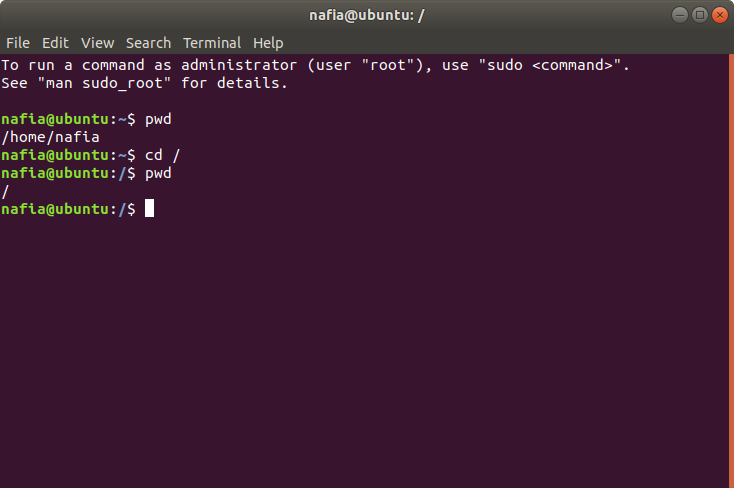
## Linux Commands

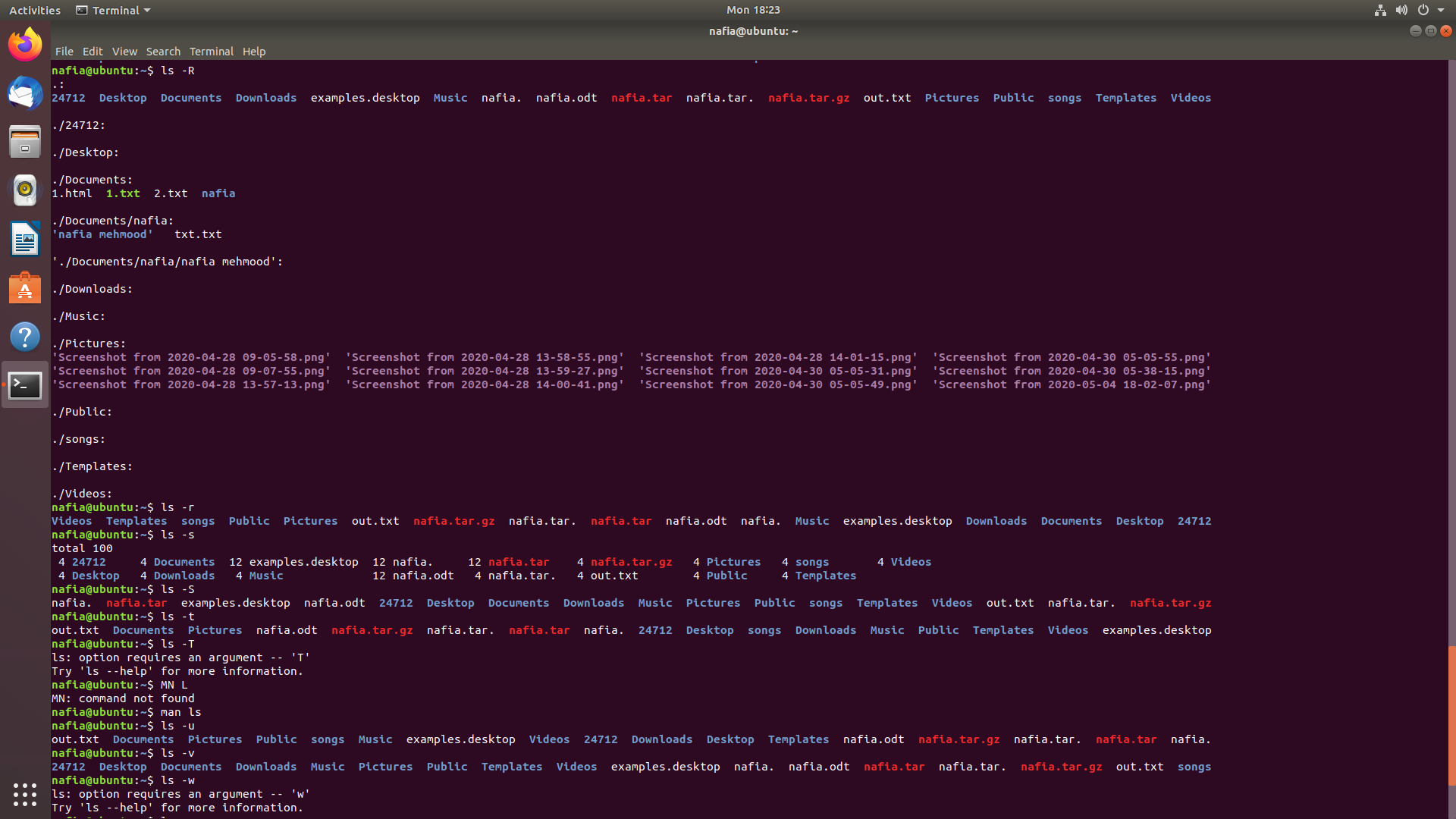
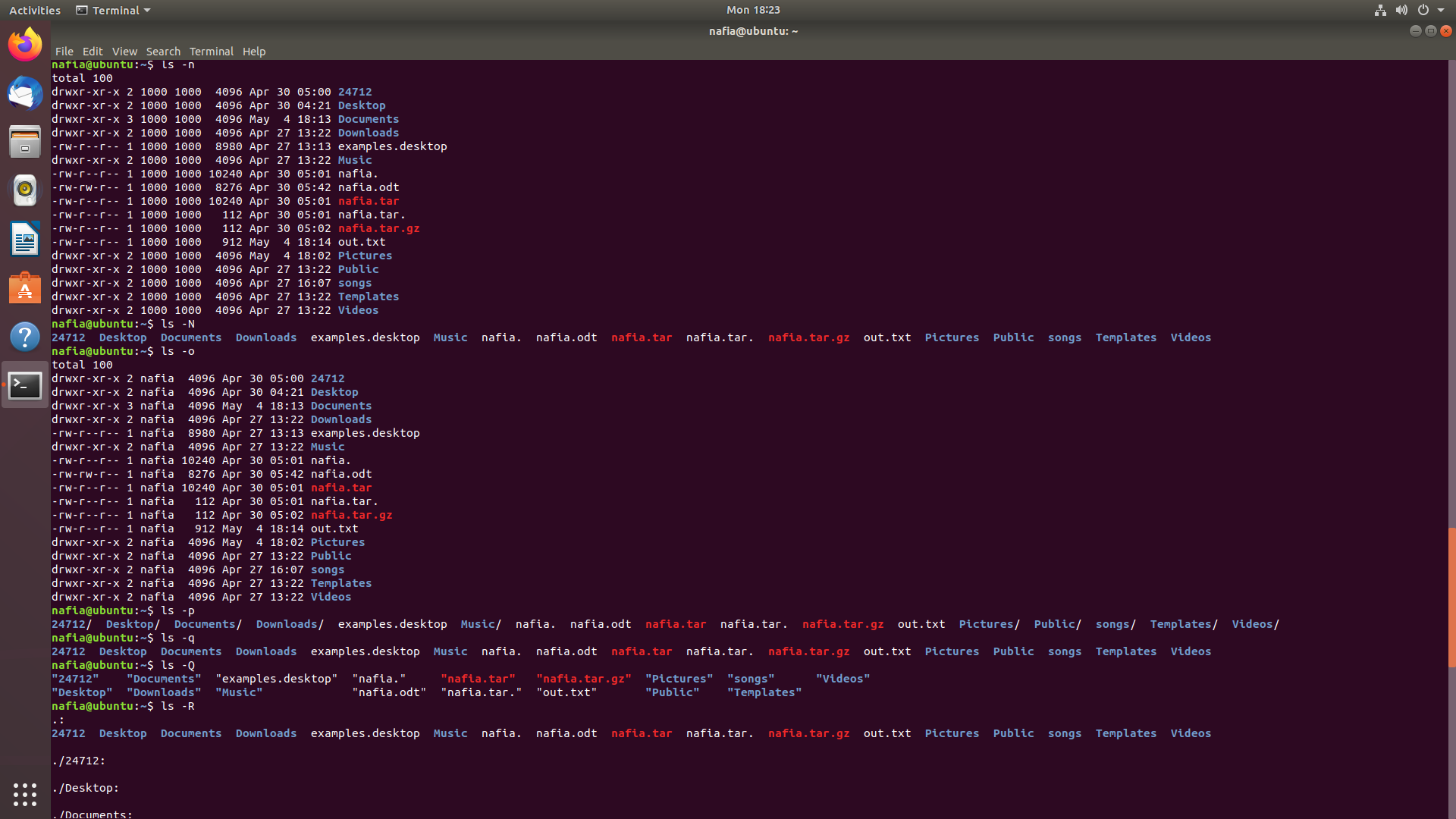
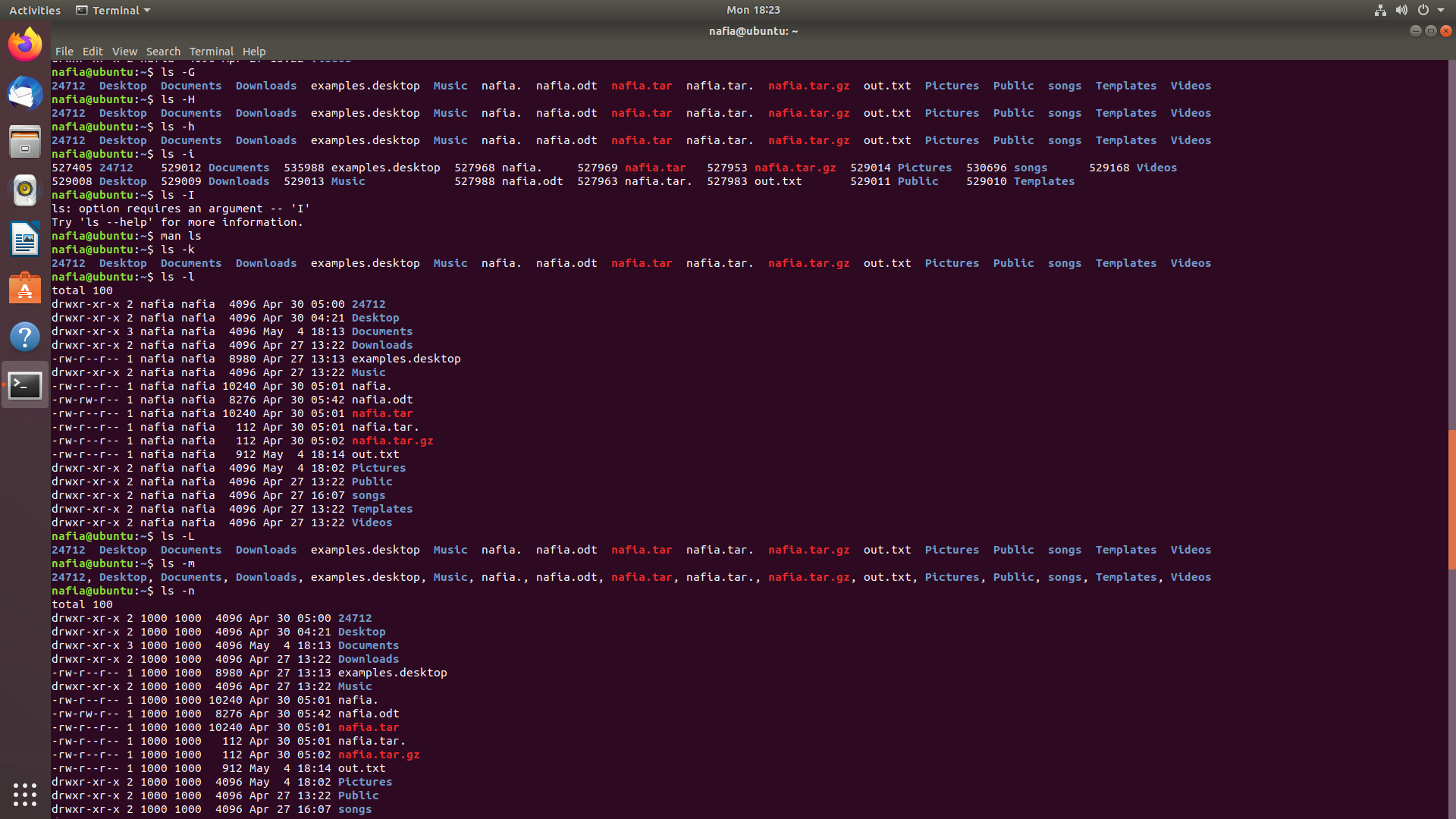
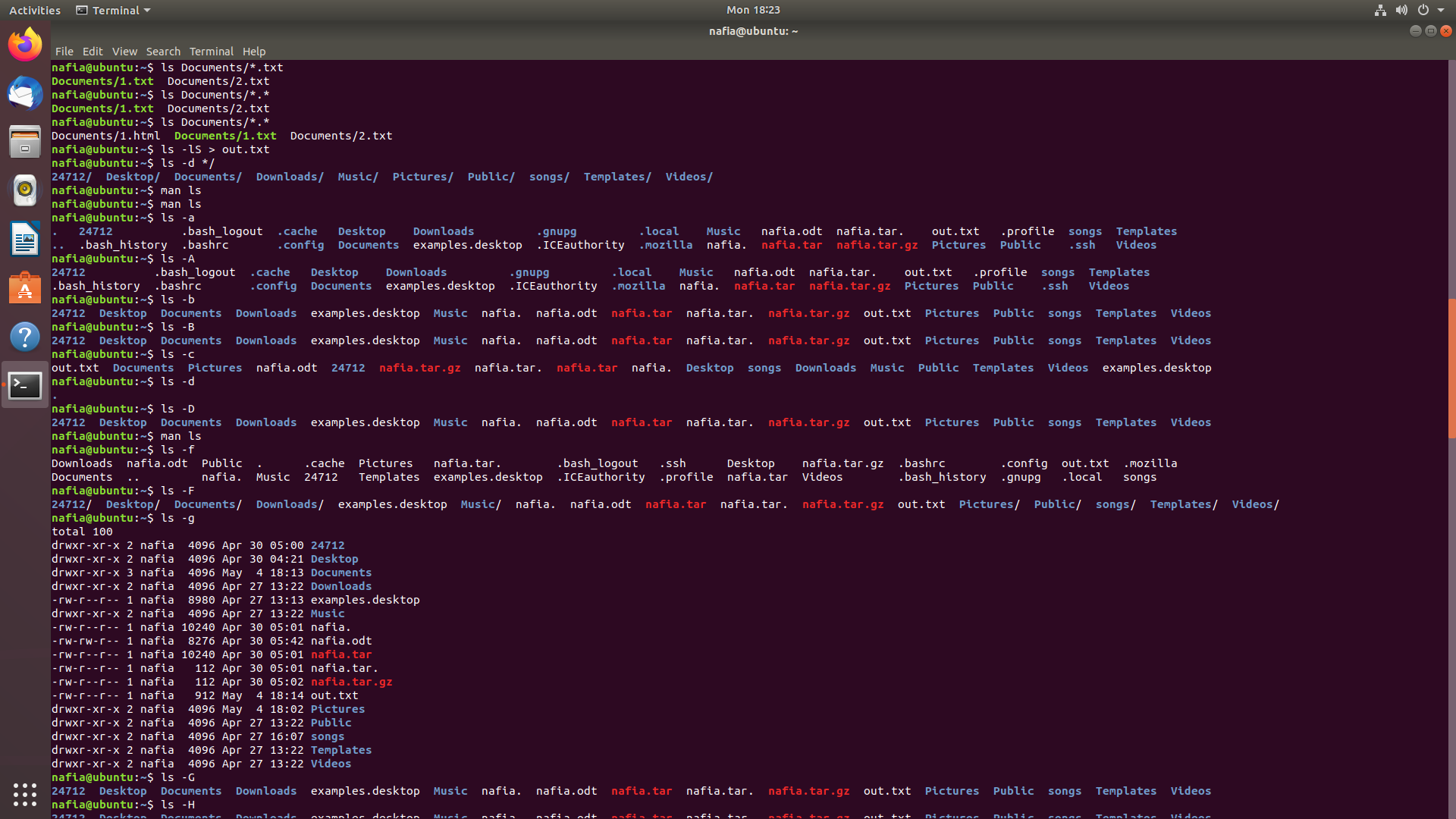
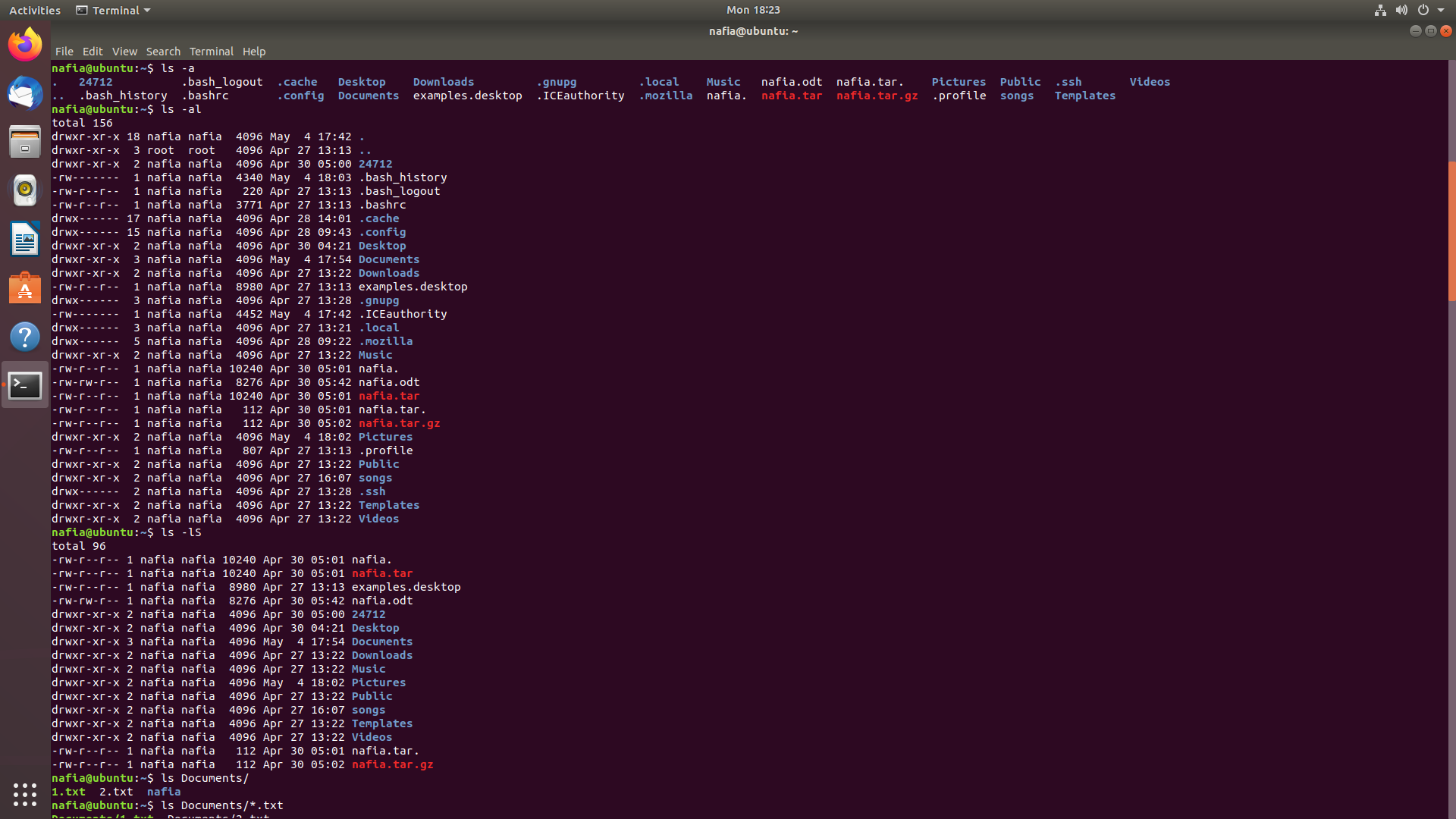
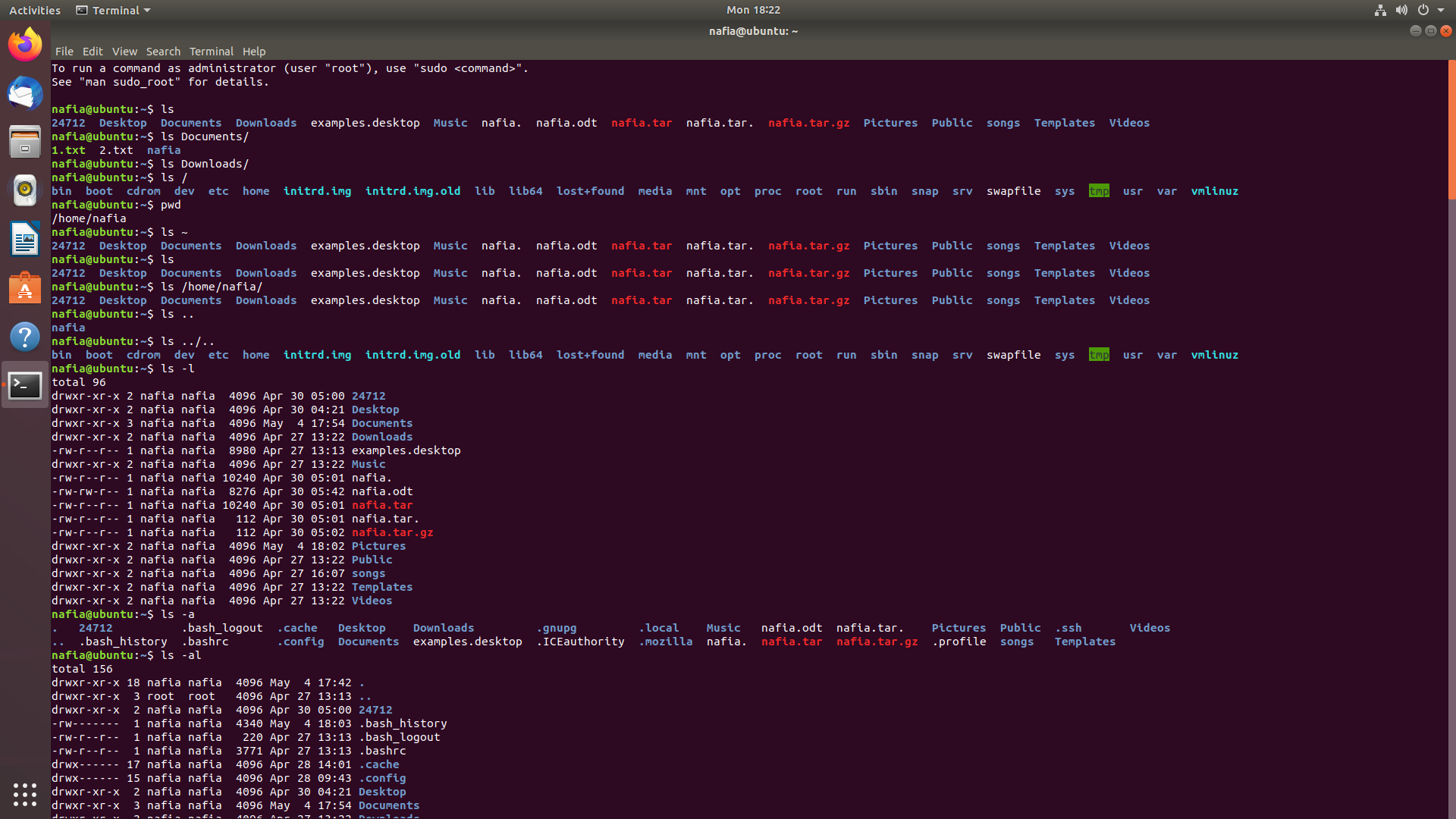
**1. pwd**

pwd stands for Print Work directory and does exactly what you think – it shows the directory you’re currently in. This is one of the handiest Linux terminal commands that aims to make new user’s life peaceful by ensuring they don’t get lost in that seemingly cryptic terminal window.



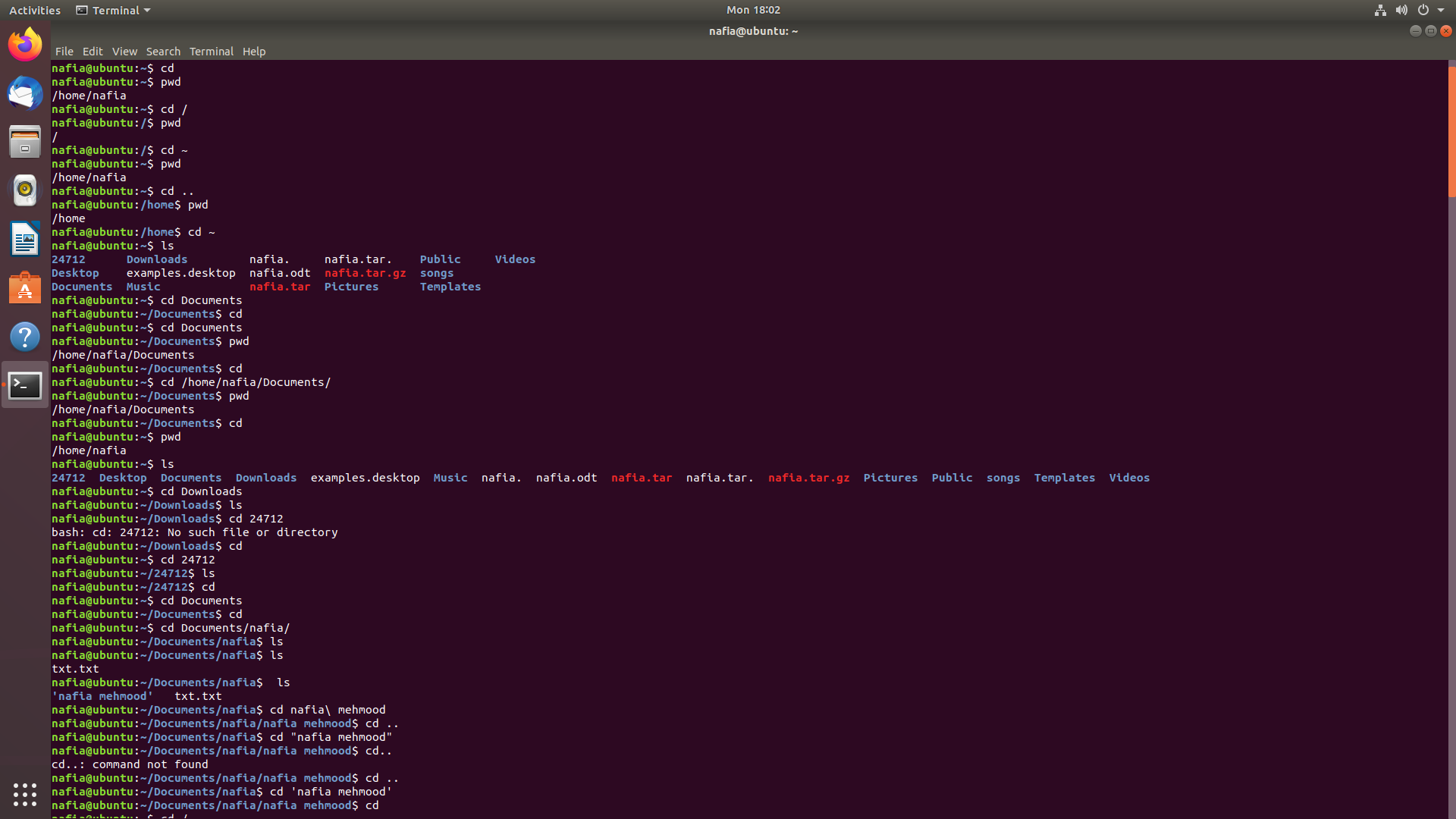
**2. ls**

The ls command is probably one of the most widely used commands in the Unix world. It presents to you the contents of a particular directory – both files and directories. You will use this command alongside pwd to navigate your ways inside the mighty Unix filesystem.



