Linux Commands (Basic)

1. pwd : to know the current working directory ['print working directory'].

```
beryl@beryl-ThinkPad-L412:~/practice$ pwd
/home/beryl/practice
```

----

2. cd : to change the working directory ['change directory']

    2.1 cd : - when used without any options change the directory to root directory

```
beryl@beryl-ThinkPad-L412:~/practice$ cd
beryl@beryl-ThinkPad-L412:~$ pwd
/home/beryl
```

    2.2 cd direcory_name : - changes the directory to diretory mentioned inplace of 'directory_name'

```
beryl@beryl-ThinkPad-L412:~$ cd practice
beryl@beryl-ThinkPad-L412:~/practice$
```

    2.3 cd .. : - [..] double dots are used to refer the parent directory of current directory

```
beryl@beryl-ThinkPad-L412:~/practice$ cd ..
beryl@beryl-ThinkPad-L412:~$
```

----

3. ls : to list out the current directory items

```
beryl@beryl-ThinkPad-L412:~$ ls
Desktop     'Git and Github'   Pictures    snap        training
Documents   Music              practice    Templates   Videos
Downloads   output.txt         Public      Tishu
```

    ls -[options] :

    3.1   -a, --all : - to also show hidden files

```
beryl@beryl-ThinkPad-L412:~$ ls -a
.                Documents          .mozilla      snap
..               Downloads          Music         .ssh
.bash_history    .exrc              output.txt    .sudo_as_admin_successful
.bash_logout     .gconf             Pictures      Templates
.bashrc          'Git and Github'   .pki          .thunderbird
.cache           .gitconfig         practice      Tishu
.config          .gnupg             .profile      training
Desktop          .local             Public        Videos
```

3.2   **-A, --almost-all : -** works exactly like [-a] but do not show [.] and [..]

```
beryl@beryl-ThinkPad-L412:~/practice$ ls -A
count.txt  file.txt  .git  .hello.py  new.txt  song  song.txt  tasks.txt
```

3.3   **--author : -** gives author of file when used with [-l]

permissions   hardl inks   owner   group   author   block size   modify dat e   file name
```
beryl@beryl-ThinkPad-L412:~/practice$ ls -l --author
total 24
-rw-rw-r-- 1 beryl beryl beryl  113 Jul 11 17:38 count.txt
-rw-rw-r-- 1 beryl beryl beryl  360 Jul 11 15:24 file.txt
-rw-rw-r-- 1 beryl beryl beryl    3 Jul 11 16:30 new.txt
drwxrwxr-x 2 beryl beryl beryl 4096 Jul 11 17:03 song
-rw-rw-r-- 1 beryl beryl beryl    3 Jul 11 16:31 song.txt
-rw-rw-r-- 1 beryl beryl beryl   32 Jul 12 10:18 tasks.txt
```

3.4   **-C : -** list enteries by column

```
beryl@beryl-ThinkPad-L412:~/practice$ ls -C
count.txt  file.txt  new.txt  song  song.txt  tasks.txt
```

3.5   **--color=[*parameter*] : -** colorize the output.
                    Parameters – always (*default*), never and auto

```
beryl@beryl-ThinkPad-L412:~/practice$ ls --color
count.txt  file.txt  new.txt  song  tasks.txt
beryl@beryl-ThinkPad-L412:~/practice$ ls --color=auto
count.txt  file.txt  new.txt  song  tasks.txt
beryl@beryl-ThinkPad-L412:~/practice$ ls --color=never
count.txt  file.txt  new.txt  song  tasks.txt
```

3.6   **-d, --directory : -** List directory entries instead of contents.

```
beryl@beryl-ThinkPad-L412:~/practice$ ls -d
.
```

3.6.1 ls -d */ : - to print only directory names

```
beryl@beryl-ThinkPad-L412:~/practice$ ls -d */
folder/  song/
```

3.7 -f : Do not sort, enable -aU, and disablels –color.
Prints the same ordr as file system with hidden files.

```
beryl@beryl-ThinkPad-L412:~/practice$ ls -f
.hello.py  new.txt  tasks.txt  folder  file.txt  count.txt  ..  .  song  .git
```

