TOP 100 LINUX COMMANDS WITH EXAMPLE

> _A MASTER KEY FOR BEGINNERS

Top 100+ Linux Commands

Linux is one of the most popular open-source operating systems, widely used for its flexibility, stability, and security. It provides a robust command-line interface (CLI) that allows users to interact with the system directly, offering powerful features for file manipulation, system administration, networking, and automation.
This guide is designed to help users harness the full potential of Linux through a comprehensive list of commands. Whether you are a novice or an experienced Linux user, mastering these commands will significantly enhance your productivity and ability to work efficiently within the Linux environment.

Throughout this document, you will find examples of commonly used Linux commands, complete with explanations, options, and practical use cases.

Top 100+ Linux Commands With Example

From simple file management tasks to complex system administration, each command has been carefully selected to provide a thorough overview of what Linux has to offer.

By the end of this guide, you will have gained the knowledge to:

- Navigate and manipulate the Linux file system with ease.
- Monitor system resources and processes.
- Manage users and permissions securely.
- Perform networking tasks and manage data efficiently.

The goal of this document is not just to list commands, but to help you understand how they work, how to combine them, and how to use them effectively in real-world scenarios.

Abstract

This document provides an extensive list of 100+ Linux commands, including essential commands for managing files, monitoring system performance, handling user accounts, networking, and more. Each command is explained in simple terms, accompanied by practical examples, to help users, from beginners to advanced, make the most out of their Linux environment.

Linux commands are powerful tools for administrators, developers, and everyday users to perform a wide range of tasks. Whether you're navigating the file system, editing text files, managing processes, or automating tasks, this guide covers it all. With this collection, users will not only learn the syntax and options of each command but also gain a deeper understanding of how they interact within the Linux ecosystem.

Table of Contents

Basic Linux Commands

- **ls** – Lists directory contents.
- **cd** – Changes the current directory.
- **pwd** – Prints the current working directory.
- **mkdir** – Creates a new directory.
- **rm** – Removes files or directories.
- **cp** – Copies files or directories.
- **mv** – Moves or renames files or directories.
- **touch** – Creates a new empty file or updates the timestamp of an existing file.
- **cat** – Displays the contents of a file.
- **more** – Views the contents of a file one screen at a time.
- **less** – Allows backward and forward navigation through a file.
- **head** – Displays the first few lines of a file.
- **tail** – Displays the last few lines of a file.

File Manipulation Commands

- **ln** – Creates hard and symbolic links.
- **chmod** – Changes file permissions.
- **chown** – Changes file ownership.
- **chgrp** – Changes group ownership.
- **stat** – Displays detailed information about a file or directory.
- **cp -r** – Copies directories recursively.
- **rm -r** – Removes directories and their contents recursively.
- **mv -u** – Moves or renames files only when the source is newer than the destination.
- **rename** – Renames multiple files at once based on a pattern.
- **split** – Splits a file into multiple smaller.
- **join** – Joins two files based on a common field.

File Manipulation Commands

- **tar** – Archives files into a tarball.
- **gzip** – Compresses files using the gzip algorithm.
- **gunzip** – Decompresses files that were compressed using gzip.
- **zip** – Compresses files into a zip archive.
- **unzip** – Extracts files from a zip archive.
- **bzip2** – Compresses files using the bzip2 algorithm.
- **bunzip2** – Decompresses files that were compressed using bzip2.
- **xz** – Compresses files using the xz algorithm.
- **unxz** – Decompresses files that were compressed using xz.
- **wc** – Counts the number of lines, words, and bytes in a file.

Search and Text Processing

- **grep**: Search for patterns in files.
- **find**: Search for files and directories.
- **awk**: Process and manipulate text.
- **sed**: Edit text in streams.
- **cut**: Extract parts of lines.
- **sort**: Sort lines of text.
- **uniq**: Remove or identify duplicate lines.
- **wc**: Count words, lines, or characters.
- **head**: Show the first few lines of a file.
- **tail**: Show the last few lines of a file.
- **tr**: Translate or delete characters.
- **diff**: Compare two files line by line.
- **comm**: Compare sorted files line by line.
- **xargs**: Build and execute command lines from input.
- **tee**: Read from stdin and write to stdout and files.

System Information and Monitoring

- **grep**: Search for patterns in files.
- **find**: Search for files and directories.
- **awk**: Process and manipulate text.
- **sed**: Edit text in streams.
- **cut**: Extract parts of lines.
- **sort**: Sort lines of text.
- **uniq**: Remove or identify duplicate lines.
- **wc**: Count words, lines, or characters.
- **head**: Show the first few lines of a file.
- **tail**: Show the last few lines of a file.
- **tr**: Translate or delete characters.
- **diff**: Compare two files line by line.
- **comm**: Compare sorted files line by line.
- **xargs**: Build and execute command lines from input.
- **tee**: Read from stdin and write to stdout and files.

