Linux Command

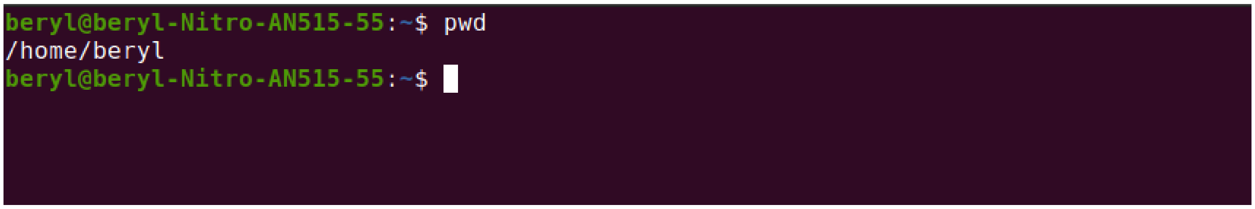
Command name: `pwd` **— Print Working Directory**

Purpose: **Displays the absolute path of the current working directory.**

Syntax: `pwd [options]`

Example: `pwd`

Output:



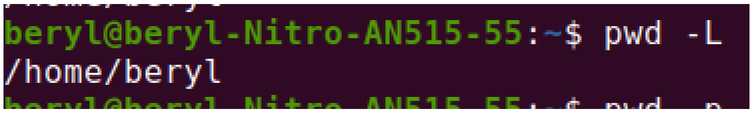
```
beryl@beryl-Nitro-AN515-55:~$ pwd
/home/beryl
beryl@beryl-Nitro-AN515-55:~$ 
```

`pwd` Options

-L → Displays the logical path, including symbolic **links (default behavior)**

Example: `pwd -L`

Output :

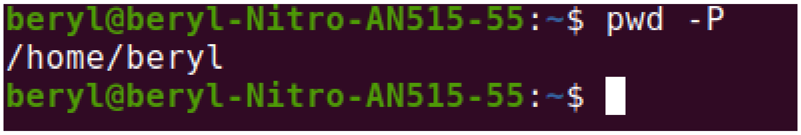


```
beryl@beryl-Nitro-AN515-55:~$ pwd -L
/home/beryl
```

-P → Displays the physical path, resolving all **symbolic links**

Example: **`pwd -P`**

Output :



```
beryl@beryl-Nitro-AN515-55:~$ pwd -P
/home/beryl
beryl@beryl-Nitro-AN515-55:~$ 
```

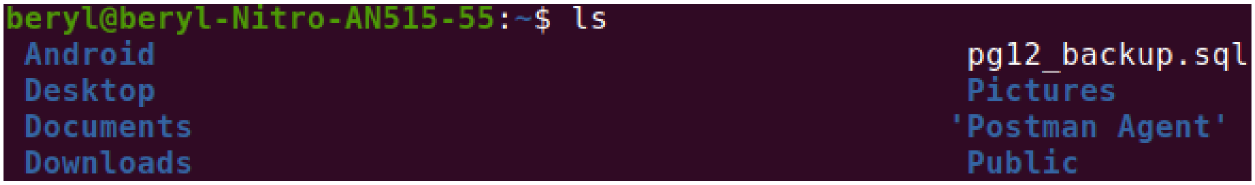
Command name :   ls – List directory contents

Purpose: Lists files and directories in a directory.

Syntax: ls [options] [path]

Example: ls

Output :



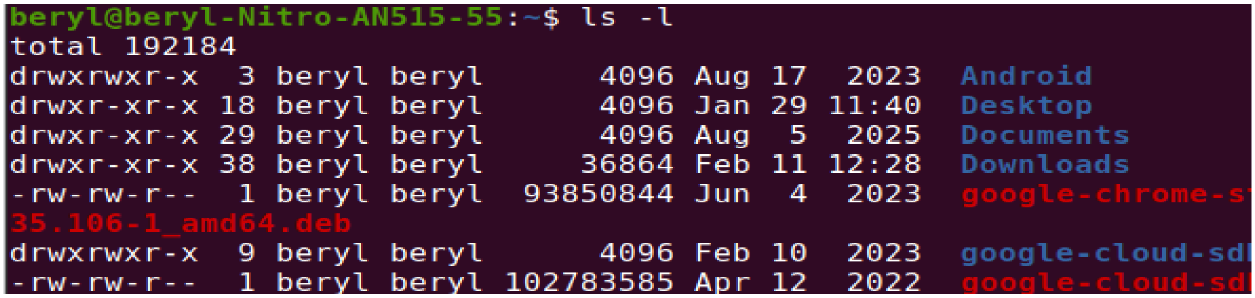
```
beryl@beryl-Nitro-AN515-55:~$ ls
 Android                                          pg12_backup.sql
 Desktop                                          Pictures
 Documents                                       'Postman Agent'
 Downloads                                        Public
```

ls Options

-l →   Displays files in long format (permissions, owner, size, date)

Example: ls -l

Output :

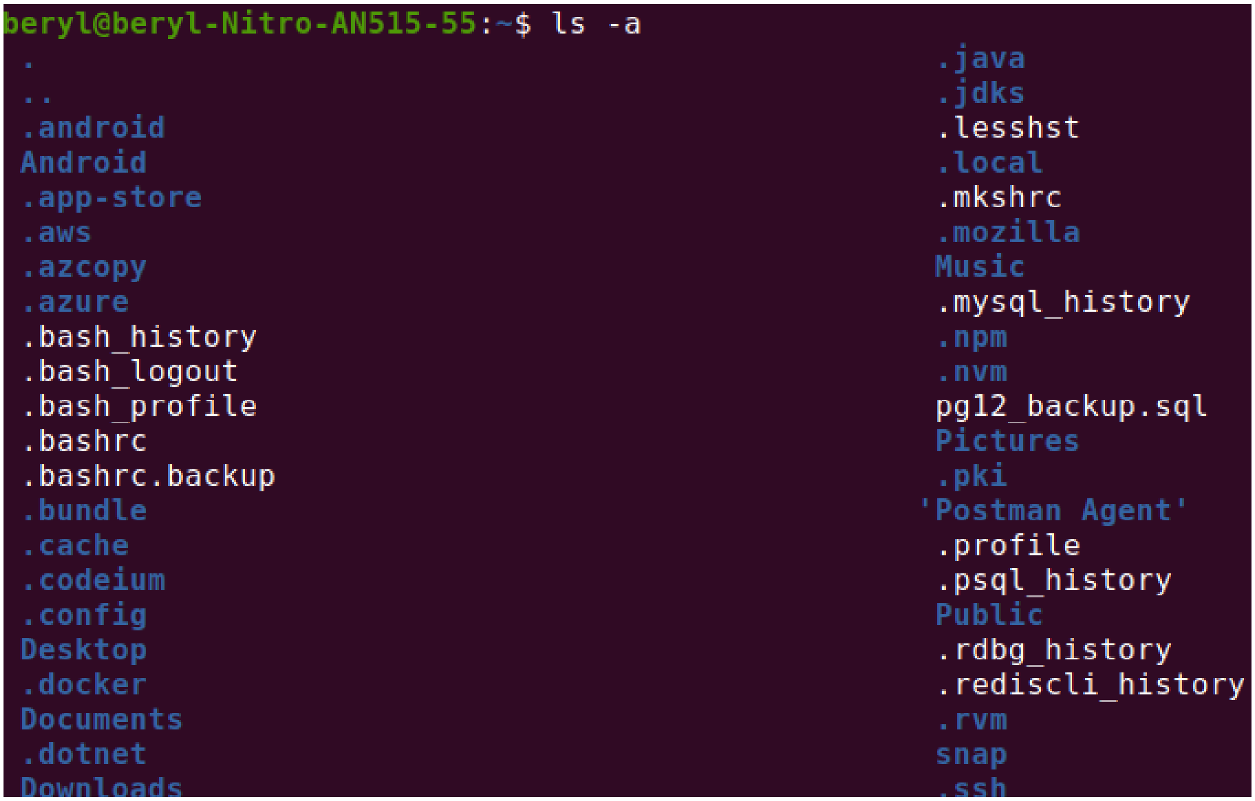


```
beryl@beryl-Nitro-AN515-55:~$ ls -l
total 192184
drwxrwxr-x  3 beryl beryl      4096 Aug 17  2023  Android
drwxr-xr-x 18 beryl beryl      4096 Jan 29 11:40  Desktop
drwxr-xr-x 29 beryl beryl      4096 Aug  5  2025  Documents
drwxr-xr-x 38 beryl beryl     36864 Feb 11 12:28  Downloads
-rw-rw-r--  1 beryl beryl  93850844 Jun  4  2023  google-chrome-s
35.106-1_amd64.deb
drwxrwxr-x  9 beryl beryl      4096 Feb 10  2023  google-cloud-sdk
-rw-rw-r--  1 beryl beryl 102783585 Apr 12  2022  google-cloud-sdk
```

-a → Shows all files, including hidden files (. files)

Example: `ls -a`

Output :

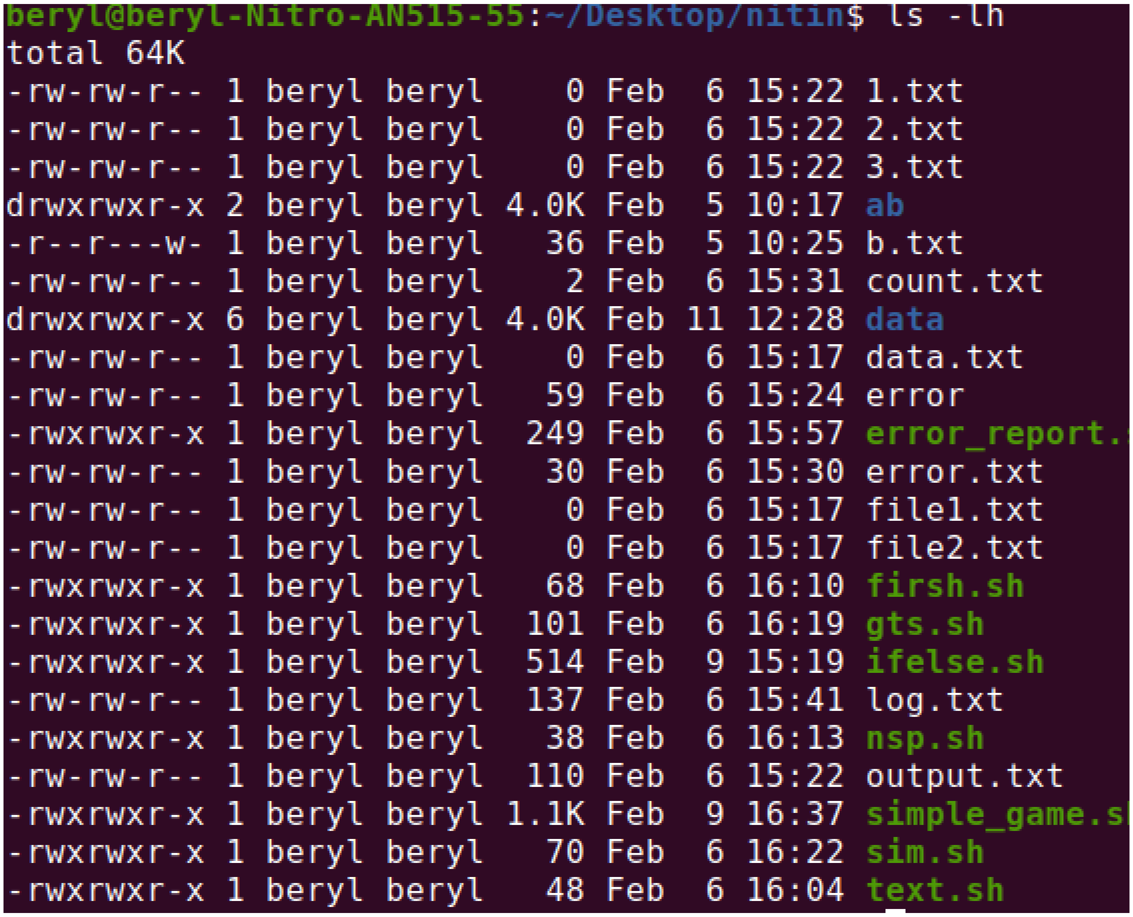


```
beryl@beryl-Nitro-AN515-55:~$ ls -a
 .                                                .java
 ..                                               .jdks
 .android                                         .lesshst
 Android                                          .local
 .app-store                                       .mkshrc
 .aws                                             .mozilla
 .azcopy                                          Music
 .azure                                           .mysql_history
 .bash_history                                    .npm
 .bash_logout                                     .nvm
 .bash_profile                                    pg12_backup.sql
 .bashrc                                          Pictures
 .bashrc.backup                                   .pki
 .bundle                                         'Postman Agent'
 .cache                                           .profile
 .codeium                                         .psql_history
 .config                                          Public
 Desktop                                          .rdbg_history
 .docker                                          .rediscli_history
 Documents                                        .rvm
 .dotnet                                          snap
 Downloads                                        .ssh
```

-h → Shows file sizes in human-readable format (KB, MB, GB)

Example: `ls -lh`

Output :

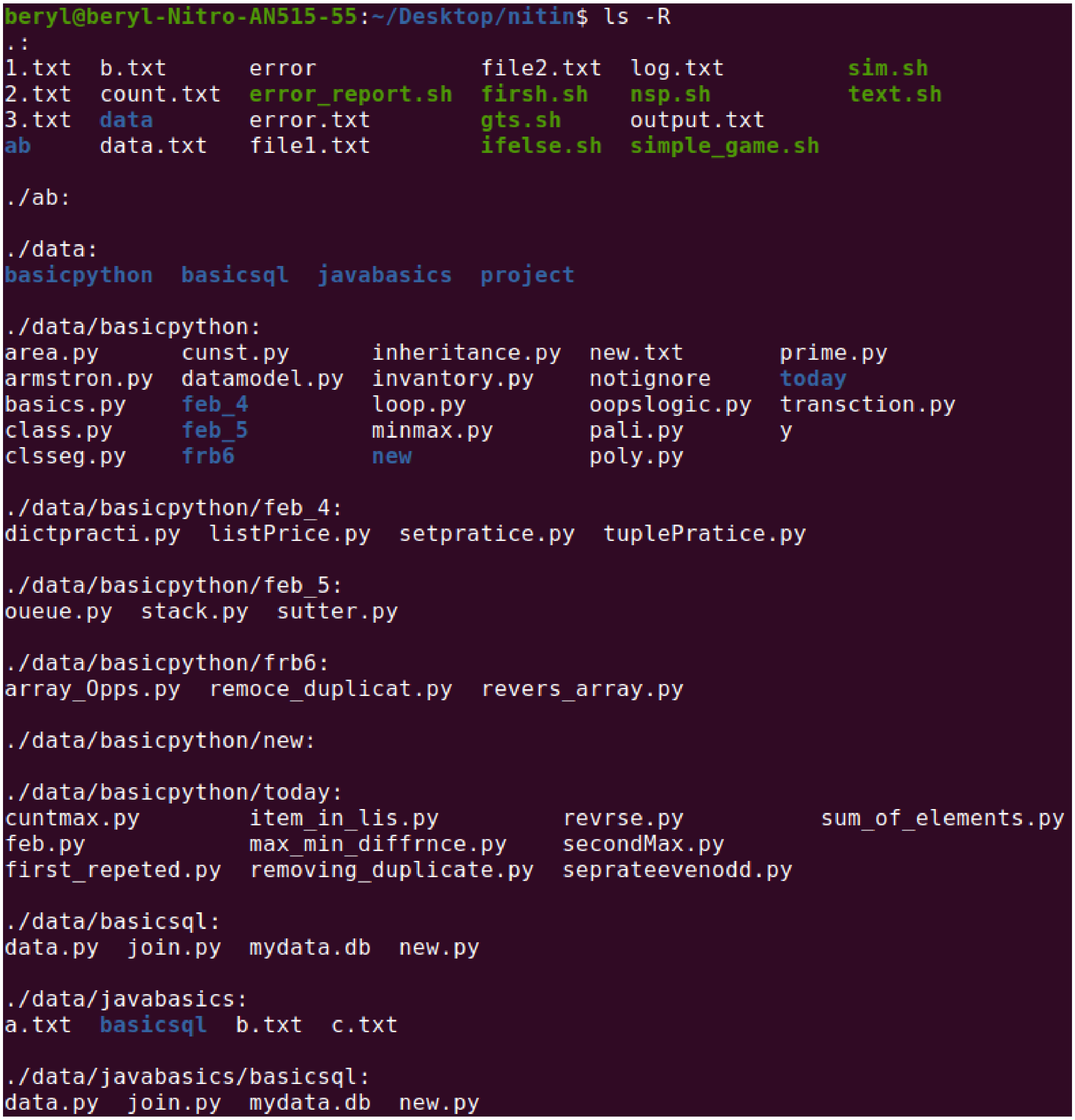


```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ ls -lh
total 64K
-rw-rw-r-- 1 beryl beryl    0 Feb  6 15:22 1.txt
-rw-rw-r-- 1 beryl beryl    0 Feb  6 15:22 2.txt
-rw-rw-r-- 1 beryl beryl    0 Feb  6 15:22 3.txt
drwxrwxr-x 2 beryl beryl 4.0K Feb  5 10:17 ab
-r--r---w- 1 beryl beryl   36 Feb  5 10:25 b.txt
-rw-rw-r-- 1 beryl beryl    2 Feb  6 15:31 count.txt
drwxrwxr-x 6 beryl beryl 4.0K Feb 11 12:28 data
-rw-rw-r-- 1 beryl beryl    0 Feb  6 15:17 data.txt
-rw-rw-r-- 1 beryl beryl   59 Feb  6 15:24 error
-rwxrwxr-x 1 beryl beryl  249 Feb  6 15:57 error_report.
-rw-rw-r-- 1 beryl beryl   30 Feb  6 15:30 error.txt
-rw-rw-r-- 1 beryl beryl    0 Feb  6 15:17 file1.txt
-rw-rw-r-- 1 beryl beryl    0 Feb  6 15:17 file2.txt
-rwxrwxr-x 1 beryl beryl   68 Feb  6 16:10 firsh.sh
-rwxrwxr-x 1 beryl beryl  101 Feb  6 16:19 gts.sh
-rwxrwxr-x 1 beryl beryl  514 Feb  9 15:19 ifelse.sh
-rw-rw-r-- 1 beryl beryl  137 Feb  6 15:41 log.txt
-rwxrwxr-x 1 beryl beryl   38 Feb  6 16:13 nsp.sh
-rw-rw-r-- 1 beryl beryl  110 Feb  6 15:22 output.txt
-rwxrwxr-x 1 beryl beryl 1.1K Feb  9 16:37 simple_game.s
-rwxrwxr-x 1 beryl beryl   70 Feb  6 16:22 sim.sh
-rwxrwxr-x 1 beryl beryl   48 Feb  6 16:04 text.sh
```

-R → Lists files and directories

Recursively

Example: **ls -R**

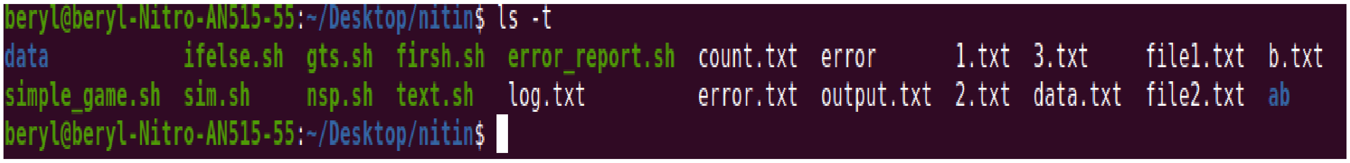


```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ ls -R
.:
1.txt  b.txt      error            file2.txt  log.txt          sim.sh
2.txt  count.txt  error_report.sh  firsh.sh   nsp.sh           text.sh
3.txt  data       error.txt        gts.sh     output.txt
ab     data.txt   file1.txt        ifelse.sh  simple_game.sh

./ab:

./data:
basicpython  basicsql  javabasics  project

./data/basicpython:
area.py      cunst.py      inheritance.py  new.txt       prime.py
armstron.py  datamodel.py  invantory.py    notignore     today
basics.py    feb_4         loop.py         oopslogic.py  transction.py
class.py     feb_5         minmax.py       pali.py       y
clsseg.py    frb6          new             poly.py

./data/basicpython/feb_4:
dictpracti.py  listPrice.py  setpratice.py  tuplePratice.py

./data/basicpython/feb_5:
oueue.py  stack.py  sutter.py

./data/basicpython/frb6:
array_Opps.py  remoce_duplicat.py  revers_array.py

./data/basicpython/new:

./data/basicpython/today:
cuntmax.py        item_in_lis.py         revrse.py          sum_of_elements.py
feb.py            max_min_diffrnce.py    secondMax.py
first_repeted.py  removing_duplicate.py  seprateevenodd.py

./data/basicsql:
data.py  join.py  mydata.db  new.py

./data/javabasics:
a.txt  basicsql  b.txt  c.txt

./data/javabasics/basicsql:
data.py  join.py  mydata.db  new.py
```