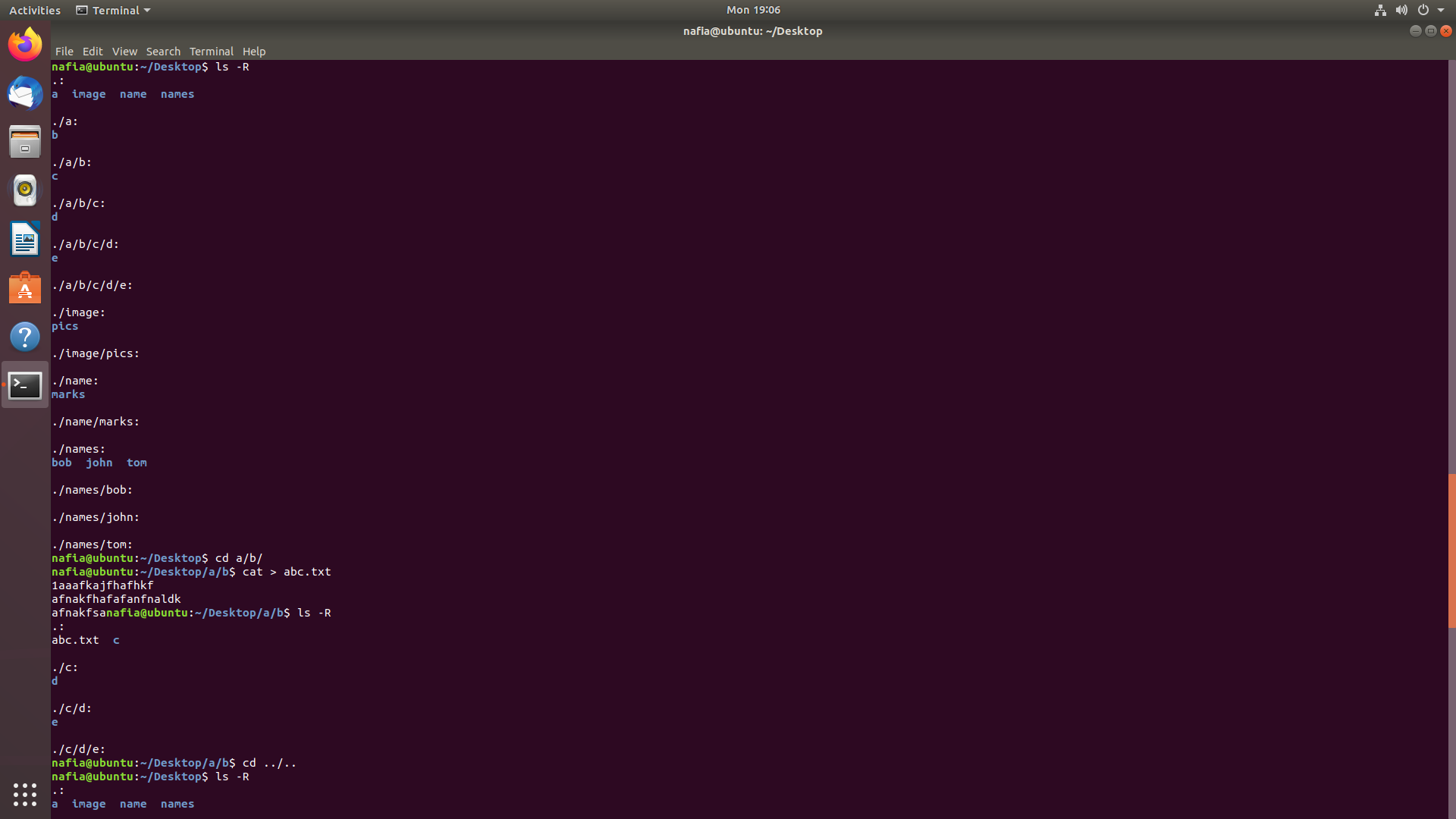
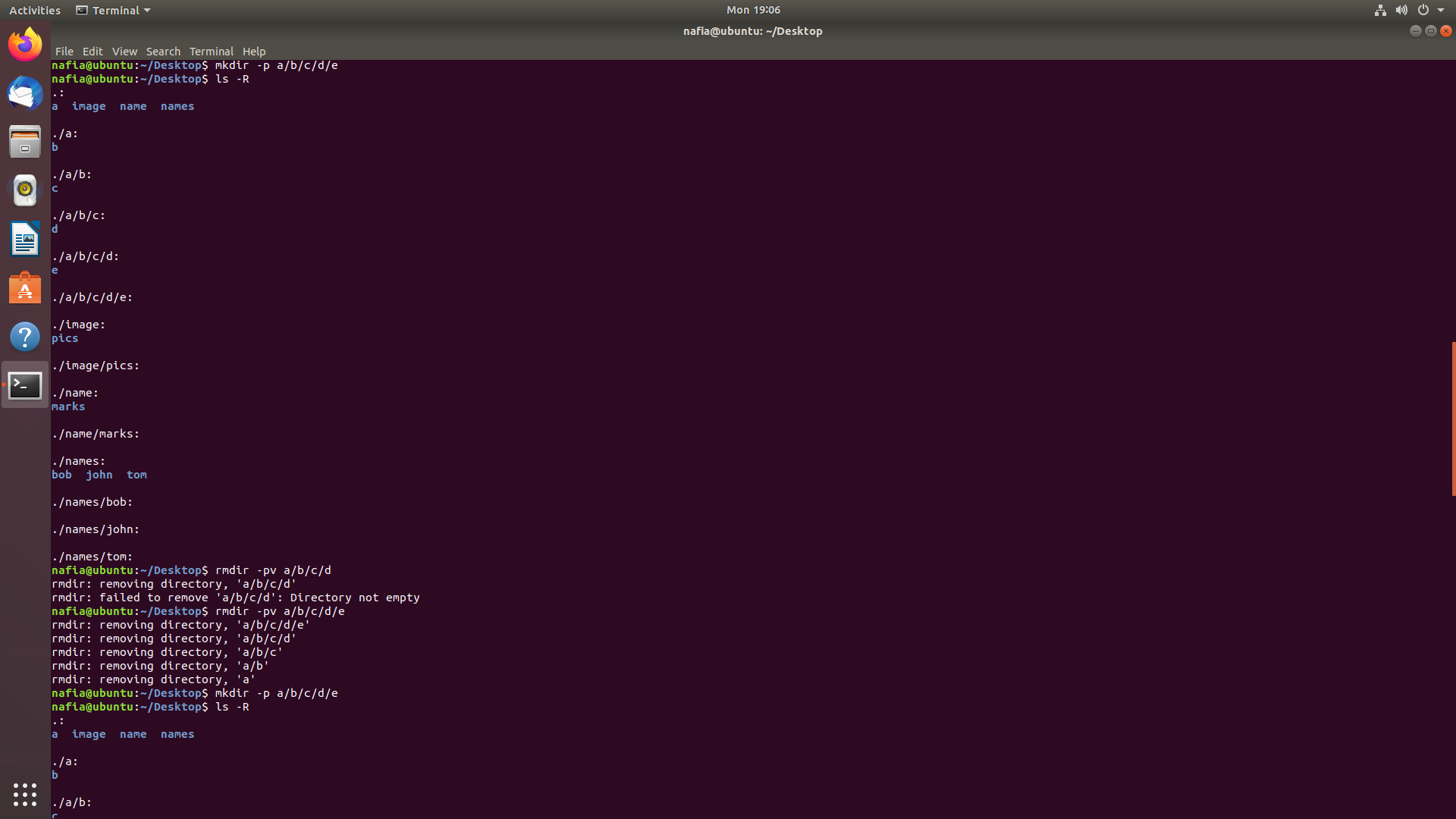
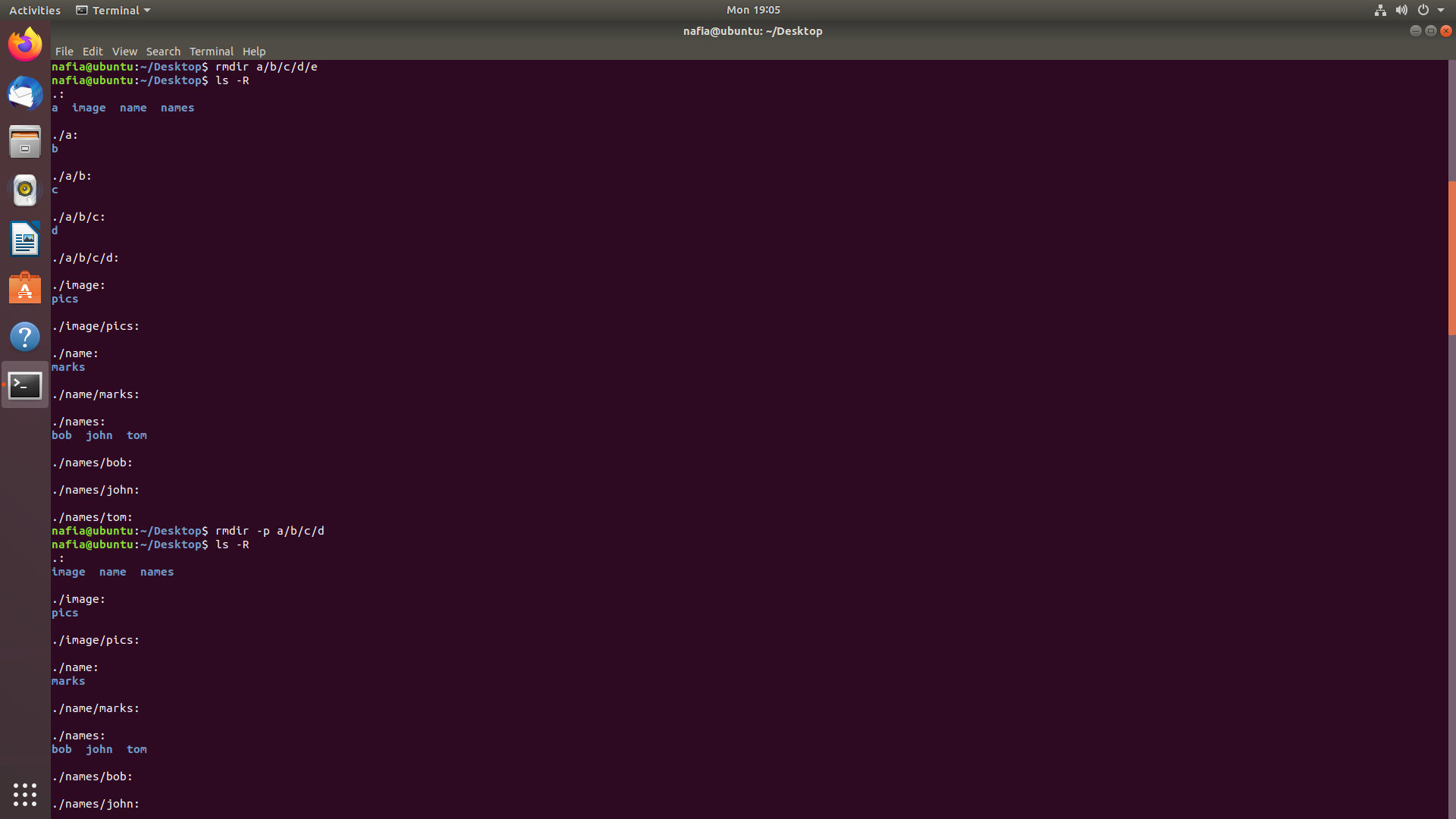
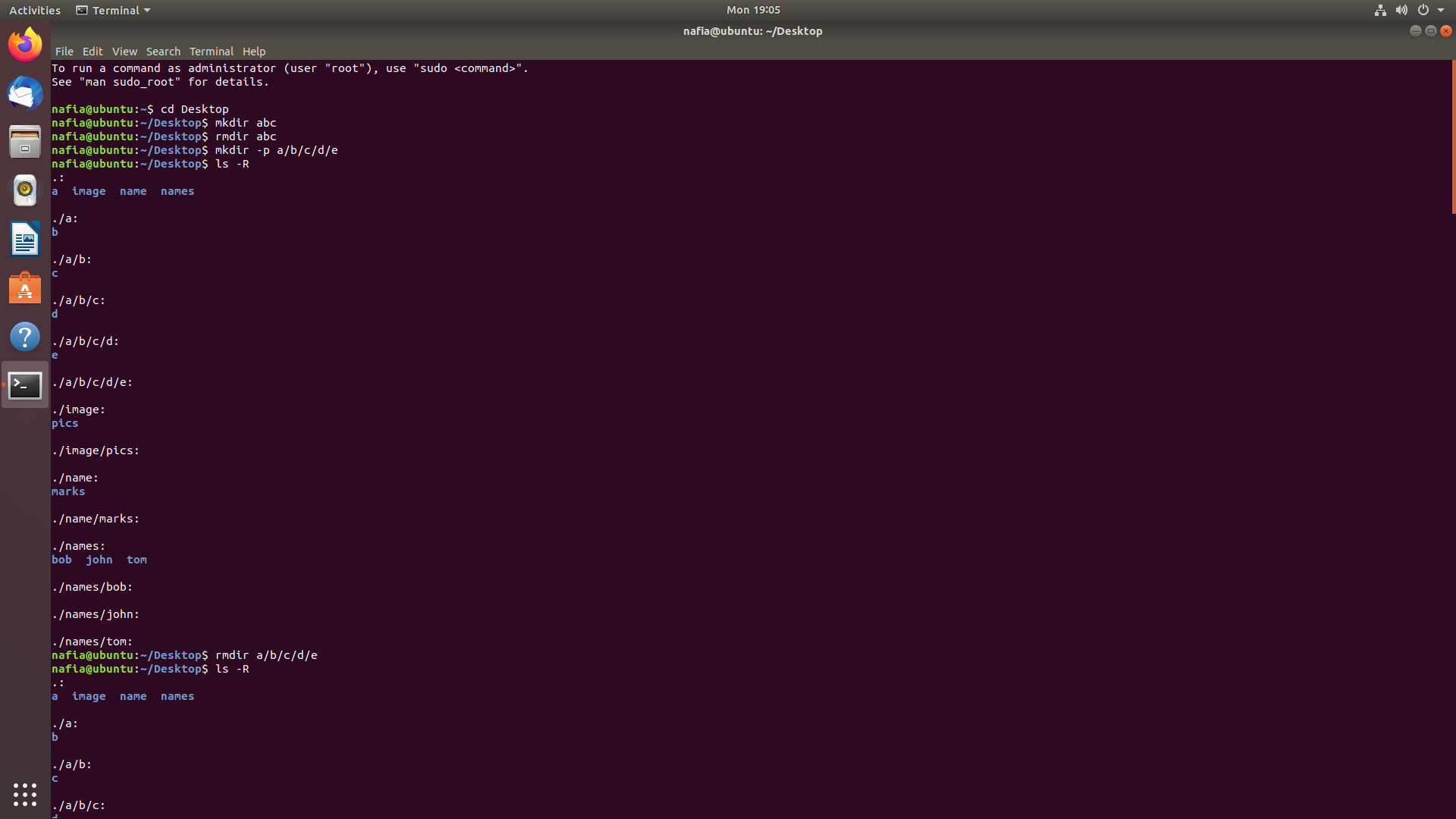
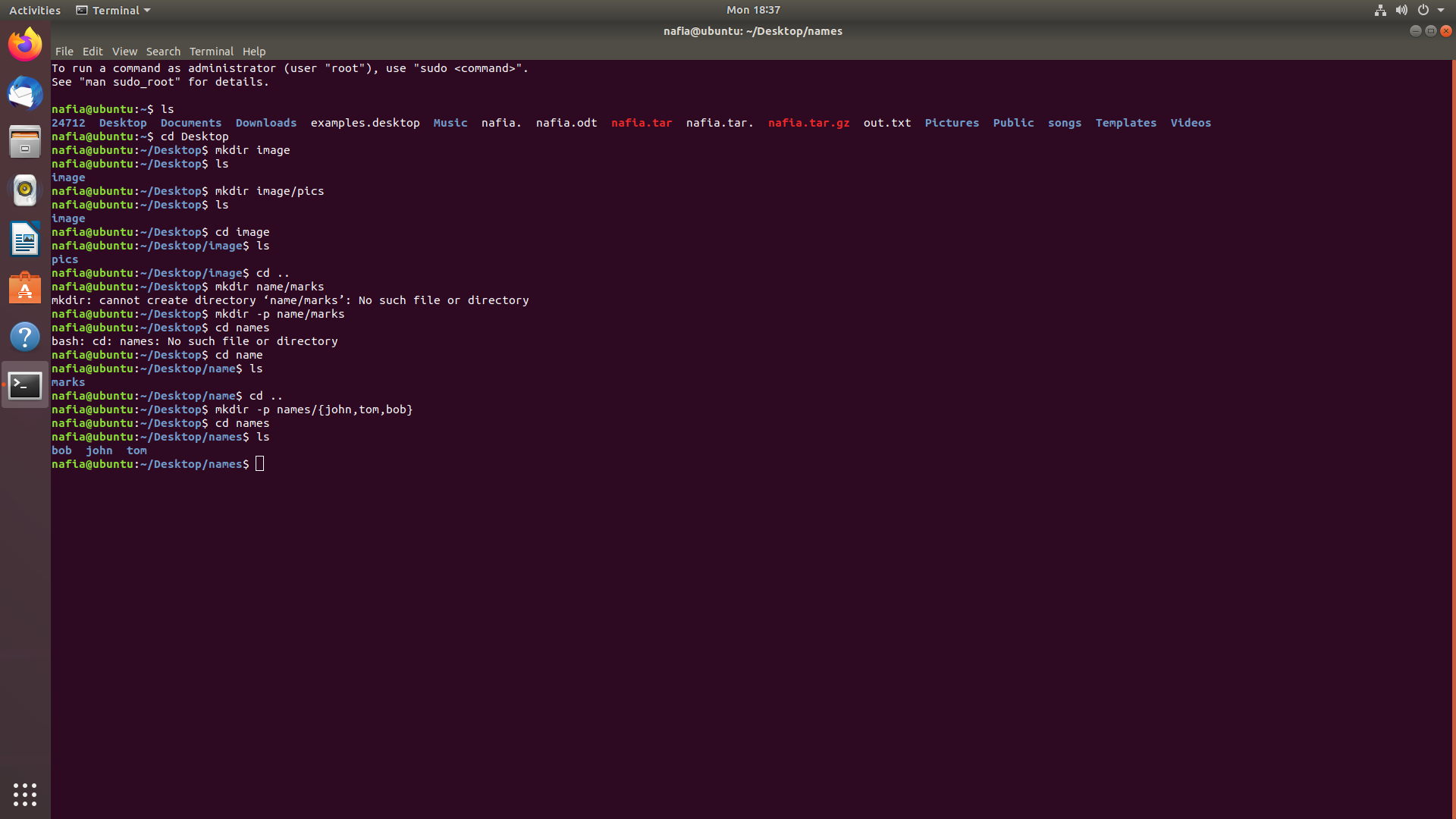
**3. cd**

Short for Change Directory, the cd command is behind your movement from one directory to another. It’s one of the few Linux commands that you’re *bound* to use throughout your stint with the Linux system. This command makes life in front of the terminal less scary for beginners while providing a standard method to browse the entire filesystem of your device.



**4. mkdir**

Want to create a new folder through the terminal? The mkdir command is created for just this specific purpose. It lets you create folders anywhere you like in your Linux system – given you have got the necessary permission, of course!



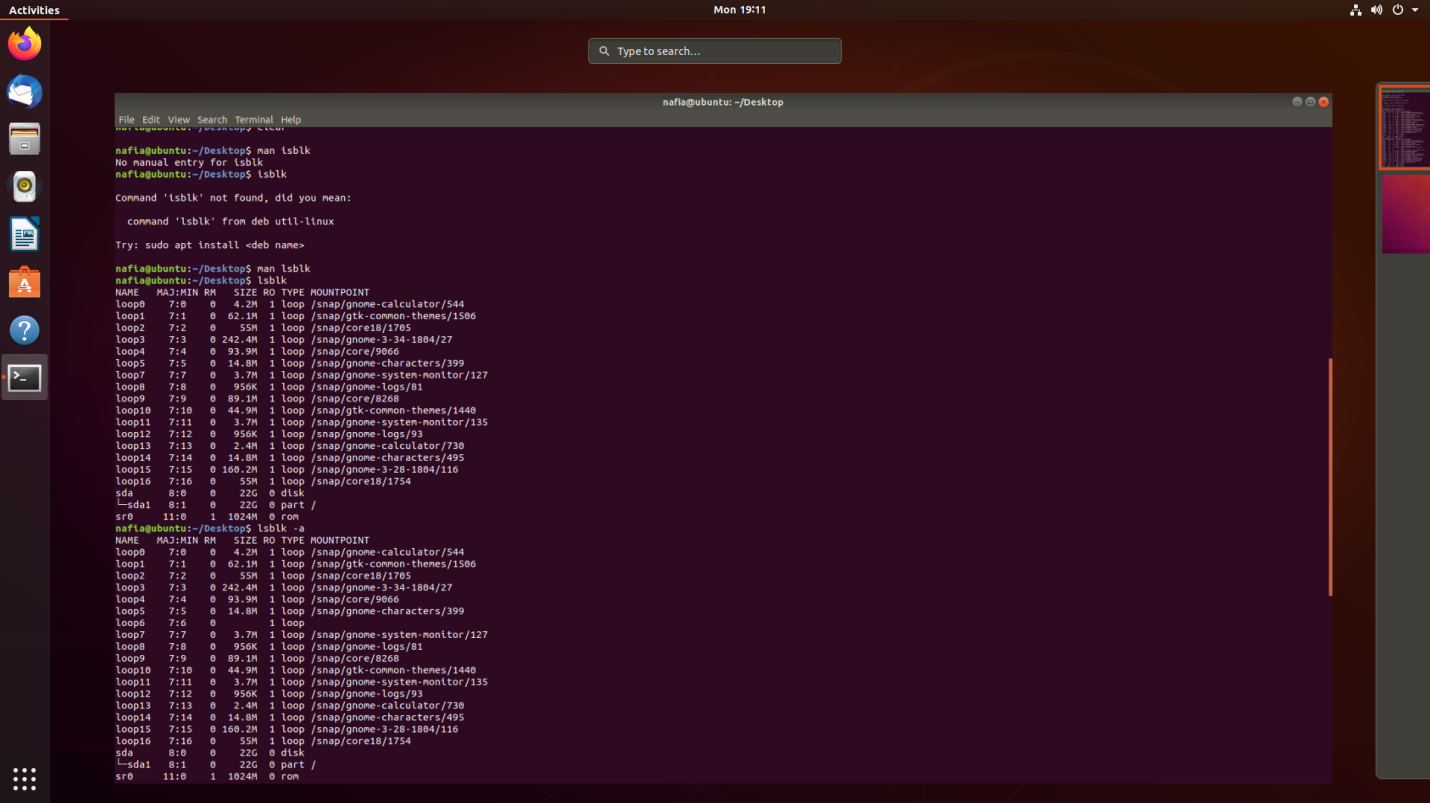
**5. rmdir**

The arch rival of the mkdir command, the rmdir command allows you to delete specific folders from your system without any hassles. Although many utilize the rm command for this purpose, screwing up parameters or even a single character with rm can do things you wouldn’t even dream. So, stick with rmdir for now.