System Information and Monitoring

- **top**: Displays real-time system resource usage.
- **htop**: Interactive version of top.
- **uptime**: Shows system uptime and load averages.
- **free**: Displays memory usage.
- **df**: Shows disk space usage.
- **du**: Displays disk usage for files and directories.
- **ps**: Lists running processes.
- **vmstat**: Shows system performance statistics.
- **lscpu**: Displays CPU information.
- **lsblk**: Lists block devices (disks, partitions).

System Information and Monitoring

- **lsblk**: Lists block devices (disks, partitions).
- **dmesg**: Displays kernel messages.
- **iostat**: Shows CPU and I/O statistics.
- **netstat**: Displays network connections and stats.
- **ss**: Socket statistics (modern netstat).
- **who**: Lists logged-in users.
- **w**: Displays user info and activities.

User Management

- **useradd**: Adds a new user to the system.
- **usermod**: Modifies an existing user account.
- **userdel**: Deletes a user account.
- **groupadd**: Adds a new user group.
- **groupdel**: Deletes a user group.
- **passwd**: Changes a user's password.
- **chage**: Changes user password expiration information.
- **id**: Displays user and group IDs.
- **whoami**: Displays the current logged-in user.
- **groups**: Shows the user groups.
- **last**: Shows the last login details for users.
- **finger**: Displays information about a user.
- **sudo**: Executes a command as another user, typically root.

User Management

- **useradd**: Adds a new user to the system.
- **usermod**: Modifies an existing user account.
- **userdel**: Deletes a user account.
- **groupadd**: Adds a new user group.
- **groupdel**: Deletes a user group.
- **passwd**: Changes a user's password.
- **chage**: Changes user password expiration information.
- **id**: Displays user and group IDs.
- **whoami**: Displays the current logged-in user.
- **groups**: Shows the user groups.
- **last**: Shows the last login details for users.
- **finger**: Displays information about a user.
- **sudo**: Executes a command as another user, typically root.

Networking Commands

- **ifconfig**: Displays or configures network interfaces (deprecated).
- **ip**: Configures network settings (IP addresses, routes).
- **ping**: Tests network connectivity.
- **traceroute**: Traces the path to a remote host.
- **netstat**: Displays network connections and stats.
- **ss**: Shows socket statistics (modern netstat).
- **route**: Views or modifies the routing table.
- **nslookup**: Queries DNS records.
- **dig**: Advanced DNS query tool.
- **curl**: Transfers data to/from a server.
- **wget**: Downloads files from the web.

Networking Commands

- **nmap**: Scans networks for hosts and services.
- **hostname**: Displays or sets the system's hostname.
- **scp**: Securely copies files between hosts.
- **ssh**: Connects to a remote machine securely.
- **ftp**: Transfers files via FTP.

Permissions and Ownership

- **chmod**: Changes file permissions.
- **chown**: Changes file ownership (user and group).
- **chgrp**: Changes the group ownership of a file.
- **umask**: Sets default file creation permissions.
- **ls -l**: Displays detailed file information, including permissions.
- **getfacl**: Displays file access control list (ACL) information.
- **setfacl**: Sets file access control lists (ACLs) for more granular permissions.

Permissions and Ownership

- **tmux**: Terminal multiplexer for managing multiple sessions.
- **screen**: Another terminal multiplexer for running multiple sessions.
- **find**: Search for files based on criteria.
- **xargs**: Builds and executes command lines from input.
- **strace**: Traces system calls of a process.
- **lsof**: Lists open files and their associated processes.
- **nc (netcat)**: Utility for network connections.
- **awk**: Text processing and pattern matching tool.
- **sed**: Stream editor for text manipulation.
- **dd**: Low-level file copying and conversion.

Advanced Commands

- **tmux**: Terminal multiplexer for managing multiple sessions.
- **screen**: Another terminal multiplexer for running multiple sessions.
- **find**: Search for files based on criteria.
- **xargs**: Builds and executes command lines from input.
- **strace**: Traces system calls of a process.
- **lsof**: Lists open files and their associated processes.
- **nc (netcat)**: Utility for network connections.
- **awk**: Text processing and pattern matching tool.
- **sed**: Stream editor for text manipulation.
- **dd**: Low-level file copying and conversion.

Conclusion

Mastering Linux commands is essential for efficient system administration, automation, and troubleshooting. The commands covered in this guide provide a solid foundation for managing files, users, networks, system resources, and more. By becoming familiar with both basic and advanced commands, you can significantly improve your productivity and ability to handle a wide range of tasks in a Linux environment.

Whether you're just starting out or looking to deepen your knowledge, practicing these commands will help you gain confidence and become more proficient in using Linux for both personal and professional purposes.

Conclusion

Feel free to explore and experiment with these commands to unlock the full potential of your Linux system!
Let me know if you need further clarifications or examples for any command!

Conclusion

https://eeebuntu.org/linux-commands/
A comprehensive list of common Linux commands with examples and explanations.

Ubuntu Manual
https://ubuntu-manual.org/
An official manual for Ubuntu users, covering installation, commands, and system administration.

Linux.org Wiki
https://www.linux.org/
A community-driven site with articles and tutorials on various Linux topics, including commands and usage.