Basic Linux Commands Cheat Sheet With Examples [Updated]

Basic Linux Commands Cheat Sheet With Examples [Updated]

Master Linux commands with our comprehensive guide. Learn to navigate the file system, manage users, and optimize your Linux system with our cheat sheet and real-world examples. Whether you're a sysadmin, developer, or data scientist, take your Linux skills to the next level.

Server Tutorials - Aug 30, 21 - by Susith Nonis - 17 min Read

List of content you will read in this article:

- 1. What are Linux commands?
- 2. What is a Command-line interface (CLI)?
- 3. Basic Linux Commands Cheat Sheet
- 4. Advanced Linux Commands List
- 5. Networking Linux Commands
- 6. Linux File Management Commands
- 7. Linux System Monitoring Commands
- 8. Linux Security Commands
- 9. Linux Commands for Compressing Files and Folders
- 10. Linux Commands for User Management
- 11. Some Additional Linux Commands
- 12. Real-World Scenarios Where Linux Commands Can Be Helpful
- 13. Conclusion

What are Linux commands?

Linux commands are a set of text-based instructions that are used to interact with a Linux operating system. They are executed through the command-line interface (CLI), which allows users to enter text commands to perform various tasks, such as file management, system monitoring, networking, and security.

Linux commands are an essential part of Linux and are used by system administrators, developers, and everyday users to manage and operate the system. They provide a powerful and flexible way to interact with the operating system and can be used to automate tasks, troubleshoot issues, and improve productivity.

Some common Linux commands include ls (list files and directories), cd (change directory), cp (copy files), mv (move or rename files), rm (remove files), mkdir (create directories), ping (check network connectivity), and top (view system processes).

By learning Linux commands, users can become more proficient in working with Linux and take advantage of the many benefits of the command-line interface.

What is a Command-line interface (CLI)?

A command-line interface (CLI) is a text-based way of interacting with a computer's operating system. It allows users to enter commands as text and execute them by pressing the "Enter" key.

The CLI is a powerful and flexible way to interact with a computer's operating system and is used by many operating systems, including Linux, macOS, and Windows. It provides a way to perform tasks that may not be possible or practical to do through a graphical user interface (GUI).

In the context of Linux, the CLI is an essential part of the system and is often used for tasks such as system administration, software development, and networking. Linux commands are entered through the CLI and are used to perform a wide variety of tasks, from basic file management to more advanced system administration and networking tasks.

Using the CLI can take some getting used to, as it requires users to remember and type commands and may not have the visual cues and feedback of a GUI. However, once users become familiar with the CLI, they may find that it is a more efficient and flexible way to work with a computer's operating system.

Basic Linux Commands Cheat Sheet

Several basic Linux commands are essential for users to know. These commands allow users to navigate the file system, create and delete directories and files, and check their current working directories.

ls: List files and directories in the current directory
Example: ls

```
ls
```

cd: Change directory to a different directory
Example: cd /home/user/documents

```
cd /home/user/documents
```

pwd: Print the current working directory
Example: pwd

pwd

mkdir: Create a new directory
Example: mkdir new_directory

```
mkdir new_directory
```

touch: Create a new empty file or update the access and modification time of an existing file
Example: touch file.txt

```
touch file.txt
```

cp: Copy files or directories from one location to another
Example: cp file.txt /home/user/documents

```
cp file.txt /home/user/documents
```

mv: Move or rename files and directories
Example: mv file.txt /home/user/documents

```
mv file.txt /home/user/documents
```

rm: Remove files and directories
Example: rm file.txt

```
rm file.txt
```

rmdir: Remove an empty directory
Example: rmdir empty_directory

```
rmdir empty_directory
```

man: Display the manual pages for a specific command
Example: man ls

```
man ls
```

Advanced Linux Commands List

These are just a few advanced Linux commands that are commonly used by system administrators and developers to perform complex tasks and automate workflows. By

mastering these commands, users can work more efficiently and effectively with Linux systems.

