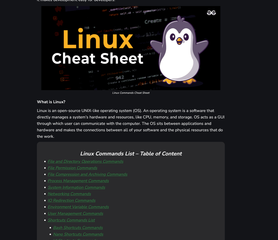
Linux Commands Cheat Sheet

Last Updated : 11 Aug, 2023

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, combining with other commands, and creating shell interfaces, can make it hard to switch back to older operating systems. If you're a developer, you can likely relate the advantages to your Linux journey.

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 Linux commands, including file and directory commands, file permission commands, file compression and archiving, process management, system information, networking, and more. We've grouped these commands and provided descriptions, in addition to that we provide all the most used Linux shell shortcuts which includes Bash shortcuts, Nano shortcuts, VI & Vim shortcuts, Keyboard shortcuts. It provides a solid foundation in 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 Commands 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 the 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 Commands
- File Permission Commands
- File Compression & Archiving Commands
- Process Management Commands
- System Information Commands
- Networking Commands
- Organization Commands
- Environment Variable Commands
- User Management Commands
- Shortcuts Commands List
  - Bash Shortcuts Commands
  - Nano Shortcuts Commands
  - V & 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.	• -l: Long format listing. • -a: Include hidden files hidden ones. • -h: Human-readable file sizes.	• **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.		• **cd /path/to/directory**: changes the current directory to the specified path.
cwd	Print current working directory.		• **pwd**: displays the current working directory.
mkdir	Create a new directory.		• **mkdir my_directory**: creates a new directory named "my_directory".
rm	Remove files and directories.	• -r: Remove directories recursively. • -f: Force removal without confirmation.	• **rm file.txt**: deletes the file named "file.txt". • **rm -r dir**: recursively deletes the directory "my_directory" and its contents. • **rm -f file.txt**: forcefully deletes the file "file.txt" without confirmation.
cp	Copy files and directories.	• -r: Copy directories recursively.	• **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.		• **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 timestamp.		• **touch file.txt**: creates an empty file named "file.txt".
cat	View the contents of a file.		• **cat file.txt**: displays the contents of the file "file.txt".
head	Display the first few lines of a file.	• -n: Specify the number of lines to display.	• **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".
tail	Display the last few lines of a file.	• -n: Specify the number of lines to display.	• **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".
tar	Create Unix between files.	• -c: Create symbolic (soft) links.	• **ln -s source_file link_name**: creates a symbolic link named "link_name" pointing to "source_file".
find	Search for files and directories.	• **-name**: Search by filenames. • **-type**: Search by file type.	• **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.	• **u**: User/owner permissions. • **g**: Group permissions. • **o**: Other permissions. • **+**: Add permissions. • **-**: Remove permissions. • **=**: Set permissions explicitly.	• **chmod u+rwx file.txt**: grants read, write, and execute permissions to the owner of the file.
chown	Change file ownership.		• **chown user file.txt**: changes the owner of "file.txt" to the specified user.
chgrp	Change group ownership.		• **chgrp group file.txt**: changes the group ownership of "file.txt" to the specified group.
umask	Set default file permissions.		• **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.	• **-c**: Create a new archive. • **-x**: Extract files from an archive. • **-f**: Specify the archive file name. • **-v**: Verbose output. • **-z**: Compress the archive with gzip. • **-j**: Compress the archive with bzip2.	• **tar -czvf archive.tar.gz files/**: creates a compressed tar archive named "archive.tar.gz" containing the files in the "files/" directory.
gzip	Compress files.	• **-d**: Decompress files.	• **gzip file.txt**: compresses the file "file.txt" and replaces it with "file.txt.gz".
zip	Create compressed zip archives.	• **-r**: Recursively include directories.	• **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.	• **aux**: Show all processes.	• **ps aux**: shows all running processes with detailed information.
top	Monitor system processes in real-time.		• **top**: displays a dynamic view of system processes and their resource usage.
kill	Terminate a process.	• **-9**: Forcefully kill a process.	• **kill PID**: terminates the process with the specified process ID.
killall	Terminate processes based on their name.		• **pkill process_name**: Terminates all processes with the specified name.
pgrep	List processes based on their name.		• **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.	• -i: Ignore case sensitivity when matching patterns. • -v: Invert the match, displaying non-matching lines. • **-r or -R**: Recursively search directories for matching patterns. • -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. • **-o**: 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.	• **grep -i "hello" file.txt** • **grep -v "error" file.txt** • **grep -n "pattern" file.txt** • **grep -c "keyword" file.txt** • **grep -o "pattern" file.txt** In these examples, we are extracting our desire output from filename (file.txt)