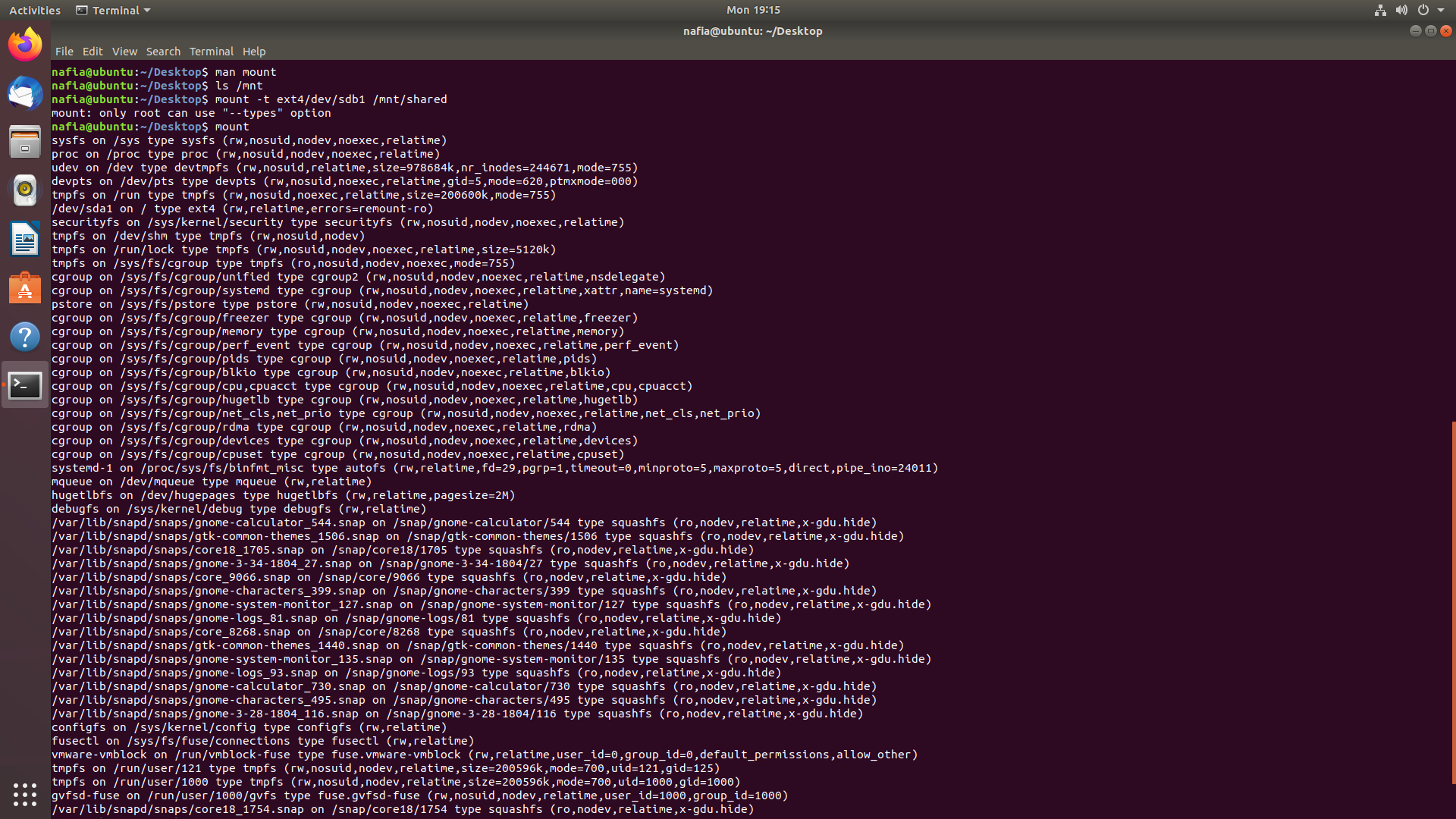
**6. lsblk**

Often you will find the need to list the available block devices of your Linux system. The lsblk is one of the most used Linux commands for this purpose. This handy terminal command will present you with a tree structure of your block devices and is used heavily by professional users.

****

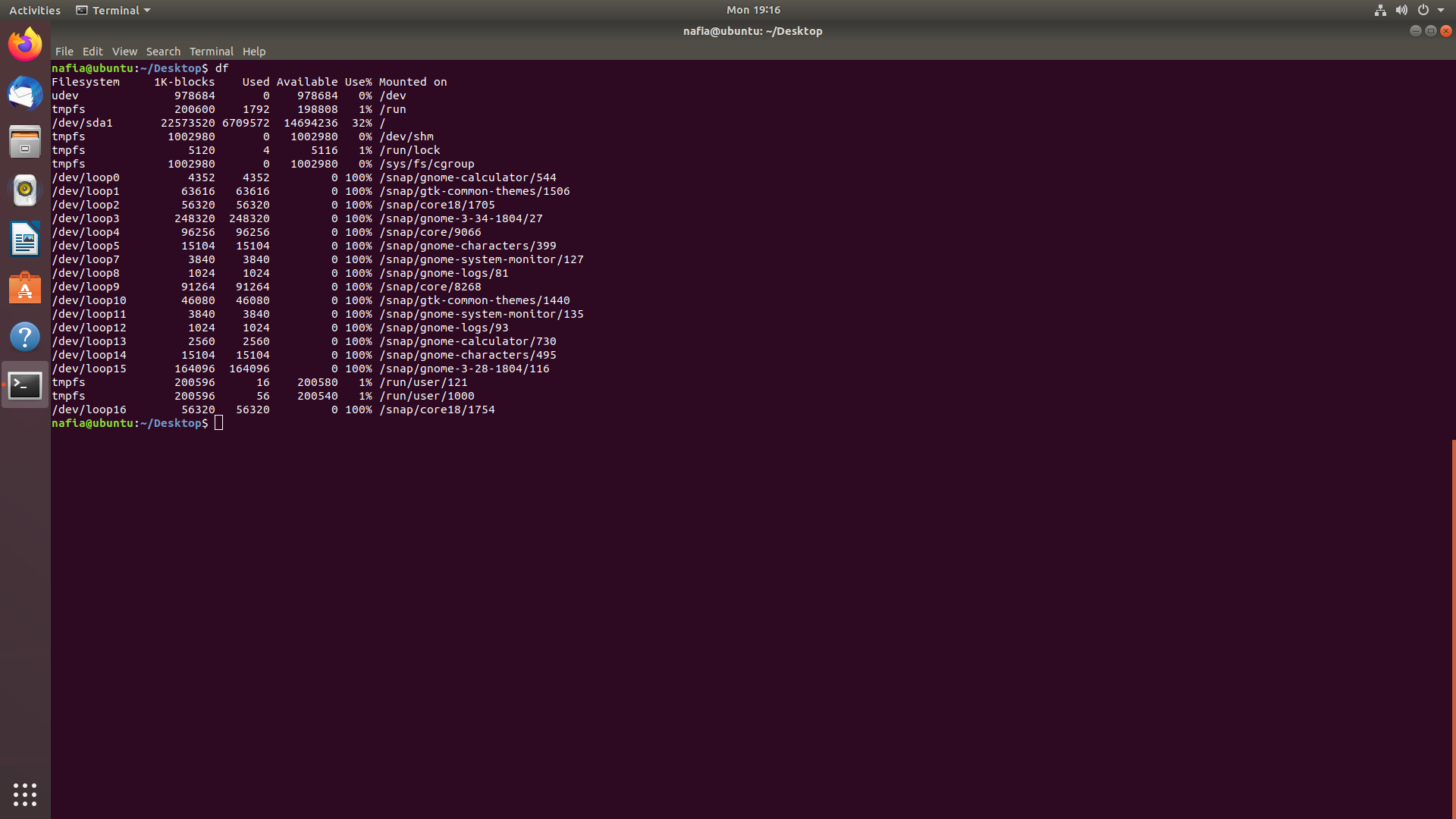
**7. mount**

Contrary to Windows, whenever you plug in an SD card or a USB, chances are your distro won’t show them directly at the start. You need to mount it with your existing filesystem using the mount command. This Linux command is one of the most powerful terminal commands out there.



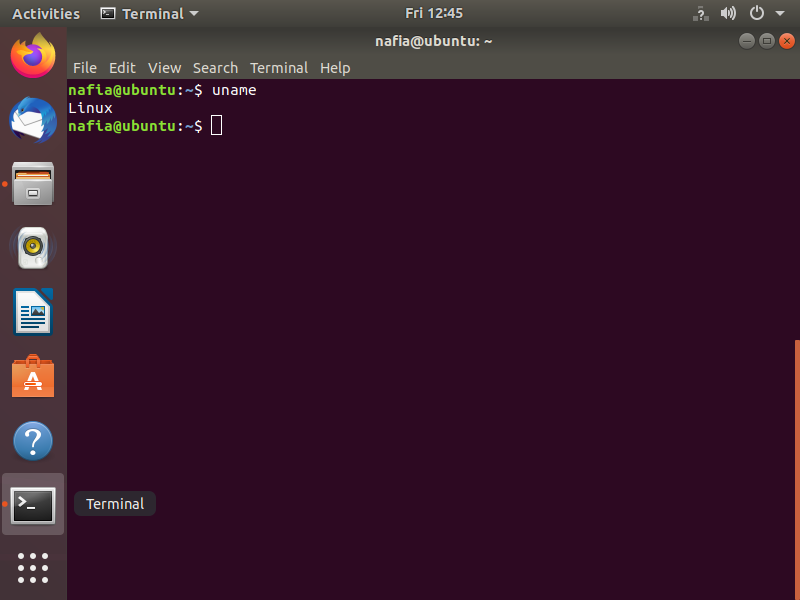
**8. df**

The [df command](https://www.ubuntupit.com/practical-and-productive-example-of-linux-df-commands/" \o "40 Practical and Productive Example of Linux df Commands" \t "_blank) is one of the most empowering Linux terminal commands that display essential information about the disk space on your filesystem. It is used widely by system administrators to monitor and analyze real-time server or network-oriented systems.



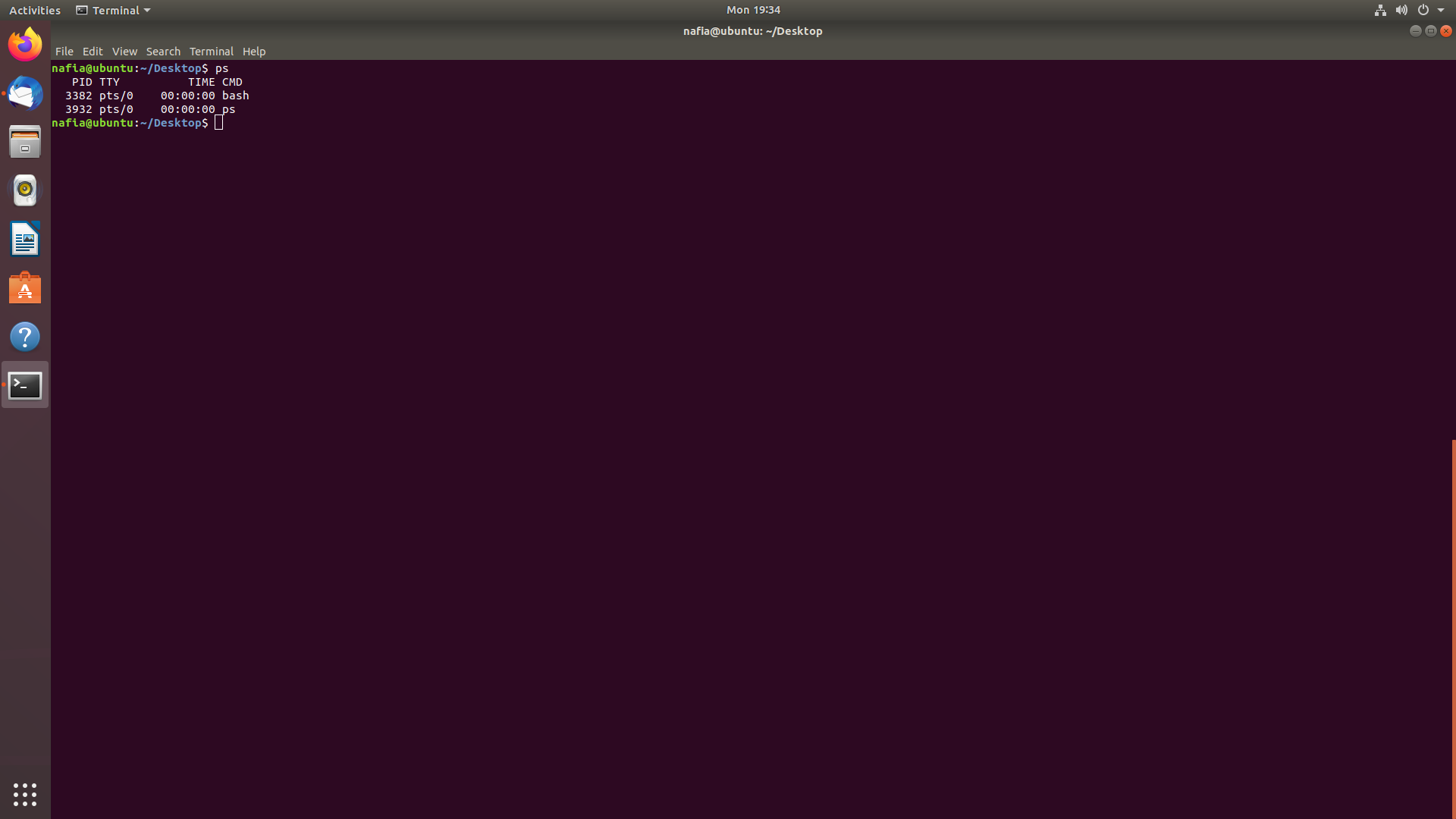
**9. uname**

The uname command is an elementary Linux command for obtaining system information like name, version, and other system-specific details. You can quickly check your OS and kernel version with this command and can know for sure the instruction length of your machine.



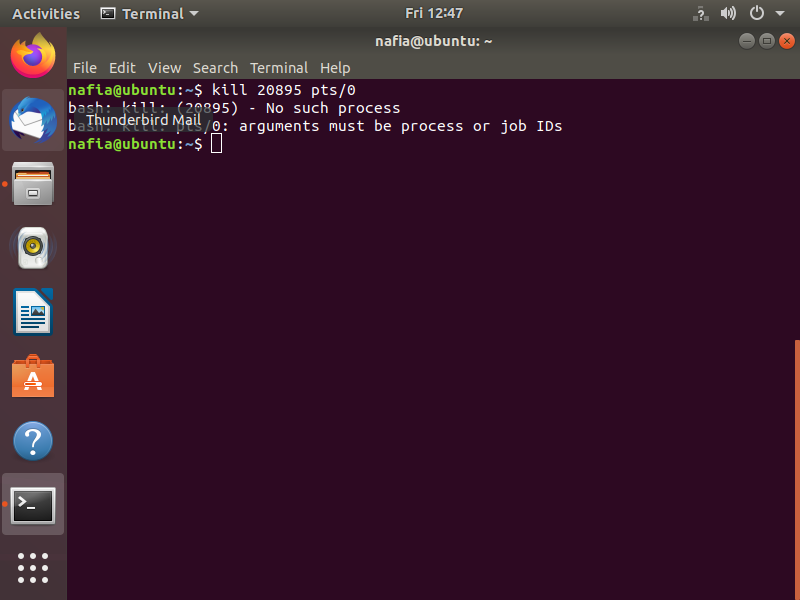
**10. ps**

This is among other straightforward terminal commands that will allow you to visualize what processes are currently run by your machine. It’s a very elegant yet flexible mean to analyze system resources and will enable you to manipulate system processes natively through the terminal. This command is considered as one of the basic and [best Linux monitoring tools](https://www.ubuntupit.com/most-comprehensive-list-of-linux-monitoring-tools-for-sysadmin/) available for Linux nerds.