-t → Sorts files by modification time

(newest first)

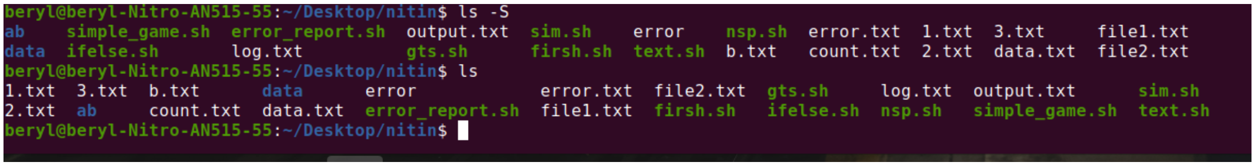
Example: **ls -t**



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ ls -t
data           ifelse.sh  gts.sh  firsh.sh  error_report.sh  count.txt  error       1.txt  3.txt     file1.txt  b.txt
simple_game.sh  sim.sh     nsp.sh  text.sh   log.txt          error.txt  output.txt  2.txt  data.txt  file2.txt  ab
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$
```

-S → Sorts files by size (largest first)

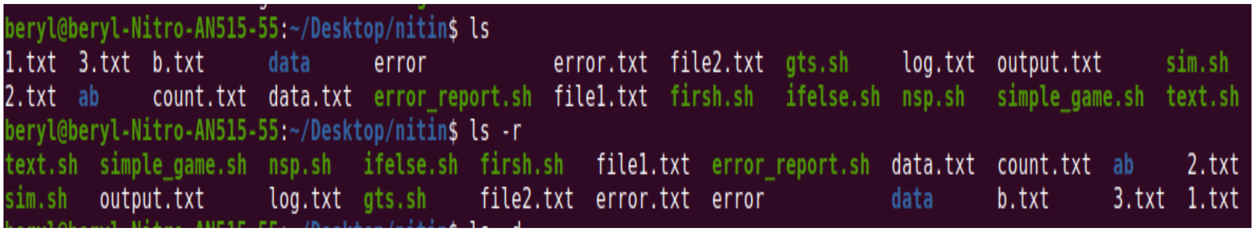
Example: **ls -S**



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ ls -S
ab    simple_game.sh  error_report.sh  output.txt  sim.sh    error    nsp.sh  error.txt  1.txt  3.txt     file1.txt
data  ifelse.sh       log.txt          gts.sh      firsh.sh  text.sh  b.txt   count.txt  2.txt  data.txt  file2.txt
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ ls
1.txt  3.txt  b.txt      data      error            error.txt  file2.txt  gts.sh     log.txt  output.txt      sim.sh
2.txt  ab     count.txt  data.txt  error_report.sh  file1.txt  firsh.sh   ifelse.sh  nsp.sh   simple_game.sh  text.sh
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$
```

-r → Reverses the sorting order

Example: **ls -r**



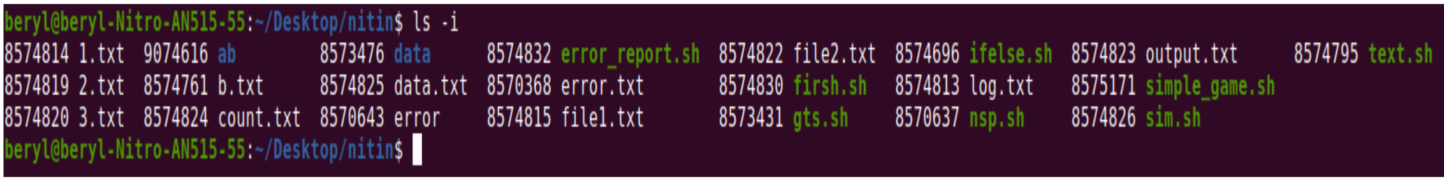
```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ ls
1.txt  3.txt  b.txt      data      error            error.txt  file2.txt  gts.sh     log.txt  output.txt      sim.sh
2.txt  ab     count.txt  data.txt  error_report.sh  file1.txt  firsh.sh   ifelse.sh  nsp.sh   simple_game.sh  text.sh
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ ls -r
text.sh  simple_game.sh  nsp.sh   ifelse.sh  firsh.sh   file1.txt  error_report.sh  data.txt  count.txt  ab     2.txt
sim.sh   output.txt      log.txt  gts.sh     file2.txt  error.txt  error            data      b.txt      3.txt  1.txt
```

