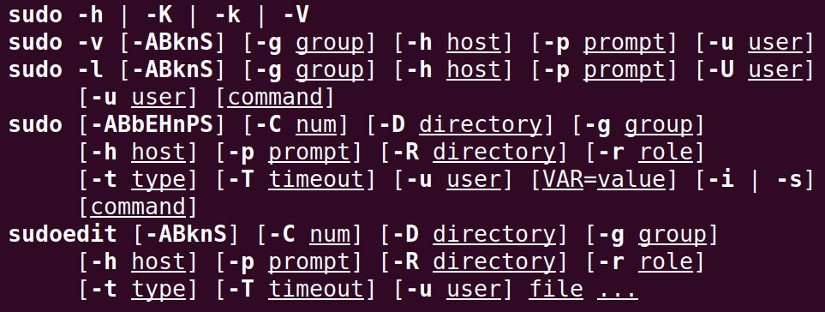
50 Most Used Linux Commands

Suppose, you've started to use **Linux** recently, and you’re quite new to the command line interface. But one thing is certain: if you want to learn about some frequently used **Linux** commands and practice them on your own then you’ve found the right place. Here, we’ve discussed the **50 most used Linux commands** with a brief description and practical examples.

# List of 50 Most Used Linux Commands

## sudo command

**Sudo** can be referred as the supreme command. It is the abbreviation for **“S**uper **U**ser **DO”.** It allows a user to act as a **superuser** and run commands accordingly. One can run certain commands prefixed by **sudo** with boosted rights. It is considered analogous to the “**run as administrator**” process of Windows.

**Synopsis**

**Useful Options**

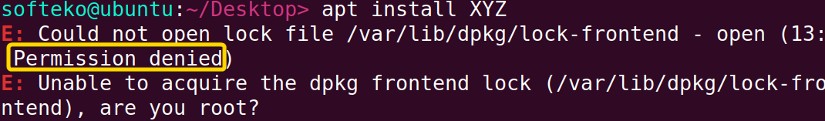
**-D directory, --chdir=directory** (executes the command in the specific directory)

**-e** (edits one or multiple files instead of executing commands)

**-l** (runs specific commands as the root user)

**-u user, --user=user** (executes the command as a user other than the specific default user)

### Example



Any general user cannot install any packages on the machine. However, with **sudo** prefixed with the command, the user can execute his/her task by providing his/her password.

## pwd command

**Pwd** is the abbreviation for **P**rint **W**orking **D**irectory. As the name suggests, it prints the name of the **current/working** directory all the way beginning from the root(/) directory.

### Synopsis

**Useful Options**

**-L, --logical** (Even as it carries symlinks, PWD utilizes from the environment. )

**-P, --physical** (avoid the symlinks)

When no option is mentioned, it is assumed that option -P is being used.

### Example

Generally, Terminal prompts have a complete directory in the name. Otherwise, pwd becomes a handy command to get insights about the current working directory.

## cd command

**C**hange **D**irectory(**cd**) allows one to change one’s current directory to the desired directory within the **terminal**.

### Synopsis

**Note: cd** is a shell built-in command, it doesn’t have a dedicated man(manual) page. However, you can get help using command help!

### Useful Options

**cd ~[username]** — change the directory to the home directory of the specified user.

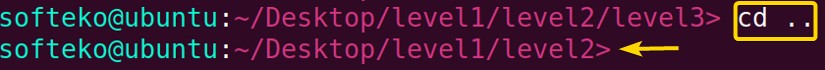
**cd ..** — changes directory one directory up the current directory.

**cd -** — changes the directory to the previously changed directory.

### Example

In our Desktop directory, we have a 3-level nested directory. we want to leap forward to the level 3 directory by jumping one level at a time.



We have changed our mind, now we want to be at level2. Therefore, we just need to go one directory backward which can easily be done by executing the command **cd ..**

Lastly, we have come to the conclusion that we want to be in the home(**~**) directory. This can be simply done by executing **cd ~** (tilde represents home directory).

## ls command

**Lists** the contents, both files and subdirectories of the current directory by default. It is one of the most used commands, as one can view the contents of a directory without exiting the terminal and perform their desired tasks on the specific contents.

### Synopsis

**Useful Options**

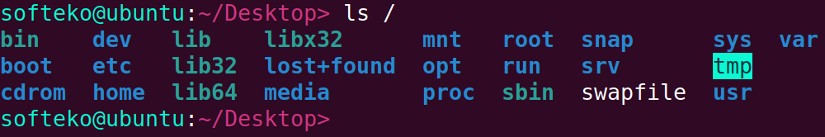
**-a** — doesn’t ignore the hidden files (files named with **.**(dot) at the beginning).

**-h** — print sizes in human-readable forms.

**-l** — lists in a long form.

**-S** — Sorts according to file size, largest first.

### Example

After being at the **root(/)** directory, if we run the **ls** command we can view the contents of the root directory.

## cat command

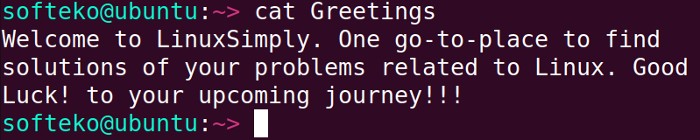
Prints the contents of the file specified. Generally, **cat** (con**cat**enates) reads the contents of the files fed to its arguments and prints them serially on the terminal.

### Synopsis

**Useful Options**

**-n, --number** — Displays line numbers when utilized.

### Example

We can display the contents of a file simply by using the **cat** command followed by the file name.

## mv command

**Mv** is the abbreviation for **m**o**v**e. As the name suggests it moves things from one place to another place. **mv** moves one or multiple files to the specified destination directory. If the directory doesn’t exist it just renames the files. **mv** can also be used to move directories and their contents.

### Synopsis

**Useful Options**

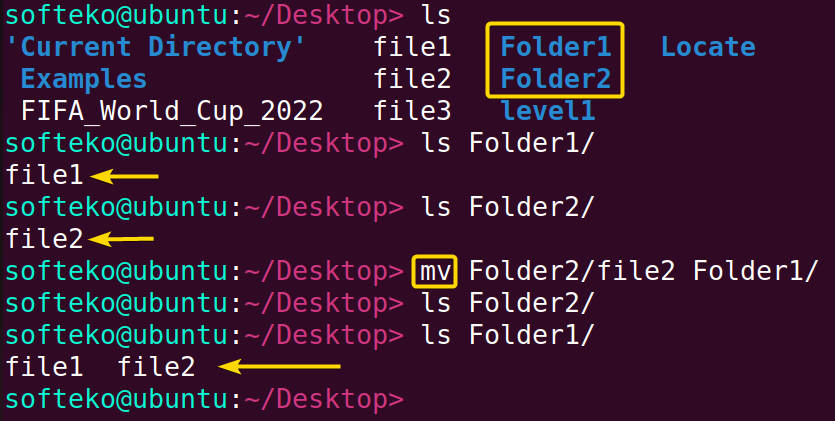
**-i, --interactive** (Displays interactive prompt before completing the modification)

**-t, --target-directory=DIRECTORY** (Moves every specified file to the targeted DIRECTORY)

**-v, --verbose** (Prints message of what is being performed.)

### Example

In the desktop directory, there are two subirectories named Folder1 & Folder2 which respectively contains two files named file1 & file2. Now lets **m**o**v**e file2 to folder1.



## cp command

**Cp** is the abbreviation for **c**o**p**y. As the name suggests it copies things from one place to another place. **cp** copies one or multiple files to the specified destination directory. If the directory doesn’t exist it just renames the files. **cp** can also be used to copy directories and their contents.

### Synopsis

**Useful Options**

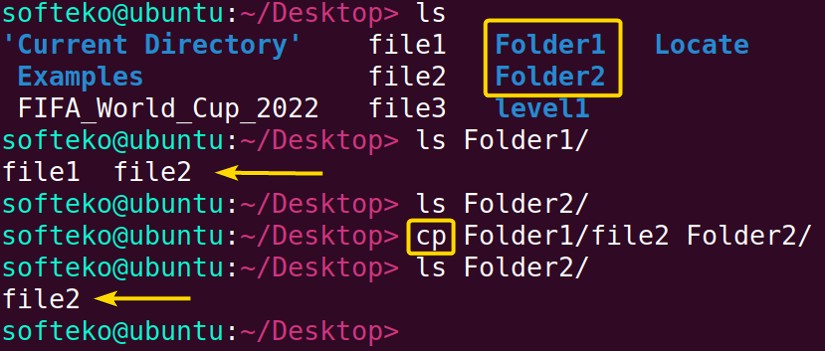
**-i, --interactive** (Displays interactive prompt before completing the modification)

**-R, -r, --recursive** (Copies the directories as well as its contents recursively)

**-v, --verbose** (Prints message of what is being performed.)

### Example

In the desktop directory, there are two subdirectories named Folder1 & Folder2. In folder1 there is a file named file2. Let's make a **c**o**p**y of file2 in the Folder2 directory.



## rm command

**Rm** is the abbreviation for **r**e**m**ove. As the name suggests it removes things and the removal is **permanent**, so be cautious while using it. **rm** can also be used to remove directories and their contents permanently.

### Synopsis

**Useful Options**

**-i** (Displays interactive prompt before completing the deletion each time)

**-I** (Less intuitive than **-i**, Only shows prompt while deleting 3 or more files or deleting recursively.)

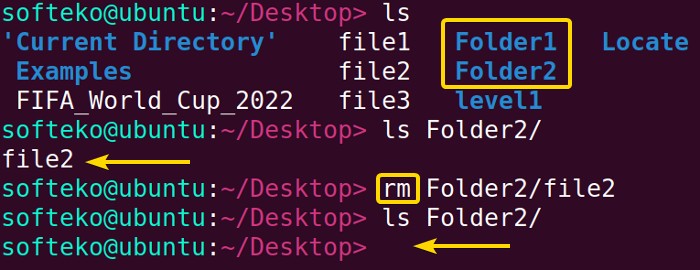
**-d, --dir** (Removes the empty directories)

**-R, -r, --recursive** (Removes the directories as well as its contents recursively)

**-v, --verbose** (Prints message of what is being performed.)

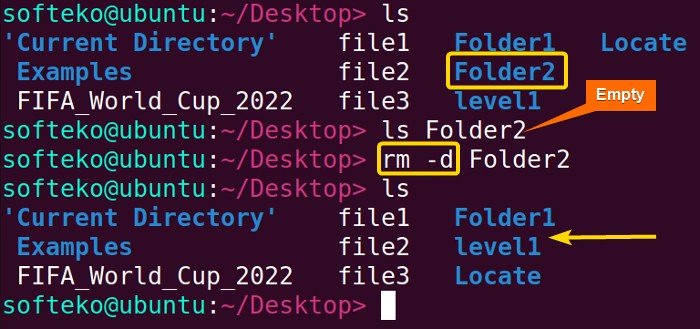
### Example

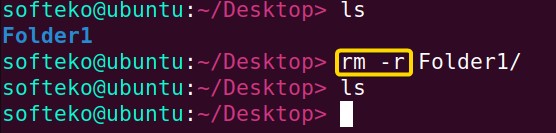
In the desktop directory, there are two subdirectories named Folder1 & Folder2. Folder2 contains a file named file2. let's **r**e**m**ove file2.



Now, Folder2 is empty let's remove this directory. To remove an empty directory option **-d**

has to be used.



Finally, we have decided that Folder1 is also of no use to us. Let's remove this directory. However, it is not an empty directory so **-d** will not work. Instead, we have to use **-r**.

**Disclaimer:** You cannot retrieve anything after removing it through the **rm** command. So, better be careful with **rm**.

## touch command

The **touch** command allows us to update a file's access or modification time. However, if the file doesn’t exist we can **create** that file. This ability to create files makes the **touch** command one of the most useful commands.

### Synopsis

**Useful Options**

**-a** (changes only the access time)

**-m** (changes only the modification time)

### Example

We created 3 files in our current directory, Desktop named file1, file2 and file3 with the help of the **touch** command.

## mkdir command

**mkdir** is the abbreviation for **m**a**k**e **dir**ectory. As the name suggests the **mkdir** command can be used to create one or more directories.

### Synopsis

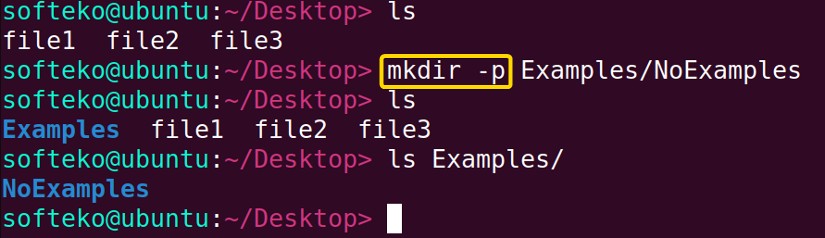
**Useful Options**

**-p - -parents** (Creates the necessary parent directories if required)

**-v, --verbose** (Prints message of what is being performed.)

### Example

Let’s make a directory named ‘Examples’ in our desktop directory which will contain one subdirectory named ‘NoExamples’.



Note: Option **-p** is used to create the necessary parent directory, in this case, which is ‘Examples’.

## locate command

The **locate** command performs the search operation from an **existing database** and prints the results with the exact directory path.

### Synopsis

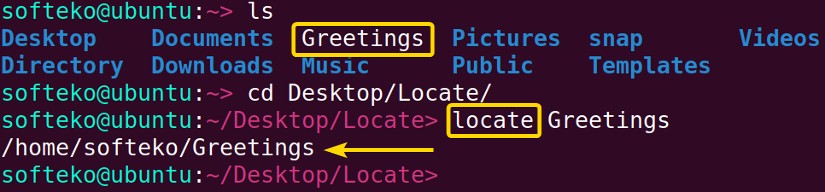
**Useful Options**

**-c** (Does not print any matches rather total occurring number)

**-e** (Only prints the existing match)

**-i** (Ignores the sensitivity of cases)

### Example

We have a file named Greetings in our ‘/home’ directory. After changing our current directory from ‘/home’ to ‘/Desktop/Locate’ we operated the **locate** command.

## find command

The **find** command searches in real-time not like the **locate** command from an existing file.

### Synopsis

**Useful Options**

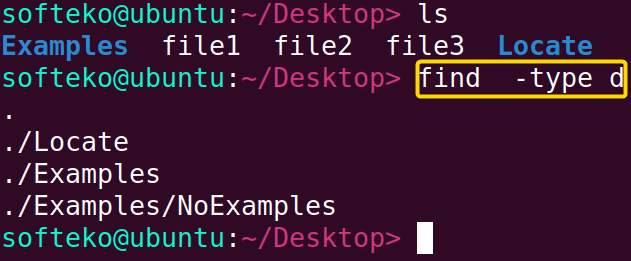
**-type d**/**f** (**d**(limits the search to only directories), **f**(limits the search to only files)).

**-size +n**,**n**,**-n** (Finds for a specific size **n**.)

**-name** pattern (Will search for the given pattern)

**-exec** (can be used to perform our customized tasks on the matches.)

### Example

We will only limit the find operation to directory type.

## grep command

The **grep** command can be used to search patterns in specific files or each file. Then prints the entire line containing the match.

### Synopsis

**Useful Options**

**-c** (Does not print any matches rather total occurring number)

**-i** (Ignores the sensitivity of cases)

**-w** (search for a whole word)

### Example

We have a text file named “[FIFA\_World\_Cup\_2022](https://en.wikipedia.org/wiki/2022_FIFA_World_Cup)” in our desktop directory. Let’s search for ‘FIFA’ in the file.



Now, if we want to just know about the occurrence number, we have to use the **-c** option.



## head command

The **head** command prints the **first**(by default 10 lines) few lines of a file.

### Synopsis

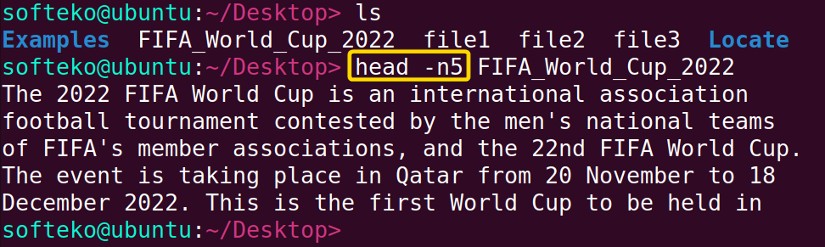


**Useful Options**

**-n** (Prints the first n lines.)

**-v, --verbose** (Prints message of what is being performed.)

### Example

We want to print the first 5 lines of the “[FIFA\_World\_Cup\_2022](https://en.wikipedia.org/wiki/2022_FIFA_World_Cup)” text file. It can be simply done using the **head** command with **-n5** option

## tail command

The **tail** command prints the **last**(by default 10 lines) few lines of a file.

### Synopsis

**Useful Options**

**-n** (Prints the last n lines.)

**-v, --verbose** (Prints message of what is being performed.)

### Example

We want to print the first 5 lines of the “[FIFA\_World\_Cup\_2022](https://en.wikipedia.org/wiki/2022_FIFA_World_Cup)” text file. It can be simply done using the **tail** command with **-n3** option

## man command

You are whether a beginner or a professional Linux CLI(command line interface) user, the command you will interact with for the most is certainly the **man** command. The **man** command enables the user to learn more about a specific command i.e. syntax, options and arguments of that command.

### Synopsis

**Useful Options & Shortcuts**

**-k, --apropos** (Displays the short manual page on the terminal)

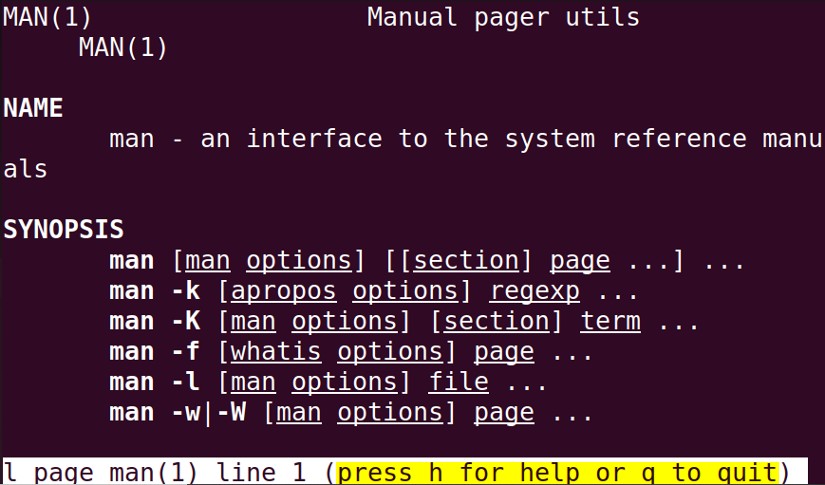
**-I, --match-case** (Performs case sensitive manual page search)

**/pattern** (It is a useful search shortcut built-in inside the man page.)

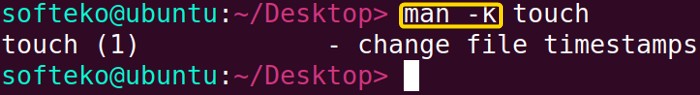
### Example

You can **navigate** through the man page using arrow keys and other specific keys. To know more about the **man** page navigation try pressing **h** on the **man** page of any command.

