

Here are some file compression and archiving commands in Linux:

Commands	Description	Options	Examples
tar	Create or extract archive files.	<ul style="list-style-type: none"> • -c: Create a new archive. • -x: Extract files from an archive. 	<ul style="list-style-type: none"> • tar -czvf archive.tar.gz files/ creates a compressed tar archive named "archive.tar.gz"

Commands	Description	Options	Examples
		<ul style="list-style-type: none"> • -f: Specify the archive file name. • -v: Verbose mode. • -z: Compress the archive with gzip. • -j: Compress the archive with bzip2. 	containing the files in the "files/" directory.
gzip	Compress files.	<ul style="list-style-type: none"> • -d: Decompress files. 	<ul style="list-style-type: none"> • gzip file.txt compresses the file "file.txt" and renames it as "file.txt.gz".
zip	Create compressed zip archives.	<ul style="list-style-type: none"> • -r: Recursively include directories. 	<ul style="list-style-type: none"> • zip archive.zip file1.txt file2.txt creates a zip archive named "archive.zip" containing "file1.txt" and "file2.txt".

4. Process Management Commands

In Linux, process management commands allow you to monitor and control running processes on the system. Here are some commonly used process management commands:

Commands	Description	Options	Examples
ps	Display running processes.	<ul style="list-style-type: none"> • -aux: Show all processes. 	<ul style="list-style-type: none"> • ps aux shows all running processes with detailed information.

Commands	Description	Options	Examples
top	Monitor system processes in real-time.		<ul style="list-style-type: none"> top displays a dynamic view of system processes and their resource usage.
kill	Terminate a process.	<ul style="list-style-type: none"> -9: Forcefully kill a process. 	<ul style="list-style-type: none"> kill PID terminates the process with the specified process ID.
pkill	Terminate processes based on their name.		<ul style="list-style-type: none"> pkill process_name terminates all processes with the specified name.
pgrep	List processes based on their name.		<ul style="list-style-type: none"> pgrep process_name lists all processes with the specified name.
grep	used to search for specific patterns or regular expressions in text files or streams and display matching lines.	<ul style="list-style-type: none"> -i: Ignore case distinctions while searching. -v: Invert the match, displaying non-matching lines. -r or -R: Recursively search directories for 	<ul style="list-style-type: none"> grep -i "hello" file.txt grep -v "error" file.txt grep -r "pattern" directory/ grep -l "keyword" file.txt grep -n "pattern" file.txt In these examples we are extracting our desired output

Commands	Description	Options	Examples
		<p>matching patterns.</p> <ul style="list-style-type: none">• -l: Print only the names of files containing matches.• -n: Display line numbers alongside matching lines.• -w: Match whole words only, rather than partial matches.• -c: Count the number of matching lines instead of displaying them.• -e: Specify multiple patterns to search for.• -A: Display lines after the matching line.• -B: Display lines before the matching line.• -C: Display lines both before and after the matching line.	<p>from filename (file.txt)</p>

5. System Information Commands

In Linux, there are several commands available to gather system information. Here are some commonly used system information commands:

Command	Description	Options	Examples
uname	Print system information.	<ul style="list-style-type: none"> • -a: All system information. 	<ul style="list-style-type: none"> • uname -a displays all system information.
whoami	Display current username.		<ul style="list-style-type: none"> • whoami shows the current username.
df	Show disk space usage.	<ul style="list-style-type: none"> • -h: Human-readable sizes. 	<ul style="list-style-type: none"> • df -h displays disk space usage in a human-readable format.
du	Estimate file and directory sizes.	<ul style="list-style-type: none"> • -h: Human-readable sizes. • -s: Display total size only. 	<ul style="list-style-type: none"> • du -sh directory/ provides the total size of the specified directory.
free	Display memory usage information.	<ul style="list-style-type: none"> • -h: Human-readable sizes. 	<ul style="list-style-type: none"> • free -h displays memory usage in a human-readable format.
uptime	Show system uptime.		<ul style="list-style-type: none"> • uptime shows the current system uptime.
lscpu	Display CPU information.		<ul style="list-style-type: none"> • lscpu provides detailed CPU information.

Command	Description	Options	Examples
lspci	List PCI devices.		<ul style="list-style-type: none"> • lspci List PCI devices.
<u>lsusb</u>	List USB devices.		<ul style="list-style-type: none"> • lsusb lists all connected USB devices.

6. Networking Commands

In Linux, there are several networking commands available to manage and troubleshoot network connections. Here are some commonly used networking commands:

Command	Description	Examples
<u>ifconfig</u>	Display network interface information.	<ul style="list-style-type: none"> • ifconfig shows the details of all network interfaces.
<u>ping</u>	Send ICMP echo requests to a host.	<ul style="list-style-type: none"> • ping google.com sends ICMP echo requests to "google.com" to check connectivity.
<u>netstat</u>	Display network connections and statistics.	<ul style="list-style-type: none"> • netstat -tuln shows all listening TCP and UDP connections.
ss	Display network socket information.	<ul style="list-style-type: none"> • ss -tuln shows all listening TCP and UDP connections.
<u>ssh</u>	Securely connect to a remote server.	<ul style="list-style-type: none"> • ssh user@hostname initiates an SSH connection to the specified hostname.

Command	Description	Examples
<code>scp</code>	Securely copy files between hosts.	<ul style="list-style-type: none"> • <code>scp file.txt user@hostname:/path/to/destination</code> securely copies "file.txt" to the specified remote host.
<code>wget</code>	Download files from the web.	<ul style="list-style-type: none"> • <code>wget http://example.com/file.txt</code> downloads "file.txt" from the specified URL.
<code>curl</code>	Transfer data to or from a server.	<ul style="list-style-type: none"> • <code>curl http://example.com</code> retrieves the content of a webpage from the specified URL.

7. IO Redirection Commands

In Linux, IO (Input/Output) redirection commands are used to redirect the standard input, output, and error streams of commands and processes. Here are some commonly used IO redirection commands:

Command	Description
<code>cmd < file</code>	Input of cmd is taken from file.
<code>cmd > file</code>	Standard output (stdout) of cmd is redirected to file.
<code>cmd 2> file</code>	Error output (stderr) of cmd is redirected to file.
<code>cmd 2>&1</code>	stderr is redirected to the same place as stdout.
<code>cmd1 <(cmd2)</code>	Output of cmd2 is used as the input file for cmd1.
<code>cmd > /dev/null</code>	Discards the stdout of cmd by sending it to the null device.
<code>cmd &> file</code>	Every output of cmd is redirected to file.
<code>cmd 1>&2</code>	stdout is redirected to the same place as stderr.

Command	Description
<code>cmd >> file</code>	Appends the stdout of cmd to file.

8. Environment Variable Commands

In Linux, environment variables are used to store configuration settings, system information, and other variables that can be accessed by processes and shell scripts. Here are some commonly used environment variable commands:

Command	Description
export VARIABLE_NAME=value	Sets the value of an environment variable.
echo \$VARIABLE_NAME	Displays the value of a specific environment variable.
env	Lists all environment variables currently set in the system.
unset VARIABLE_NAME	Unsets or removes an environment variable.
export -p	Shows a list of all currently exported environment variables.
env VAR1=value COMMAND	Sets the value of an environment variable for a specific command.
printenv	Displays the values of all environment variables.

9. User Management Commands

In Linux, user management commands allow you to create, modify, and manage user accounts on the system. Here are some commonly used user management commands:

Command	Description
who	Show who is currently logged in.
sudo adduser username	Create a new user account on the system with the specified username.
finger	Display information about all the users currently logged into the system, including their usernames, login time, and terminal.
sudo deluser USER GROUPNAME	Remove the specified user from the specified group.
last	Show the recent login history of users.
finger username	Provide information about the specified user, including their username, real name, terminal, idle time, and login time.
sudo userdel -r username	Delete the specified user account from the system, including their home directory and associated files. The -r option ensures the removal of the user's files.
sudo passwd -l username	Lock the password of the specified user account, preventing the user from logging in.
su - username	Switch to another user account with the user's environment.
sudo usermod -a -G GROUPNAME USERNAME	Add an existing user to the specified group. The user is added to the group without removing them from their current groups.

10. Shortcuts Commands

There are many shortcuts commands in Linux that can help you be more productive. Here are a few of the most common ones:

10.1: Bash Shortcuts Commands:

Navigation	Description	Editing	Description	History	Description
Ctrl + A	Move to the beginning of the line.	Ctrl + U	Cut/delete from the cursor position to the beginning of the line.	Ctrl + R	Search command history (reverse search).
Ctrl + E	Move to the end of the line.	Ctrl + K	Cut/delete from the cursor position to the end of the line.	Ctrl + G	Escape from history search mode.
Ctrl + B	Move back one character.	Ctrl + W	Cut/delete the word before the cursor.	Ctrl + P	Go to the previous command in history.
Ctrl + F	Move forward one character.	Ctrl + Y	Paste the last cut text.	Ctrl + N	Go to the next command in history.
Alt + B	Move back one word	Ctrl + L	Clear the screen.	Ctrl + C	Terminate the current command.
Alt + F	Move forward one word.				

10.2: Nano Shortcuts Commands:

File Operations	Description	Navigation	Description	Editing	Description
Ctrl + O	Save the file.	Ctrl + Y	Scroll up one page.	Ctrl + K	Cut/del from the cursor position to the end of the line.
Ctrl + X	Exit Nano (prompt to save if modified).	Ctrl + V	Scroll down one page.	Ctrl + U	Uncut/re the last text.
Ctrl + R	Read a file into the current buffer.	Alt + \	Go to a specific line number.	Ctrl + 6	Mark a block of text copying cutting.
Ctrl + J	Justify the current paragraph.	Alt + ,	Go to the beginning of the current line.	Ctrl + K	Cut/del the marked block of text.
		Alt + .	Go to the end of the current line.	Alt + 6	Copy the marked block of text.

10.3: VI Shortcuts Commands:

Command	Description
cw	Change the current word. Deletes from the cursor position to the end of the current word and switches to insert mode.
dd	Delete the current line.

Command	Description
x	Delete the character under the cursor.
R	Enter replace mode. Overwrites characters starting from the cursor position until you press the Escape key.
o	Insert a new line below the current line and switch to insert mode.
u	Undo the last change.
s	Substitute the character under the cursor and switch to insert mode.
dw	Delete from the cursor position to the beginning of the next word.
D	Delete from the cursor position to the end of the line.
4dw	Delete the next four words from the cursor position.
A	Switch to insert mode at the end of the current line.
S	Delete the current line and switch to insert mode.
r	Replace the character under the cursor with a new character entered from the keyboard.
i	Switch to insert mode before the cursor.
3dd	Delete the current line and the two lines below it.
ESC	Exit from insert or command-line mode and return to command mode.
U	Restore the current line to its original state before any changes were made.
~	Switch the case of the character under the cursor.

Command	Description
a	Switch to insert mode after the cursor.
C	Delete from the cursor position to the end of the line and switch to insert mode.

10.4: Vim Shortcuts Commands:

Normal Mode	Description	Command Mode	Description	Visual Mode	Description
i	Enter insert mode at the current cursor position.	:w	Save the file.	v	Enter visual mode to select text.
x	Delete the character under the cursor.	:q	Quit Vim.	y	Copy the selected text.
dd	Delete the current line.	:q!	Quit Vim without saving changes.	d	Delete the selected text.
yy	Copy the current line.	:wq or :x	Save and quit Vim.	p	Paste the copied or deleted text.
p	Paste the copied or deleted text below the current line.	:s/old/new/g	Replace all occurrences of "old" with "new" in the file.		

