**Files and Directory management**

**Files**

Different files are available such as flat files, compressed files, hidden files and system files.

**File Listing Commands:**

ls : list directory contents. Short list.

Ex:

$ls

ls -lart

ls <options> <directory>

if you don’t mention the directory, ls command will take/list the current directory contents.

**Options:**

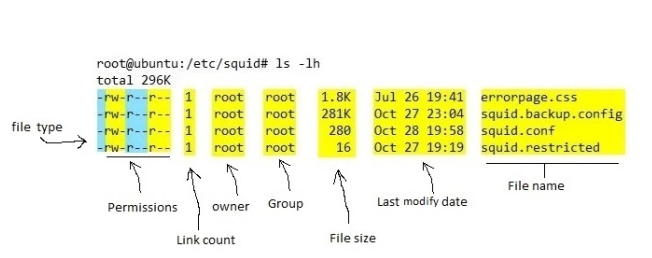
-lart:

l : use a long listing format

a : show all files including hidden and . files/directories

r : reverse the order while sorting

t : sort by modification time. Default is descending



**Total**: show total size of the folder.

**File type**: First field in the output is file type. If the there is a – it means it is a plain file. If there is d it means it is a directory, c represents a character device , b represents a block device.

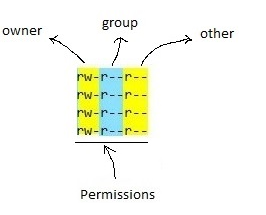
**Owner:** This field provide info about the creator of the file.

**Group**: This filed provide info about who all can access the file.

**File size:** This field provide info about the file size. By default the ls -l command give output in bytes.

Other fields are self explanatory.

Now permission can be divided into three parts: Owner, Group, Other



r: read permission  
w: write permission  
x: execute permission

Octal values are used to represent permissions.

4 -> read permission  
2 -> write permission  
1 -> execute permission

**Example**: If a file have 764 permission it means owner can do all operation on file, group can do read and write operation on file and other can read only the file

**How to Use the ls Command**

The syntax for the ls command is as follows:

ls [OPTIONS] [FILES]

To list files in a specific directory, pass the path to the directory as an argument to the ls command. For example, to list the contents of the /etc directory you would type type:

$ls /etc

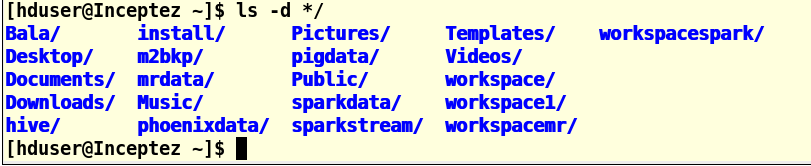
You can also pass multiple directories and files to the ls command separated by space:

$ls /etc /var /home/hduser

**List only the directories:**

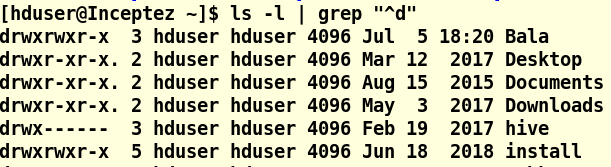
**Option 1:**

$ls -d \*/



**Option 2:**

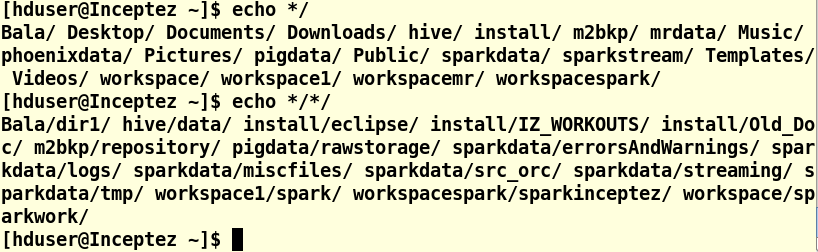
$ls -l | grep “^d”



**Option 3:**

$echo \*/

$echo \*/\*/



**VI Text Editor with Commands: Linux/Unix Tutorial**

The VI editor is the most popular and classic text editor in the Linux family. Below, are some reasons which make it a widely used editor –

1) It is available in almost all Linux Distributions

2) It works the same across different platforms and Distributions

3) It is user-friendly. Hence, millions of Linux users love it and use it for their editing needs

Nowadays, there are advanced versions of the vi editor available, and the most popular one is **VIM**which is **V**i **Im**proved. Some of the other ones are Elvis, Nvi, Nano, and Vile. It is wise to learn vi because it is feature-rich and offers endless possibilities to edit a file.