**$ softeko@ubuntu:~/Desktop> man man**



You can print short information from the **man** page using option **-k**.

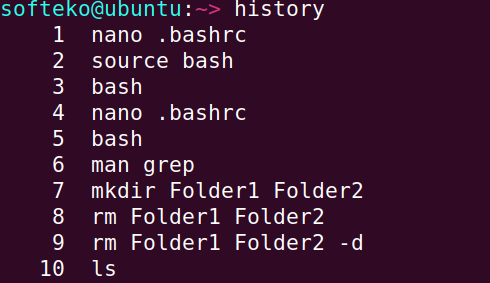


## history command

Executing this command one can view the previously used commands in the terminal.

### Synopsis

**Example**

let’s view some of our previously executed commands using the **history** command.

## chmod command

**chmod** is the abbreviation for **ch**ange **mod**e. The **chmod** command can be used to alter the permission attributes of system contents.

### Synopsis

**Useful Options**

**u** represents **user** (**u+x** will empowers user with executable permission.**)**

**g** represents **groups** (**g-w** will revoked the modification power of members of the group.)

**o** represents **others** (**o+r** will empower others to read the contents.)

**a** represents **all** (**a=r** will grant everyone accessing power however it will revoke everyone from write and execution permissions.)

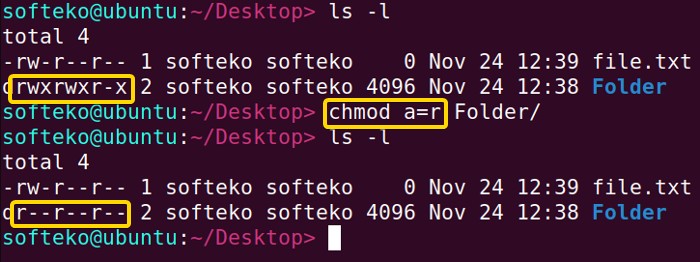
**-c, --change** (reports the occurrence of change.)

### Example

The modification permission represented by character w of groups for filem named “file.txt” can be revoked using the command **chmod g-w**.



Now, for the directory named “Folder”, we want that everyone only be able to read the file. This canbe simply be done by using the command **chmod a=r**.



## chown command

The **chown (ch**ange **own**er**)** command can be used to alter the owner of system files & directories.

### Synopsis

**Useful Options**

**-c, --change** (reports the occurrence of the change.)

**-R, --recursive** (operates recursivelyon the desired files & directories)

### Example

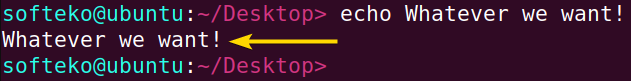
We want to change the ownership of a directory named “Folder”. It can be easily done using the **chown** command. However, we need to use **sudo** as a prefix of **chown** as only the **root** user has the permission to do so.

## echo command

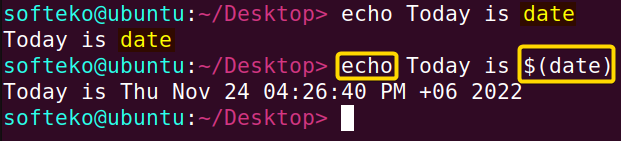
The **echo** common is one of the simplest commands, it prints whatever we give to it.

### Synopsis

**Example**

Let's print whatever we want!!!.

Let's print today’s date. However, if we feed ‘**date**’ to **echo** it will just literally print ‘date’ so we have to use special syntax **$(command)**.



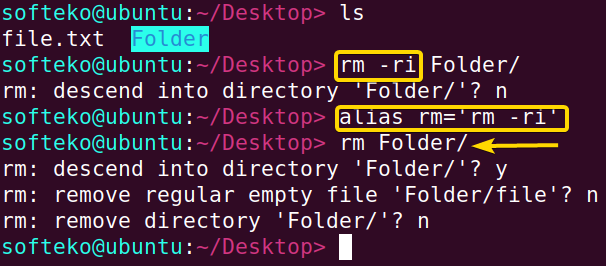
## alias command

The **alias** command can be used to replace a command with user-modified instructions while running the command.

### Synopsis

Note: Shell builtin commands don't have **man** pages. The **help** command can be used here.

### Example

If we want to remove directories/files with the **rm** command while adding a confirmation prompt we need to run **rm -ri**. Now, we can make just **rm** sufficient enough to do the task of **rm -ri**, using the **alias** command.

## passwd command

The **passwd** command can be used to change the **password** of a specific user.

### Synopsis

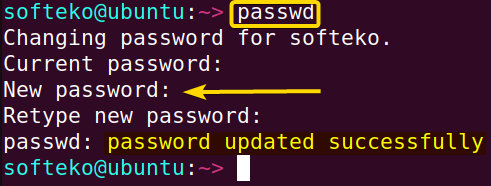
**Useful Options**

**-d --delete** User’s password can be deleted.

**-e --expire** Immediately makes the password expired.

**-i --inactive INACTIVE** makes the password inactivate after a specific INACTIVATE period.

### Example

We can simply change our password for the current user with **passwd** command.

## less command

The **less** command is used to display the contents of a file on the terminal screen in page by page manner.

### Synopsis

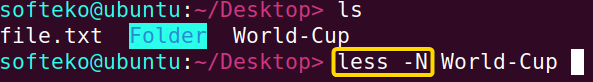
**Useful Options**

**-n --line-numbers** (when enabled it stops showing line numbers)

**-N --LINE-NUMBERS** (displays line numbers at starting points of each line.)

### Example

We want to view the file named “World\_Cup” with line numbers. This can be done by using the **less -N** command.



Arrow keys can be used to navigate through the less view.

**Press q** to exit the less display.

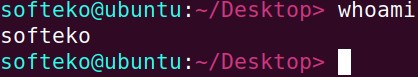


## whoami command

The **whoami** command simply displays the currently logged-in user.

### Synopsis

**Example**

The **whomai** command displays the current user in our case which is softeko.

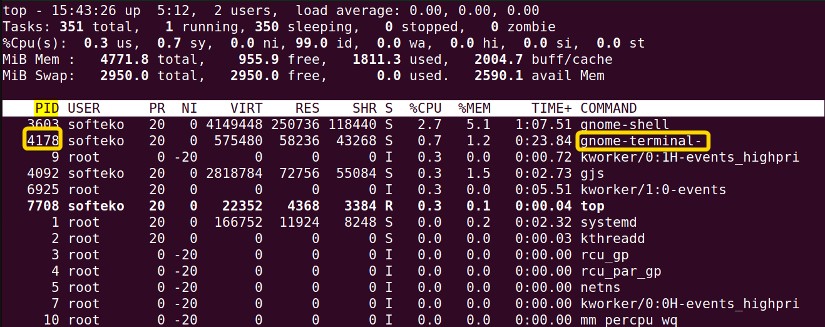
## kill command

### Synopsis

**Example**



The **top** command can be referred to as the task manager of linux. It shows information regarding CPU and memory optimization.



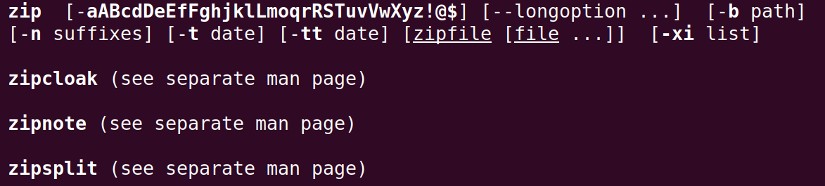
The process ID(**PID**) of the terminal is 4178 in this case. Now using the **kill** command with the PID of the terminal as its argument we can close the terminal.



## zip Command

**zip** command is used to compress files or folders into a .**zip** file in UNIX and UNIX-like operating systems. This allows for reducing the size of files and disk usage. Converting many files, and folders into a .zip file allows sharing and maintaining disk location by reducing file zie without any loss.

### Synopsis



**Useful Options**

**-e, --encrypt** (encrypt a file with a password which is entered through the prompt)

**-i \\*.txt** (include the only files with some conditions)

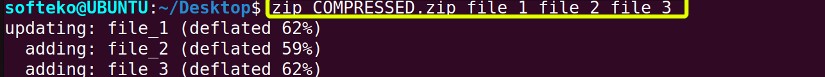
**-m** (moves the files into the zip files, deletes the original files after compression)

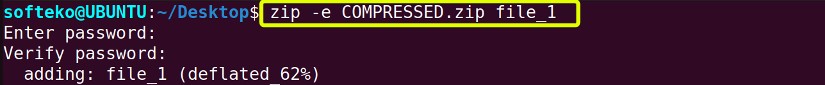
**-r, --recurse-paths** (compression throughout all the files and folders recursively)

**-s, --split-size SIZE** (the zip file is split into many different files into a specific size)

### Example

We can use **zip** command to compress a single file.

We can also compress multiple files.

We can encrypt a file with password with **-e** option.

We can move the file into a .zip file and delete the original with **-m** option.

## unzip Command

**unzip** command is used to extract files from a .zip file. This command creates a new folder in the current directory where the field inside the **zip** files is extracted.

### Synopsis

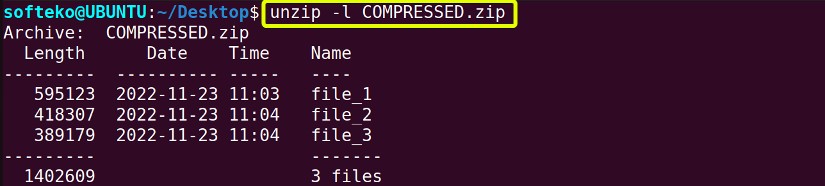
**Useful Options**

**-u** (update existing files inside the folder and create new ones if required)

**-i** (shows the files or folders inside the compressed document)

### Example

To simply extract files from a compressed folder, we use **unzip** command.

We can use **-l** option too view the files inside the compressed files.

## wget Command

**wget** is a download command which downloads files or webpages non-interactively from the network.

### Synopsis

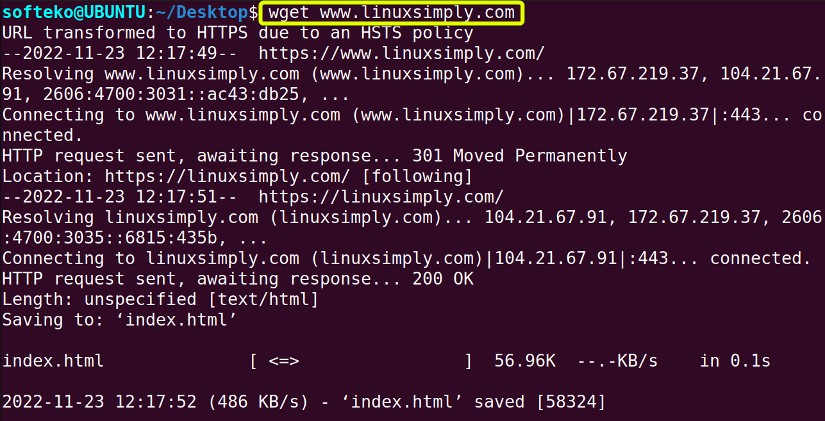
**Useful Options**

**-b** (downloads a file in the background)

**-c** (continues a partially downloaded file)

### Example

To download a webpage, we can use **wget** command.



To download a webpage in the background requires **-b** option.

## df command

**df** (**d**isk **f**ree) command shows the size, used, available space and mounted on the information of the filesystem.

### Synopsis

**Useful Options**

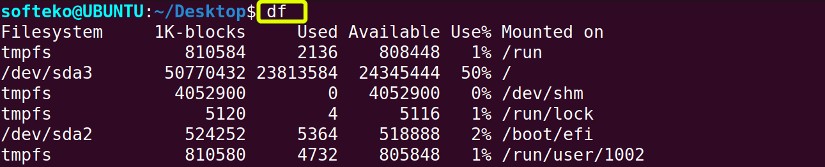
**-a, --all** (displays all file systems including inaccessible or hidden files)

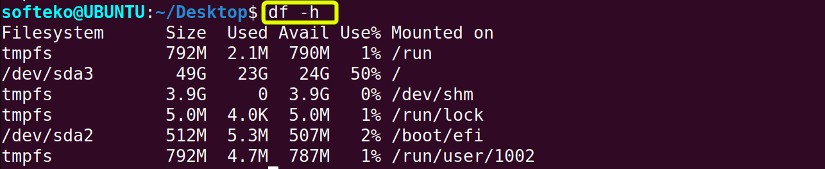
**-h, --human-readable** (displays information in human-readable format)

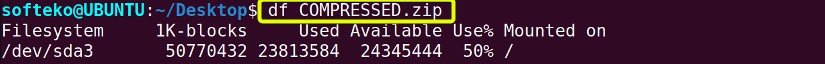
**-t, --type=file\_type** (displays files of a certain file type)

**-l, --local** (displays local file system)

### Example

Using only **df** command shows default information.

Using **df -h** shows us the file size in a human-readable format.

Using **df file\_name** shows information about only the **file\_name.**

## ping command

**ping** (**P**acket **In**ternet **G**roper) command shows the information about the network information about the host and the server. It can check the internet connection and show the latency between the host and server.

### Synopsis

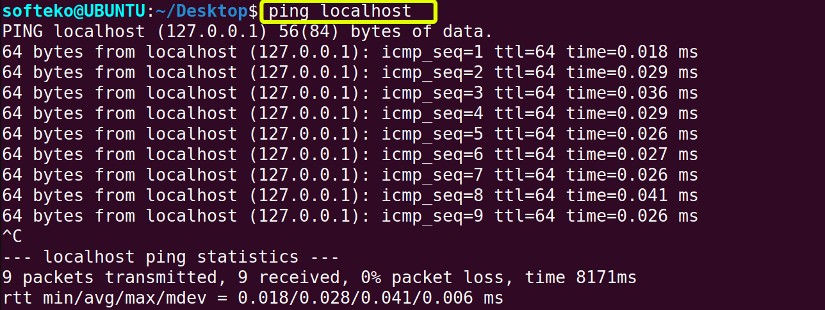
**Useful Options**

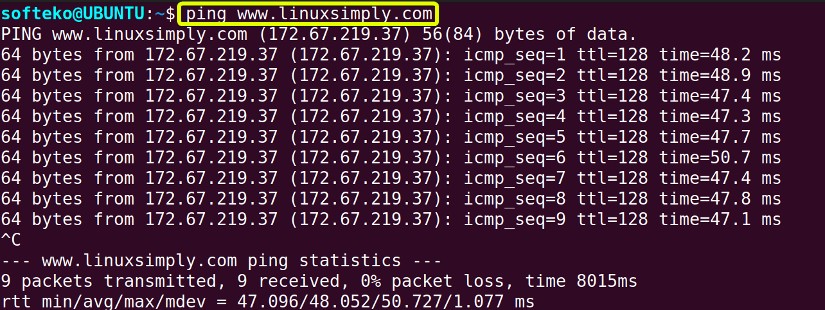
**-c COUNT** (sends COUNT number of ECHO\_REQUEST)

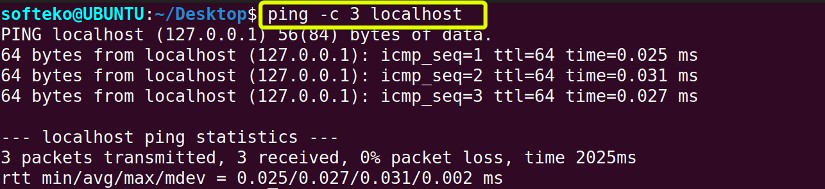
**-i COUNT** (sends a packet in every COUNT second interval)

**-f** (flood ping with rapid display, **sudo** permission is needed)

### Example

Finds information about the **localhost**, exiting the command needed, using ctrl+c.

Finds **ping** information about a website exiting the command needed, using **ctrl+c**.

We can add **-c NUM** option to limit the number of packets using ping.

## diff command

**diff** command finds checks 2 files and shows the difference between 2 files. Normally, it does not change the content of files, but it can generate a script.

### Synopsis

**Useful Options**

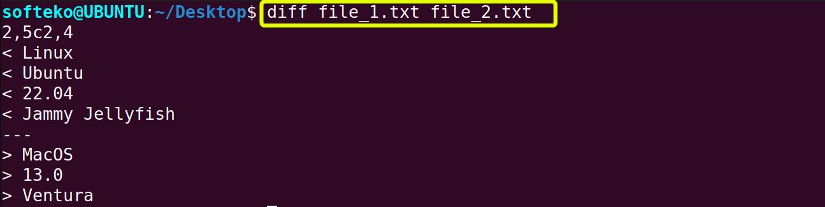
**-c** (shows the differences between the 2 files in context mode)

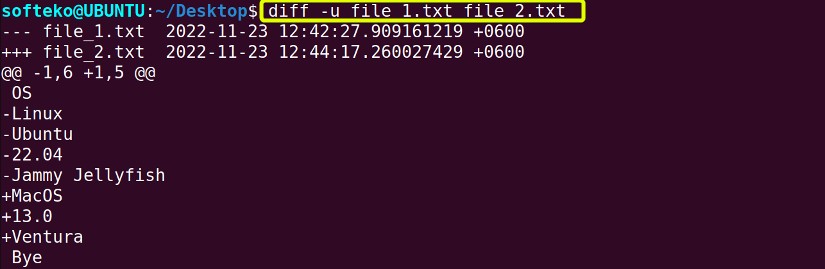
**-u, -U, --unified** (shows the unified context of the 2 files)

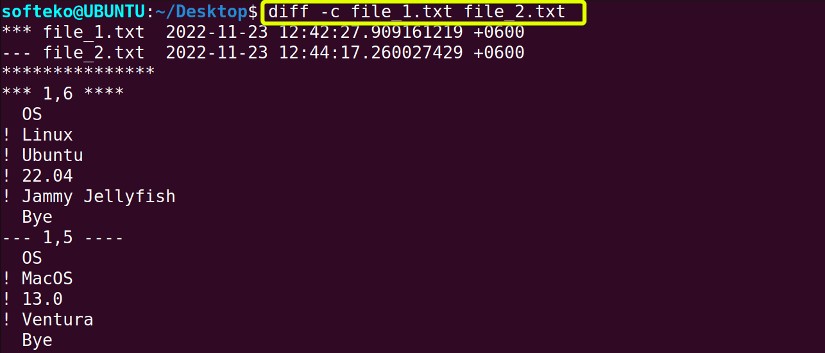
**-r, --recursive** (compares files inside any subdirectories)

**-i, --ignore-case** (ignore cases)

### Example

Using the **diff** command with 2 files shows the difference between those 2 files.