Normal Mode	Description	Command Mode	Description	Visual Mode	Description
u	Undo the last change.	:set nu or :set number	Display line numbers.		
Ctrl + R	Redo the last undo.				

Conclusion

In conclusion, Linux is a widely used operating system for development, and as a developer, you should have knowledge of Linux and its basic commands. In this Cheat Sheet, we covered all commands like creating directories, file compression and archiving, process management, system information, networking and more. In addition to that, this Linux Cheat Sheet is organized and categorized, making it easy for developers to quickly find the commands they need for specific use cases. By utilizing this resource, developers can enhance their productivity and efficiency in working with Linux, leading to smoother and more successful development projects.

PS. Don't miss our other Python cheat sheet for data science that covers [Scikit-Learn](#), [Bokeh](#), [Pandas](#) and [Python basics](#).

[Comment](#)
[More info](#)
[Advertise with us](#)
[Next Article](#)

Pandas Cheat Sheet for Data
Science in Python

Similar Reads

Geeksforgeeks Cheatsheets - All Coding Cheat Sheets Collections

Cheatsheets are short documents that contain all the most essential information about a specific technology in short, such as its syntax, commands, functions, or its features. Sheets are designed to help...

🕒 4 min read

Subnet Mask Cheat Sheet

A Subnet Mask is a numerical value that describes a computer or device's how to divide an IP address into two parts: the network portion and the host portion. The network element identifies the network to which...

🕒 9 min read

Git Cheat Sheet

Git Cheat Sheet is a comprehensive quick guide for learning Git concepts, from very basic to advanced levels. By this Git Cheat Sheet, our aim is to provide a handy reference tool for both beginners and...

🕒 10 min read

NumPy Cheat Sheet: Beginner to Advanced (PDF)

NumPy stands for Numerical Python. It is one of the most important foundational packages for numerical computing & data analysis in Python. Most computational packages providing scientific functionality use...

🕒 15+ min read

Linux Commands Cheat Sheet

Linux, often associated with being a complex operating system primarily used by developers, may not necessarily fit that description entirely. While it can initially appear challenging for beginners, once you...

🕒 13 min read

Pandas Cheat Sheet for Data Science in Python

Pandas is a powerful and versatile library that allows you to work with data in Python. It offers a range of features and functions that make data analysis fast, easy, and efficient. Whether you are a data scientist,...

🕒 15+ min read

Java Cheat Sheet

Java is a programming language and platform that has been widely used since its development by James Gosling in 1991. It follows the Object-oriented Programming concept and can run programs written on a...

🕒 15+ min read

C++ STL Cheat Sheet

The C++ STL Cheat Sheet provides short and concise notes on Standard Template Library (STL) in C++. Designed for programmers that want to quickly go through key STL concepts, the STL cheatsheet covers...

🕒 15+ min read

Docker Cheat Sheet : Complete Guide (2024)

Docker is a very popular tool introduced to make it easier for developers to create, deploy, and run applications using containers. A container is a utility provided by Docker to package and run an applicatio...

🕒 11 min read

C++ Cheatsheet

This is a C++ programming cheat sheet. It is useful for beginners and intermediates looking to learn or revise the concepts of C++ programming. While learning a new language, it feels annoying to switch...

🕒 15+ min read



Corporate & Communications Address:

A-143, 7th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305)



Registered Address:

K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305



Advertise with us

Company

- About Us
- Legal
- Privacy Policy
- In Media
- Contact Us
- Advertise with us
- GFG Corporate Solution
- Placement Training Program

DSA

- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- Top 100 DSA Interview Problems
- DSA Roadmap by Sandeep Jain
- All Cheat Sheets

Languages

- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial
- Tutorials Archive

Data Science & ML

- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Python Web Scraping
OpenCV Tutorial
Python Interview Question
Django

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

Interview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

GeeksforGeeks Videos

DSA
Python
Java
C++
Web Development
Data Science
CS Subjects