

# LINUX

## Basic Linux Commands for Beginners

### What Is Linux?

Linux is an operating system's kernel. You might have heard of UNIX. Well, Linux is a UNIX clone. But it was actually created by Linus Torvalds from Scratch. Linux is free and open-source, that means that you can simply change anything in Linux and redistribute it in your own name! There are several Linux Distributions, commonly called “distros”.

At Royal Holloway, you will generally use Ubuntu Linux.

Generally you will use ‘NoMachine’ to connect to a Linux server; this provides you with a remote desktop, i.e. a windows-like interface to a Unix server.

### Linux Shell or “Terminal”

So, basically, a shell is a program that receives commands from the user and gives it to the OS to process, and it shows the output. Linux's shell is its main part. Its distros come in GUI (graphical user interface), but basically, Linux has a CLI (command line interface). In this tutorial, we are going to cover the basic commands that we use in the ‘bash’ shell of Linux.

To open a terminal in NoMachine, select Applications > Terminal Emulator (“Applications” is similar to the Windows start button). You will also find Terminal on your taskbar by default.

## Basic Commands

### Directories and files

1. **pwd** — When you first open the terminal, you are in the home directory of your user. To know which directory you are in, you can use the “pwd” command. It gives us the absolute path, which means the path that starts from the root. The root is the base of the Linux file system. It is denoted by a forward slash(/). The user directory is usually something like “/home/username”. The tilda character is a shortcut to the home directory.

Example: cim-ts-node-03\$ **pwd**

2. **ls** — Use the “ls” command to know what files are in the directory you are in. You can see all the hidden files by using the command “ls -a”.

Example: cim-ts-node-03\$ **ls -ltrF**

3. **cd** — Use the “cd” command to go to a directory. For example, if you are in the home folder, and you want to go to the downloads folder, then you can type in “cd Downloads”. Remember, this command is case sensitive, and you have to type in the name of the folder exactly as it is.

If you have spaces in your file or directory names the shell will see this as two separate commands or arguments, so put them in double quotes.

Example: cim-ts-node-03\$ **cd ~/Documents**

4. **mkdir & rmdir** — Use the **mkdir** command when you need to create a folder or a 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. But **rmdir** can only be used to delete an empty directory.

Example: `cim-ts-node-03$ mkdir foo`

5. **rm** - Use the **rm** command to delete files. Beware that “**rm \*.txt**” will delete all files.

Example: `cim-ts-node-03$ rm foo/*`

6. **touch** — The **touch** command will update a file’s access time to the current time; if the file doesn’t exist, it will be created.

Example: `cim-ts-node-03$ touch foo/README`

7. **man** and **--help** — To know more about a command and how to use it, use the **man** command. It shows the manual pages of the command. 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**).

Example: `cim-ts-node-03$ man rm`

8. **cp** — Use the **cp** command 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.

Example: `cim-ts-node-03$ cp f1 f2 f3 f4 d1`

9. **mv** — Use the **mv** command to move files through the command line. We can also use the **mv** command to rename a file. For example, if we want to rename the file “text” to “new”, we can use “**mv text new**”. It takes two arguments.

Example: `cim-ts-node-03$ mv foo/README .`

10. **locate** — The **locate** command is used to locate a file in a Linux system, just like the **search** command in Windows. This command is useful when you don't know where a file is saved or the actual name of the file. Using the **-i** argument with the command helps to ignore the case (it doesn't matter if it is uppercase or lowercase).

Example: `cim-ts-node-03$ locate eclipse stylesheet`

## Intermediate Commands

1. **echo** — The “**echo**” command helps us move some data, usually text into a file. For example, if you want to create a new text file or add to an already made text file, you just need to type in, “**echo hello world >> new.txt**”.

2. **cat** — Use the **cat** command to display the contents of a file. It is usually used to easily view programs.

3. **nano, vi** — **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).

Alternatively, NoMachine has a perfectly serviceable text editor on the task bar or the Application list.

4. **sudo** — A widely used command in the Linux command line, **sudo** stands for “**SuperUser Do**”. So, if you want any command to be done with administrative or root privileges, you can use the **sudo** command.

More accurately, you can’t, because you don’t have admin privileges on the college systems.

5. `df` — Use the `df` command to see the available disk space in each of the partitions in your system. You can just type in `df` in the command line and you can see each mounted partition and their used/available space in % and in KBs. If you want it shown in megabytes, you can use the command “`df -m`”.

6. `du` — Use `du` to know the disk usage of a file in your system. If you want to know the disk usage for a particular folder or file in Linux, you can type in the command `du` and the name of the folder or file. For example, if you want to know the disk space used by the documents folder in Linux, you can use the command “`du Documents`”. You can also use the command “`ls -lah`” to view the file sizes of all the files in a folder.

7. `tar` — Use `tar` to work with tarballs (or files compressed in a tarball archive) in the Linux command line. It has a long list of uses. It can be used to compress and uncompress different types of tar archives like `.tar`, `.tar.gz`, `.tar.bz2`, etc. It works on the basis of the arguments given to it. For example, “`tar -cvf`” for creating a `.tar` archive, “`tar -xvf`” to untar a tar archive, “`tar -tvf`” to list the contents of the archive, etc. Since it is a wide topic, here are some examples of tar commands.

8. `zip`, `unzip` — Use `zip` to compress files into a zip archive, and `unzip` to extract files from a zip archive.

9. `uname` — Use `uname` to show the information about the system your Linux distro is running. Using the command “`uname -a`” prints most of the information about the system. This prints the kernel release date, version, processor type, etc.

10. `chmod` — Use `chmod` to make a file executable and to change the permissions granted to it in Linux. Imagine you have a python code named `numbers.py` in your computer. You'll need to run “`python numbers.py`” every time you need to run it. Instead of that, when you make it executable, you'll just need to run “`python numbers.py`” in the terminal to run the file. To make a file executable, you can use the command “`chmod +x numbers.py`” in this case.

11. `hostname` — Use `hostname` to know your name in your host or network. Basically, it displays your hostname and IP address. Just typing “`hostname`” gives the output. Typing in “`hostname -I`” gives you your IP address in your network.

12. `ping` — Use `ping` to check your connection to a server. Wikipedia says, “Ping is a computer network administration software utility used to test the reachability of a host on an Internet Protocol (IP) network”. Simply, when you type in, for example, “`ping google.com`”, it checks if it can connect to the server and come back. It measures this round-trip time and gives you the details about it. The use of this command for simple users like us is to check your internet connection. If it pings the Google server (in this case), you can confirm that your internet connection is active!

## Additional notes

‘`clear`’ will clear the display (like ‘`cls`’).

TAB can be used for filename completion.

Ctrl+C will try to stop a running foreground process.

You can exit from the terminal by using the `exit` command or control-D.

Learn about NoMachine

Learn about how the shell expands wildcards

Learn about regular expressions

Definitely learn about pipelines, as in this not very useful example:

```
ls -l | grep -v '^d' | sort +4n | cat -E > foo/files
```