**What is Linux?**

Linux is an open-source operating system (OS). It was initially designed to be similar to UNIX but gradually evolved to run on various hardware from phones to supercomputers.

Linux is widely adopted across the world due to the following reasons:

* It is a free and open-source operating system.
* It is secure.
* It is flexible.
* It supports almost all the programming languages.

**Why Learn Linux along with DevOps?**

With so many new tools and developments in the market, the developers and engineers must learn some basic **Linux commands for DevOps**. It will help them organise, troubleshoot, and optimise their applications – when things go wrong.

Here is a sneak-peek of the Linux tool along with its scope and also demand in the job market:

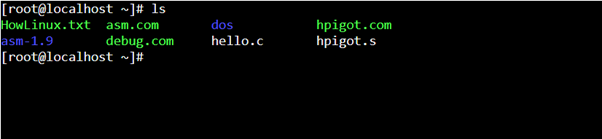
* Infrastructure Automation is amongst the major practices done in most IT companies. Linux is highly used in the domain of infrastructure automation.
* With Linux’s help, you can save time through instance creation and increase the efficiency of operations.
* By 2021, 47% of the companies will opt for Linux for major infrastructure versioning and infra-automation.

Sounds exciting? If yes, it’s time to learn some of the widely used **Linux commands for DevOps**. This blog is a compilation of some of the handy Linux commands to make your DevOps life easier and simpler.

So, without any further ado, let’s get started.

* **Ls (listing files)**

If you wish to see the list of files on your Linux system, the *ls*command will help you. Use it to see the files in your current directory.



[*source*](https://www.linuxfordevices.com/tutorials/linux/ls-command-linux-unix)

Also, to view the subdirectories, you can use the *ls-R* command.

**Note**: The command is case-sensitive. You will get an error if you write *ls-r*instead of *ls-R*.

* **tr**

The *tr*command is used to translate, delete, or squeeze the repeated characters. This command will read from STDIN and write to STDOUT.

Its syntax is:

*$ tr [flag] SET1 [SET2]*

It can also be used for counting the words. To convert lowercase to uppercase, you can run the following command:

*$ tr a-z A-Z  
Abcdef  
ABCDEF  
(abcdef)  
(ABCDEF)  
^C  
$*

* **cat**

The *cat*command is used to concatenate and print files. Developers can issue *cat*to check the contents of their dependencies file or to confirm the version of the application that they already built locally.

**Our Learners also read**: [Devops career path](https://www.upgrad.com/blog/devops-career-path-job-roles/)!

For example, to check whether a Python Flask application has Flask listed, you can run the following command:

*$ cat requirements.txt  
flask  
flask\_pymongo*

* **cut**

The *cut*command is used to extract a part of a file with the help of columns and delimiters. If you wish to list everything in a selected column, you can use the “-c” flag with the cut command.

Let’s understand this with the help of an example. To select the first two columns from a file named abc.txt, we can use the following command:

*cut -c1-2 abc.txt*

Additionally, to extract specific strings from the file, the delimiter “-d” flag is used. The “-f” flag is used to select the field.

For example, if you wish to extract all the names from the file named abc.txt, enter the following command in the Linux Command Line Input:

*cut -d’ ‘ -f2 abc.txt*

* **alias**

The *alias*command is used to personalise and organise all your commands. Amazing, isn’t it? You can use it to designate a name to a single command or even a string of commands.

For instance,

*alias cls = clear*

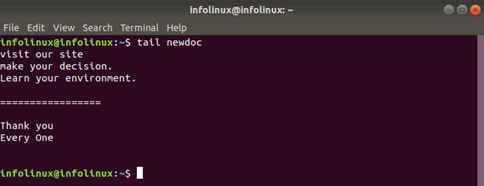
The above command will imply that *‘cls’*is the new name for the *clear*command. It is helpful for you if you are amongst the ones who frequently misspell.

* **tail**

The *tail*command is complementary to the *head*command. As the name suggests, this Linux command is used to print the last N number of data of the given input.

By default, this command prints the last 10 lines of the specified file or data. In case you input more than one filename, you will get data from each file preceded by its file name.

Its syntax is:

*tail [OPTION]… [FILE]…*

[source](https://infolinux.com/2019/03/14/linux-tail-command-for-file-system/)

* **curl**

The *curl*command is a highly useful tool to recover data from URLs or internet repositories. This command is not available by default on various Linux distributions. To install it, enter the following command in the command line:

*sudo apt-get install curl*

For example, to retrieve a particular file from a GitHub repository, you can run the following command:

*curl https://raw.githubusercontent.com/smiths/linux/master/kernel/events/core.c -o core.c*

The -o (output) option must be used to save the files on your computer.

* **chown**

The *chown* command is used to change the owner as well as the group owner of a file. To do this, you need to enter the name of the owner and the group, separated by a “:” character. You will also need to take the help of the *sudo*command.

Here is how you can use the *chown*command:

*sudo chown dave:mary abc.txt*

* **sudo**

The *sudo*command is amongst the most commonly used **Linux commands for DevOps**. *sudo*is the abbreviation for ‘SuperUser Do.’ If you wish to run any file in the Linux with administrative or root privileges, *sudo*will help in the mission.

For instance, if you wish to alter any file and need root permission to do so, *sudo*can be used to open this file as a root:

*sudo vi nginx.conf*

* **grep**

To show content (*tail*), scan for anomalies (*cat*), or find the right process (*ps aux*), you need to apply some kind of filter. To make your work simpler, you can combine pipe | and *grep.*