-d → Lists directories themselves, not their contents

Example: `ls -d folder`

-i → Displays the inode number of files

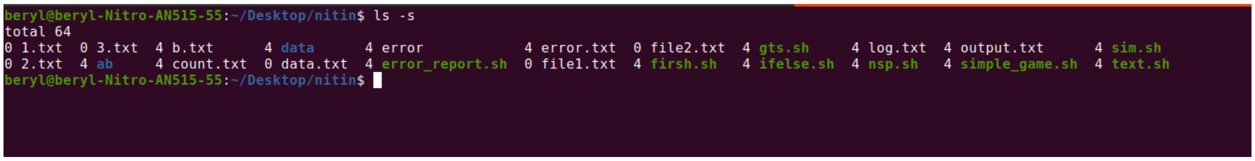
Example: `ls -i`



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ ls -i
8574814 1.txt  9074616 ab         8573476 data      8574832 error_report.sh  8574822 file2.txt  8574696 ifelse.sh  8574823 output.txt       8574795 text.sh
8574819 2.txt  8574761 b.txt      8574825 data.txt  8570368 error.txt        8574830 firsh.sh   8574813 log.txt    8575171 simple_game.sh
8574820 3.txt  8574824 count.txt  8570643 error     8574815 file1.txt        8573431 gts.sh     8570637 nsp.sh     8574826 sim.sh
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ 
```

-s → Displays file size in blocks before file name

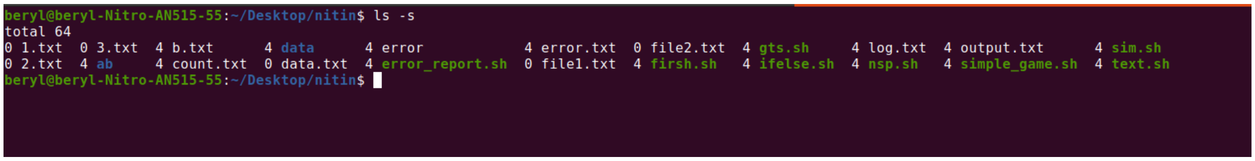
Example: `ls -s`



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ ls -s
total 64
0 1.txt  0 3.txt  4 b.txt      4 data      4 error            4 error.txt  0 file2.txt  4 gts.sh     4 log.txt  4 output.txt      4 sim.sh
0 2.txt  4 ab     4 count.txt  0 data.txt  4 error_report.sh  0 file1.txt  4 firsh.sh   4 ifelse.sh  4 nsp.sh   4 simple_game.sh  4 text.sh
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ 
```

