**Linux Commands**

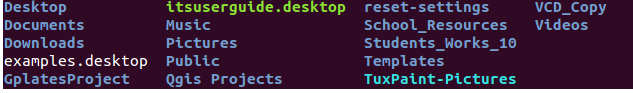
Linux is a family of open source Unix-like operating systems based on the Linux kernel, an operating system kernel first released on September 17, 1991, by Linus Torvalds. Linux is typically packaged in a Linux distribution. The Linux console is a system console internal to the Linux kernel (a system console is the device which receives all kernel messages and warnings and which allows logins in single user mode). The Linux console provides a way for the kernel and other processes to send text output to the user, and to receive text input from the user. The Linux kernel supports virtual consoles - consoles that are logically separate, but which access the same physical keyboard and display. The Linux console (and Linux virtual consoles) are implemented by the VT subsystem of the Linux kernel, and do not rely on any user space software. This is in contrast to a terminal emulator, which is a user space process that emulates a terminal, and is typically used in a graphical display environment. Every Linux system includes a command line of one sort or another.

**Some essential Linux commands:**

* **ls**
* **pwd**
* **cd**
* **mkdir & rmdir**
* **rm**
* **touch**
* **man & --help**
* **cp**
* **mv**
* **locate**
* **Echo**
* **Cat**
* **nano, vi, jed**
* **Sudo**
* **Clear**
* **Alias**
* **chmod**

The commands are described below –

**1. ls**

The ls command - the list command - functions in the Linux terminal to show all of the major directories filed under a given file system. 

**2. pwd**

pwd stands for Print Working Directory. It prints the path of the working directory, starting from the root. pwd is shell built-in command(pwd) or an actual binary(/bin/pwd). $PWD is an environment variable which stores the path of the current directory. This command has two flags. 

**3. cd**

The cd command - change directory - will allow the user to change between file directories. As the name command name suggest, you would use the cd command to circulate between two different directories.

For example you are in the home folder,and you want to go to the downloads folder,then you can type in **“cd Downloads”.**



**4. mkdir & rmdir**

The mkdir - make directory - command allows the user to make a new directory. For example, if you want to make a directory called “DIY”, then you can type **“mkdir DIY**”. Remember, as told before, if you want to create a directory named “DIY Hacking”, then you can type “mkdir **DIY\ Hacking**”. Use **rmdir** to delete a directory.

The rmdir - remove directory - command allows the user to remove an existing command using the Linux CLI.



**5. rm**

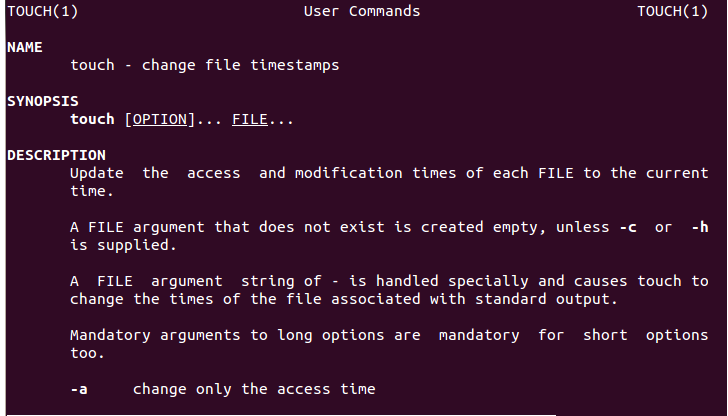
The rm command - remove - like the rmdir command is meant to remove files from your Linux OS. 

**6. touch**

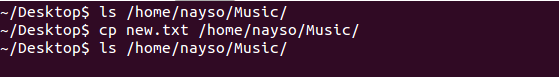
The touch command - a.k.a. the make file command - allows users to make files using the Linux CLI.  


**7. man & –help**

The man command - the manual command - is used to show the manual of the inputted command. Just like a film on the nature of film, the man command is the meta command of the Linux CLI. For example, “**man cd**” shows the manual pages of the **cd**command. Typing in the command name and the argument helps it show which ways the command can be used (e.g., **cd –help**).



