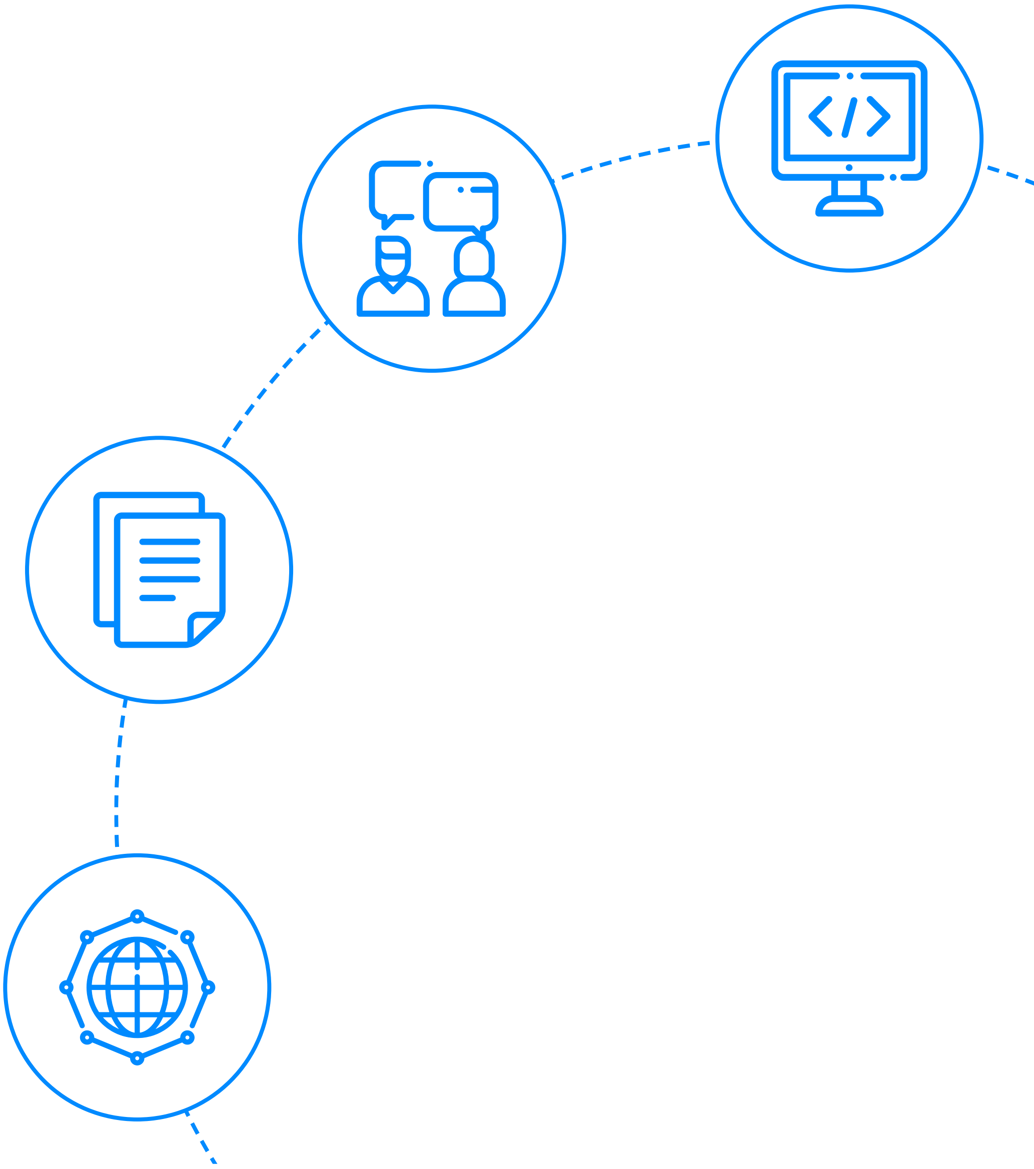
InterviewBit

Linux Commands Cheat Sheet



To view the live version of the page, click here.

Contents

Linux Commands Tutorial: Basics to Advanced

Let's get Started

Introduction

Linux is a very famous, open-source operating system. Many developers use Linux for development purposes because of its high throughput. As a student or even a professional in the software industry, it is very essential to have knowledge of the Linux OS. Many programmers prefer Linux over Windows OS for development purposes due to a variety of reasons such as the security of the Linux Operating System is better than Windows, the Linux terminal is way superior to the windows command line in many ways, etc.