5. System Information Commands

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

nuCommand	Description	Options	Examples
uname	Print system information.	• **-a**: Act system information.	• **uname -a**: displays all system information.
whoami	Display current username.		• **whoami**: shows the current username.
df	Show disk space usage.	• **-h**: Human-readable sizes.	• **df -h**: displays disk space usage in a human-readable format.
du	Estimate file and directory sizes.	• **-h**: Human-readable sizes. • **-s**: Display total size only.	• **du -sh directory**: provides the total size of the specified directory.
free	Display memory usage statistics.	• **-h**: Human-readable sizes.	• **free -h**: displays memory usage in a human-readable format.
uptime	Show system uptime.		• **uptime**: shows the current system uptime.
lscpu	Display CPU information.		• **lscpu**: provides detailed CPU information.
lspci	List PCI devices.		• **lspci**: List PCI devices.
lsusb	List USB devices.		• **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
ifconfig	Display network interface information.	• **ifconfig**: shows the details of all network interfaces.
ping	Send ICMP echo requests to a host.	• **ping google.com**: sends ICMP echo requests to "google.com" to check connectivity.
netstat	Display network connections and statistics.	• **netstat -tulnp**: shows all listening TCP and UDP connections.
ss	Display network socket information.	• **ss -tulnp**: Shows all listening TCP and UDP connections.
ssh	Securely connect to a remote server.	• **ssh user@hostname**: initiates an SSH connection to the specified hostname.
ftp	Securely copy files between hosts.	• **sftp file.txt user@hostname /path/to/destination**: securely copies "file.txt" to the specified remote host.
wget	Download files from the web.	• **wget http://example.com/file.txt**: downloads "file.txt" from the specified URL.
curl	Transfer data to or from a server.	• **curl http://example.com**: 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
cmd < file	Input of cmd is taken from file.
cmd > file	Standard output (stdout) of cmd is redirected to file.
cmd 2> file	Error output (stderr) of cmd is redirected to file.
cmd 2>&1	stderr is redirected to the same place as stdout.
cmd1 | cmd2	Output of cmd1 is used as the input file for cmd1.
cmd > /dev/null	Discards the stdout of cmd by sending it to the null device.
cmd 0> file	Every output of cmd is redirected to file.
cmd 1>&2	stdout is redirected to the same place as stderr.
cmd >> file	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	Search and Replace	Desc
Ctrl + O	Save the file.	**Ctrl + Y**	Scroll up one page.	**Ctrl + K**	Cut/delete from the cursor position to the end of the line.	**Ctrl + W**	Sear strin t
Ctrl + X	Exit Nano (prompt to save if modified).	**Ctrl + V**	Scroll down one page.	**Ctrl + U**	Uncut/restore the last cut text.	**Alt + W**	Sear rep strin t
Ctrl + R	Read a file into the current buffer.	**Alt + **	Go to a specific line number.	**Ctrl + 6**	Mark a block of text for copying or cutting.	**Alt + R**	Rep Last
Ctrl + J	Justify the current paragraph.	**Alt + .**	Go to the beginning of the current line.	**Ctrl + K**	Cut/delete 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.
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.
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.		
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, Keras, Pandas and Python basics.*

FAQs on Linux Commands Cheat Sheet

1. What is Linux Cheat Sheet?

*When your memory fails or you prefer not to rely on '**linux --help?**' in the Terminal, this linux cheat sheet comes to the rescue. It is hard to memorize all the important linux Commandsby heart, so print this out or save it to your desktop to resort to when you get stuck.*

2.What are the basics of Linux?

- ***Kernel**. The base component of the OS. Without it, the OS doesn't work. ...*
- ***System user space**. The administrative layer for system-level tasks like configuration and software install. ...*
- ***Applications**. A type of software that lets you perform a task.*

3. What is 777 in Linux command?

*You might have heard of chmod 777. This command will **give read, write and execute permission to the owner, group and public**.*

4. How do I see what users are doing in Linux?

*Using the **w Command**, w command in Linux shows logged-in users and their activities.*

Here's a complete roadmap for you to become a developer: **Learn DSA -> Master Frontend/Backend/Full Stack -> Build Projects -> Keep Applying to Jobs**

And why go anywhere else when our DSA to Development: Coding Guide helps you do this in a single program! Apply now to our DSA to Development Program and our counsellors will connect with you for further guidance & support.

15

Suggest improvement

Previous
NumPy Cheat Sheet: Beginner to Advanced (PDF)

Next
Pandas Cheat Sheet for Data Science in Python

Share your thoughts in the comments

Add Your Comment

Similar Reads

jQuery Cheat Sheet - A Basic Guide to jQuery

Tkinter Cheat Sheet

CSS Cheat Sheet - A Basic Guide to CSS

Git Cheat Sheet

ggplot2 Cheat Sheet

Subnet Mask Cheat Sheet

C++ STL Cheat Sheet

Python OpenCV Cheat Sheet

C Cheat Sheet

React Cheat Sheet

jayesh962

+ Follow

Article Tags : Cheat Sheet, GFG Sheets, linux-commands, Linux Unix

GeeksforGeeks

A-143, 9th Floor, Sovereign Corporate Tower, Sector-136, Noida, Uttar Pradesh - 201305

Company
About Us
Legal
Careers
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program

Explore
Hack-A-Thons
GFG Weekly Contest
DSA in JAVA/C++
Master System Design
Master CP
GeeksforGeeks Videos
Geeks Community

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

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

Data Science & ML
Data Science With Python
Data Science For Beginner
Machine Learning Tutorial
ML Maths
Data Visualization Tutorial
Pandas Tutorial
NumPy Tutorial
NLP Tutorial
Deep Learning Tutorial

HTML & CSS
HTML
CSS
Web Templates
CSS Frameworks
Bootstrap
Tailwind CSS
SASS
LESS
Web Design
Synge Tutorial

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

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

DevOps
Git
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

Competitive Programming
Top DS or Algo for CP
Top 50 Tree
Top 50 Graph
Top 50 Array
Top 50 String
Top 50 DP
Top 15 Websites for CP

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

JavaScript
JavaScript Examples
TypeScript
ReactJS
NextJS
AngularJS
NodeJS
Lodash
Web Browser

Preparation Corner
Company Wise Recruitment Process
Resume Templates
Aptitude Preparation
Puzzles
Company-Wise Preparation

School Subjects
Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
World GK

Management & Finance
Management
HR Management
Finance
Income Tax
Organizational Behaviour
Marketing

Free Online Tools
Typing Test
Image Editor
Code Formatters
Code Converters
Currency Converter
Random Number Generator
Random Password Generator

More Tutorials
Software Development
Software Testing
Product Management
SAP
SEO - Search Engine Optimization
Linux
Excel

GeeksforGeeks Videos
DSA
Python
Java
C++
Data Science
CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved