**20 Basic Ubuntu Commands And Terminal Shortcuts Every Beginner Must Know**



If you are Ubuntu newbie, then this post about **basic Ubuntu commands** is going to help you to get familiar with the [Terminal emulator](https://techlog360.com/linux-terminal-emulators/" \t "_self).

As we all know, Ubuntu is one of the most customizable Linux distribution available today. In fact, when we heard Linux, the first thing came to our would-be Ubuntu. Almost every Linux enthusiast start with Ubuntu when they enter the Linux world, as it’s one of the chosen [Linux distros for beginners](https://techlog360.com/linux-distro-beginners/" \t "_self).

And this post is for those Ubuntu beginners. Here we are going to list some important basic Ubuntu commands every beginner must know.

I think Ubuntu’s incredible flexibility is due mainly to its powerful shell, which is at the core of all Linux variants. Through the user-friendly “Terminal” interface, you can easily interact with this shell using a wide variety of commands.

Ubuntu uses BASH as its default shell and there are a lot of [bash commands](https://techlog360.com/a-z-bash-command-line-linux/" \t "_self) — that confuses every Ubuntu beginners. But you don’t have to learn all these bash commands if you want to familiar with Ubuntu Terminal in your early Linux days. Else you only have to understand basic Ubuntu commands which will help you to do some basic operations. Unlike [CMD commands on Windows](https://techlog360.com/all-windows-cmd-commands/" \t "_self), here on Ubuntu and other Linux distros we use commands to do most of our tasks.

So to do this task easier, here are the most basic starters Ubuntu commands and some Terminal shortcuts, that will help you to execute commands more quickly, to ease up your Linux journey.

Contents

**Basic Ubuntu Commands For Beginner:**

**1. Sudo**

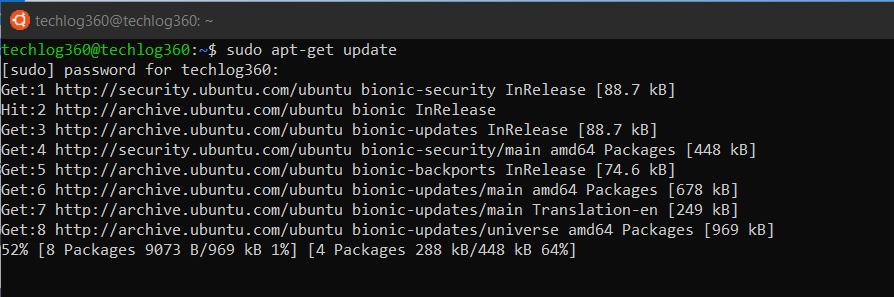
**sudo** (SuperUser DO) Linux command allows you to run programs or other commands with administrative privileges, just like “Run as administrator” in Windows. This is useful when, for example, you need to modify files in a directory that your user wouldn’t normally have access to.

**2. Apt-Get**

**apt-get** is the one of the most important Ubuntu commands every beginner must know. It is used to install, update, upgrade and remove any package. apt-get basically works on a database of available packages. Here is the list of different apt-get commands:

**1. sudo apt-get update**

apt-get update with super user privileges is the first command you need to run in any Linux system after a fresh install. This command updates the database and let your system know if there are newer packages available or not.



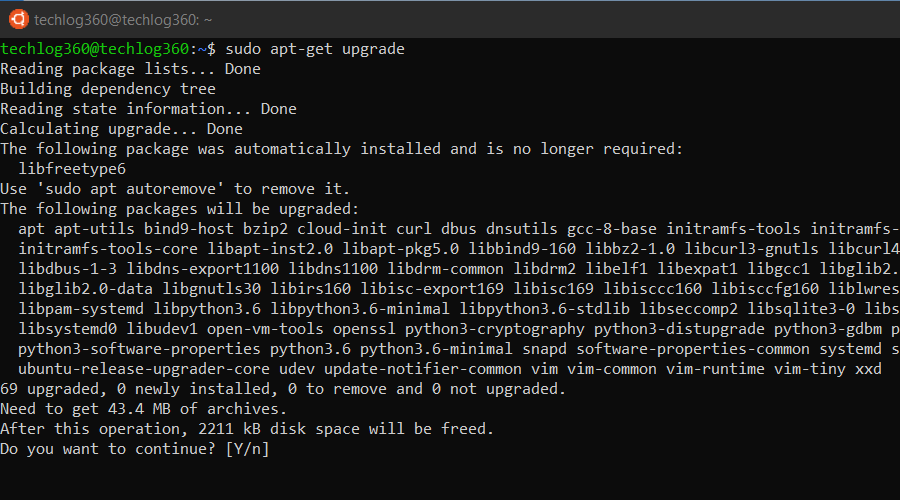
**2. sudo apt-get upgrade**

After updating the package database, next step is to to upgrade the installed packages. For upgrading all the packages with available updates you can use this command.

And if you like to upgrade a particular package, you should tweak the above command a little:

sudo apt-get upgrade <package-name>

Replace the *<package-name>* with your desired package.

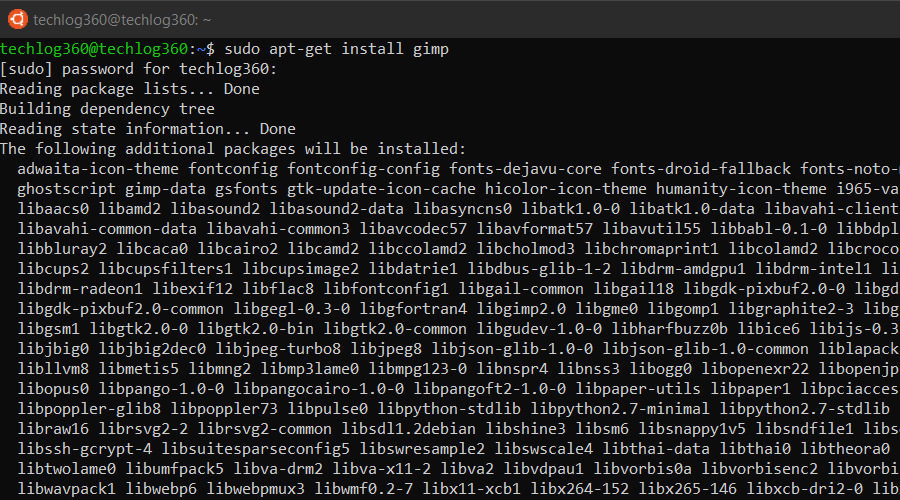


**3. sudo apt-get install**

If you know the name of the package, then you can easily install a program using this command:

sudo apt-get install <package-name>

Replace the *<package-name>* with your desired package.



If you are not sure about the package name, you can type a few letters and press tab and it will suggest all the packages available with those letters. Thanks for auto-completion feature.

**4. sudo apt-get remove**

When it comes to removing the installed program apt-get remove command suits your need. You only have to know the exact package name of the software you want to uninstall.

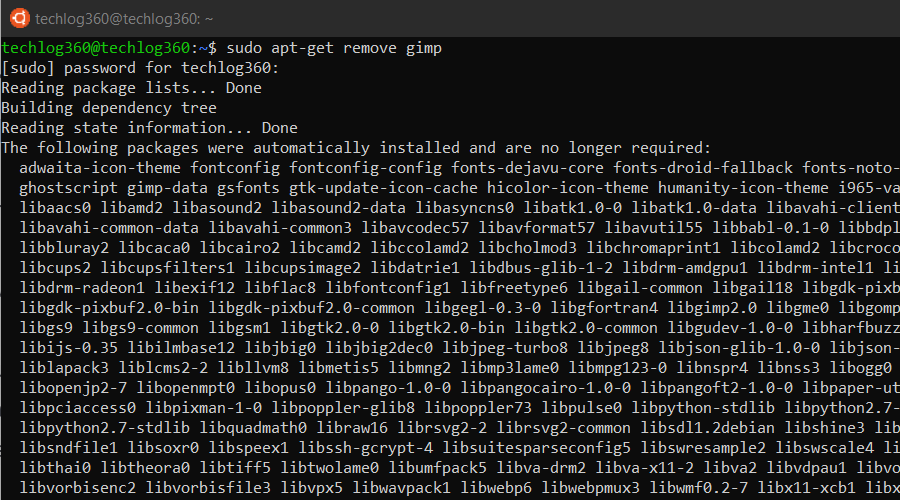
If you don’t know the package name, use below ubuntu basic command to list all the packages installed on your system and then copy the package name from the list:

dpkg --list

Now run the *apt-get remove* command as sudo in order to remove the software:

sudo apt-get remove <package-name>

Replace the *<package-name>* with the one you copied from the dpkg list.



apt-get remove command only removes the software from your system but not the configuration or data files of the package. These files help in keeping the same settings when you want to reinstall the same software.

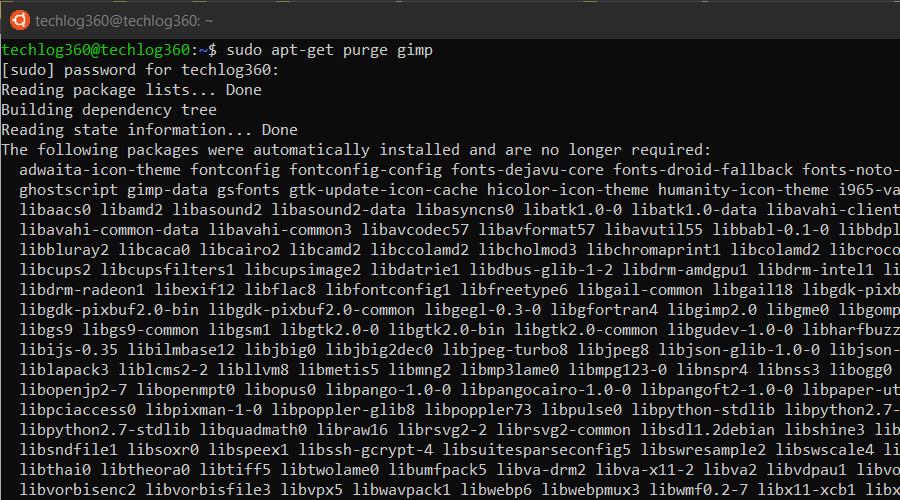
**5. sudo apt-get purge**

apt-get purge command is used when you want to remove a software completely from your system with its configuration or data files so that no longer personalized settings will be available during reinstallation.

Run the *apt-get purge* command as sudo in order to remove the software completely:

sudo apt-get purge <package-name>

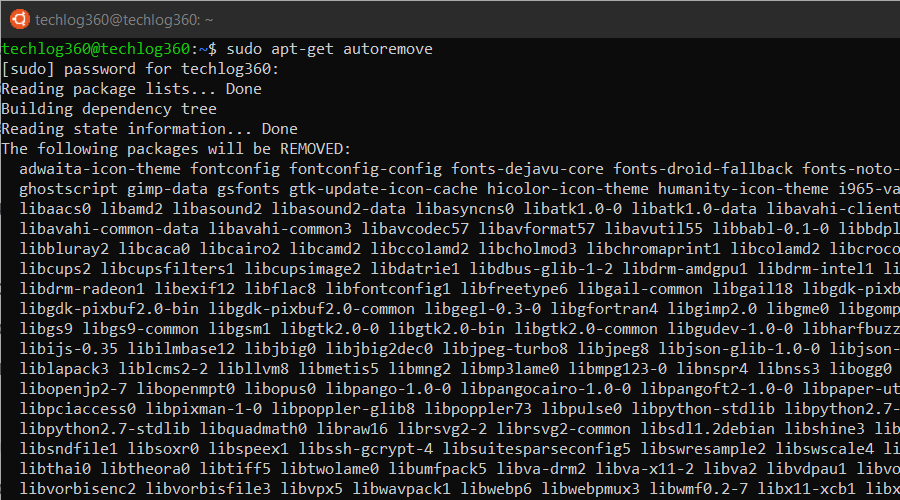
Replace the *<package-name>* with the application that you want to remove or copied from the dpkg list.



**6. sudo apt-get autoremove**

apt-get autoremove command is used to remove any unnecessary packages. Unnecessary means, whenever you install an application, the system will also install the software that this application depends on. It is common in Ubuntu that applications share the same libraries. When you remove the application the dependency will stay on your system.

So run *apt-get autoremove* as sudo after uninstalling a package to remove unwanted software dependencies.



So apt-get autoremove will remove those dependencies that were installed with applications and that are no longer used by anything else on the system.

**3. Ls**

**ls** (list) command lists all files and folders in your current working directory. You can also specify paths to other directories if you want to view their contents.

**4. Cd**

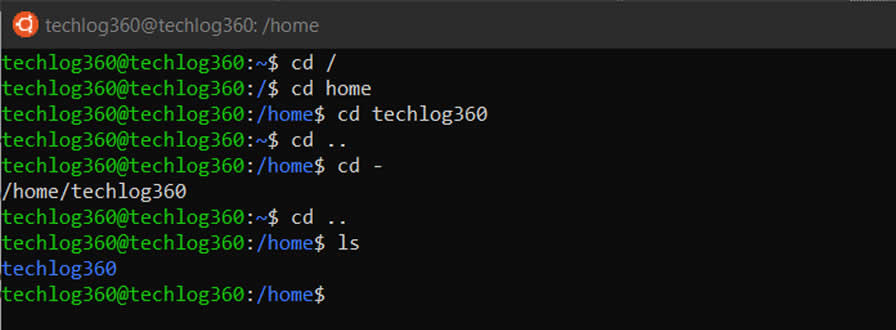
**cd** (change director”) Linux command also known as chdir used to change the current working directory. It’s one of the most used basic Ubuntu commands. Using this command is easy, just type cd followed by the the folder name. You can use full paths to folders or simply the name of a folder within the directory you are currently working. Some common uses are:

* cd /  – Takes you to the root directory.
* cd .. – Takes you up one directory level.
* cd –  – Takes you to the previous directory.

Here are some examples to how to use cd command in Ubuntu:

Example 1: ***cd home*** – open home folder in current directory.

Example 2: ***cd Linux\ Drive*** – open Linux Drive named folder in directory. Here you can see I use backslash because the folder name has spaces so *for each space you use “backslash+space”*. Like, if your folder name is “am a programmer” then the cd command will be, “cd am\ a\ programmer”.



**5. Pwd**

**pwd** (print working directory) Ubuntu command displays the full pathname of the current working directory.

**6. Cp**

**cp** (copy) Linux command allows you to copy a file. You should specify both the file you want to be copied and the location you want it copied to – f*or example, cp*xyz*/home/*myfiles*would copy the file “*xyz*” to the directory “/home/*myfiles*”*.

**7. Mv**

**mv** (move) command allows you to move files. You can also rename files by moving them to the directory they are currently in, but under a new name. The usage is the same as cp – f*or example mv*xyz*/home/*myfiles*would move the file “*xyz*” to the directory “/home/*myfiles*”*.

**8. Rm**

**rm** (remove) command removes the specified file.

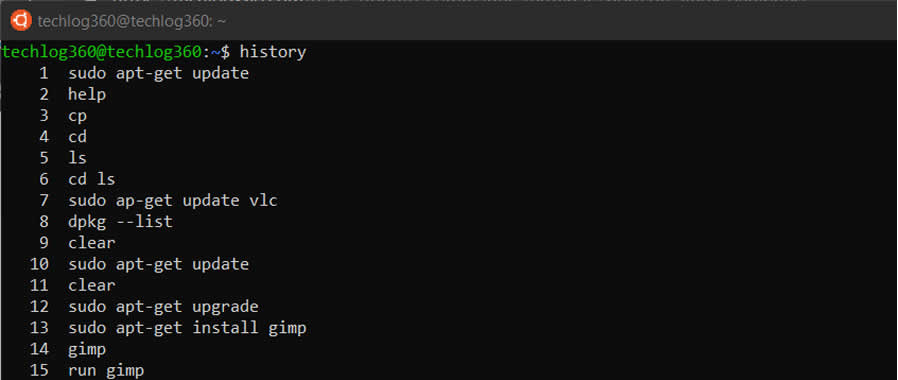
* rmdir (“remove directory”) – Removes an empty directory.
* rm -r (“remove recursively”) – Removes a directory along with its content.

**9. Mkdir**

**mkdir** (make directory) command allows you to create a new directory. You can specify where you want the directory created – if you do not do so, it will be created in your current working directory.

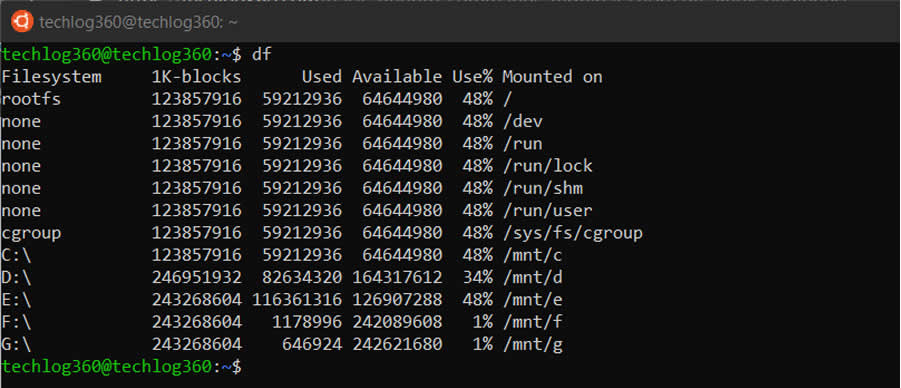
**10. History**

**history** command displays all of your previous commands up to the history limit.



**11. Df**

**df** (display filesystem) command displays information about the disk space usage of all mounted filesystems.



**12. Du**

**du** (directory usage) command displays the size of a directory and all of its subdirectories.

**13. Free**

**free** – Displays the amount of free space available on the system.

**14. Uname -A**

**uname -a** – Provides a wide range of basic information about the system.

**15. Top**

**top** – Displays the processes using the most system resources at any given time. “q” can be used to exit.

**16. Man**

**man** command displays a “manual page”. Manual pages are usually very detailed, and it’s recommended that you read the man pages for any command you are unfamiliar with. Some uses are :

* man man – Provides information about the manual itself.
* man intro – Displays a brief introduction to Linux commands.

**17. Info**

Similar to man, but often provides more detailed or precise information.

**18. <Command Name> -H Or <Command Name> –Help**

This command is a third alternative to get help. While not as detailed as the info or man pages, this will provide a quick overview of the command and its uses.

For example: *man -h* or *man -help*

**19. Passwd**

passwd Ubuntu basic command is used to change user password using Terminal. What you have to do is run the below command, where is the username whose password has to change:

passwd <user>

**20. Whatis**

whatis command shows a brief description of what is the functionality of specific built-in Linux command.

whatis <command>

Some examples are:

* whatis cd
* whatis man
* whatis help

Above commands will display the purpose of cd, man and help commands.

**Ubuntu Terminal Shortcuts:**

To further ease up your skill, these Ubuntu Terminal keyboard shortcuts would help.

|  |  |
| --- | --- |
| **Ubuntu Terminal Shortcuts** | **Function** |
| Ctrl + Shift + T | Open new tab on current terminal |
| Ctrl + Shift + W | Close the current tab |
| Ctrl + A | Move cursor to beginning of line |
| Ctrl + E | Move cursor to end of line |
| Ctrl + U | Clears the entire current line |
| Ctrl + K | Clears the command from the cursor right |
| Ctrl + W | Delete the word before the cursor |
| Ctrl + R | Allows you to search your history for commands matching what you have typed |
| Ctrl + C | Kill the current process |
| Ctrl + Z | Suspend the current process by sending the signal SIGSTOP |
| Ctrl + L | Clears the terminal output |
| Alt + F | Move forward one word |
| Alt + B | Move backward one word |
| Ctrl + Shift + C | Copy the highlighted command to the clipboard |
| Ctrl + Shift + V or Shift + Insert | Paste the contents of the clipboard |
| Up/Down Arrow keys | To scroll through your command history, allowing you to quickly execute the same command multiple times |
| TAB | Used to complete the command you are typing. If more than one command is possible, you can press it multiple times to scroll through the possible completions. If a very wide number of commands are possible, it can output a list of all possible completions. |