grep: Search for a specific string or pattern in a file or output
Example: grep "error" log.txt

```
grep "error" log.txt
```

sed: Stream editor for text transformation and processing
Example: sed 's/old_text/new_text/g' file.txt

```
sed 's/old_text/new_text/g' file.txt
```

awk: Pattern scanning and processing language for text files
Example: awk '{print $1}' file.txt

```
awk '{print $1}' file.txt
```

find: Search for files or directories that match a specific pattern or criteria
Example: find /home/user -name "*.txt"

```
find /home/user -name "*.txt"
```

tar: Archive files into a single file and compress it
Example: tar -czvf archive.tar.gz /home/user/documents

```
tar -czvf archive.tar.gz /home/user/documents
```

rsync: Efficiently transfer and synchronize files between two locations
Example: rsync -avz /home/user/documents/ user@remote_host:/home/user/documents/

```
rsync -avz /home/user/documents/ user@remote_host:/home/user/documents/
```

curl: Transfer data from or to a server using a variety of protocols
Example: curl http://example.com/file.txt

```
curl http://example.com/file.txt
```

ssh: Secure shell protocol for remote login to a server
Example: ssh user@remote_host

```
ssh user@remote_host
```

ping: Test network connectivity to a server or website
Example: ping google.com

```
ping google.com
```

top: Monitor system resource usage and running processes
Example: top

```
top
```

Networking Linux Commands

These are just a few networking commands in Linux that system administrators and networking professionals commonly use to manage and troubleshoot networks. By mastering these commands, users can more effectively manage and optimize network performance on Linux systems.

ifconfig: Configure and view network interface settings
Example: ifconfig eth0

```
ifconfig eth0
```

ip: Configure and view IP addresses and routes
Example: ip addr show

```
ip addr show
```

netstat: View network connections and statistics
Example: netstat -a

```
netstat -a
```

ping: Test network connectivity to a server or website
Example: ping google.com

```
ping google.com
```

traceroute: Trace the path of network packets between a local and remote host
Example: traceroute google.com

```
traceroute google.com
```

route: Configure and view IP routing tables
Example: route -n

```
route -n
```

nslookup: Query DNS servers for domain name resolution
Example: nslookup google.com

```
nslookup google.com
```

dig: Domain Information Groper, query DNS servers for domain name resolution and DNS-related information
Example: dig google.com

```
dig google.com
```

wget: Retrieve files from the internet using HTTP, HTTPS, or FTP protocols
Example: wget https://example.com/file.txt

```
wget https://example.com/file.txt
```

curl: Transfer data from or to a server using a variety of protocols
Example: curl http://example.com/file.txt

```
curl http://example.com/file.txt
```

Linux File Management Commands

These are just a few file management commands in Linux that system administrators and everyday users commonly use to manage and organize files and directories. By mastering these commands, users can perform various file management tasks, from creating and deleting files and directories to monitoring disk space usage.

ls: List files and directories in the current directory
Example: ls

```
ls
```

cd: Change directory to a different directory
Example: cd /home/user/documents

```
cd /home/user/documents
```

pwd: Print the current working directory
Example: pwd

```
pwd
```

mkdir: Create a new directory
Example: mkdir new_directory

```
mkdir new_directory
```

touch: Create a new empty file or update the access and modification time of an existing file

Example: touch file.txt

```
touch file.txt
```

cp: Copy files or directories from one location to another
Example: cp file.txt /home/user/documents

```
cp file.txt /home/user/documents
```

mv: Move or rename files and directories
Example: mv file.txt /home/user/documents

```
mv file.txt /home/user/documents
```

rm: Remove files and directories
Example: rm file.txt

```
rm file.txt
```

rmdir: Remove an empty directory
Example: rmdir empty_directory

```
rmdir empty_directory
```

du: Display the disk space used by files and directories
Example: du -h file.txt

```
du -h file.txt
```

df: Display the disk space used and available for a file system
Example: df -h /dev/sda1

```
df -h /dev/sda1
```

cat: Display the contents of a file
Example: cat file.txt

```
cat file.txt
```

less: Display the contents of a file one page at a time
Example: less file.txt

```
less file.txt
```

head: Display the first few lines of a file
Example: head file.txt

```
head file.txt
```

tail: Display the last few lines of a file
Example: tail file.txt

```
tail file.txt
```