**8. cp**

The **cp**command is used to copy files through the command line. It takes two arguments: The first is the location of the file to be copied, the second is where to copy.  


**9. mv**

The mv command - move - allows a user to move a file to another folder or directory. Just like dragging a file located on a PC desktop to a folder stored within the "Documents" folder, the mv command functions in the same manner. An example of the mv command is: **mv ~/Documents/Monir/os.pdf**

**10. Locate**

The locate - a.k.a. find - command is meant to find a file within the Linux OS. If you don't know the name of a certain file or you aren't sure where the file is saved and stored, the locate command comes in handy. Using the -i argument with the command helps to ignore the case (it doesn't matter if it is uppercase or lowercase). So, if you want a file that has the word “hello”, it gives the list of all the files in your Linux system containing the word "hello" when you type in “**locate -i hello**”. If you remember two words, you can separate them using an asterisk (\*). For example, to locate a file containing the words "hello" and "this", you can use the command “**locate -i \*hello\*this”.**

**11. Echo**

The echo command prints (echoes) a string of text to the terminal window. The command below will print the words “A string of text” on the terminal window.

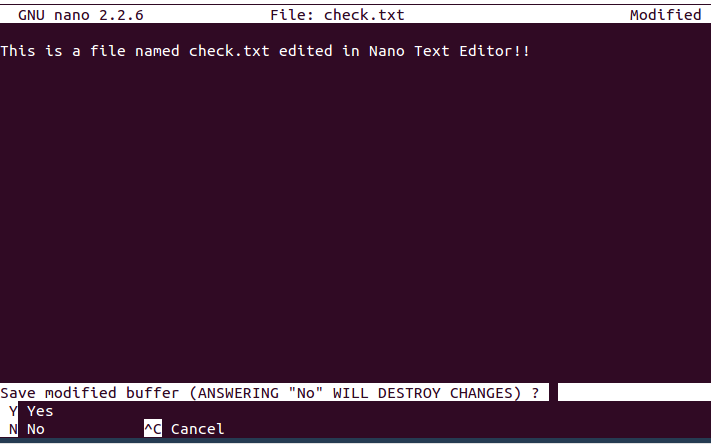
**echo A string of text**

**12. cat**

The cat command (short for “concatenate”) lists the contents of files to the terminal window. This is faster than opening the file in an editor, and there’s no chance you can accidentally alter the file. To read the contents of your .bash\_log\_out file, type the following command while the home directory is your current working directory, as it is by default: **cat .bash\_logout**

**13. nano, vi, jed**

**nano** and **vi** are already installed text editors in the Linux command line. The **nano** command is a good text editor that denotes keywords with color and can recognize most languages. And **vi** is simpler than **nano**. You can create a new file or modify a file using this editor. For example, if you need to make a new file named **"check.txt**", you can create it by using the command “**nano check.txt**”. You can save your files after editing by using the sequence Ctrl+X, then Y (or N for no). In my experience, using **nano**for HTML editing doesn't seem as good, because of its color, so I recommend **jed**text editor. We will come to installing packages soon.



**14. sudo**

The sudo command is required when performing actions that require root or superuser permissions, such as changing the password for another user: **sudo passwd 1234**

**15. clear**

The clear command does exactly what it says. When your Linux CLI gets all mucked up with various readouts and information, the clear command clears the screen and wipes the board clean. Using the clear command will take the user back to the start prompt of whatever directory you are currently operating in. To use the clear command simply type clear.

**16. alias**

The alias command lets you give your own name to a command or sequence of commands. You can then type your short name, and the shell will execute the command or sequence of commands for you.

**alias cls=clear**

This sets up an alias called cls . It will be another name for **clear**. When you type **cls**, it will clear the screen just as though you had typed **clear**.

**17. chmod**

The chmod command sets the file permissions flags on a file or folder. The flags define who can read, write to or execute the file.