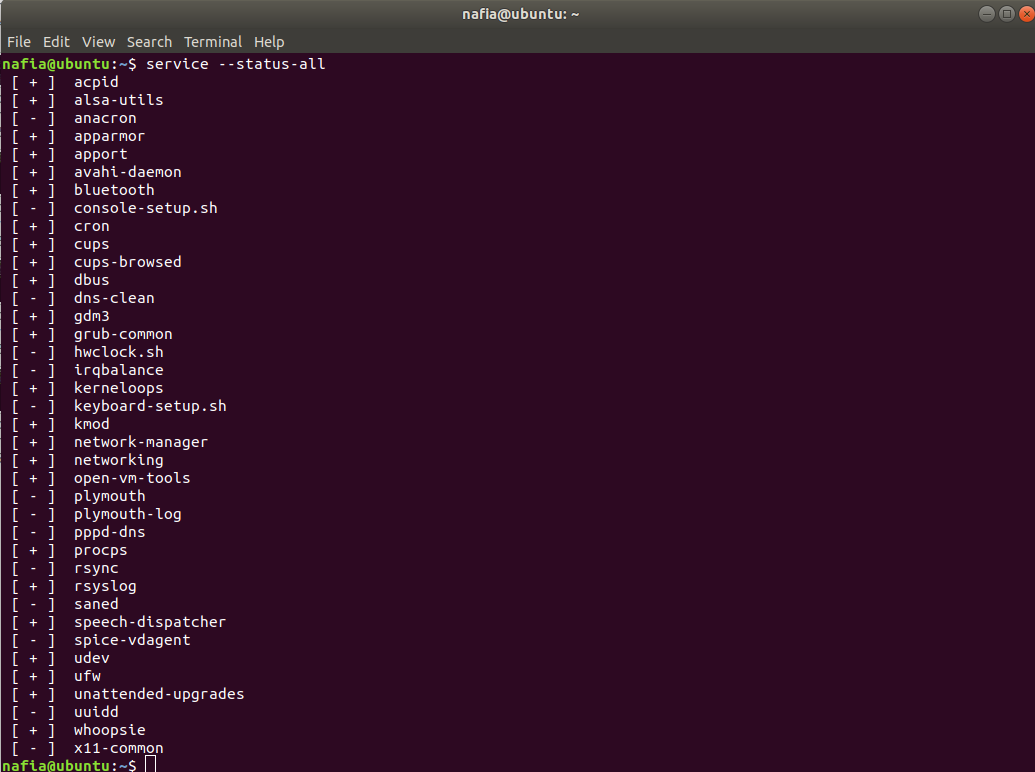
**11. kill**

The kill command is a powerful way to stop processes that are stuck due to resource constraints. As you grow your Linux system skills, you will come to know the essence and importance of this command. Often presented in lists of funny Linux commands, the kill command is as mighty as its name.



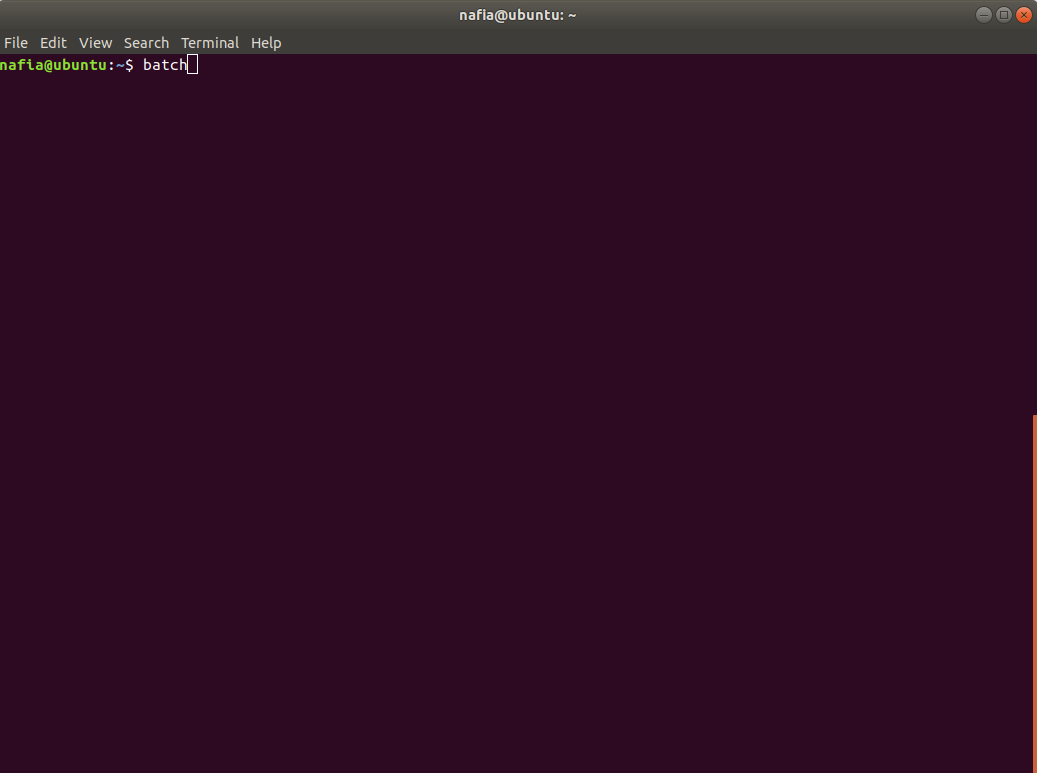
**12. service**

The service command is the de-facto command to invoke system-wide services from the terminal. A powerful Linux terminal command for manipulating the system; you can leverage this command for running any *System V init* script directly from the terminal window.



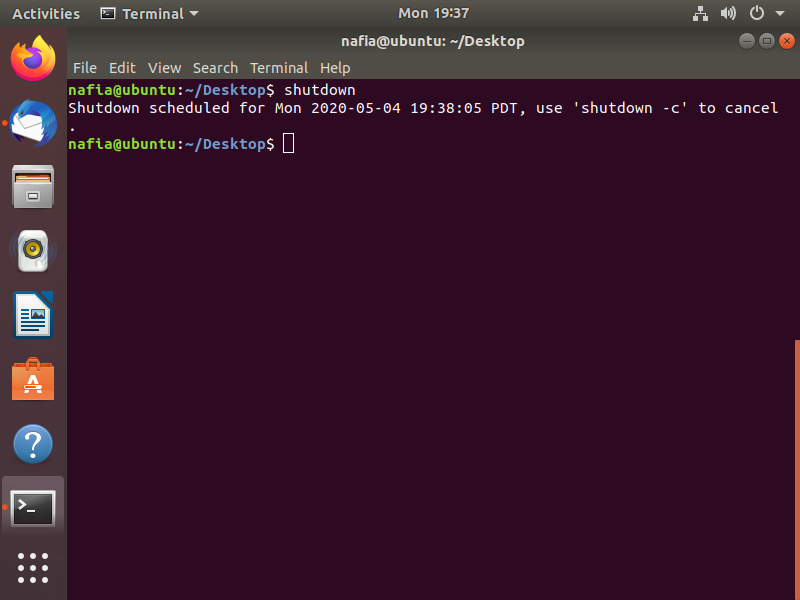
**13. batch**

If you are looking for a neat tool that will run system services in a pre-defined schedule, the batch command is here for your redemption. Another powerful Linux command for writing automation shell scripts; this flexible little command can aid to your [Linux productivity](https://www.ubuntupit.com/top-15-linux-productivity-software-and-tools-that-worth-your-attention/) significantly.



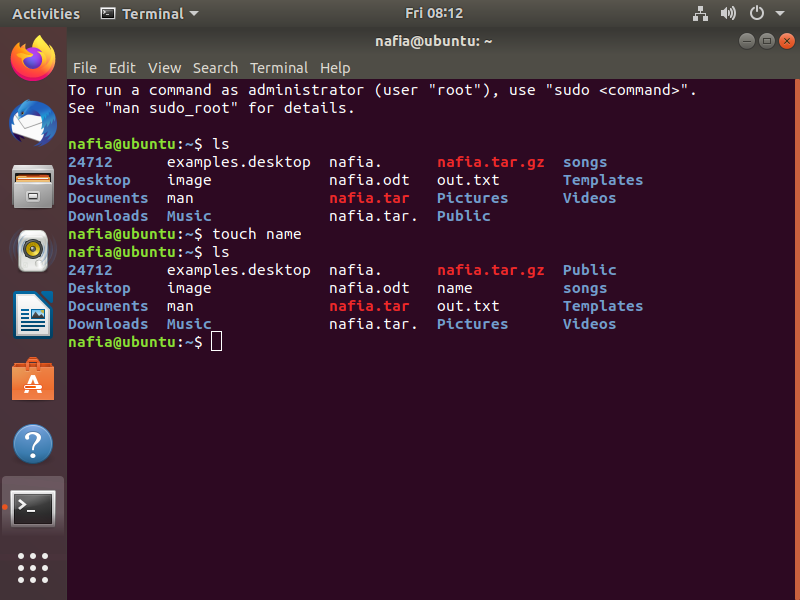
**14. shutdown**

Feeling a bit lazy and don’t want to click a few GUI elements at the end of a long night? The shutdown command is here for empowering your Linux terminal commands skills to a whole new level. Just don’t type it now while reading this post!



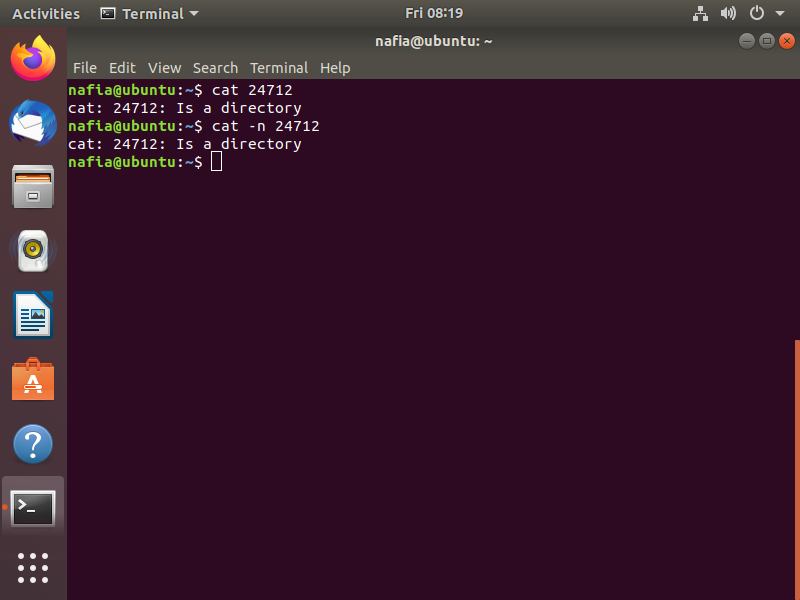
**15. touch**

The touch command is an essential Linux command for creating a valid empty file. You can create files on the go in your terminal and fill them up later or in real-time – based on your requirements. It’s also the go-to command for changing the timestamps.



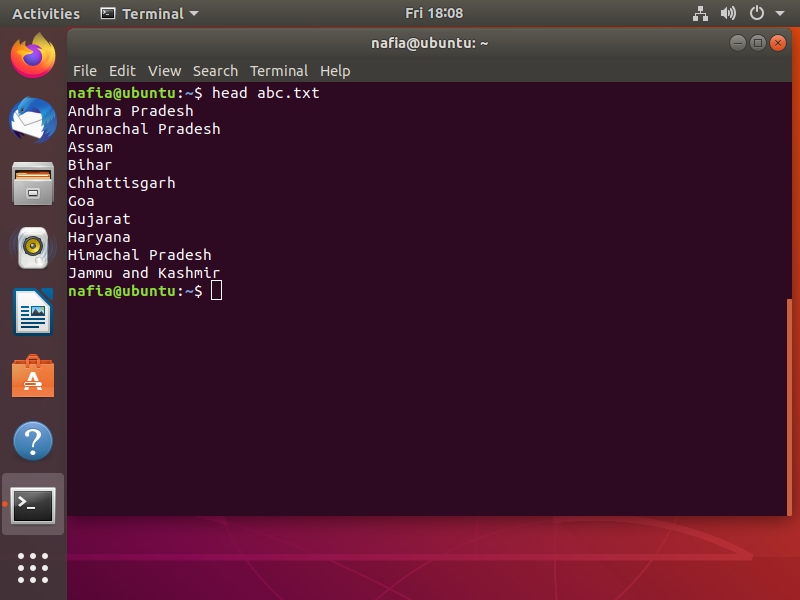
**16. cat**

Designed initially for concatenating multiple files, the cat command is used for numerous other purposes since. This is among other Linux commands you will use to create new files, view file contents in the terminal, and redirect output to another command-line tool or file.



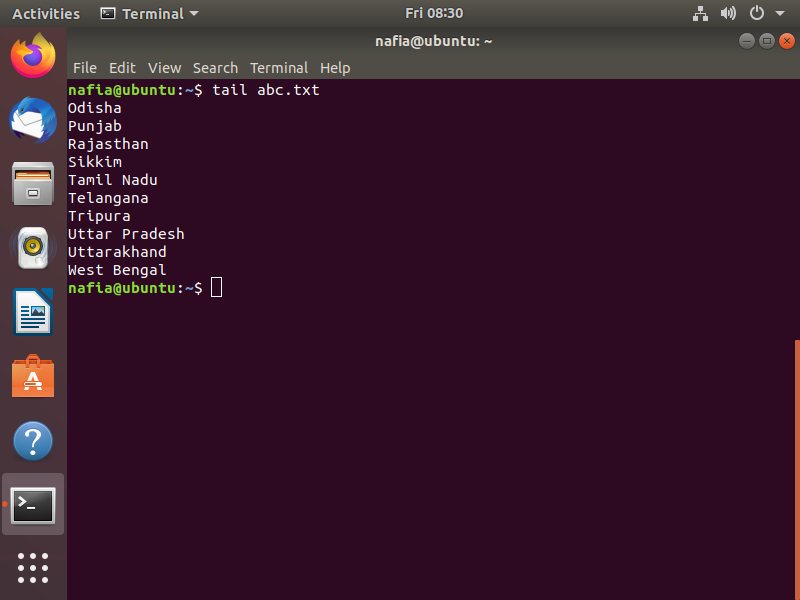
**17. head**

The head command allows you to view the beginning of a file or piped data directly from the terminal. It’s one of the most widely used Linux commands by users who works heavily with [text processing](https://www.ubuntupit.com/best-latex-editor-top-33-reviewed-for-linux-nerds/). Use this command whenever you are going through a lot of files in the terminal to increase your productivity.



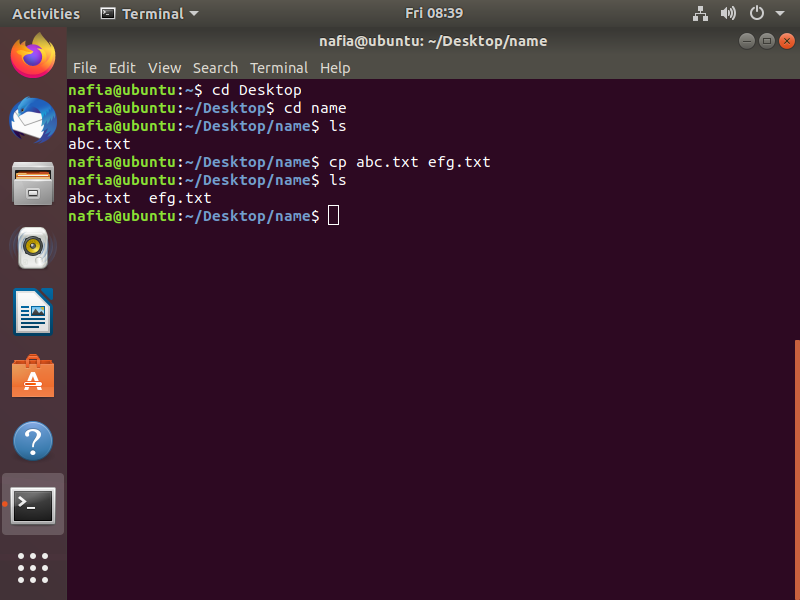
**18. tail**

A compliment to the previous command, chances are you will use the tail command much more than the header commands. A basic Linux terminal command, tail, mixed with cat and echo can do things you wouldn’t even imagine.



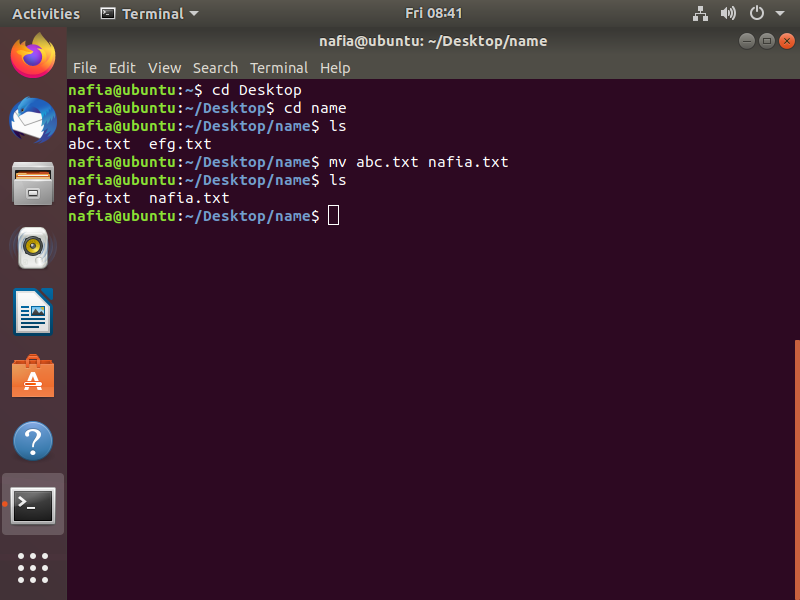
**19. cp**

The cp command is just a short way of telling your machine to copy a file or directory from one folder to another. It is among other de-facto Linux commands you can’t live without. You can copy multiple files to a directory right from your terminal with this neat command.