To work on VI editor, you need to understand **its operation modes**. They can be divided into two main parts.

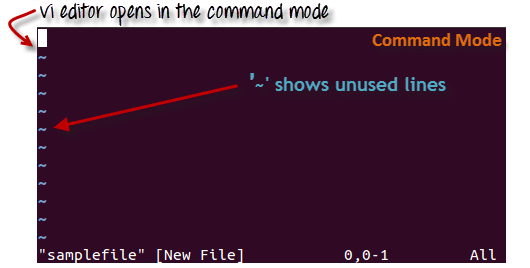
**Starting the vi editor**

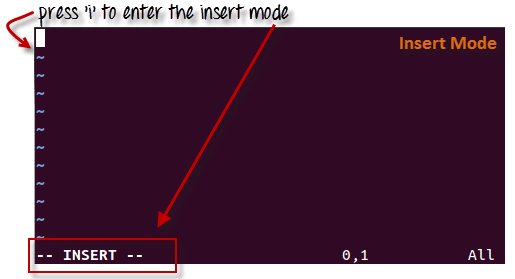
To launch the VI Editor -Open the Terminal (CLI) and type

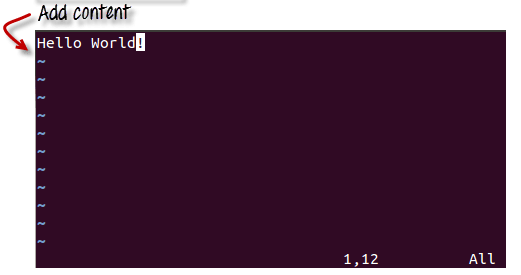
$vi <filename\_NEW> or <filename\_EXISTING>

And if you specify an existing file, then the editor would open it for you to edit. Else, you can create a new file.

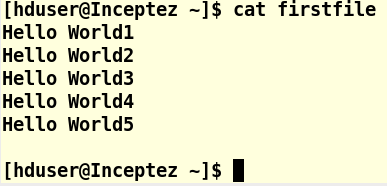
Creating a new file:











**VI Editing commands**

* i - Insert at cursor (goes into insert mode)
* a - Write after cursor (goes into insert mode)
* A - Write at the end of line (goes into insert mode)
* ESC - Terminate insert mode
* u - Undo last change
* U - Undo all changes to the entire line
* o - Open a new line (goes into insert mode)
* dd - Delete line
* 3dd - Delete 3 lines.
* D - Delete contents of line after the cursor
* C - Delete contents of a line after the cursor and insert new text. Press ESC key to end insertion.
* dw - Delete word
* 4dw - Delete 4 words
* cw - Change word
* x - Delete character at the cursor
* r - Replace character
* R - Overwrite characters from cursor onward
* s - Substitute one character under cursor continue to insert
* S - Substitute entire line and begin to insert at the beginning of the line
* ~ - Change case of individual character

**Moving within a file**

* k - Move cursor up
* j - Move cursor down
* h - Move cursor left
* l - Move cursor right

You need to be in the command mode to move within a file. The default keys for navigation are mentioned below else; You can **also use the arrow keys on the keyboard**.

**Saving and Closing the file**

* Shift+zz - Save the file and quit
* :w - Save the file but keep it open
* :q - Quit without saving
* :wq - Save the file and quit

**echo command**

**echo command** in linux is used to display line of text/string that are passed as an argument

**Create/Replace a file with few content**

**echo 'welcome to unix' > filename1**

**Append content to a file**

**echo "this is the second line" >> filename1**

**Create Empty file**

**touch filename2**

**> filename3**

**Display content of a file**

**Display whole content**

$cat filename

**Display incremental content**

$more filename

**Display first 10 lines**

$head filename

**Display last 10 lines**

$tail filename

**wc command**

**wc** stands for word count. As the name implies, it is mainly used for counting purpose. It is used to find out number of lines, word count, byte and characters count in the files specified in the file arguments.

