Here is a list of basic Linux commands:

**1. pwd command**

Use the **pwd** command to find out the path of the current working directory (folder) you’re in. An example of an absolute path is **/home/username**.

**2. cd command**

To navigate through the Linux files and directories, use the **cd** command. It requires either the full path or the name of the directory, depending on the current working directory that you’re in.

Let’s say you’re in **/home/username/Documents** and you want to go to **Photos**, a subdirectory of **Documents**. To do so, simply type the following command: **cd** **Photos**.

Another scenario is if you want to switch to a completely new directory, for example,**/home/username/Movies**. In this case, you have to type **cd** followed by the directory’s absolute path: **cd /home/username/Movies**.

There are some shortcuts to help you navigate quickly:

1. **cd ..** (with two dots) to move one directory up
2. **cd** to go straight to the home folder
3. **cd-** (with a hyphen) to move to your previous directory

On a side note, Linux’s shell is case sensitive. So, you have to type the name’s directory exactly as it is.

**6. ls command**

The **ls** command is used to view the contents of a directory. By default, this command will display the contents of your current working directory.

If you want to see the content of other directories, type **ls** and then the directory’s path. For example, enter **ls** **/home/username/Documents** to view the content of **Documents**.

There are variations you can use with the **ls** command:

1. **ls -R** will list all the files in the sub-directories as well
2. **ls -a** will show the hidden files
3. **ls -al** will list the files and directories with detailed information like the permissions, size, owner, etc.

**10. cat command**

**cat** (short for concatenate) is one of the most frequently used commands in Linux. It is used to list the contents of a file on the standard output (sdout). To run this command, type **cat** followed by the file’s name and its extension. For instance: **cat file.txt**.

Here are other ways to use the **cat** command:

1. **cat > filename** creates a new file
2. **cat filename1 filename2>filename3** joins two files (1 and 2) and stores the output of them in a new file (3)
3. to convert a file to upper or lower case use, **cat filename | tr a-z A-Z >output.txt**

**14. cp command**

Use the **cp** command to copy files from the current directory to a different directory. For instance, the command **cp scenery.jpg** **/home/username/Pictures** would create a copy of **scenery.jpg** (from your current directory) into the **Pictures** directory.

**15. mv command**

The primary use of the **mv** command is to move files, although it can also be used to rename files.

The arguments in mv are similar to the cp command. You need to type **mv**, the file’s name, and the destination’s directory. For example: **mv file.txt /home/username/Documents**.

To rename files, the Linux command is **mv oldname.ext newname.ext**

**16. mkdir command**

Use **mkdir** command to make a new directory — if you type **mkdir Music** it will create a directory called **Music**.

**17. rmdir command**

If you need to delete a directory, use the **rmdir** command. However, rmdir only allows you to delete empty directories.

**18. rm command**

The **rm** command is used to delete directories and the contents within them. If you only want to delete the directory — as an alternative to rmdir — use **rm -r**.

**19. touch command**

The **touch** command allows you to create a blank new file through the Linux command line. As an example, enter touch **/home/username/Documents/Web.html** to create an HTML file entitled **Web** under the **Documents** directory.

**20. find command**

Similar to the **locate** command, using **find** also searches for files and directories. The difference is, you use the **find** command to locate files within a given directory.

As an example, find **/home/ -name notes.txt** command will search for a file called **notes.txt** within the home directory and its subdirectories.

### 21. chmod command

**chmod** is another Linux command, used to change the read, write, and execute permissions of files and directories

### 22. kill command

If you have an unresponsive program, you can terminate it manually by using the **kill** command

### 23. uname command

The **uname** command, short for Unix Name, will print detailed information about your Linux system like the machine name, operating system, kernel, and so on.

### 24. man command

Confused about the function of certain Linux commands? Don’t worry, you can easily learn how to use them right from Linux’s shell by using the **man** command. For instance, entering **man tail** will show the manual instruction of the tail command.

### 25. jobs command

**jobs** command will display all current jobs along with their statuses. A job is basically a process that is started by the shell.

### 26. diff command

Short for difference, the **diff** command compares the contents of two files line by line. After analyzing the files, it will output the lines that do not match. Programmers often use this command when they need to make program alterations instead of rewriting the entire source code.The simplest form of this command is **diff file1.ext file2.ext**

### 27. head command

The **head** command is used to view the first lines of any text file. By default, it will show the first ten lines, but you can change this number to your liking. For example, if you only want to show the first five lines, type **head -n 5 filename.ext**.

### 28. sudo command

Short for “**SuperUser Do**”, this command enables you to perform tasks that require administrative or root permissions. However, it is not advisable to use this command for daily use because it might be easy for an error to occur if you did something wrong.

### 29. echo command

This command is used to move some data into a file. For example, if you want to add the text, “Hello, my name is John” into a file called name.txt, you would type **echo Hello, my name is John** **>> name.txt**

### 30. zip, unzip command

Use the **zip** command to compress your files into a zip archive, and use the **unzip** command to extract the zipped files from a zip archive.