What is Linux and Why use it?

Before we jump into studying a lot of Linux commands, it is very important to address this question **What is Linux** and why is it preferred over Windows OS. Linux is an open-source operating system whose source code is available for modification and commercial and non-commercial distribution under the guidelines of the GNU General Public License.

Linux has a number of advantages over the Windows operating system and is used widely because of these advantages Below are a few listed:

- The major advantage of the Linux Operating System is that it is a freely available open-source OS. This means that the source code of the Linux Operating System is available openly in the market (on the internet) and anyone can study it, modify it and even send it forward for commercial uses too. However, the major use case of it being an Open Source OS is the study of the Operating System itself. If anyone has to understand the Operating System as a core subject, it's working, and what are the things kept in mind while designing an operating system, you can simply look up the code of Linux OS and get a plethora of ideas.
- Privacy and Security are very important for the users, especially nowadays, when hacking and other cyber malpractices have become so common. Linux maintains the privacy of the user and is a lot more secure than Windows OS. However, this does not mean that Linux is completely secured and can never be attacked. This simply means that it is more secure than Windows and it does not even require any antivirus software to be installed within it.
- Linux Operating System is highly stable and does not hang or requires to be restarted again and again.
- Linux is fast and easy to install and can be installed very easily from the internet.
- Linux is network friendly.

So, now that we have a fair idea about the Linux Operating System and its advantages over the Windows OS, let us now move on to the Linux Cheat Sheet and study the Linux Commands.

Here, we have a cheat sheet prepared for you to refer to all the important **Linux Commands with Examples**.

Linux Commands Tutorial: Basics to Advanced

1. File and Directory CRUD Navigation Commands

CRUD stands for Create, Read, Update, and Delete. CRUD operations are said to be the basic operations on any file or directory or database. Even if you are not a Linux User, file and directory CRUD operations are something that you should be comfortable with.

COMMAND	MEANING	EXAMPLE & SYNTAX
`ls (list all directories)`	Lists all the files and directories inside the current directory in which you are.	Syntax: `$ ls`
`ls -R`	Lists all the files and directories inside the current directory as well as all the files and directories of the sub-directories as well.	Syntax: `$ ls -R`
`ls -a`	Lists all the files and directories in the current directory and also lists the hidden files (such as .git files). However, this command does not list the files and directories of	Syntax: `$ ls -a`

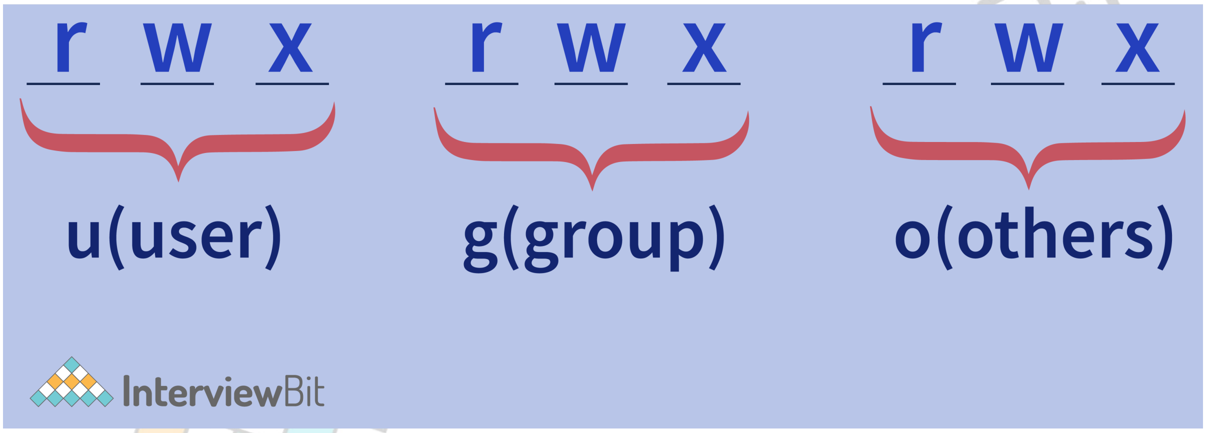
2. System Information Commands

These are some of the general-purpose system information commands that are important to know and easy to remember.