3.8 -F, --classify : Append indicator (one of */=>@|) to entries.

```
beryl@beryl-ThinkPad-L412:~/practice$ ls -F
count.txt  file.txt  folder/  new.txt  song/  tasks.txt
```

3.9 --format=word : - formats the output according.
across -x, commas -m, horizontal -x, long -l,
single-column -1, verbose -l, vertical -C

```
beryl@beryl-ThinkPad-L412:~/practice$ ls --format=commas
count.txt, file.txt, folder, new.txt, song, tasks.txt
beryl@beryl-ThinkPad-L412:~/practice$ ls -m
count.txt, file.txt, folder, new.txt, song, tasks.txt
```

3.10 --group-directories-first :- shows directories before files

```
beryl@beryl-ThinkPad-L412:~/practice$ ls --group-directories-first
folder  song  count.txt  file.txt  new.txt  tasks.txt
```

3.11 -h, --human-readable : - With -l, print sizes in human-readable format
(e.g., 1K, 234M, 2G).

```
beryl@beryl-ThinkPad-L412:~/practice$ ls -l -h
total 24K
-rw-rw-r-- 1 beryl beryl  113 Jul 11 17:38 count.txt
-rw-rw-r-- 1 beryl beryl  360 Jul 11 15:24 file.txt
drwxrwxr-x 2 beryl beryl 4.0K Jul 12 13:04 folder
-rw-rw-r-- 1 beryl beryl    3 Jul 11 16:30 new.txt
drwxrwxr-x 2 beryl beryl 4.0K Jul 11 17:03 song
-rw-rw-r-- 1 beryl beryl   32 Jul 12 10:18 tasks.txt
```

3.12   -i, -inode : - print the index number of each file

```
beryl@beryl-ThinkPad-L412:~/practice$ ls -i
12868204 count.txt   13109283 folder    13108211 song
12852080 file.txt    12860461 new.txt   12853286 tasks.txt
```

3.13   -n, --numeric-uid-gid :- Like **-l**, but list numeric user and group IDs

```
beryl@beryl-ThinkPad-L412:~/practice$ ls -n
total 24
-rw-rw-r-- 1 1000 1000  113 Jul 11 17:38 count.txt
-rw-rw-r-- 1 1000 1000  360 Jul 11 15:24 file.txt
drwxrwxr-x 2 1000 1000 4096 Jul 12 13:04 folder
-rw-rw-r-- 1 1000 1000    3 Jul 11 16:30 new.txt
drwxrwxr-x 2 1000 1000 4096 Jul 11 17:03 song
-rw-rw-r-- 1 1000 1000   32 Jul 12 10:18 tasks.txt
```

3.14   -p : - apped "/" indicator to directories

```
beryl@beryl-ThinkPad-L412:~/practice$ ls -p
count.txt  file.txt  folder/  new.txt  song/  tasks.txt
```

3.15   -Q, --quote-name : - enclose entry names in double quotes

```
beryl@beryl-ThinkPad-L412:~/practice$ ls -Q
"count.txt"  "file.txt"  "folder"  "new.txt"  "song"  "tasks.txt"
```

3.16   -R, --recursie : - prints subdirecteries recursively

```
beryl@beryl-ThinkPad-L412:~/practice$ ls -R
.:
count.txt  file.txt  folder  new.txt  song  tasks.txt

./folder:

./song:
jj.txt
```

3.17 --sort=*word* **: -** Sort by *word* instead of name: none (-U),
extension (-X), size (-S), time (-t), version (-v).

```
beryl@beryl-ThinkPad-L412:~/practice$ ls --sort=size
folder   song   file.txt   count.txt   tasks.txt   new.txt
beryl@beryl-ThinkPad-L412:~/practice$ ls -S
folder   song   file.txt   count.txt   tasks.txt   new.txt
```

3.18 -r, --reverse : - reverse the order when sorting

```
beryl@beryl-ThinkPad-L412:~/practice$ ls -S -r
new.txt   tasks.txt   count.txt   file.txt   song   folder
```

3.19 -1: - prints eqach filein a new line

```
beryl@beryl-ThinkPad-L412:~/practice$ ls -1
count.txt
file.txt
folder
new.txt
song
tasks.txt
```