We can use **-u** option to see the difference in the unified context of 2 files.

Option **-c** shows both files in context mode.

## ps command

**ps** (**P**rocess **S**tatus) command shows the process status and information about that process.

### Synopsis

**Useful Options**

**-a, --all** (shows all the processes including hidden processes)

**-r** (shows all the running processes)

**--pid PID** (shows the specific process according to PID)

### Example

Simply typing **ps** shows the process status of the Linux system.

Option **-a** shows all the processes including hidden processes.

To look into a specific process, **--pid** command is used.

## apt command

**apt** (**A**dvanced **P**ackage **T**ool) command manages different packages including install, remove, update, etc.

### Synopsis

**Useful Options**

**apt [install, update, upgrade]** (downloads package information and install, update or upgrade the package, **sudo** permission is required)

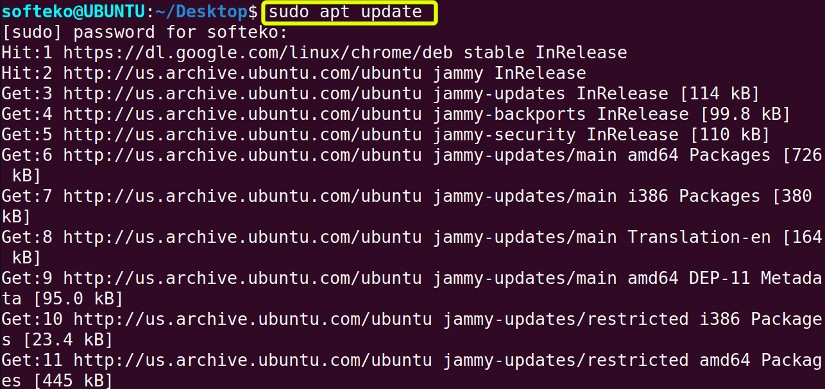
**apt full-upgrade** ( upgrade all installed packages, even remove some if required)

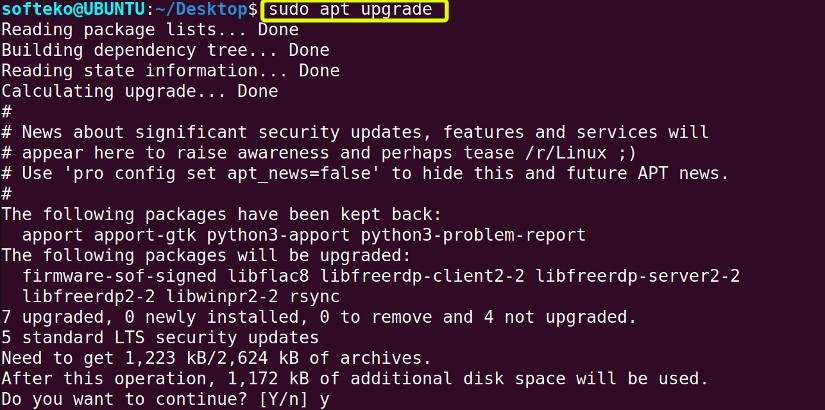
**apt remove Package\_Name** (removes a specific package)

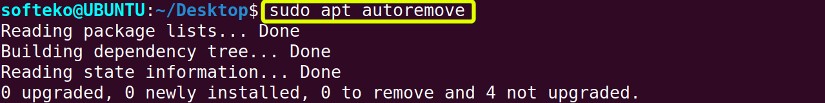
**apt autoremove** (removes unneeded dependencies)

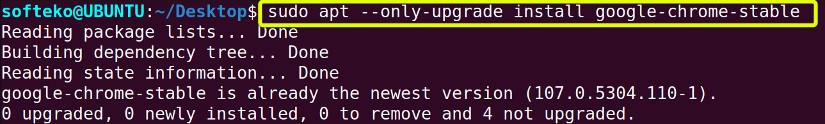
**apt --only-upgrade install Package\_Name** (installs or upgrades a specific package)

### Example

**sudo apt update** command updates all packages.

We can use **sudo apt upgrade** command to upgrade all available packages.

The **autoremove** option removes all unneeded dependencies.

The **--only-upgrade Packae\_name** option updates a specific package.

## dd command

**dd** command converts and copies a file to another directory. This command can be used to create a backup inside the hard drive or an external hard drive.

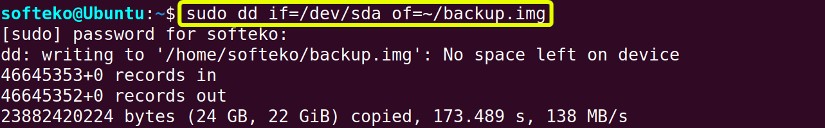
### Synopsis

**Useful Options**

**if** (read the file instead of standard input)

**of** (write the file instead of standard output)

### Example

**dd** command is used to back up a folder into another directory as **backup.img** file.

## top command

**top** (**T**able **O**f **P**rocesses) command shows the currently running process inside linux. It gives a dynamic but not interactive view of the process.

### Synopsis

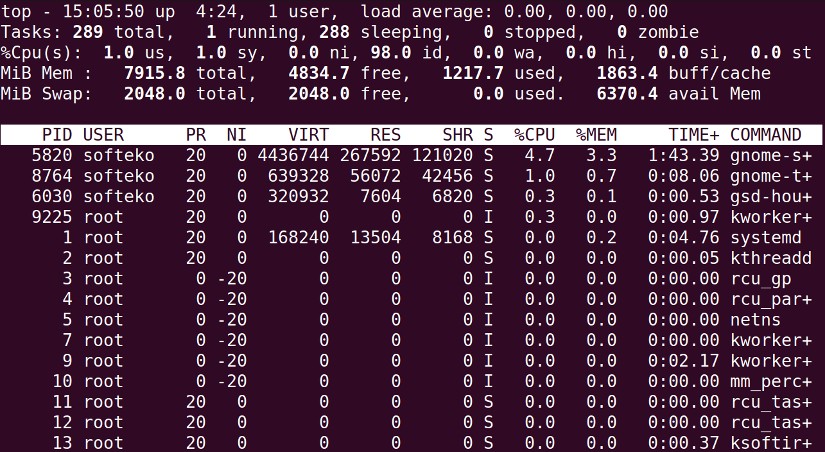
**Useful Options**

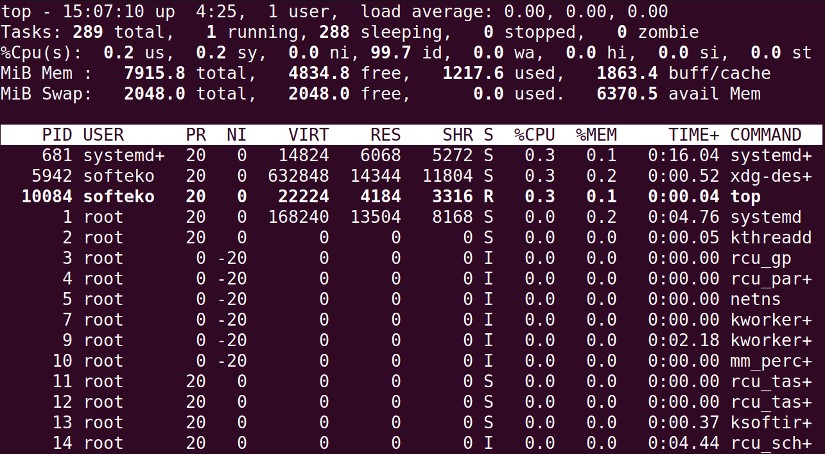
**-n NUM** (shows the top NUM number of processes)

**-u PARAS** (shows the processes according to PID or User as given as paras)

**-d TIME** (shows a dynamic view which updates in TIME tenth of seconds)

### Example

Simply typing **top** shows the processes dynamically.

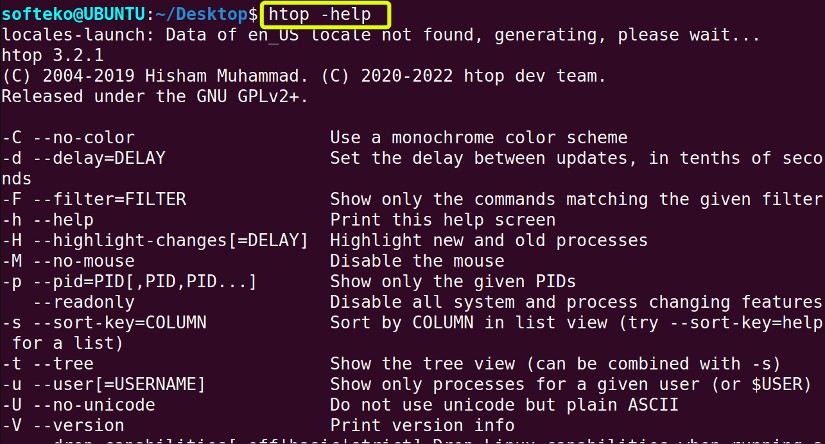
We can use option **-n 10** to find the top 10 running processes according to cpu usage.