Linux System Monitoring Commands

These are just a few system monitoring commands in Linux that are commonly used by system administrators and developers to monitor system performance and resource usage. By mastering these commands, users can more effectively manage and optimize system performance on Linux systems.

top: Display system processes and their resource usage
Example: top

```
top
```

ps: Display running processes and their status
Example: ps aux

```
ps aux
```

free: Display memory usage and available memory
Example: free -h

```
free -h
```

vmstat: Display system virtual memory statistics
Example: vmstat 1

```
vmstat 1
```

sar: Collect and report system resource usage
Example: sar -u 1 5

```
sar -u 1 5
```

iostat: Display system input/output statistics
Example: iostat -x 1

```
iostat -x 1
```

uptime: Display system uptime and load average
Example: uptime

```
uptime
```

netstat: Display network connections and statistics

Example: netstat -a

```
netstat -a
```

iftop: Display network interface bandwidth usage
Example: iftop -i eth0

```
iftop -i eth0
```

htop: Interactive process viewer and system monitor
Example: htop

```
htop
```

Linux Security Commands

These are just a few security commands in Linux that system administrators and security professionals commonly use to secure Linux systems and protect them against security threats. By mastering these commands, users can more effectively manage and optimize system security on Linux systems.

passwd: Change the password for the current user
Example: passwd

```
passwd
```

su: Switch to another user account
Example: su root

```
su root
```

sudo: Run a command with superuser privileges
Example: sudo ls

```
sudo ls
```

chroot: Change the root directory for a command or process
Example: chroot /mnt/chroot/

```
chroot /mnt/chroot/
```

chmod: Change file or directory permissions
Example: chmod 644 file.txt

```
chmod 644 file.txt
```

chown: Change the owner of a file or directory
Example: chown user:group file.txt

```
chown user:group file.txt
```

iptables: Configure firewall rules
Example: iptables -L

```
iptables -L
```

fail2ban: Intrusion prevention system that blocks IP addresses after failed login attempts
Example: fail2ban-client status sshd

```
fail2ban-client status sshd
```

openssl: Tool for managing public key infrastructures and SSL/TLS connections
Example: openssl x509 -in cert.pem -text

```
openssl x509 -in cert.pem -text
```

auditd: System audit daemon that records system events and logs them for review
Example: auditctl -w /etc/passwd -p rwxa -k password_change

```
auditctl -w /etc/passwd -p rwxa -k password_change
```

Linux Commands for Compressing Files and Folders

These are just a few Linux commands for compressing and decompressing files and directories. By using these commands, users can reduce the size of large files and directories for storage or transfer, and extract compressed files when needed.

gzip: Compress files using the gzip algorithm
Example: gzip file.txt

```
gzip file.txt
```

gunzip: Decompress files compressed with gzip
Example: gunzip file.txt.gz

```
gunzip file.txt.gz
```

tar: Create an archive of files and directories, optionally compressed
Example: tar -czvf archive.tar.gz /home/user/documents

```
tar -czvf archive.tar.gz /home/user/documents
```

xz: Compress files using the xz algorithm
Example: xz file.txt

```
xz file.txt
```

unxz: Decompress files compressed with xz
Example: unxz file.txt.xz

```
unxz file.txt.xz
```

zip: Compress files into a zip archive
Example: zip archive.zip file.txt

```
zip archive.zip file.txt
```

unzip: Extract files from a zip archive
Example: unzip archive.zip

```
unzip archive.zip
```

Linux Commands for User Management

These are just a few Linux commands for managing users and groups on a Linux system. By using these commands, system administrators can create and delete user accounts, modify user and group settings, and control access to system resources.

useradd: Create a new user account
Example: useradd newuser

```
useradd newuser
```

passwd: Change the password for a user account
Example: passwd newuser

```
passwd newuser
```

usermod: Modify an existing user account
Example: usermod -s /bin/bash newuser

```
usermod -s /bin/bash newuser
```

userdel: Remove a user account
Example: userdel olduser

```
userdel olduser
```

groupadd: Create a new group
Example: groupadd newgroup

```
groupadd newgroup
```

groupmod: Modify an existing group
Example: groupmod -n newgroup oldgroup

```
groupmod -n newgroup oldgroup
```

groupdel: Remove a group
Example: groupdel oldgroup

```
groupdel oldgroup
```

chown: Change the owner of a file or directory
Example: chown user:group file.txt

```
chown user:group file.txt
```

chmod: Change file or directory permissions
Example: chmod 644 file.txt

```
chmod 644 file.txt
```

su: Switch to another user account
Example: su - newuser

```
su - newuser
```

sudo: Run a command with superuser privileges
Example: sudo ls

```
sudo ls
```

id: Display user and group information for the current user or a specified user
Example: id

```
id
```