**wc [OPTION]... [FILE]...**

**$ wc state.txt**

**5 7 63 state.txt**

**OR**

**$ wc capital.txt**

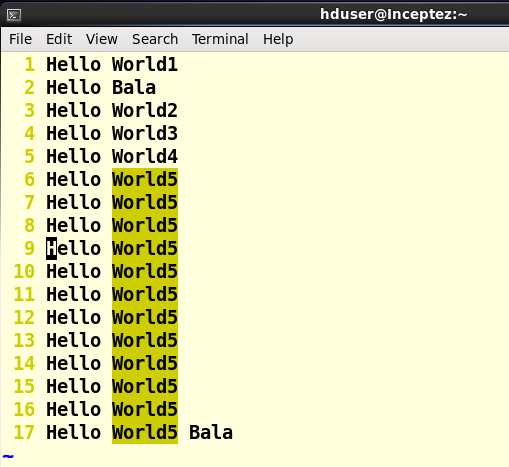
**5 5 45 capital.txt**

**$ wc state.txt capital.txt**

**5 7 63 state.txt**

**5 5 45 capital.txt**

**10 12 108 total**



$wc -l firstfile

17 firstfile

17 – number of lines

firstfile - file name

$wc firstfile

17 35 224 firstfile

17 – number of lines

35 – number of words

224 – count of bytes

firstfile - file name

wc --version

It gives the version number

**File operation (copy, move, rename, delete)**

cp filename file2

mv file2 file3

rm file3

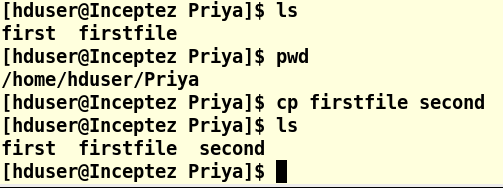
cp stands for copy. This command is used to copy files or group of files or directory

The general syntax for the cp command is as follows:

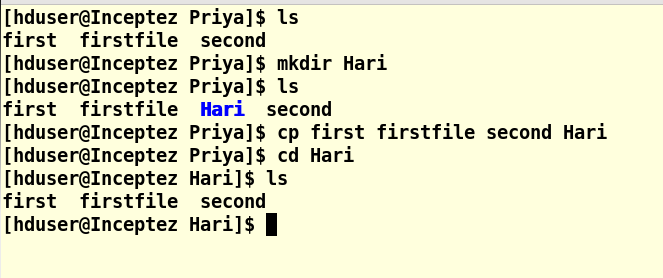
cp [OPTIONS] SOURCE... DESTINATION

The SOURCE can contain one or more files or directories as arguments, and the DESTINATION argument can be a single file or directory.

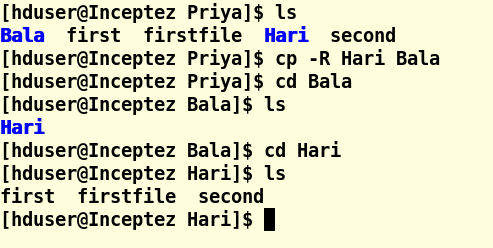
* When the SOURCE and DESTINATION arguments are both files, the cp command copies the first file to the second one. If the file doesn’t exists the command creates it.



* When the SOURCE has multiple files or directories as arguments, the DESTINATION argument must be a directory. In this situation, the SOURCE files and directories are moved to the DESTINATION directory.



* When the SOURCE and DESTINATION arguments are both directories, the cp command copies the first directory into the second one.