-F → Appends indicators (/, *, @) to file names

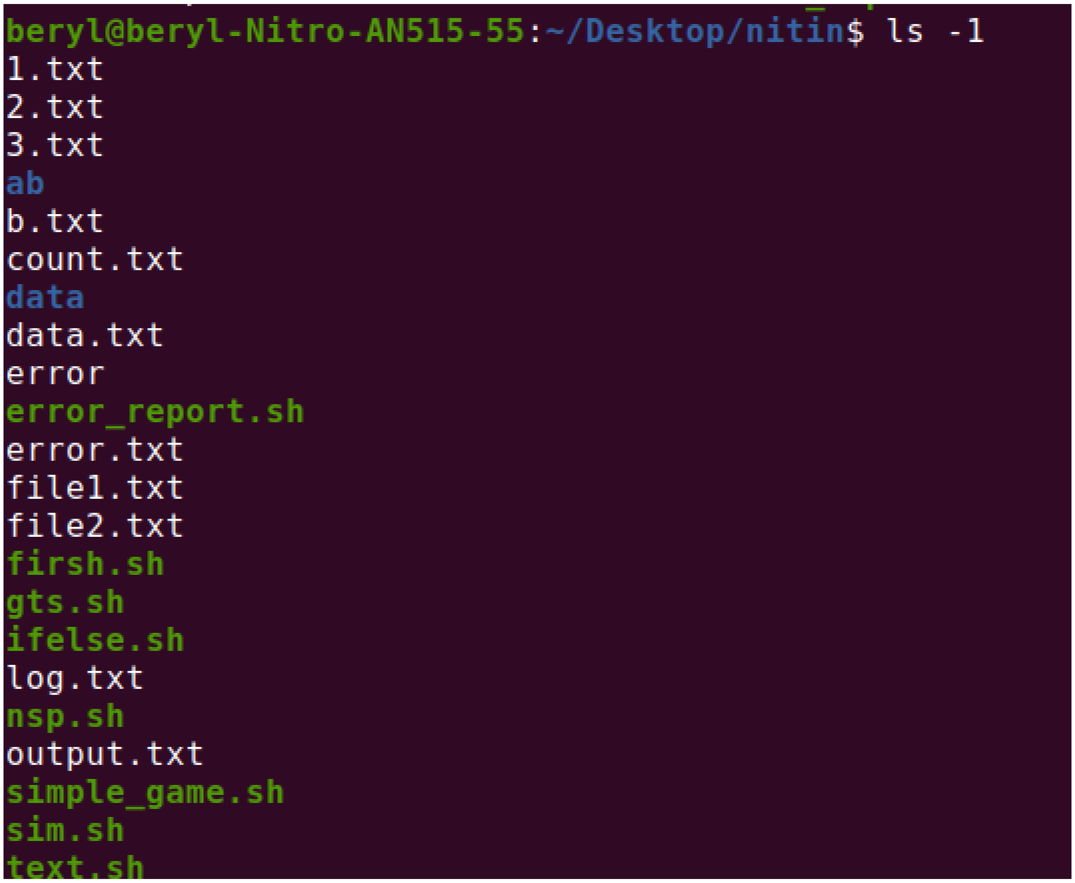
Example: **ls -F**



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ ls -s
total 64
0 1.txt  0 3.txt  4 b.txt      4 data      4 error            4 error.txt  0 file2.txt  4 gts.sh     4 log.txt  4 output.txt      4 sim.sh
0 2.txt  4 ab     4 count.txt  0 data.txt  4 error_report.sh  0 file1.txt  4 firsh.sh   4 ifelse.sh  4 nsp.sh   4 simple_game.sh  4 text.sh
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ 
```

-1 → Lists one file per line

Example: **ls -1**



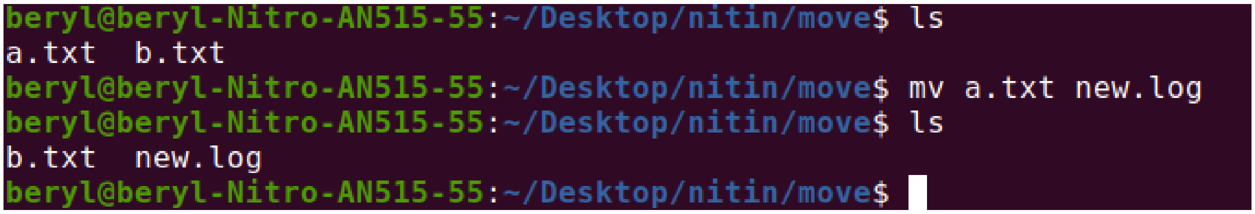
```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ ls -1
1.txt
2.txt
3.txt
ab
b.txt
count.txt
data
data.txt
error
error_report.sh
error.txt
file1.txt
file2.txt
firsh.sh
gts.sh
ifelse.sh
log.txt
nsp.sh
output.txt
simple_game.sh
sim.sh
text.sh
```

Command name: mv — Move or rename files/directories

Purpose : **Moves files or directories from one location to another, or renames them.**

Syntax: `mv [options] source destination`

Example: `mv file1.txt file2.txt`

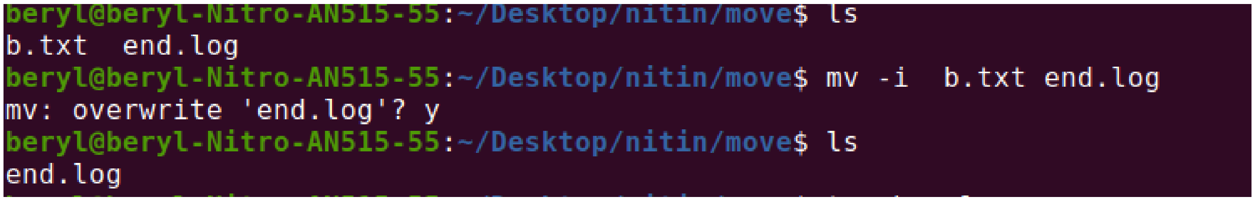


```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ ls
a.txt  b.txt
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ mv a.txt new.log
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ ls
b.txt  new.log
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ 
```

`mv` Options

`-i` → Prompts before overwriting an existing file

Example: `mv  -i  a.txt  b.txt`

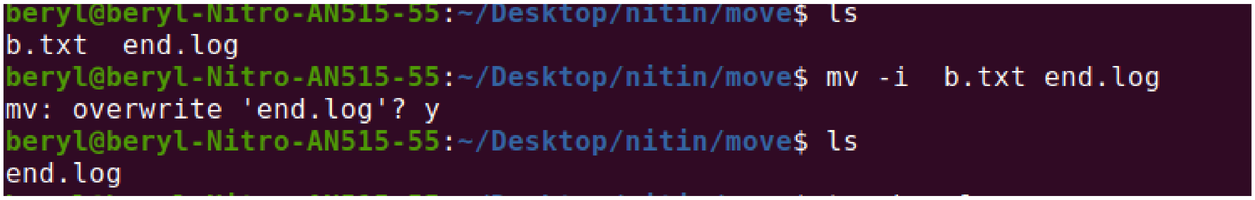


```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ ls
b.txt  end.log
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ mv -i  b.txt end.log
mv: overwrite 'end.log'? y
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ ls
end.log
```

`-f` → Forces move without confirmation

(overwritesfiles)

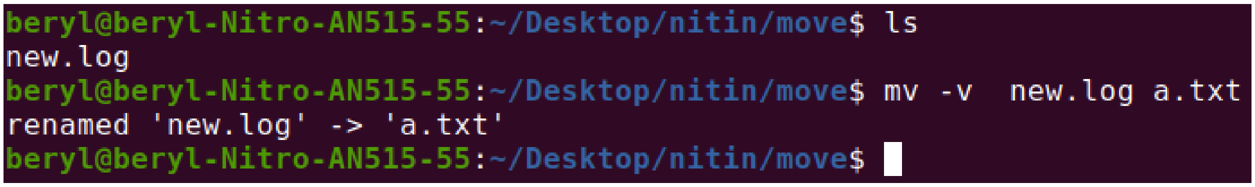
Example: `mv -f a.txt b.txt`



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ ls
b.txt   end.log
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ mv -i  b.txt end.log
mv: overwrite 'end.log'? y
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ ls
end.log
```