----

4. echo : the echo command prints text to standard output

4.1 echo " " > : to create a text file in current directory
'>' - redirection is used to transfer output to a command or a file

```
beryl@beryl-ThinkPad-L412:~/practice$ echo "My name is Hero" > hero.txt
beryl@beryl-ThinkPad-L412:~/practice$ ls
count.txt   file.txt   folder   hero.txt   new.txt   song   tasks.txt
```

----

5. cat : to join two or more text files or to view the file contents['catenate']

    5.1 cat file_name : - view content of file

```
beryl@beryl-ThinkPad-L412:~/practice$ cat hero.txt
My name is Hero
```

    5.2   **cat file1 > file2 : -** tranfers data from one file to other file
                           (file2's data will be deleted as '>' overwrites)

    5.3   **cat file1 >> file2 : -** tranfers data from one file to other file
                           (file1's is added with file2's data in file2 as ">>" is used)

----

6. <<,<, >, >> Redirection : - used to redirect the input or ouputs from commands.

    6.1 : - Above in echo and cat we can see the working of redirection (>).

    6.2 : - We can also use the redirection to directly create files
         with any extension or no extension

```
beryl@beryl-ThinkPad-L412:~/practice/kite$ ls
file.txt
beryl@beryl-ThinkPad-L412:~/practice/kite$ > note.txt
beryl@beryl-ThinkPad-L412:~/practice/kite$ > empty
beryl@beryl-ThinkPad-L412:~/practice/kite$ > code.py
beryl@beryl-ThinkPad-L412:~/practice/kite$ ls
code.py  empty  file.txt  note.txt
```

----

7. mkdir : to create a folder in current working directory ['make directory']

   7.1 mkdir dir_name : -

```
beryl@beryl-ThinkPad-L412:~/practice$ ls
count.txt  file.txt  folder  hero.txt  new.txt  song  tasks.txt
beryl@beryl-ThinkPad-L412:~/practice$ mkdir kite
beryl@beryl-ThinkPad-L412:~/practice$ ls
count.txt  file.txt  folder  hero.txt  kite  new.txt  song  tasks.txt
```

7.2   mkdir -p multipe_parent_dir : - create a multilevel parent empty directory

```
beryl@beryl-ThinkPad-L412:~/practice$ ls
count.txt  file.txt  folder  hero.txt  kite  new.txt  song  tasks.txt
beryl@beryl-ThinkPad-L412:~/practice$ mkdir -p good/better/best
beryl@beryl-ThinkPad-L412:~/practice$ ls
count.txt  file.txt  folder  good  hero.txt  kite  new.txt  song  tasks.txt
beryl@beryl-ThinkPad-L412:~/practice$ ls good/
better
beryl@beryl-ThinkPad-L412:~/practice$ ls good/better
best
```

----

8. rm : to delete any file ['remove']
   [options] :

   8.1   rm file_name : - delete any file except folders

```
beryl@beryl-ThinkPad-L412:~/practice$ ls
count.txt  file.txt  folder  good  hero.txt  kite  new.txt  song  tasks.txt
beryl@beryl-ThinkPad-L412:~/practice$ rm new.txt
beryl@beryl-ThinkPad-L412:~/practice$ ls
count.txt  file.txt  folder  good  hero.txt  kite  song  tasks.txt
```

   8.2   -d, --dir : - delete empty directories
   Syntax : - *rm -d dir_name*

```
beryl@beryl-ThinkPad-L412:~/practice$ ls
count.txt  file.txt  folder  good  hero.txt  kite  song  tasks.txt
beryl@beryl-ThinkPad-L412:~/practice$
beryl@beryl-ThinkPad-L412:~/practice$ rm folder
rm: cannot remove 'folder': Is a directory
beryl@beryl-ThinkPad-L412:~/practice$
beryl@beryl-ThinkPad-L412:~/practice$ rm -d folder
beryl@beryl-ThinkPad-L412:~/practice$
beryl@beryl-ThinkPad-L412:~/practice$ ls
count.txt  file.txt  good  hero.txt  kite  song  tasks.txt
beryl@beryl-ThinkPad-L412:~/practice$
```