Some Additional Linux Commands

These are just a few additional Linux commands that can be useful for a variety of tasks and scenarios. By mastering these commands, users can work more efficiently and effectively with Linux systems.

df: Display disk space usage for mounted file systems
Example: df -h

```
df -h
```

fdisk: Manipulate disk partition table
Example: sudo fdisk -l

```
sudo fdisk -l
```

tar: Archive files into a single file and compress it
Example: tar -czvf archive.tar.gz /home/user/documents

```
tar -czvf archive.tar.gz /home/user/documents
```

crontab: Schedule recurring tasks or commands to run at specific times or intervals
Example: crontab -e

```
crontab -e
```

awk: Pattern scanning and processing language for text files
Example: awk '{print $1}' file.txt

```
awk '{print $1}' file.txt
```

sed: Stream editor for text transformation and processing
Example: sed 's/old_text/new_text/g' file.txt

```
sed 's/old_text/new_text/g' file.txt
```

du: Display the disk space used by files and directories
Example: du -h file.txt

```
du -h file.txt
```

ps: Display running processes and their status
Example: ps aux

```
ps aux
```

kill: Terminate running processes
Example: kill PID

```
kill PID
```

locate: Search for files or directories that match a specific pattern or criteria
Example: locate file.txt

```
locate file.txt
```

echo: Display text or values to the console
Example: echo "Hello, World!"

```
echo "Hello, World!"
```

history: Display a list of previously executed commands

Example: history

```
history
```

uname: Display system information, including the kernel version and operating system name
Example: uname -a

```
uname -a
```

chroot: Run a command or process with a modified root directory
Example: chroot /mnt/chroot /bin/bash

```
chroot /mnt/chroot /bin/bash
```

date: Display or set the system date and time
Example: date

```
date
```

Real-World Scenarios Where Linux Commands Can Be Helpful

1. Managing a web server: System administrators use Linux commands to manage web servers, such as Apache or Nginx, to ensure that web applications run smoothly. Commands such as ps and top can monitor server performance, while cp and mv can manage files and directories on the server.

2. Troubleshooting a network issue: Network administrators use Linux commands to troubleshoot issues like slow internet speeds or connectivity problems. Commands such as ping, traceroute, and netstat can be used to diagnose network issues and identify the source of the problem.

3. Developing software: Developers use Linux commands to write and compile code for software applications. Commands such as gcc and make can be used to compile C and C++ code, while git can be used to manage version control and collaborate with other developers.

4. Managing data centers: Data center operators use Linux commands to manage large-scale computing infrastructure and ensure that servers and storage systems are running optimally. Commands such as df, du, and mount can be used to monitor and manage storage capacity, while iptables can be used to manage firewall rules.

5. Performing data analysis: Data scientists use Linux commands to analyze and process large datasets for research and analysis. Commands such as grep, awk, and sed can be used to search and manipulate data, while sort, uniq, and cut can summarize and transform data.

These are just a few examples of how Linux commands are used in real-world scenarios. By mastering these commands, users can become more proficient in their respective fields and perform tasks more efficiently and effectively.

Conclusion

I hope this basic Linux commands manual will help you execute your many tasks and make it easier while executing. The above Linux commands list is the top pick by our community members, who are experts and Linux administrators. This article is also used for Linux terminal commands. If you feel we have missed listing any Linux commands, which is also the most helpful command, comment in the below-listed comment section.

Also, you can try to practice these commands with Linux VPS Server.

People Are Also Reading:

- CAT Command in Linux with examples
- Zip Command in Linux with Examples
- Linux VIM Commands with Examples
- The chmod command in Linux
- Basic Git Commands with Examples
- Linux Ping Command
- How to Move a File Directory in Linux