## htop command

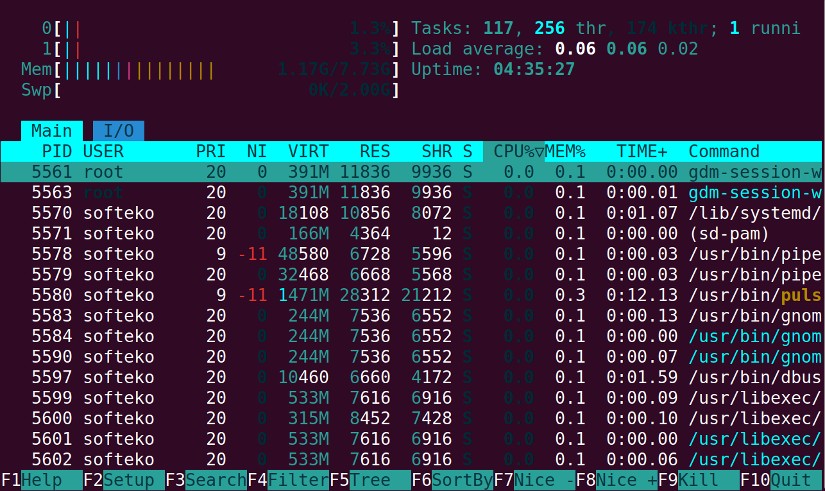
**htop** process is quite similar to the top process but in **htop**, the user can have interactive control over the top running processes in Linux. **htop** may need to be installed in the terminal first for usage.



### Synopsis

There is no manual page for **htop**, but help, there are different commands and options shown.

### Example

Simply typing **htop** shows the prompt where each process can be analyzed interactively.

## useradd command

**useradd** command is used to add a user to the Linux system. sudo command is needed to create a new user. The superuser **sudo** command is required to use this command.

### Synopsis

**Useful Options**

**-d DIRECTORY USER\_NAME** (creates a user in DIRECTORY)

**-u, --uid ID** (creates a user with the user id of ID)

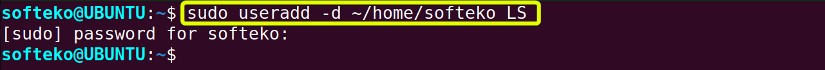
**-g --gid ID** (creates a user with a group id of ID)

**-M USER** (creates a user without any home directory)

**-e DATE USER** (creates a user with an expiry date)

### Example

Creating a simple user is possible with **useradd** command.

Option **-d** creates a user inside a specific directory.

We can also use **-u NUM** to create a user with a specific user id.

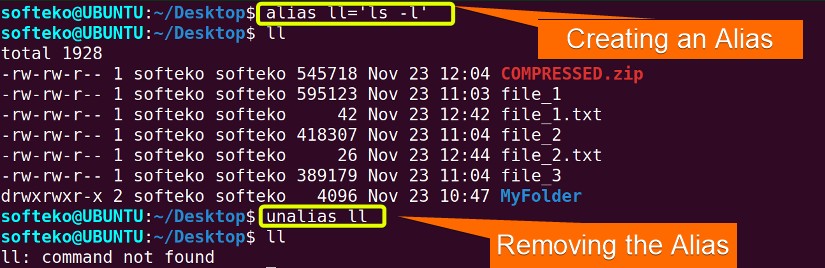
## unalias command

**unalias** command removes **alias** created before.

### Useful Options

**-a** (removes all alias)

### Example

Removes an alias can be done using **unalias ALS** command.

Removing all alias requires using **-a** option.

## uname command

**uname** command shows information about the system. Without any option it prints Linux.

### Synopsis

**Useful Options**

**-a** (displays all information about the system)

**-o** (displays the operating system name)

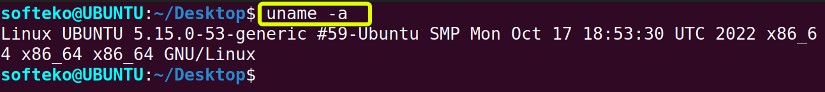
**-s** (displays the kernel name)

**-p** (displays the processor type)

### Example

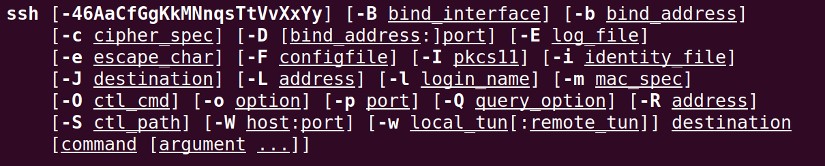
Option **-s** displays kernel name.

Option **-o** displays operating system name.

We can use **-a** to display all information about the system.

## ssh command

**ssh** (**S**ecured **Sh**ell) command is used for securely connecting to a remote server.

**Synopsis**

## traceroute command

**traceroute** command displays the packet route to reach the host. This command can be used to see the different hops or routes it takes to connect with a particular hostname.

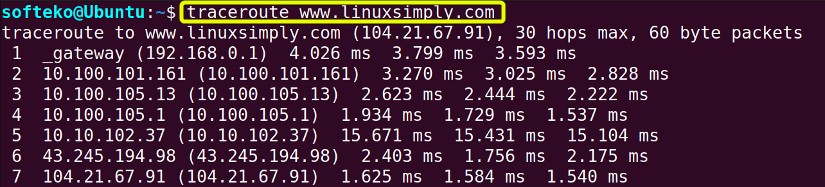
### Synopsis

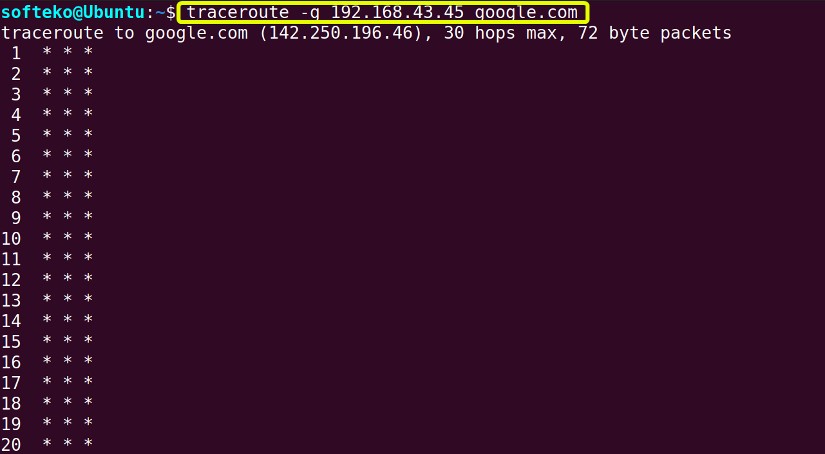
**Useful Options**

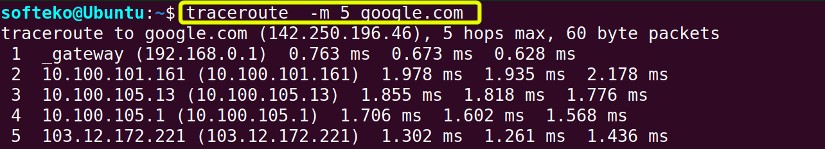
**-g Host\_name** (Routing the packet through a gate or IP address)

**-m Host\_name** (Fixing the maximum number of packets)

### Example

Simply typing traceroute shows the maximum number of hops and the routes.

To fix a gate for routing **-g** option is used.

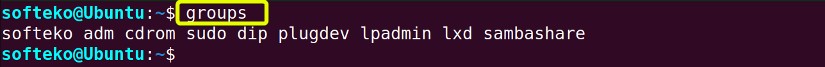
To fix the maximum number of hops, **-m** option is used.

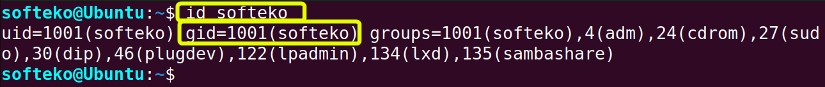
## groups command

Groups are used to manage multiple users and permissions. Several users can be members of a group and then the group can have permission for some files or folders to manage permissions.

### Synopsis

**Example**

Simply typing **groups** will display all the available groups in the machine.

We can use **id** command to find the group's id.

## shutdown command

**shutdown** command is used to shut down the system with some conditions. This command can schedule a shutdown at a specific time or cancel an already scheduled shutdown. The superuser **sudo** permission is required for this command.

### Synopsis

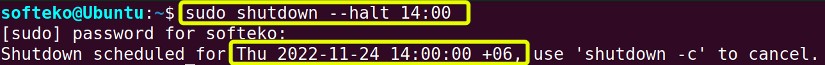
**Useful Options**

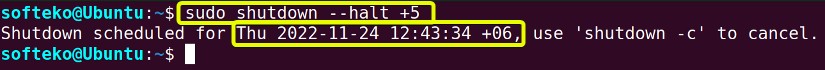
**--halt Time/Delay** (Schedule a shutdown at a specific time or after a certain delay)

**-c** (Cancel any shutdown command scheduled beforehand)

**-r Time/Delay** (Schedule a restart at a specific time or after a certain delay)

### Example

Option **--halt** can schedule a shutdown at 2:00 PM. The display will also show the date and the region.

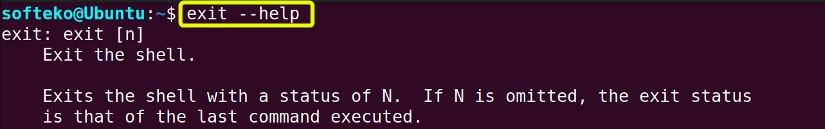
We can also schedule a shutdown after a 5 minute delay **--halt** option.