8.3 **-r, -R, --recursive :** **-** deletes directories and subfiles and subdiretories
Syntax : *rm -r dir_name*

```
beryl@beryl-ThinkPad-L412:~/practice$ ls
count.txt  file.txt  good  hero.txt  kite  song  tasks.txt
beryl@beryl-ThinkPad-L412:~/practice$ ls good/
better
beryl@beryl-ThinkPad-L412:~/practice$ rm -d good
rm: cannot remove 'good': Directory not empty
beryl@beryl-ThinkPad-L412:~/practice$
beryl@beryl-ThinkPad-L412:~/practice$ rm -r good/
beryl@beryl-ThinkPad-L412:~/practice$ ls
count.txt  file.txt  hero.txt  kite  song  tasks.txt
beryl@beryl-ThinkPad-L412:~/practice$
```

8.4 **-f, --force:** - ignore non existing files & never prompt before removing
Synatx : *rm -f file1 file2 .... fileN*

----

9. cp : to copy files or folders ['copy']
[options] :

9.1 Syntax : *cp path1 path2*

```
beryl@beryl-ThinkPad-L412:~/practice$ ls
count.txt  file.txt  good  hero.txt  kite  song  tasks.txt
beryl@beryl-ThinkPad-L412:~/practice$ ls good
better
beryl@beryl-ThinkPad-L412:~/practice$ cp hero.txt good/
beryl@beryl-ThinkPad-L412:~/practice$ ls good
better   hero.txt
beryl@beryl-ThinkPad-L412:~/practice$
```

9.2 : - creating a copy of file in same folder

```
beryl@beryl-ThinkPad-L412:~/practice$ ls
count.txt  file.txt  good  hero.txt  kite  song  tasks.txt
beryl@beryl-ThinkPad-L412:~/practice$ cat hero.txt
My name is Hero
beryl@beryl-ThinkPad-L412:~/practice$ cp hero.txt myname.txt
beryl@beryl-ThinkPad-L412:~/practice$ cat myname.txt
My name is Hero
```

9.3   -r, -R, --recursive : - to copy directories and subdirectories recursively

```
beryl@beryl-ThinkPad-L412:~/practice$ ls
count.txt  file.txt  good  hero.txt  kite  myname.txt  song  tasks.txt
beryl@beryl-ThinkPad-L412:~/practice$ ls good/
better  hero.txt
beryl@beryl-ThinkPad-L412:~/practice$ cp good bad
cp: -r not specified; omitting directory 'good'
beryl@beryl-ThinkPad-L412:~/practice$
beryl@beryl-ThinkPad-L412:~/practice$ cp -r good bad
beryl@beryl-ThinkPad-L412:~/practice$
beryl@beryl-ThinkPad-L412:~/practice$ ls -p bad
better/  hero.txt
beryl@beryl-ThinkPad-L412:~/practice$
```

----

10. mv : to move or rename files or folders ['move']

[options] :

10.1   moving a file
Syntax : *mv path1 path2*

```
beryl@beryl-ThinkPad-L412:~/practice$ ls
bad        file.txt  hello.txt  kite         song
count.txt  good      hero.txt  myname.txt  tasks.txt
beryl@beryl-ThinkPad-L412:~/practice$ mv file.txt kite/
beryl@beryl-ThinkPad-L412:~/practice$ ls
bad  count.txt  good  hello.txt  hero.txt  kite  myname.txt  song  tasks.txt
beryl@beryl-ThinkPad-L412:~/practice$ ls kite
file.txt
```

10.2 moving a file back to current directory, we can use [.] for current dir
Syntax : *mv path .*

```
beryl@beryl-ThinkPad-L412:~/practice$ ls kite
file.txt
beryl@beryl-ThinkPad-L412:~/practice$ mv kite/file.txt .
beryl@beryl-ThinkPad-L412:~/practice$ ls
bad         file.txt  hello.txt  kite        song
count.txt   good      hero.txt   myname.txt  tasks.txt
```

10.3 renaming file with mv
to rename a file with mv arg1 = original name arg2 = new name
(do not change tha path)

Syntax : *mv name1(pah1) name2(path1)*

```
beryl@beryl-ThinkPad-L412:~/practice$ ls
bad         file.txt  hello.txt  kite        song
count.txt   good      hero.txt   myname.txt  tasks.txt
beryl@beryl-ThinkPad-L412:~/practice$ mv hero.txt zero.txt
beryl@beryl-ThinkPad-L412:~/practice$ ls
bad         file.txt  hello.txt  myname.txt  tasks.txt
count.txt   good      kite       song        zero.txt
beryl@beryl-ThinkPad-L412:~/practice$
```

----

11. grep : the grep command processes text line by line, and prints any lines which match a specified pattern.