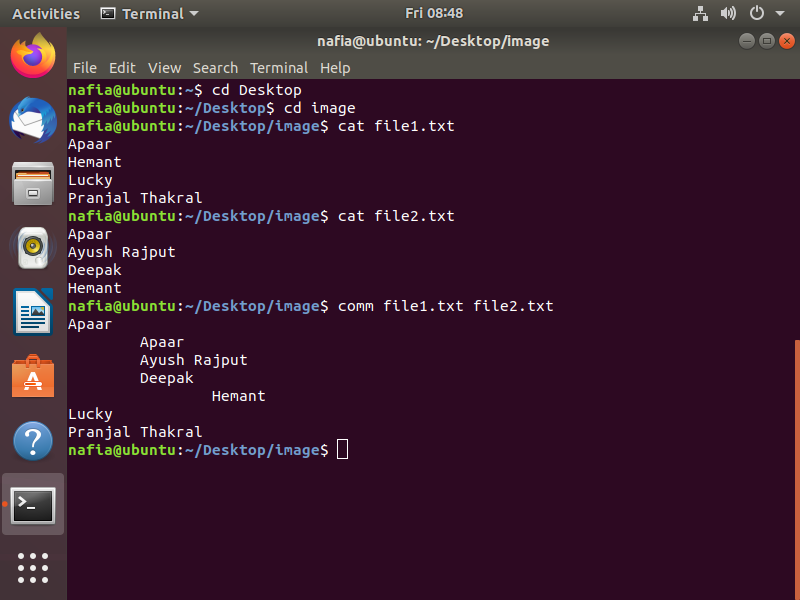
**20. mv**

Short for a move, it’s a supplement to the cut operation you perform in the GUI. Just like cp, you can use the mv command to move either single or multiple files from one location to another. You can force this Linux command to transfer large files with the -f parameter.



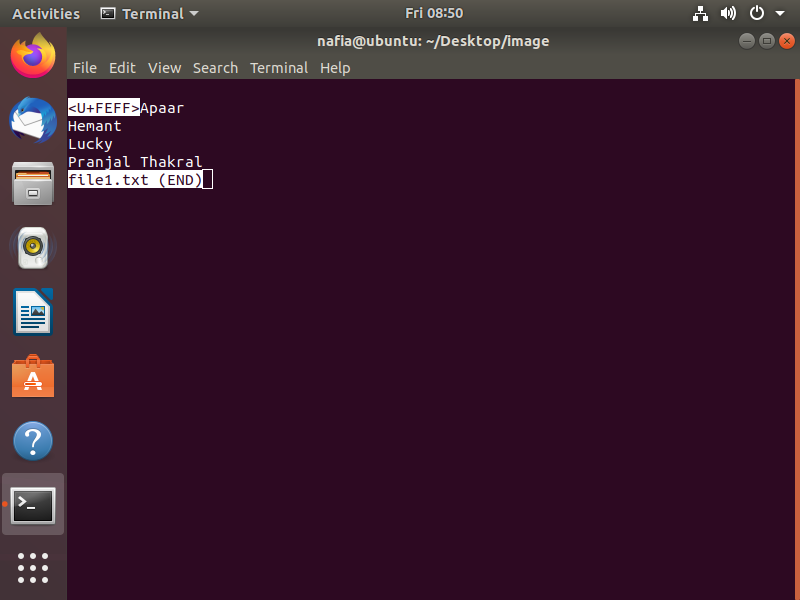
**21. comm**

One of the original Unix commands that found its way to the Linux world, you can use comm to compare two files for common and distinct lines. This is an essential tool for many that need to process large amounts of files on the terminal.



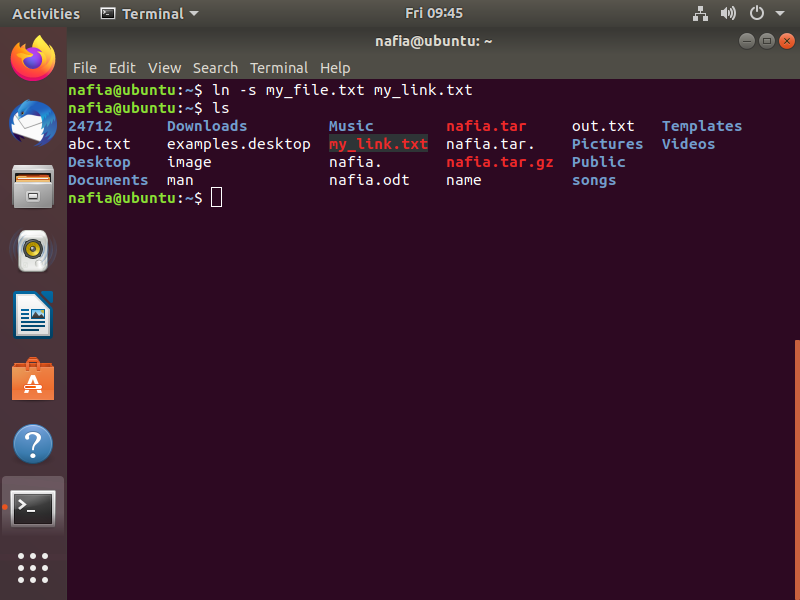
**22. less**

Another most used Linux command, the less command, is widely popular due to the convenience it brings when viewing the contents of a file. Contrary to cat, the less command allows users to navigate within a file in both directions while not interrupting the terminal session.



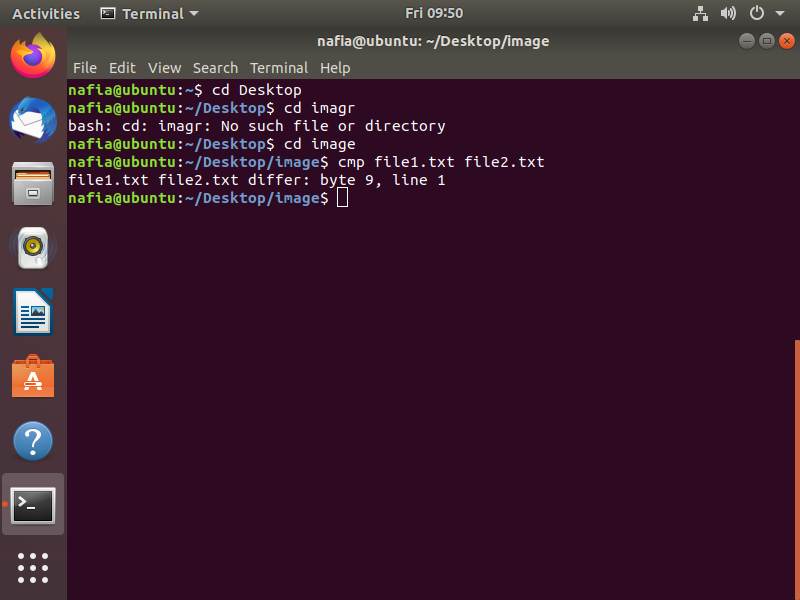
**23. ln**

The ln command is one of the handiest Linux commands for creating symbolic links to some specific file. You can use this tiny yet flexible command to produce multiple instances of a symbolic link to a particular file or directory on your disk space.



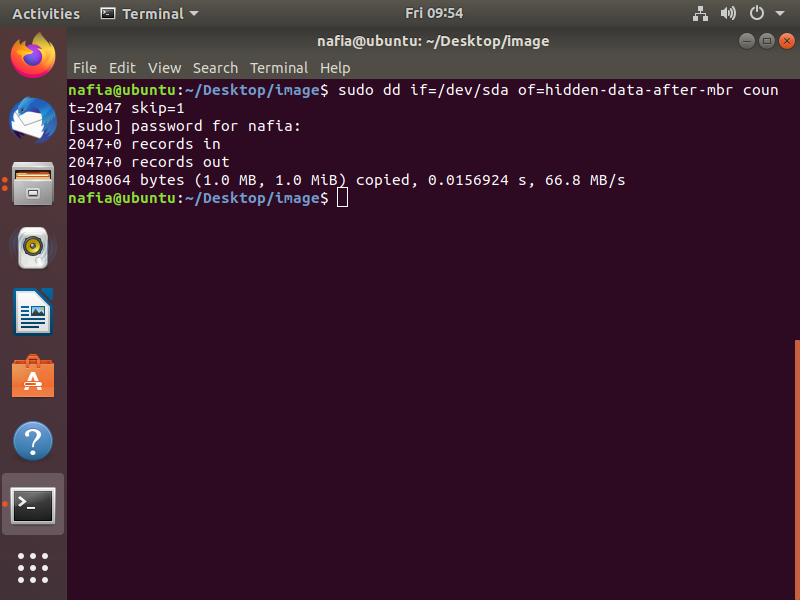
**24. cmp**

If you want to compare two files and print the result to the standard output stream, the cmp command will let you do exactly so. This, alongside the comm command, is one of the most used Linux commands by users that process large volumes of text files on a regular basis.



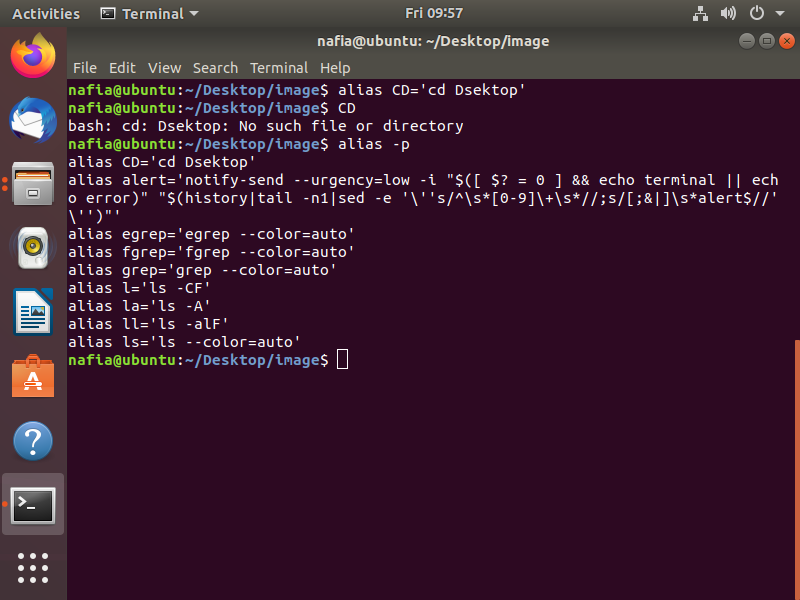
**25. dd**

This is one of the most used Linux commands by veteran users to copy and convert files from one type to another. The exciting thing about this little yet powerful command is that you will often use it among other terminal commands when creating bootable live USB sticks.



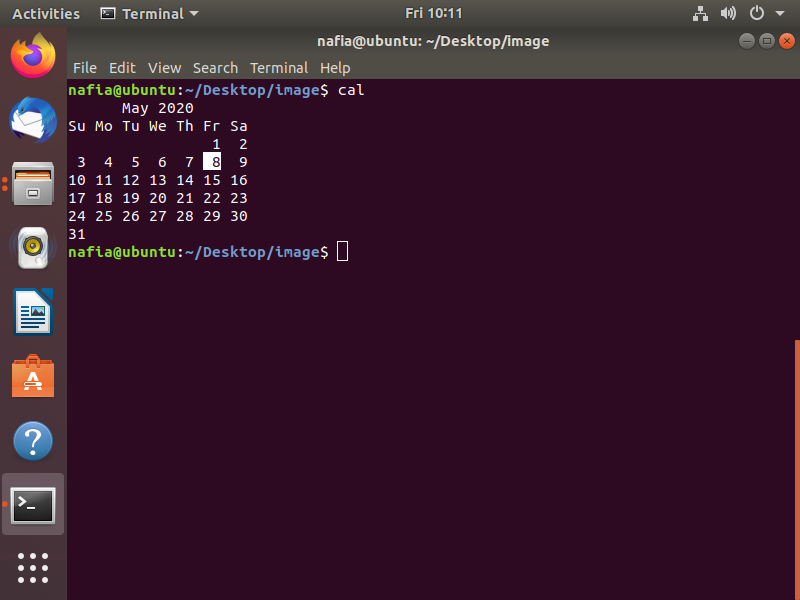
**26. alias**

It is one of the most used Linux commands by system admins as it lets them replace a word by another string in files directly from the terminal. Among other functions, this is one of the best terminal commands through which you can customize the shell and manipulate the environment variables.



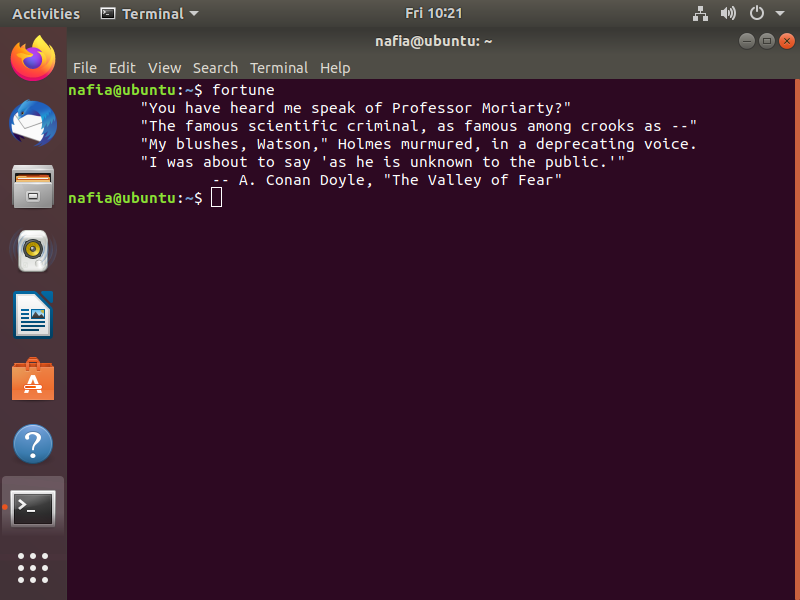
**27. cal**

Did you know Unix provides a handy little calendar tool right into the terminal? The cal command is one of those funny Linux commands that shows you the calendar in an ASCII text format. Type this command with parameters like month and year to get specified information right into the terminal.



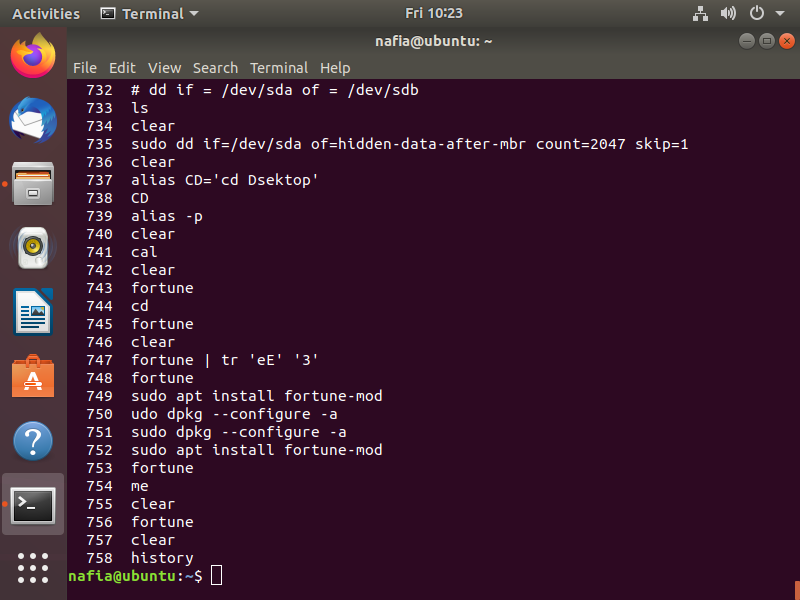
**28. fortune**

This is one of my most favorite funny Linux commands. Just type it in the terminal and see for yourself! It shows out of the blue poignant, inspirational, or silly phrases that will definitely make you laugh out loud.



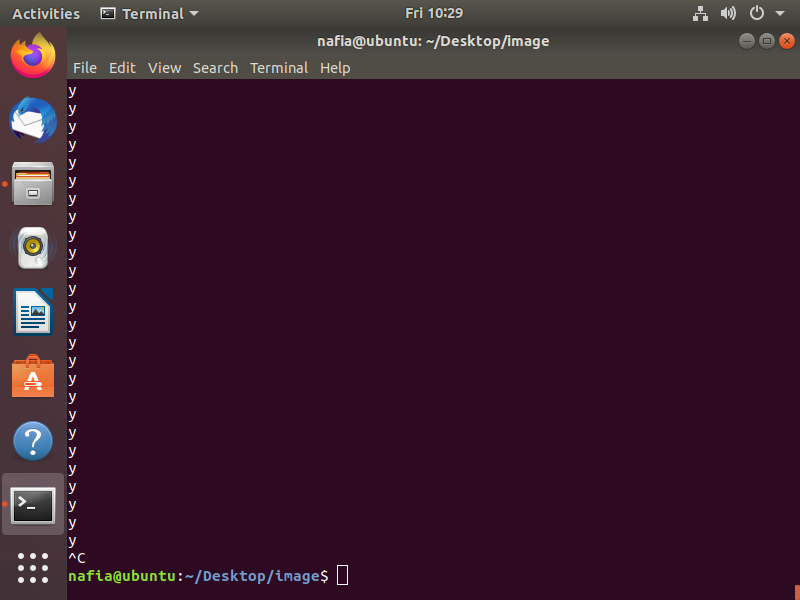
**29. history**

Want to check the history of your terminal sessions? The history command lets you do precisely this. When typed without any parameters, this will print out the bash history of your terminal session right in the tiny window. It’s not only among the most useful funny Linux commands you will use in the terminal but also provides a great insight into your terminal usage.