Option **-c** cancels a shutdown.

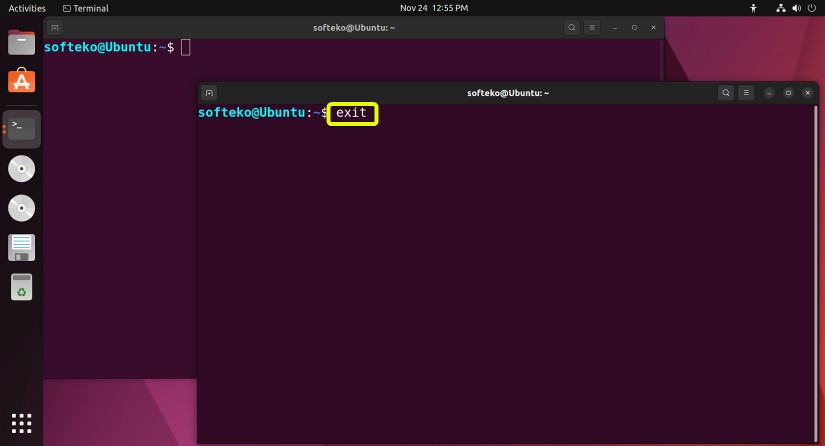
## exit command

**exit** command exits the terminal. It is a shell command, which means it doesn’t have any man page.

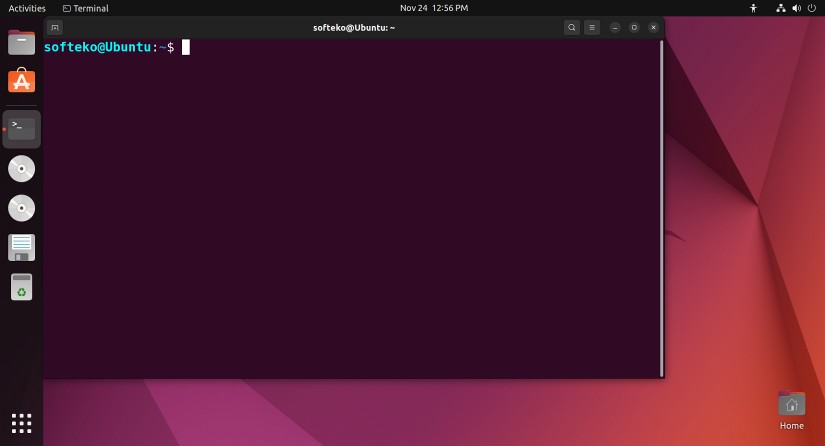
### Synopsis

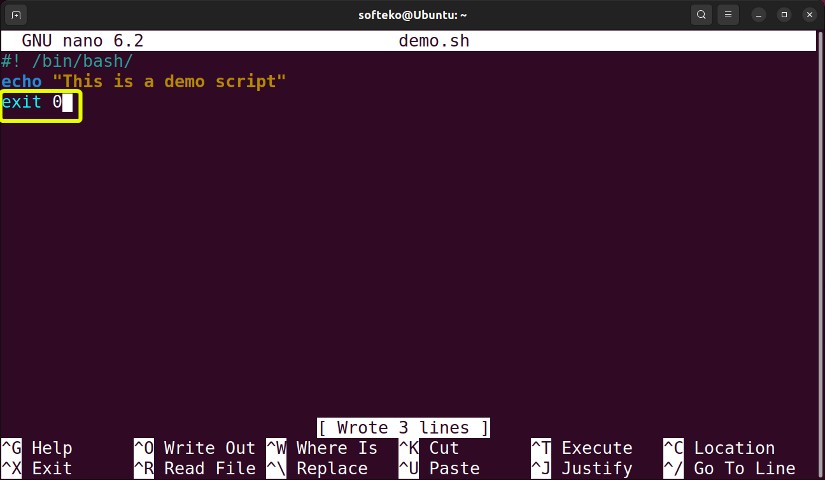


**Example**

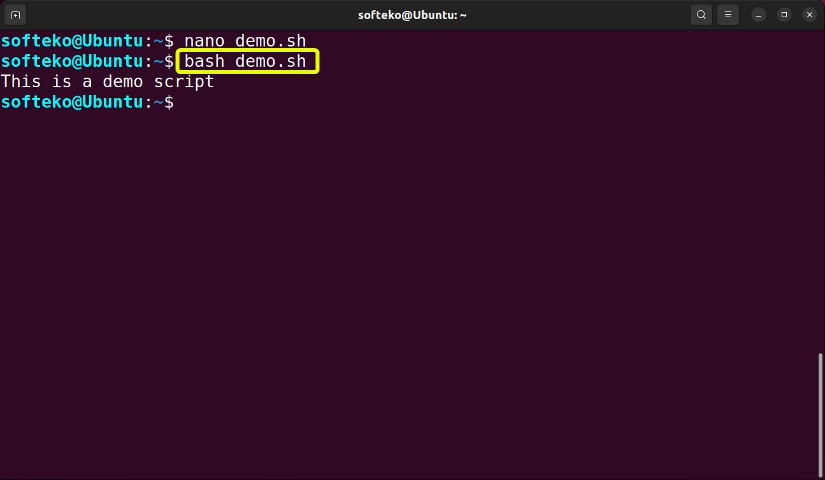
**exit** command terminates or exits a terminal.

Now that terminal is not available anymore.



The command **exit** is also used to exit a script. First, a demo script is created and **exit 0** is present at the end.

Now running the script using **bash** command, the script will exit at the end.



## wc command

**wc** (**W**ord **c**ount) command is used to count the number of characters or the number of lines in a file.

### Synopsis



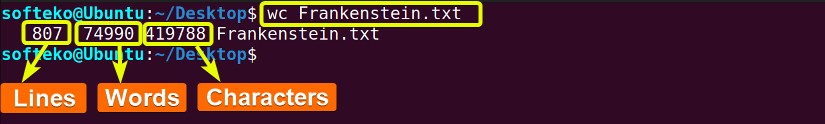
**Useful Options**

**-l File.txt** (shows the number of lines)

**-w File.txt** (shows the number of words)

**-c File.txt** (shows the number of characters)

### Example

Simply typing **wc Filename.txt** shows the line, word, and character count of the file.

Using **-l** option shows the number of lines of that file.

Using **-w** option shows the number of words in that file.



Using **-c** options shows the number of characters in the file.

## sort command

**sort** command sorts the content inside a file and shows it in the display. By default **sort**

command sort alphabetically. But different conditions can be applied for flexibility.

### Synopsis

**Useful Options**

**-o Main\_file Sorted\_File** (sort and create a new sorted file)

**-r, --reverse File** (sort in reverse order for a specific line, sort in horizontal order)

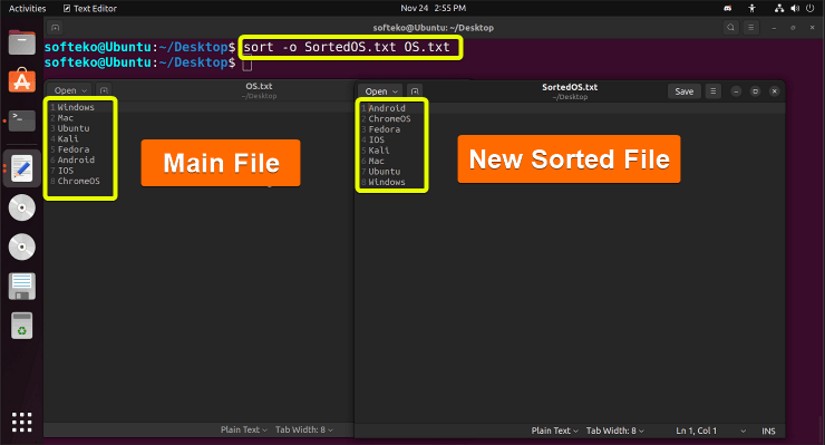
**-n, --numeric-sort File** (sort in numeric order)

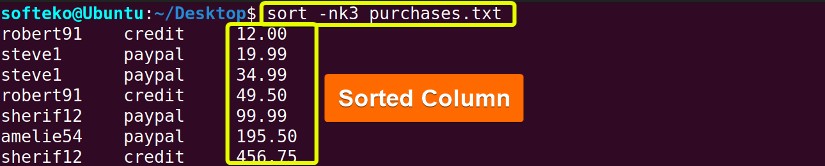
**-kNUM** (sort by using a key **NUM**, sorting by **NUM** column is possible using this option)

### Example

Simply typing this command sorts the contents inside a file alphabetically.

Sorts the content inside a file and creates a new sorted file.