"global regular expression print,"

11.1 Syntax : *grep [search_term] [file_name]*

11.2 -i : insensitive search

```
beryl@beryl-ThinkPad-L412:~/practice$ cat home.txt
Hi my name is Dipanshu.
I live in faridabad.
Faridabad is an urban area.
My home is in the NIT area.
beryl@beryl-ThinkPad-L412:~/practice$ grep faridabad home.txt
I live in faridabad.
beryl@beryl-ThinkPad-L412:~/practice$ grep -i faridabad home.txt
I live in faridabad.
Faridabad is an urban area.
beryl@beryl-ThinkPad-L412:~/practice$
```

----

12. locate : the locate command finds files by name.

[options] :

12.1 locate (file/folder name) :

12.2 -c, --count : - count the results

12.3 -i, --ignore-case : - insensitive search

```
beryl@beryl-ThinkPad-L412:~$ locate better
/home/beryl/practice/bad/better
/home/beryl/practice/bad/better/best
/home/beryl/practice/good/better
/home/beryl/practice/good/better/best
beryl@beryl-ThinkPad-L412:~$ locate -c better
4
beryl@beryl-ThinkPad-L412:~$ locate Myname
beryl@beryl-ThinkPad-L412:~$ locate -i Myname
/home/beryl/practice/myname.txt
beryl@beryl-ThinkPad-L412:~$
```

----

13 chmod : the chmod command sets the permissions of files or directories.

[Permission Groups] :		[Permission Types] :		
owner	: u	Read	: r	: 4
group	: g	Write	: w	: 2
other	: o	Execute	: x	: 1
all users	: a			

Syntax : *chmod [group] = [pemissions], …. [file_name]*

```
beryl@beryl-ThinkPad-L412:~/practice$ ls -l count.txt
-rw-rw-r-- 1 beryl beryl 0 Jul 13 14:24 count.txt
beryl@beryl-ThinkPad-L412:~/practice$ chmod u=rwx,g=rwx count.txt
beryl@beryl-ThinkPad-L412:~/practice$ ls -l count.txt
-rwxrwxr-- 1 beryl beryl 0 Jul 13 14:24 count.txt
```

13.1

You can also use the **numerical (octal representation)** of **Permission types** to **change the permissions.**

Just **simply add the permission octals** according to **group rank.**

Eg: If you want to set permissions as – rw – rw - r

Total for owner = r + w i.e., 4 + 2 = 6
Total for group = r + w i.e., 4 + 2 = 6
Total for others = r i.e., 4 = 4
i.e., 662

```
beryl@beryl-ThinkPad-L412:~/practice$ chmod 664 count.txt
beryl@beryl-ThinkPad-L412:~/practice$ ls -l count.txt
-rw-rw-r-- 1 beryl beryl 0 Jul 13 14:24 count.txt
```

----

14 zip : to create zip archive/commpressed file of multiple files or folders

      Syntax : *zip [archive_name] [file1] [file2] .... [fileN]*

```
beryl@beryl-ThinkPad-L412:~/practice$ ls
bad          file.txt   hello.txt   hubble.txt   list.txt       song          tt.mp3
count.txt  good         home.txt    kite                      myname.txt  tasks.txt  zero.txt
beryl@beryl-ThinkPad-L412:~/practice$
beryl@beryl-ThinkPad-L412:~/practice$ zip zipper bad file.txt hello.txt
  adding: bad/ (stored 0%)
  adding: file.txt (deflated 92%)
  adding: hello.txt (stored 0%)
beryl@beryl-ThinkPad-L412:~/practice$ ls
bad          good         hubble.txt   myname.txt   tt.mp3
count.txt  hello.txt   kite          song          zero.txt
file.txt    home.txt    list.txt      tasks.txt    zipper.zip
```