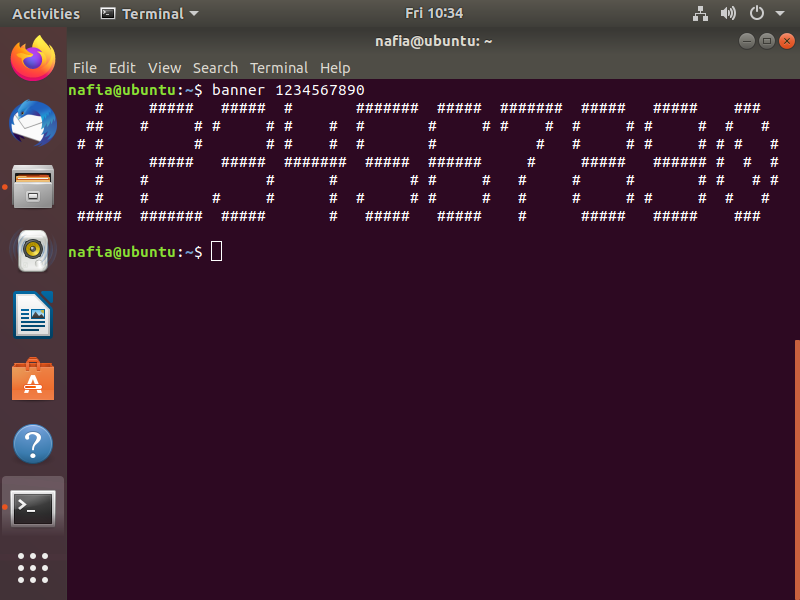
**30. yes**

Okay, you want some real funny Linux commands to try right now? Just open up a terminal type in yes, then some texts of your choosing and press enter. We’re pretty sure it’s among those terminal commands you will never forget ever since.



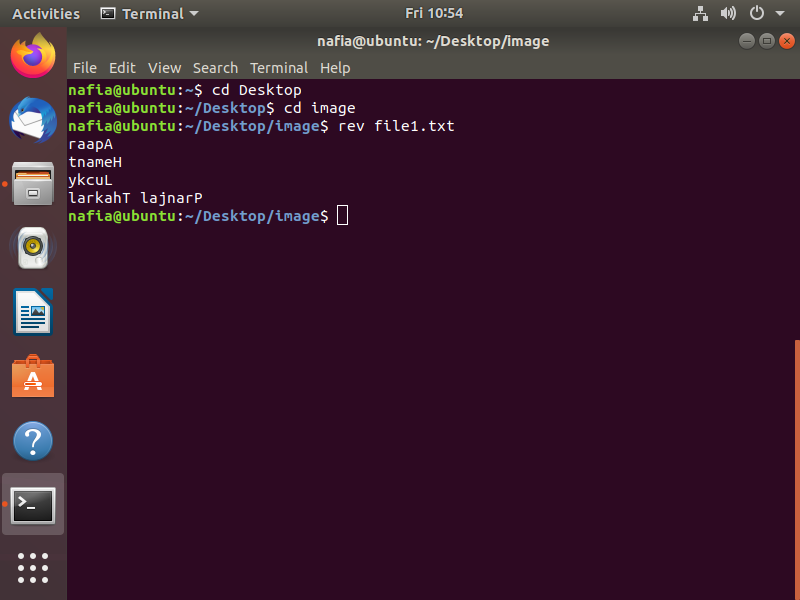
**31. banner**

Ever saw some great ASCII banners used by old school Unix legends and fall in love right away? The banner command is one of those funny Linux commands that will empower you to create your own personalized such banners. Just enter this command mixed with some texts in the terminal to get a feel.



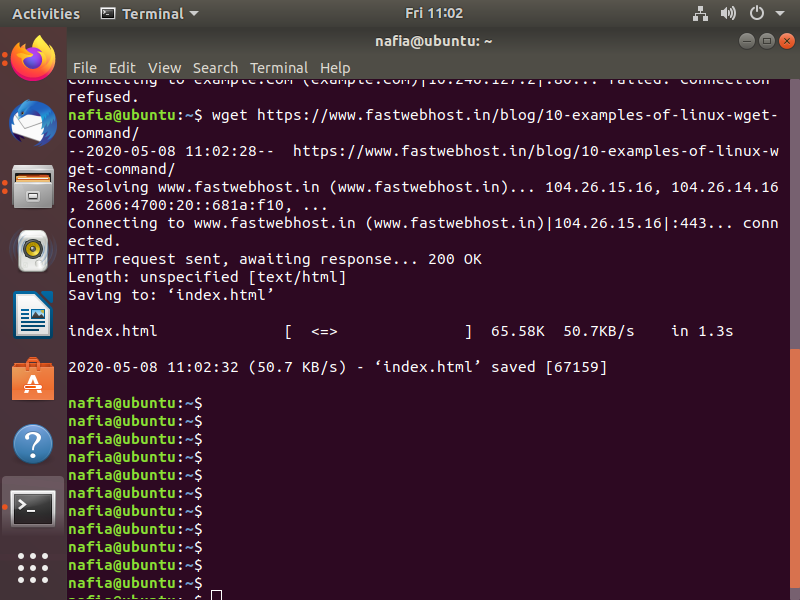
**32. rev**

This is another funny terminal command for veteran Linux users. The rev command will take your input text and write them to the standard output, reversing each character! It’s one of those terminal commands you can use to make a cryptic yet subtle impression on your friends.



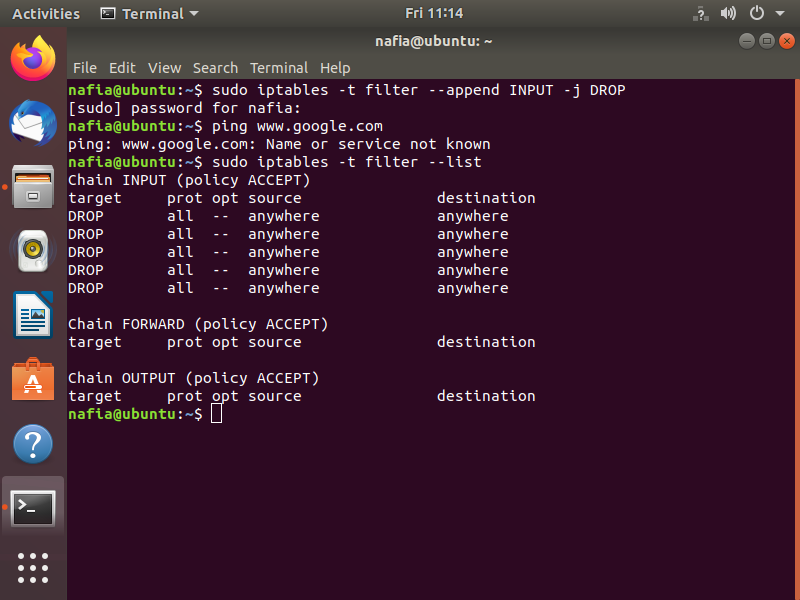
**33. wget**

This is one of the best Linux commands network admins leverage to [download files](https://www.ubuntupit.com/download-manager-for-linux-top-8-reviewed-compared/) from the web right from the terminal. This is among those handy little terminal commands that can be used in scripts or cronjobs and provides users the ability to use the HTTP, HTTPS, and FTP internet protocol.



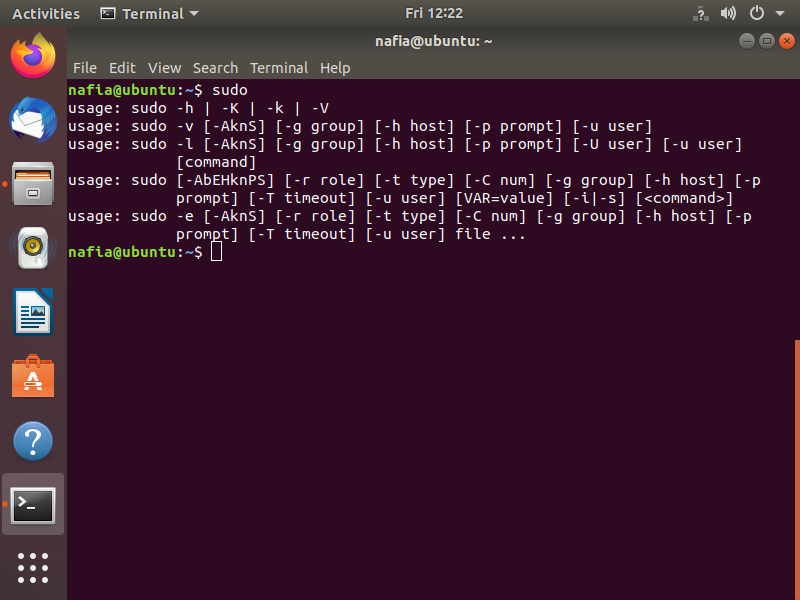
**34. iptables**

The [iptables command](https://www.ubuntupit.com/useful-and-simple-iptables-rules-for-linux-administrator/" \o "50 Useful and Simple IPtables Rules for Linux Administrator" \t "_blank) invokes a terminal utility that lets system admins control the incoming and outgoing internet traffic on a particular host machine. It is among the most used Linux commands sysadmins use on a regular basis to define authentic traffics and for blacklisting suspicious or untrusted network requests.



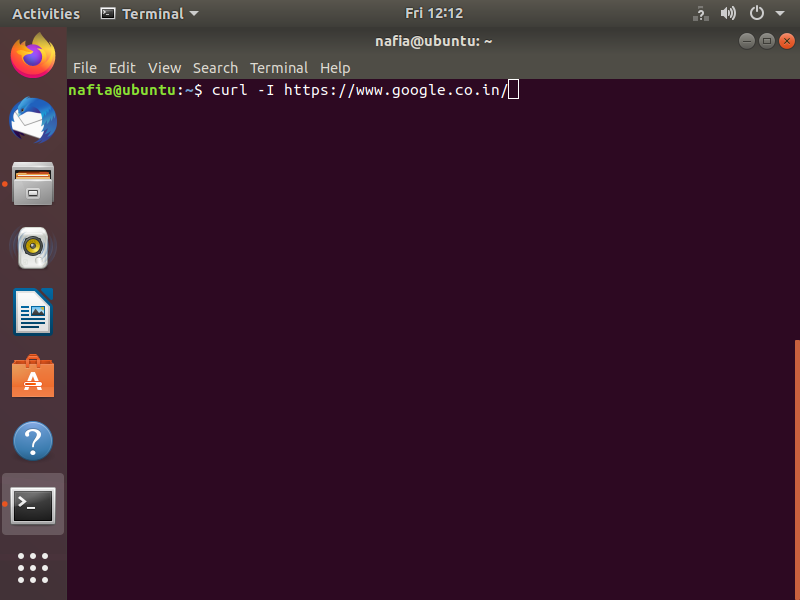
**35. traceroute**

This command is widely used by security professionals who leverage this command with other terminal commands for determining the route a network packet takes on its way from one machine to another. This is a compelling network command by using which you can safeguard your computer from a number of harmful intruders.



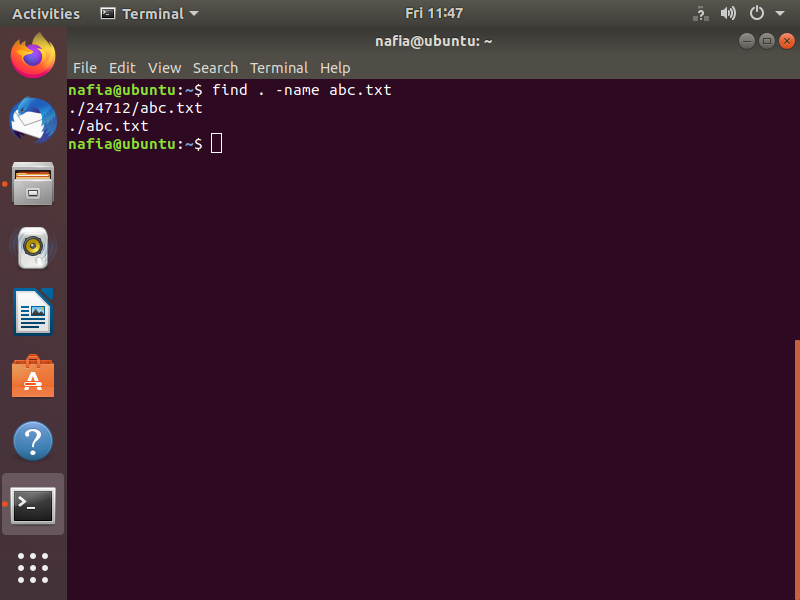
**36. cURL**

cURL is a very powerful network tool that makes transferring files over a network a child’s play for even new Linux system users. This is one of those Linux commands designed to work without user interaction and is typically employed in network-related shell scripts.



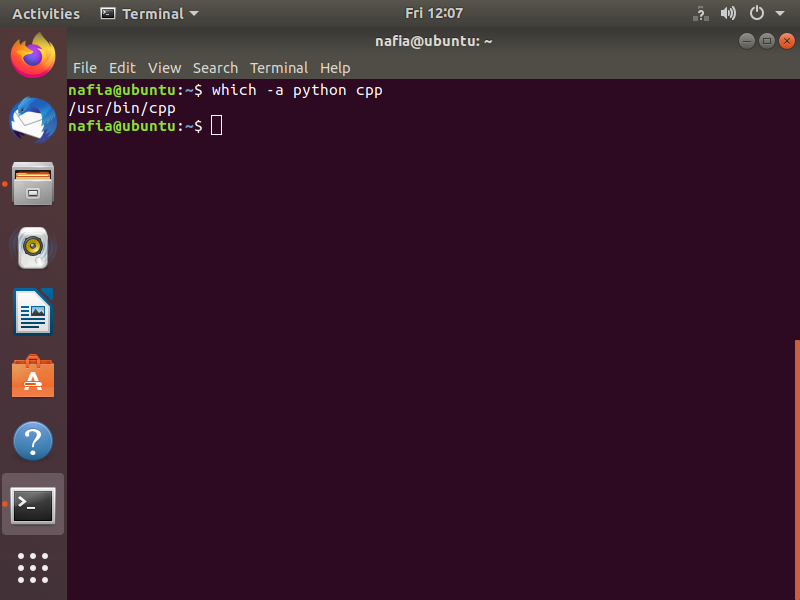
**37. find**

The [find command](https://www.ubuntupit.com/50-simple-and-useful-examples-of-find-command-in-linux/) is one of the most used Linux commands to search for files from the terminal. This compelling yet flexible terminal command allows users to search for files based on certain criteria such as file permissions, ownership, modification date, size, etc.



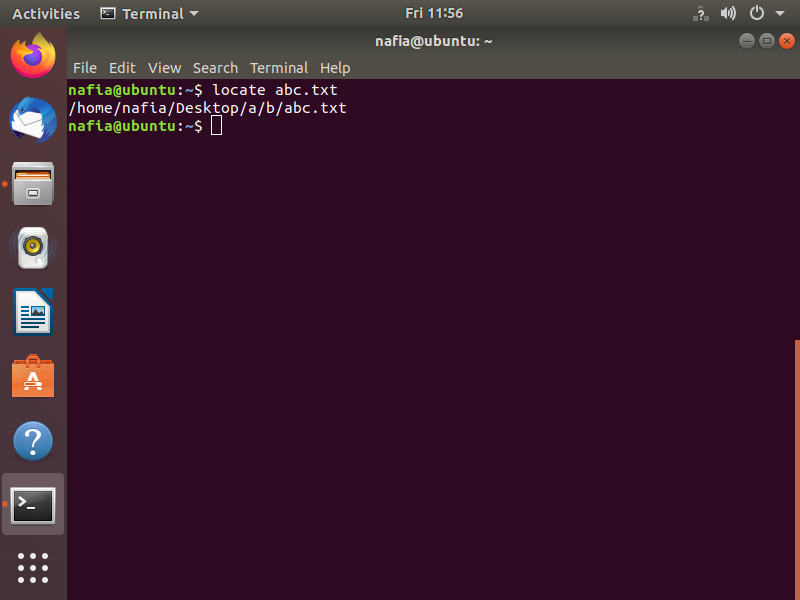
**38. which**

The which command is pretty useful if all you are trying to search are executable files. This handy little terminal command takes specific parameters and searches for binary files in the $PATH system environment variable based on them very effectively.



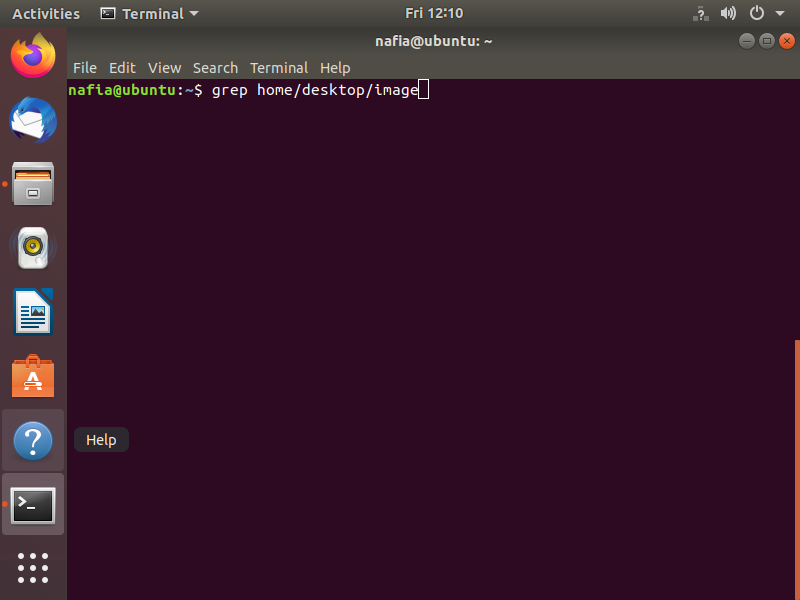
**39. locate**

The locate command is one of those Linux commands that are used for finding the location of a specific file. It is one of the most straightforward terminal commands that you can leverage when not sure about the location of a particular file on your Linux machine.