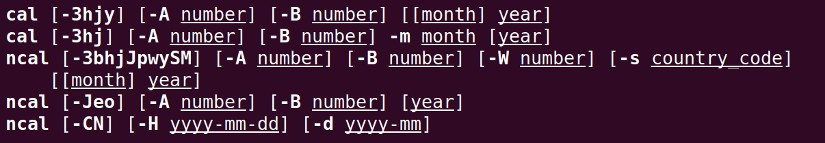


**-nk3** option sorts the using the third column numerically.

## cal command

**cal** command stands for the **cal**endar. It shows calendars in many different formats according to the condition.

### Synopsis



**Useful Options**

**-y** (shows the whole calendar of the current year)

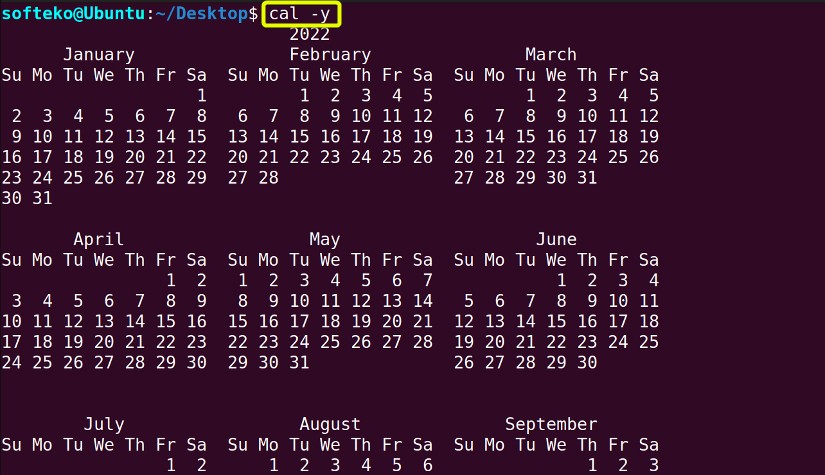
**cal MM YYYY** (shows the calendar of the **MM** month of **YYYY** year)

**cal YYYY** (shows the calendar of the **YYYY** year)

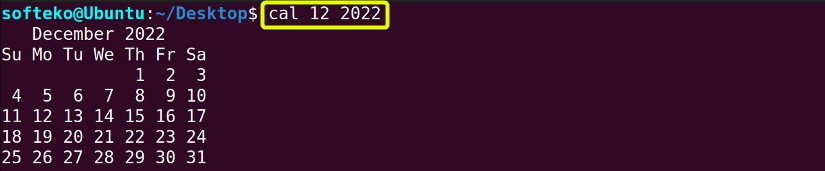
**-j** (shows the Julian calendar)

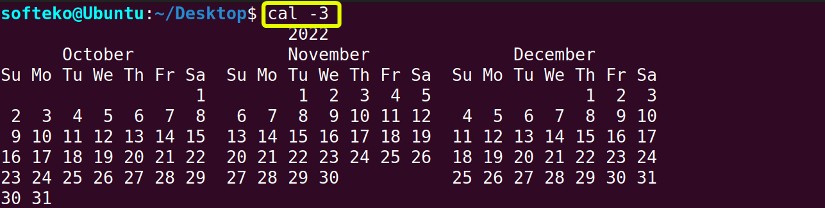
### Example

Simply typing **cal** shows the current month calendar.

Using **-y** option shows the whole year calendar.

Typing the month MM and year YYYY shows the calender of that specific month.



Using -**3** command shows the calendar of the current, previous and next month.

## nano command

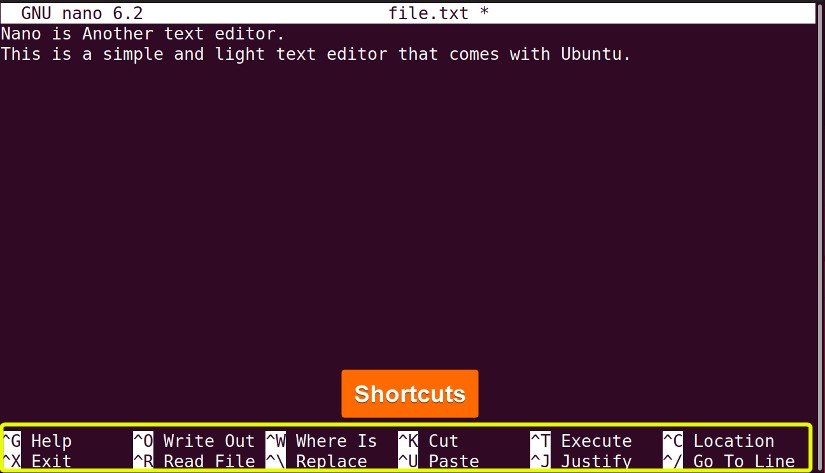
**N**ano is **ano**ther text editor. It is a simple and intuitive text editor that has many different shortcuts and is very light. It comes with the basic Ubuntu install.

### Synopsis

**Example**

Just typing **nano Filename.txt** starts the nano text editor.



Many shortcuts of nano are available at the bottom of the terminal.

## du command

**du** command means **d**isk **u**sage. Using this command the total usage of the disk and the disk usage of the different files are shown in the terminal.

### Synopsis

**Useful Options**

**-a** (shows the total disk usge)

**-ah** (shows all files disk usage)

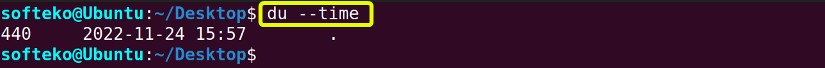
**--time** (shows the last modification time)

### Example

Just typing **du** displays the disk usage of the current directory.

**-a** option can be used to find disk usage of individual files inside the current directory.

-**ah** command is used to show all individual files’ disk usage in a human-readable format.

We can use **--time** option to show the last modification of the directory.

## apt-get command

**apt-get** command is quite similar to the **apt** command. But in **apt-get** common is generally used in the back-end and has backward compatibility. The apt-get (**A**dvanced **P**ackaging **T**ool) command is used for retrieving package and managing (install, upgrade, update, remove) the package.

### Synopsis

**Useful Options**

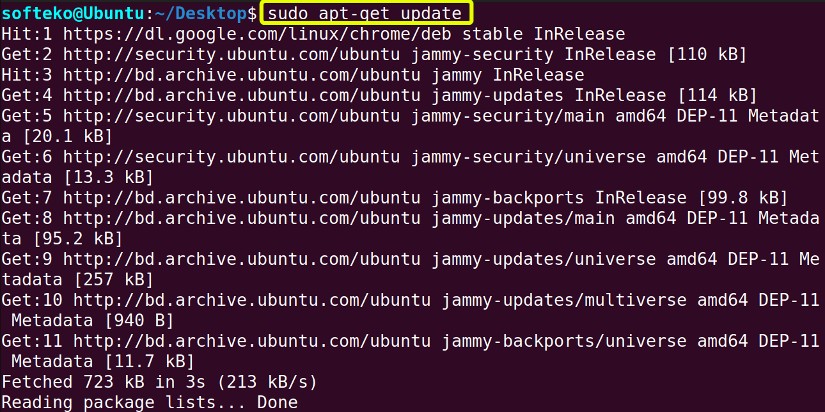
**apt-get [install, update, upgrade, remove]** (retrieve the package and install, update, upgrade or remove the package from the system)

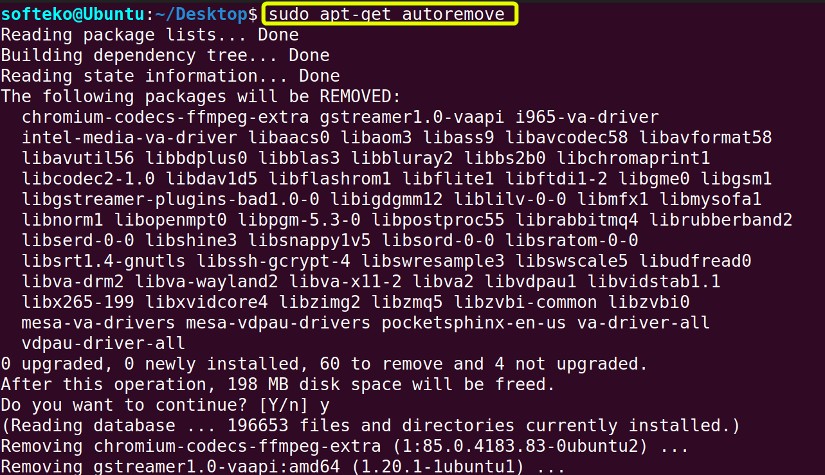
**--download-only** (downloads a package without installing or updating)

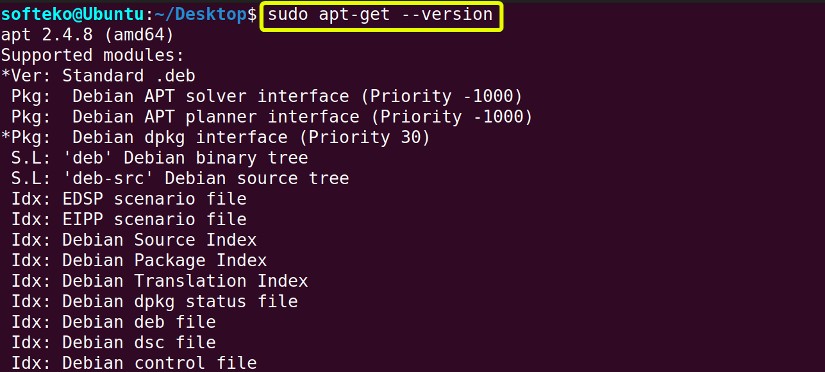
**--autoremove**, **--auto-remove** (removes unnecessary packages from the system)

**--version** (shows the version)

### Example

**update** command is used to update packages in the system.

We can use **autoremove** option to remove unnecessary packages that are not being used.

Option **--version** shows the version of the apt-get. It also shows the other possible commands on which **--version** command can be used

# Conclusion

In this article, we covered the 50 most used Linux commands that a regular user will encounter most of the time. Of course, there are many more of these commands and each of these commands has many more options and functionality available for the user.

You don’t need to memorize every one of them and we already have a free pdf document available that covers all these commands.

We will shortly cover every one of these commands individually and go into the details. In the meantime, you can also bookmark this page in case you need to have a look at any of these commands.