COMMAND	MEANING	EXAMPLE & SYNTAX
`history`	This command displays the list of all the typed commands in the current terminal session.	Syntax: `$ history`
`clear`	Clears the terminal i.e. no previous command will be visible on the screen now.	Syntax: `$ clear`
`hostname`	Shows the name of the system host.	Syntax: `$ hostname`
`hostid`	Displays the id of the host of the system.	Syntax: `$ hostid`
`sudo`	Allows a regular user to run the programs with the security privileges of a superuser or root.	Syntax: `$ sudo`
`apt-get`	This command is used to install and add new packages.	Syntax: `$ apt-get`

3. File Permission Commands

There are 3 types of people who can use a file and each type has 3 types of access to the file. This is shown in the diagram given below:



The diagram shows that there are 3 types of people accessing a file and they are:

1. User (u)
2. Group (g)
3. Others (o)

Also, the access that we want to give to each of them is of three types:

1. Read (r)
2. Write (w)
3. Execute (x)

So, each of them can have 0 or more out of these 3 permissions. Now let us understand the Linux commands that help us give these permissions to the files. One important thing to note here is that before these 9 slots of the user, group and others (read, write and execute permissions), there is also one another slot. This slot is for special files. For instance, if you something as drwxr--r--, here 'd' shows that it is a directory of which you are viewing the permissions. Further, rwx means that the user has all the three permissions where as r-- means that the group has only read permission and the write and execute permissions are not there with the group. The same is the case for others (another r--).

- **The chmod Command:**

Before we jump into the Linux file permission commands and see some examples, it is very important to understand this chmod command in detail first as understanding this command completely will clear the entire concept of file permission commands. The chmod command stands for "change-mode" which means that using this command, we can change the mode in which some user is able to access the file. This command is used to change the file permissions. The syntax can be either using symbols (characters) or numbers. We will see that in detail.

- **Symbolic Method for granting permissions:**

This is the first method of chmod command using which we can give permissions. The basic syntax is as follows:

chmod [ugoa…][-+=]perms…[,….] FILE….

Let us understand this syntax in detail.

The first set means the type of person to give access to. Here:

1. u → Stands for User
2. g → Stands for Group
3. o → Stands for Others
4. a → Stands for All the users i.e. instead of writing ugo, we can just write a.

If the user's flag is not included in the command i.e. we do not mention for which kind of people out of u, g and o, are we changing the permissions for, by default, it takes a i.e. all the users.

The second set is the set of operators. Let us see what they mean.

1. - → removes the mentioned permission
2. + → adds the mentioned permission
3. = → Changes the current permission to the mentioned permission. IF no permission is mentioned after using the = operator, all the permissions from the mentioned class are removed.

The perms stand for permission and ',' is used to separate different permissions. Let us now see the Linux commands using the symbolic notation of chmod.

COMMAND	MEANING	EXAMPLE & SYNTAX
`ls -l fileName`	This command is used to show the file permissions along with the owner and other details of the specified file.	Example: The file permissions along with the owner and other details is shown for the file file1.txt on the right. Syntax: `$ ls -l file1.txt -rw-r--r-- 1 Guneet Malhotra 197121 0 Feb 25 10:51 file1.txt`
`r`	This command represents the read permission.	Example: The command shown in the right adds the read permission to the o (other) class for the file file1.txt. Syntax: `$ chmod o+r file1.txt`
`w`	This command represents the write permission.	Example: This commands adds the write permission for a(all) i.e. user, group and others. Syntax: `$ chmod a+w file1.txt`

- **Numerical Method for granting file permissions**

There are numeric codes for each permission. They are as follows:

1. r (read) = 4
2. w (write) = 2
3. x (execute) = 1
4. No permissions = 0

The permissions number of a specific user class is represented by the sum of the values of all the permissions. For instance, if the user has read and executed permissions, but not the write permission, then the permissions number for the user will be read (4) + execute(1) = 5.

For instance, if we have to write a command to provide read and write permissions to the user, group and others, there can be many ways of doing so. Let us see one symbolic way:

- **Symbolic Way**

```
$ chmod ugo+rw file1.txt
```

We can write this in a numeric way as shown below:

- **Numeric Way**

```
$ chmod 666 file1.txt
```

Explanation: We have already studied that if we do not mention u/g/o then by default the permissions are applied to all. Also, read + write = 4 + 2 = 6. We have written 6 thrice because of applying the permissions to user, group and others. So, read and write permissions are applied to the user, group and others (666) for the file file1.txt.

