**Linux Commands**

**ls** : The **ls** command lists the content of a folder, including files and directories.

Syntax : ls [options] [directory\_or\_path]

**pwd** : To check the full path of wer current working directory, use the **pwd** command.

Syntax : pwd [options]

The **pwd** command has only two options. The**-L** option prints environment variable content, like shortcuts, instead of the actual path of wer current location. Meanwhile, **-P** outputs the exact location.

**cd** : Use **cd** to navigate between directories in wer Linux VPS. It doesn’t have any option, and the syntax is simple:

cd [path\_or\_directory]

Depending on wer location, we might only need to specify the parent directory. For example, omit **path**from **path/to/directory** if we are already inside one. The **cd** command has several shortcuts:

* **cd** – returns to the current user’s home directory.
* **cd ..** – moves a directory up.
* **cd –** – goes back to the previous directory.

**mkdir :** The mkdir command lets we create one or multiple directories. The syntax looks like this:

mkdir [options] directory\_name1 directory\_name2

To create a folder in another location, specify the full path. Otherwise, this command will make the new item in wer current working directory. By default, **mkdir** allows the current user to read, write, and execute files in the new folder. We can set custom privileges during the creation by adding the **-m** option.

**rmdir** : Run **rmdir**to delete empty directories in wer Linux system. The command syntax looks like this:

rmdir [options] directory\_name

The **rmdir** command won’t work if the directory contains subfolders. To force the deletion, add the –**p** option.

**rm** : The **rm** command deletes files from a directory. We must have the write permission for the folder or use **sudo**.

Syntax : rm [options] file1 file2

We can add the **-r** option to remove a folder and its contents, including subdirectories. Use the **-i** flag to display a confirmation message before the removal or **-f** to deactivate it completely.

**cp** : Use the **cp** command to copy files from wer current directory to another folder. The syntax looks like this:

cp file1 file2 [target\_path]

We can also use **cp** to duplicate the content of one file to another using this syntax. If the target is in another location, specify the full path like so:

cp source\_file /path/to/target\_file

Additionally, **cp** lets us duplicate a directory and its content to another folder using the **-R** option:

cp -R /path/to/folder /target/path/to/folder\_copy

**mv :** The main usage of the **mv** command is to move a file or folder to another location. Here’s the syntax:

mv file\_or\_directory [target\_directory]

For example, we will move **file1.txt** from another location to the **/new/file/directory** path using this command:

mv /original/path/file1.txt the/target/path

We can also use the **mv** command to [**rename files in wer Linux system**](https://www.hostinger.in/tutorials/how-to-rename-files-in-linux/). Here’s an example:

mv old\_name.txt new\_name.txt

If we specify the full path, we can simultaneously rename files and move them to a new location like this example:

mv old/location/of/old\_name.txt new/path/for/new\_name.txt

**touch :** Run the **touch** command to create a new empty file in a specific directory. The syntax is as follows:

touch [options] [path\_and\_file\_name]

If you omit the path, the **touch** command will create a new file in your current working directory. Here’s an example:

touch file.txt

**file**: The **file** command checks a file type, such as TXT, PDF, or other. The syntax is as follows:

file [file\_name]

If you use this command on a symbolic link, it will output the actual file connected to the shortcut. You can add the**-k** option to print more detailed information about the item.

**zip** : The **zip** command compresses one or multiple files into a **ZIP** archive, reducing their size. Here’s the syntax:

zip [options] zip\_file\_name file1 file2

**unzip** : To extract a compressed file into your current working directory, use the **unzip** command like so:

unzip [options] zip\_file\_name

**tar** : The**tar**command bundles multiple files or directories into an archive without compression. The syntax looks as follows:

tar [options] tar\_file\_name file1 file2

To create a new **TAR** file, you must add the **-c**option. Then, use the**-f**flag to specify the archive’s name.

If you want to enable compression, add a specific option based on your preferred method. For example, the following will bundle **file1.txt** and **file2.txt**with the **gzip** compression:

tar -cfz archive.tar.gz fle1.txt file2.txt

Remember that the archive’s file format will differ depending on the compression method. Regardless of the extension, you can unpack a **TAR** file using this syntax:

tar [options] tar\_file\_name

**nano**, **vi**, and **jed** commands let you edit files. They have the same syntax, except at the beginning, where you specify the name of the tool:

nano/vi/jed file\_name

If the target file doesn’t exist, these commands will create a new one. Since your system might not have these text processing utilities pre-installed, configure them using your package manager.

The **concatenate**or **cat**command has various usages. The most basic one is printing the content of a file. Here’s the syntax:

cat file\_name

To print the content in reverse order, use**tac** instead. If you add the standard output operator symbol (**>**), the **cat** command will create a new file. For example, the following will make **file.txt**:

cat > file.txt

You can also use cat with the operator to combine the content of multiple files into a new item. In this command,**file1.txt**and **file2.txt** will merge into **target.txt**:

cat file1.txt file2.txt > target.txt

**Global regular expression print** or **grep** lets you search specific lines from a file using keywords. It is useful for filtering large data like logs. The syntax looks as follows:

grep [options] keyword [file]

You can also filter data from another utility by piping it to the **grep** command. For example, the following searches **file.txt** from the **ls** command’s output:

ls | grep "file.txt"