    14.1   **-r, --recurse-paths :** In the above example "bad" directory contains sub-directories and files

             But, "zipper.zip" does not contain those sub-files
             To solve this we can use **-r** option. To include recursive paths

```
beryl@beryl-ThinkPad-L412:~/practice$ rm zipper.zip
beryl@beryl-ThinkPad-L412:~/practice$ zip -r zipper bad file.txt hello.txt
  adding: bad/ (stored 0%)
  adding: bad/hero.txt (stored 0%)
  adding: bad/better/ (stored 0%)
  adding: bad/better/best/ (stored 0%)
  adding: file.txt (deflated 92%)
  adding: hello.txt (stored 0%)
```

----

15 tail & head : to view top or bottom10 lines of any file.

```
beryl@beryl-ThinkPad-L412:~/practice$ head numbers.txt
1
2
3
4
5
6
7
8
9
10
```

```
beryl@beryl-ThinkPad-L412:~/practice$ tail numbers.txt
41
42
43
44
45
46
47
48
49
50
```

----

16.   tac : Works same like **cat** but prints file in reverse order line-by-line.
          - It also concate files in reverse order

```
beryl@beryl-ThinkPad-L412:~/practice$ cat home.txt
Hi my name is Dipanshu.
I live in faridabad.
Faridabad is an urban area.
My home is in the NIT area.
beryl@beryl-ThinkPad-L412:~/practice$ tac home.txt
My home is in the NIT area.
Faridabad is an urban area.
I live in faridabad.
Hi my name is Dipanshu.
```

----

17. **wc:** to cout the words, charachter or lines ['word count']

Syntax : *wc [option] [file_name]*

[optons] :

17.1 **-l, --lines : -** prints the line count

17.2 **-m, --chars : -** prints the charachter count

17.3 **-w, --words : -** prints the word count

```
beryl@beryl-ThinkPad-L412:~/practice$ cat home.txt
Hi my name is Dipanshu.
I live in faridabad.
Faridabad is an urban area.
My home is in the NIT area.
beryl@beryl-ThinkPad-L412:~/practice$ wc -l home.txt
4 home.txt
beryl@beryl-ThinkPad-L412:~/practice$ wc -w home.txt
21 home.txt
beryl@beryl-ThinkPad-L412:~/practice$ wc -m home.txt
101 home.txt
```

----

18. sort : to sort numbers or text line by line in a file

Syntax : *sort [option(s)] [file_name]*

```
beryl@beryl-ThinkPad-L412:~/practice$ cat names.txt
ajay
vijay
rohit mehta
rohit
pooja
sadhu
sir
sir2
asus
21fashion
beryl@beryl-ThinkPad-L412:~/practice$ sort names.txt
21fashion
ajay
asus
pooja
rohit
rohit mehta
sadhu
sir
sir2
vijay
```

[options] :

18.1   -n, --numeric-sort : compare according to their numeric value

```
beryl@beryl-ThinkPad-L412:~/practice$ sort numbers.txt
0010000
034
0.74
1
2
2.34
29
328u3
34 24
8293
laptop
beryl@beryl-ThinkPad-L412:~/practice$ sort -n numbers.txt
laptop
0.74
1
2
2.34
29
034
34 24
328u3
8293
0010000
```

18.2   -r, --reverse : reverse the results

```
beryl@beryl-ThinkPad-L412:~/practice$ sort -n -r numbers.txt
0010000
8293
328u3
34 24
034
29
2.34
2
1
0.74
laptop
```

18.3 -R, --random-sort : sort by random hash of keys

```
beryl@beryl-ThinkPad-L412:~/practice$ sort -R numbers.txt
34 24
0.74
8293
0010000
2
034
laptop
2.34
29
328u3
1
```

18.4 -h, --human-numeric-sort : compare human readable numbers (1M, 3G, 2T)
- sort according to the convention of file sizes
- K < M < G < T ....