4. Hardware Information Commands

Let us now see, some of the hardware information commands that give us the information about the hardware that we are using.

COMMAND	MEANING	EXAMPLE & SYNTAX
`cpu-info`	This command is used to display the information about your CPU. Note that this command is not available by default. It can be used after installation of the necessary package using **sudo apt install cpuinfo**.	Syntax: `$ cpu-info`
`free -h`	This command is used to display the free and used memory. -h is used for converting the information (to be displayed) to human-readable form.	Syntax: `$ free -h`
`lsusb -tv`	List all the USB connected devices.	Syntax: `$ lsusb -tv`

5. File and Directory Compression Commands

The files can be compressed and then extracted to save the storage. We see this happening many times in our daily lives that we have to compress some file to send it or we have to extract a downloaded file. There are several commands for file compression in Linux given below:

COMMAND	MEANING	EXAMPLE & SYNTAX
`gzip fileName`	This command is used to compress a file with gzip compression.	Example: The command to zip file1 using gzip compression is shown on the right. Syntax: `$ gzip file1`
`gunzip fileName.gz`	This command is used to unzip a file that has gzip compression.	Example: The command to unzip fileDemo.gz file with gz compression is shown on the right. Syntax: `$ gunzip fileDemo.gz`
`tar cf myDir.tar myDir`	This command is used to create an uncompressed tar archive.	Example: The command to create an uncompressed tar archive for the directory demoDir is shown on the right. Syntax: `$ tar cf demoDir.tar demoDir`
`tar cfz myDir.tar myDir`	This command is used to create a tar archive with gzip compression.	Example: The command to create gzip tar archive for the directory demoDir is shown on the right. Syntax: `$ tar cfz demoDir.tar demoDir`

6. Environment Variable Commands

COMMAND	MEANING	EXAMPLE & SYNTAX
`env`	This command displays all the environment variables.	Syntax: `$ env`
`echo $Variable`	This command displays the environment variable.	Example: The command at the right will display the INSTANCE environment variable. Syntax: `$ echo $INSTANCE=`
`unset`	This command removes a variable.	Syntax: `$ unset`

7. User Management Commands

COMMAND	MEANING	EXAMPLE & SYNTAX
`sudo adduser username`	This command is used to add a user.	Syntax: `$ sudo adduser username`
`sudo passwd -l ‘username’`	This command is used to change the password of a user.	Example: Command to change the password for user1 is shown Syntax: `$ sudo passwd -l 'user1'`
`sudo userdel -r ‘username’`	This command is used to remove a newly created user.	Example: Command to delete the newly created user1 Syntax: `$ sudo userdel -r 'user1'`
`sudo usermod -a -G GROUPNAME USERNAME`	This command is used to add a user to a particular group.	Example: The command to add user2 to group1 is shown. Syntax: `$ sudo usermod -a -G group1 user2`
`Sudo deluser USER GROUPNAME`	This command is used to remove a user from a group.	Example: The command to delete user1 from group1 is shown. Syntax: `$ sudo deluser user1 group1`

8. Networking Commands

COMMAND	MEANING	SYNTAX
`dir`	This command is used to display files in the current directory of a remote computer.	Syntax: `$ dir`
`put file`	This command is used to upload 'file' from local to the remote computer.	Syntax: `$ put file`
`get file`	This file is used to download 'file' from remote to the local computer.	Syntax: `$ get file`
`quit`	This command is used to log out.	Syntax: `$ quit`

9. Process Commands

COMMAND	MEANING	EXAMPLE & SYNTAX
`bg`	This command is used to send a process to the background.	Example: The process with id 1 is sent to the background by providing its id to bg. Syntax: `$ bg %1`
`fg`	This command is used to run a stopped process in the background.	Example: The process with id 1 is brought to the foreground with the help of this command. Syntax: `$ fg %1`
`top`	This command is used to get the details of all active processes.	Syntax: `$ top`
`ps`	This command is used to give the status of running for a user.	Syntax: `$ ps`
`ps PID`	This command gives the status of a particular process.	Example: Displays the status of the process with id 12230. Syntax: `$ ps 12230`