



# Bioinformatics Course

YouTube link: <https://www.youtube.com/@bioinfo byte>

## The Linux Command – Usage, Syntax, and Example

### 1. man command:

Use man <command> to get the manual of the command

```
$man ls          # get the manual of the command
```

### 2. echo command:

echo command prints its argument to the screen

```
$echo "string"    # print string to the screen
```

### 3. cd command:

cd means change directory)

```
$cd folder_name    # you are saying the computer to move to folder_name folder
$cd ..             # cd .. is used to go back 1 directory
$cd ../..          # go back 2 directory
```

### 4. ls command:

list all the files that the folder contains using the ls command

```
$ls
$ls folder_name    # list all the files (not including hidden files) that the folder contains
$ls -al folder_name # list all the files including hidden files that the folder contains
$ls *.txt           # list all the files with a pattern. pattern is looking for files ending with
the character .txt
$ls file*           # To fetch files that have the prefix "file", you can use the wildcard
pattern file*
$ls *file*          # If you're trying to list files with names containing the word "file", you
might want to use a different wildcard pattern
```

## 5. pwd command

<b>\$pwd</b>	# print the current folder path
--------------	---------------------------------

## 6. mkdir command:

<b>\$mkdir</b> folder_name	# creating a folder with folder_name
<b>\$mkdir</b> folder1 folder2	# create multiple folders with one command
<b>\$mkdir -p</b> folder3/folder4	# create multiple nested folders by adding the -p option

## 7. rmdir and rm command:

<b>\$rmdir</b> folder_name	# delete a empty folder using rmdir
<b>\$rmdir</b> folder1 folder2	# delete multiple empty folders with one command
<b>\$rm -r</b> folder_name	# remove a directory and its contents recursively, you can use the rm command with the -r

## 8. mv command:

<b>\$mv</b> file1.txt file_new1.txt	# rename files with mv command
<b>\$mv</b> folder_name new_folder_name	# rename folder with mv command
<b>\$mv</b> file1.txt file2.txt folder_name	# specify a list of files and they will all be moved in the folder path identified by the last parameter
<b>\$mv *.txt</b> folder_name	# move multiple files to a different directory using a pattern search

## 9. cp command:

<b>\$cp -r</b> folder_1 folder_2	# copy the whole folder contents of folder_1 in folder_2
<b>\$cp</b> file1.txt folder_1	# copy a file in a particular folder
<b>\$cp *.txt</b> folder_1	# copy multiple files with a pattern search

## 10. open command:

<b>\$open</b> filename	# open command, which is available on macOS systems to open files
------------------------	---

For Linux systems use Vim, cat, and less commands to open the file

<b>\$vim</b> filename	# open the file
-----------------------	-----------------

### 11. touch command:

```
$touch file_name           # create an empty file using the touch command
```

### 12. cat command:

```
$cat file                  # prints the file's content to the standard output  
$cat file1 file2           # Print the content of multiple files  
$cat file1 file2 > file3    # using the output redirection operator > you can concatenate the  
content of multiple files into a new file
```

### 13. nl command:

```
$nl file_name              # display the contents of a file with line numbers
```

### 14. find command:

```
$find . -name '*.txt'      # Find all the files under the current tree that have the .txt  
extension and print the relative path of each file that matches  
$find . -name file1.txt     # -name is case sensitive  
$find . -name fiLE1.txt     # use -iname to perform a case-insensitive search  
$find . -type d -name "folder_name" # Find directories under the current tree matching the  
name "folder_name"  
$find . -type d -name folder1 -or -name folder2 # Find directories under the current tree  
matching the name "folder1" or 'folder2'  
$find . -type f -size +100k -size -1M # Search files bigger than 100KB but smaller than 1MB.  
This is useful when you want to see which file is taking more spaces
```

### 15. alias command:

alias is used to create a new command, for example, I like to call it bioinfo\_journey, which is an alias to the cd path of

```
/drives/c/ashish_personal/bioinfo_byte/Build_course/bioinfo_course/Linux_learning
```

Once you do, you can call bioinfo\_journey just like it was a regular linux command

```
$alias bioinfo_journey='cd  
/drives/c/ashish_personal/bioinfo_byte/Build_course/bioinfo_course/Linux_learning'  
$alias # calling alias without any option will list the aliases defined
```

Note: The alias will work until the terminal session is closed.

To make it permanent, you need to add it to the shell configuration (~/.bashrc)

#### 16. tail command:

```
$tail -n 10 filename      # print the last 10 lines in a file  
$tail -n +10 filename    # print the whole file content starting from a specific line using +  
before the line number
```

#### 17. head command:

```
$head -n 10 filename      # print the content of starting 10 lines in a file
```

#### 18. wc command:

```
$wc filename             # provide 3 information, first column returned is the number of lines. The  
second is the number of words. The third is the number of bytes  
$wc -l filename         # just the lines  
$wc -w filename         # just the words  
$wc -c filename         # just the bytes
```

#### 19. clear command:

```
$clear # Type clear to clear all the previous commands that were run in the current terminal
```

#### 20. diff command:

Suppose you have 2 files, that contain almost the same information, but you can't find the difference between the two. diff will process the files and will tell you what's the difference

```
$diff file1.txt file2.txt # display differences between versions  
$diff -u file1.txt file2.txt # display differences between versions  
$diff -rq file1 file2      # In case you're interested in which files differ, rather than the content,  
use the rq options
```

-----\*\*\*\*\*-----