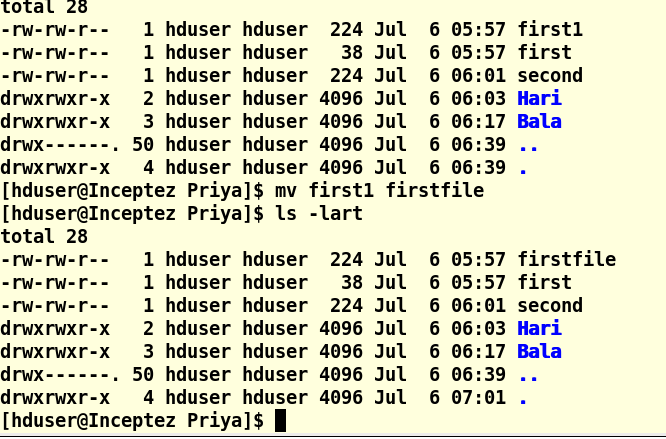


**mv Command**

The mv command (short from move) is used to rename and move and files and directories from one location to another. The syntax for the mv command is as follows:

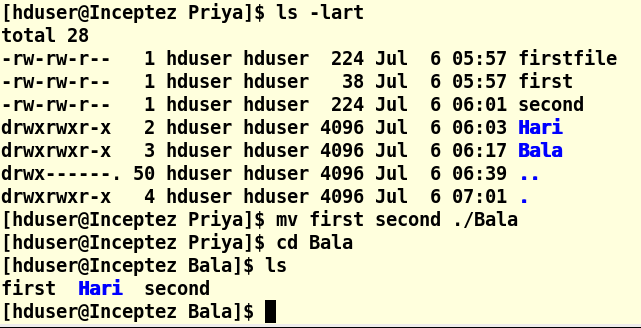
mv [OPTIONS] SOURCE DESTINATION

* If you specify a single file as SOURCE, and a single file as DESTINATION target then you’re [renaming the file](https://linuxize.com/post/how-to-rename-files-in-linux/).

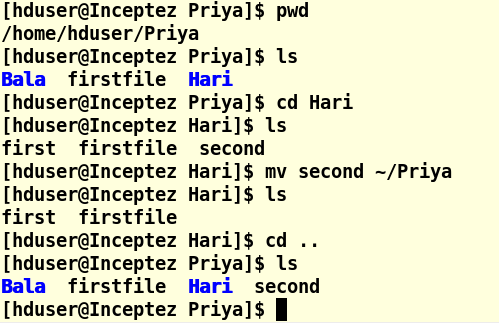


The file “first1” has been renamed/moved in the same location.

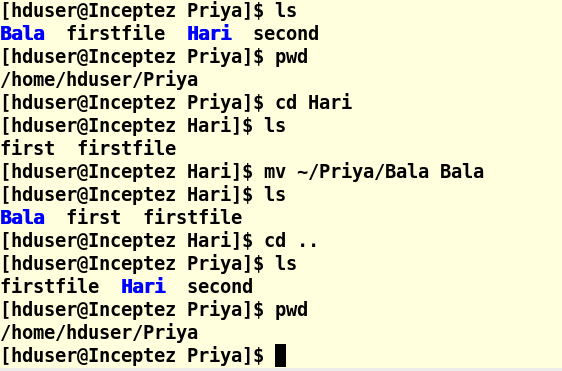
* When multiple files or directories are given as a SOURCE, the DESTINATION must be a directory. In this case, the SOURCE files are moved to the target directory.



* If you specify a single file as SOURCE, and the DESTINATION target is an existing directory, then the file is moved to the specified directory.

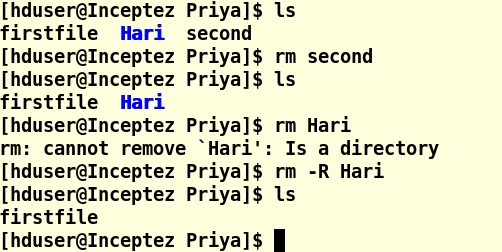


* When the SOURCE is a directory and DESTINATION doesn’t exist, SOURCE will be renamed to DESTINATION. Otherwise if DESTINATION exist, it be moved inside the DESTINATION directory.



**rm Command**

**rm** stands for remove here. **rm command** is used to remove objects such as files, directories, symbolic links and so on from the file system like UNIX.



**Important:**

* rmdir command – Delete directory only if it empty
* rm command – Delete directory and all files even if it is NOT empty