----

19. | pipe : to provide output of a command as input in other command

```
beryl@beryl-ThinkPad-L412:~/practice$ ls -1 | wc -l
20
beryl@beryl-ThinkPad-L412:~/practice$ echo "John Wick" | wc -m
10
beryl@beryl-ThinkPad-L412:~/practice$ ls -1 | sort
bad
count.txt
file.txt
ghar.txt
good
hello.txt
home.txt
hubble.txt
kite
list.txt
myname.txt
names.txt
numbers_sorted.txt
numbers.txt
num.txt
song
tasks.txt
tt.mp3
zero.txt
zipper.zip
```

20. stat : the stat command displays the detailed status of a particular file
or a file system.

```
beryl@beryl-ThinkPad-L412:~/practice$ stat bad
  File: bad
  Size: 4096          Blocks: 8          IO Block: 4096    directory
Device: 805h/2053d    Inode: 13109306    Links: 3
Access: (0775/drwxrwxr-x)  Uid: ( 1000/   beryl)   Gid: ( 1000/   beryl)
Access: 2022-07-12 18:15:04.363108559 +0530
Modify: 2022-07-12 17:58:25.571487461 +0530
Change: 2022-07-12 17:58:25.571487461 +0530
 Birth: -
beryl@beryl-ThinkPad-L412:~/practice$ stat names.txt
  File: names.txt
  Size: 66            Blocks: 8          IO Block: 4096    regular file
Device: 805h/2053d    Inode: 12866661    Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/   beryl)   Gid: ( 1000/   beryl)
Access: 2022-07-13 15:56:47.741481166 +0530
Modify: 2022-07-13 15:52:50.148276201 +0530
Change: 2022-07-13 15:52:50.204276377 +0530
 Birth: -
```