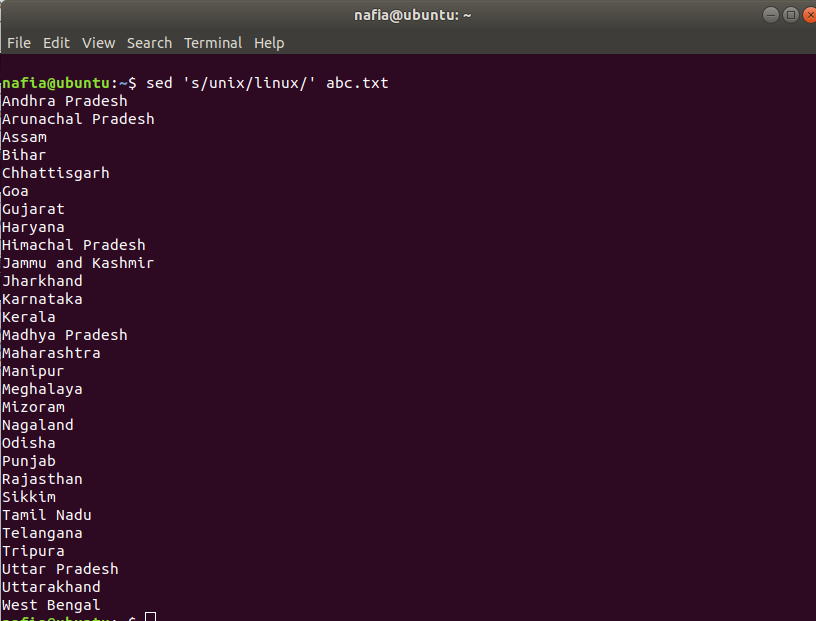
**40. grep**

[grep command](https://www.ubuntupit.com/practical-grep-command-for-linux-enthusiasts/) is among the most powerful regular expression terminal commands you can use when searching for patterns inside large volumes of text files. It will take the pattern you’re looking for as input and search the specified files for that particular pattern.



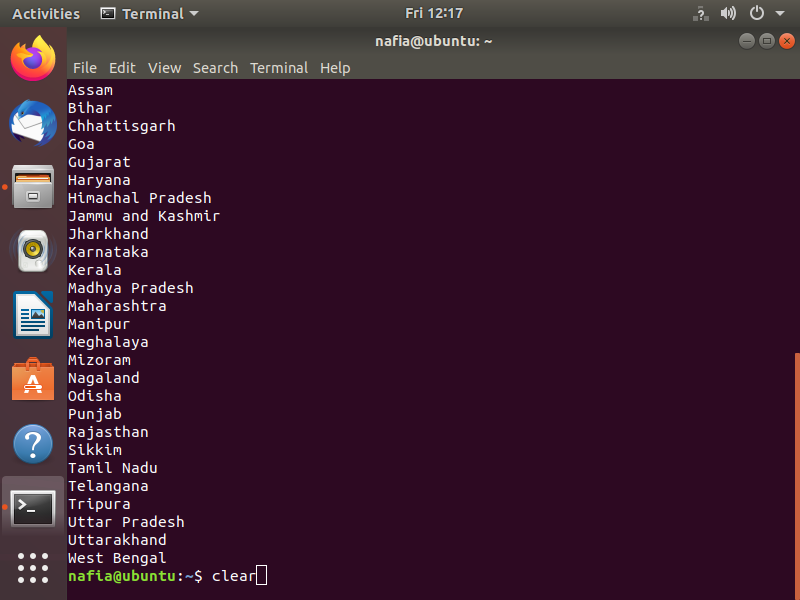
**41. sed**

This is one of the most widely used Linux commands to manipulate each line of a file or stream by replacing specified parts. It is used heavily by users that deal with large volumes of text data and need to change them on the go.



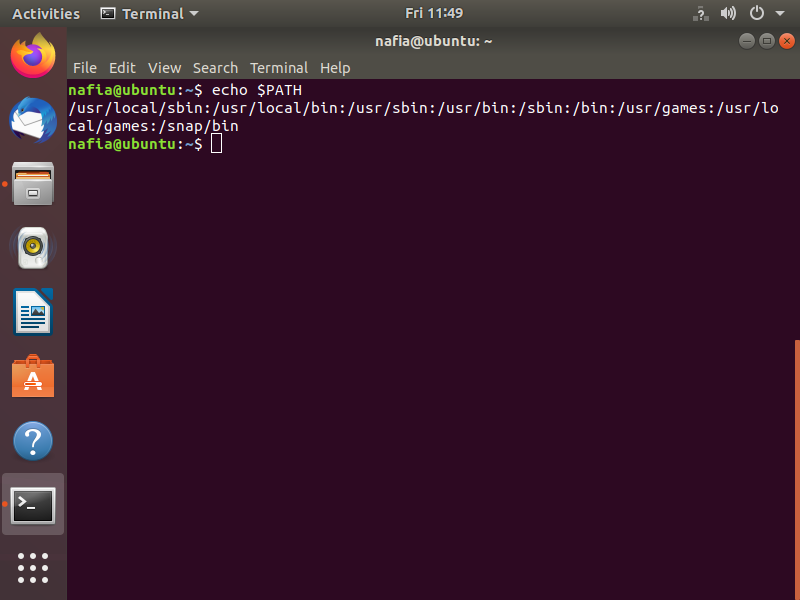
**42. clear**

The clear command is handy to clear out your existing terminal screen. Often you will find the need to wipe out the terminal screen after some earlier Linux commands leave your terminal screen with a garbled output.



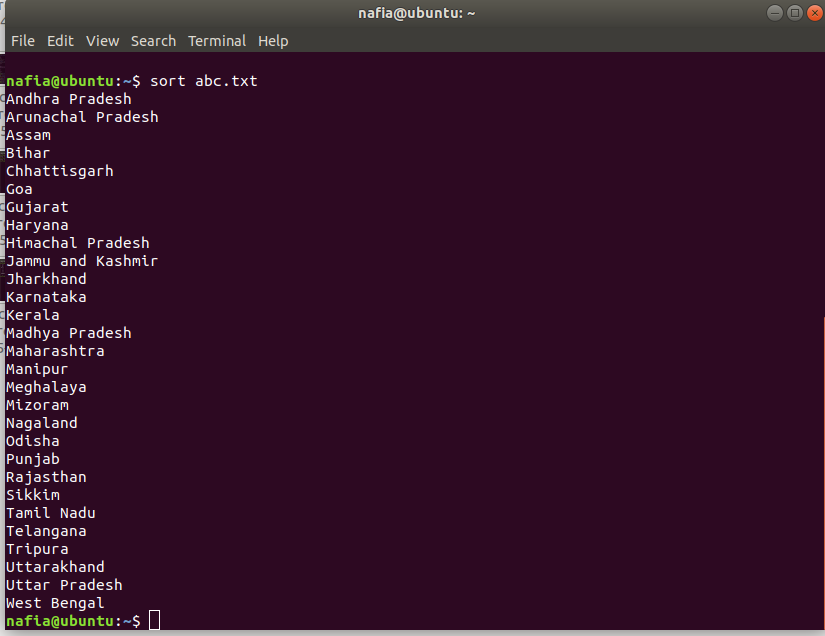
**43. echo**

The echo command is a very powerful command-line utility that lets you output a specific text to the [terminal console](https://www.ubuntupit.com/top-15-online-linux-terminal-emulators-and-bash-editors/). Type in echo followed by some texts within parentheses to find out for yourself. What’s more interesting for this command is that you can pipe the output to other terminal commands.



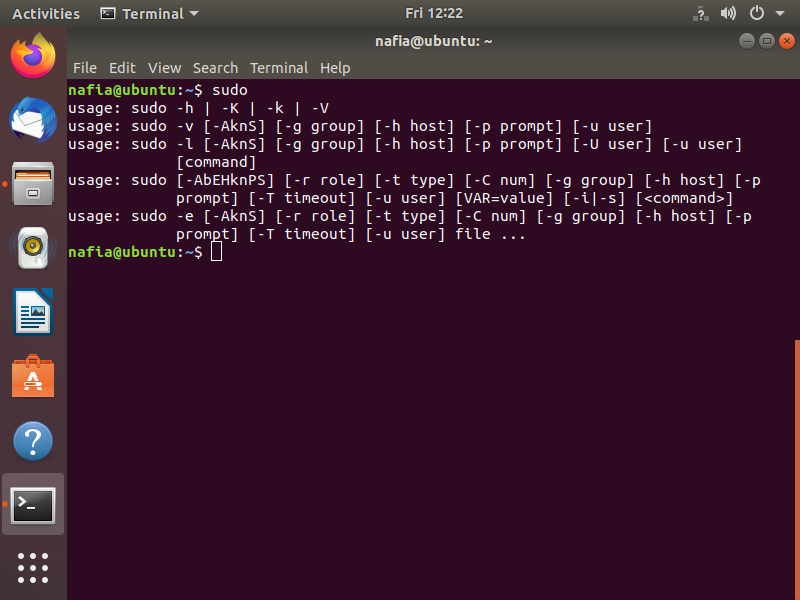
**44. sort**

The sort command is quite compelling at the things it does. Whenever you find the need to sort out a file in an alphabetical or reverse manner, utilize this command.



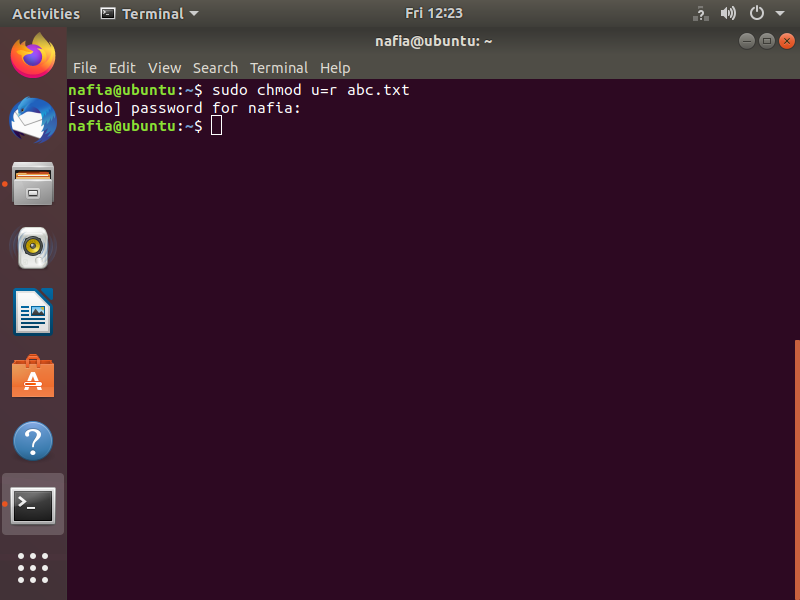
**45. sudo**

The sudo command is the holy grail of Linux commands. It lets non-privileged users access and modify files that require low-level permissions. Often you will use this command to access root from your regular user account.



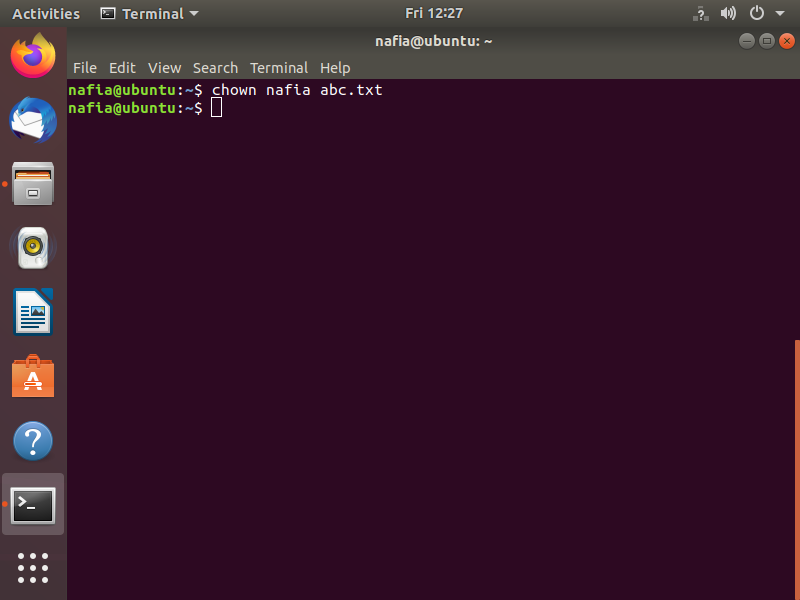
**46. chmod**

The chmod command is among the most powerful Linux commands you will use to change or modify the access permissions of system files or objects. This command can take a very diverse set of parameters from users and, based on those changes, the file permission.



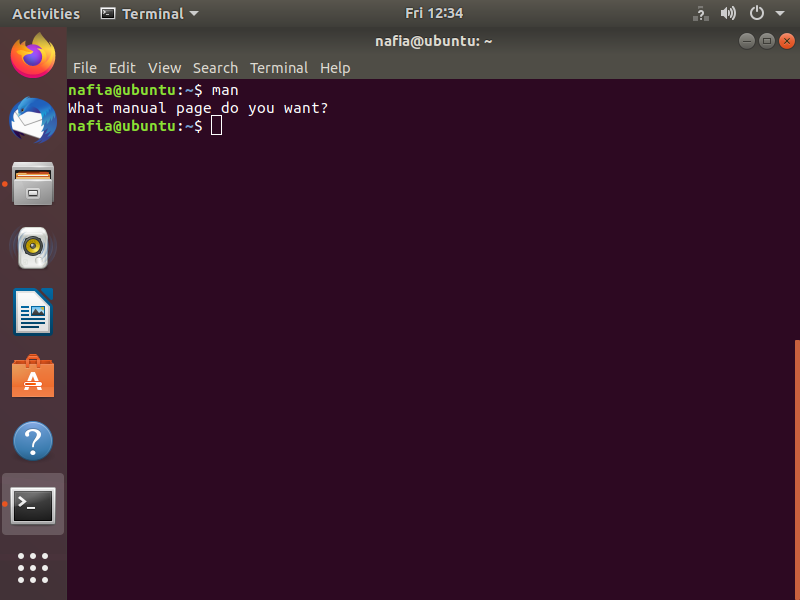
**47. chown**

The chown command is very much similar to the chmod command. But instead of changing access permissions, it enables users to change the ownership of a file or directory. Both the chmod and chown terminal commands require root privileges to run.



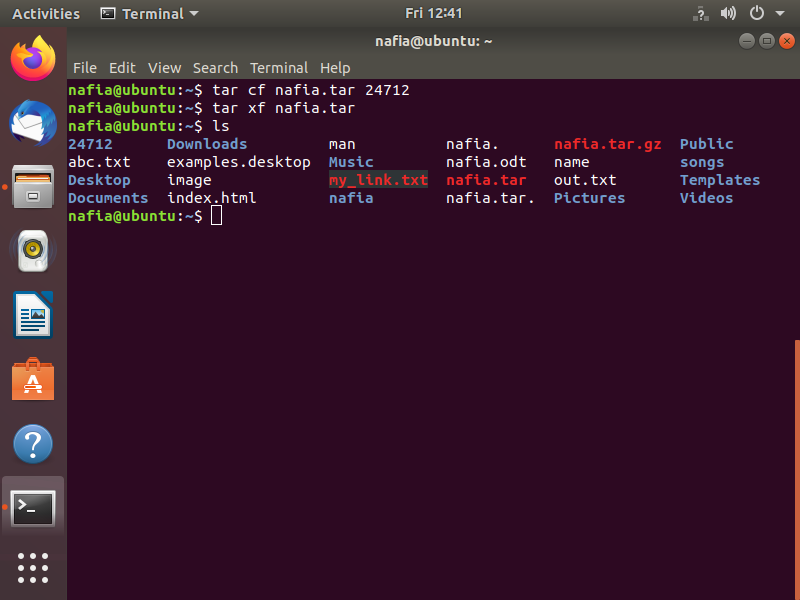
**48. man**

The man command stands for manual and is one of the most useful Linux commands you can get your hands on. This command, followed by the name of another command lists the manual or documentation page of that command. You will need to use this command often when determining how to use specific Terminal commands.



**49. tar**

The tar command is used for archiving files and extracting them. It is a very popular command for compressing files and can tackle such tasks very efficiently.



**50. whatis**

The whatis command traverses a set of databases with short descriptions provided by the user and prints out system commands that match them.