----

21. cal : - the cal commands display a formatted calendar in the terminal.

```
beryl@beryl-ThinkPad-L412:~/practice$ cal
      July 2022
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
beryl@beryl-ThinkPad-L412:~/practice$ cal -m 12
    December 2022
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

beryl@beryl-ThinkPad-L412:~/practice$ cal 12 1997
    December 1997
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

22. clear : - to clear the screen of the terminal.
            - Can also use **ctrl+l**

23. exit : - closes the terminal window.

Screenshot Commands :

gnome-screenshot   **: -** screenshot of whole screen
gnome-screenshot -w   **:** - screenshot of terminal window
gnome-screenshot -d t : - scrennshot taken after 't' seconds
gnome-screenshot -a   **:** - custom screenshot
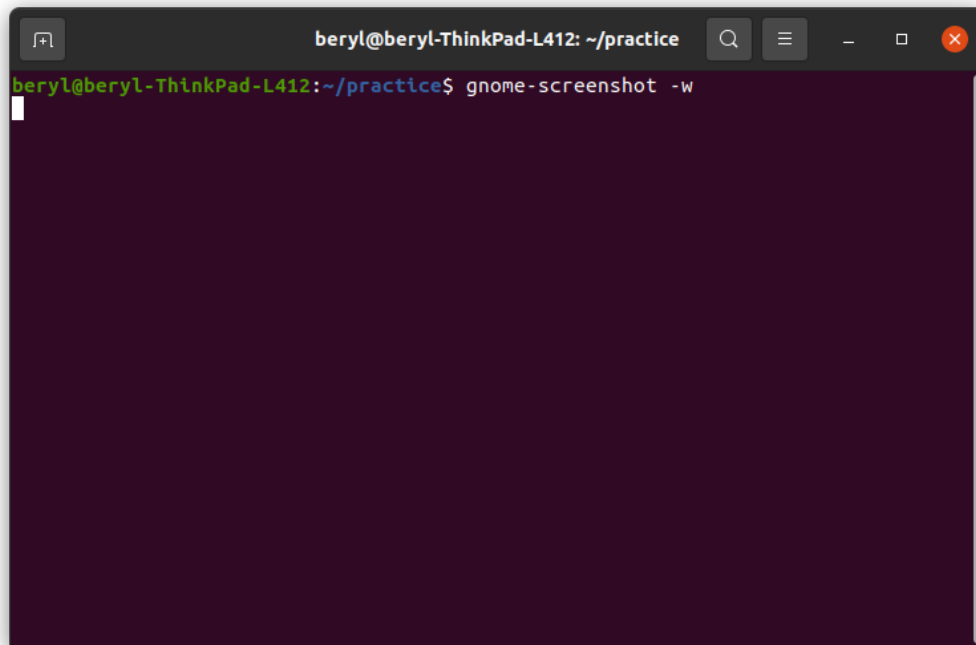
- **nano : -** nano is a simple text editor that can be run in the terminal itself.
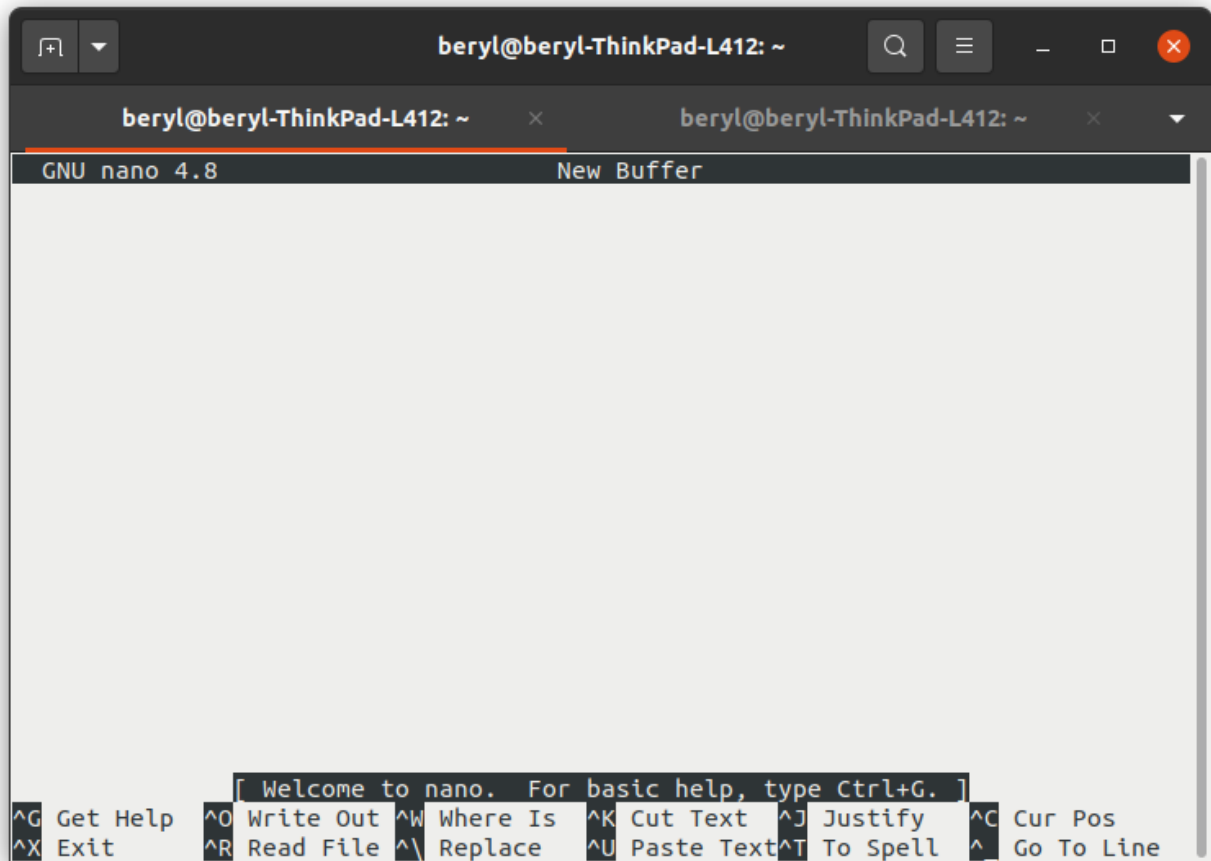
    Run / Launch : just write 'nano'

Figure 1: nano editor first run screen

    Open File : - *nano [file_name / path]*
        * If no file of that existed then nano will create an empty file of same name

    Save File : - You can type text simply like a normal text editer in nano.
        Ctrl + o is used to write out save file.

    Help Menu : - To open help shorcut menu press **ctrl + g**

    Exit nano : - You can exit nano by pressing **ctrl + x**