-v → Displays verbose output, showing files being **Moved**

Example: `mv -v old.txt new.txt`



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ ls
new.log
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ mv -v  new.log a.txt
renamed 'new.log' -> 'a.txt'
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ 
```

-n → Does not overwrite an existing file

Example: `mv -n a.txt b.txt`

-u → Moves file only if source is newer than **destination**

Example: `mv -u a.txt b.txt`

`-t DIR` → Moves all source files into the target directory

Example: `mv -t /backup file1 file2`

`--backup` → Creates a backup of the destination file before **overwriting**

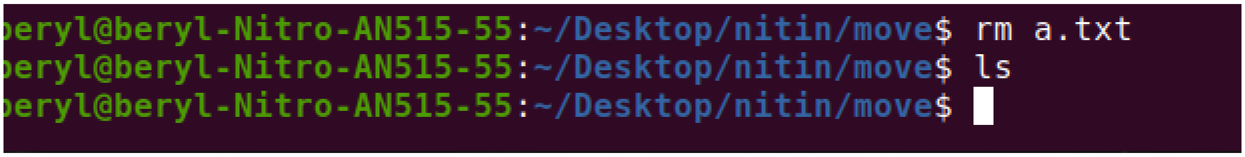
Example: `mv --backup a.txt b.txt`

Command name: `rm` – Remove files or directories

Purpose: Deletes files or directories from the filesystem.

Syntax: `rm [options] file_or_directory`

Example: `rm file.txt`

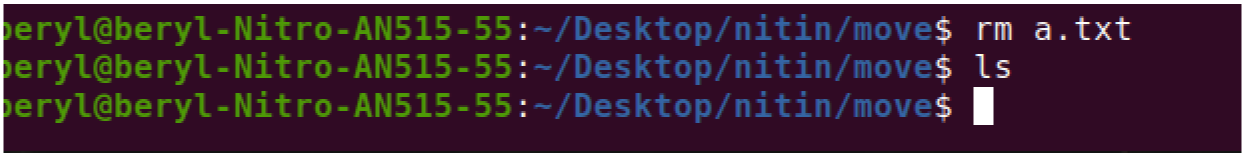


```
eryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ rm a.txt
eryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ ls
eryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ 
```

`rm` Options

`-r / -R` → Removes directories and their contents recursively

Example: `rm -r folder`



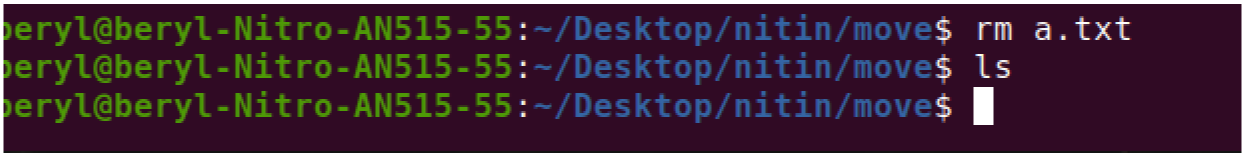
```
eryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ rm a.txt
eryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ ls
eryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ 
```

-f → Forces deletion without confirmation (ignores errors)

Example: `rm -f file.txt`

-i → Prompts before every deletion

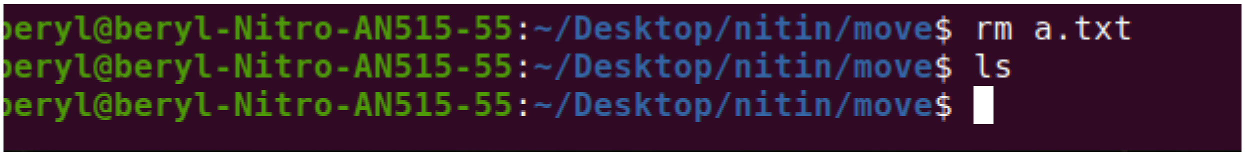
Example: `rm -i file.txt`



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ rm a.txt
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ ls
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ 
```

-v → Shows verbose output for each removed file

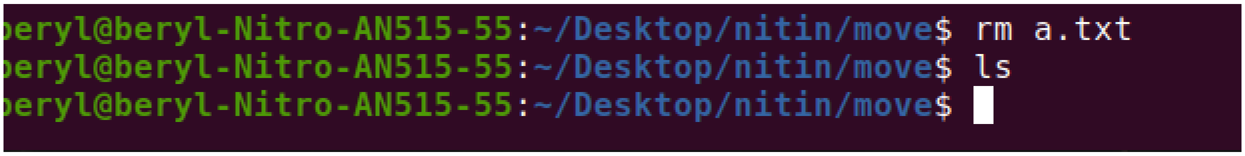
Example: `rm -v file.txt`



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ rm a.txt
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ ls
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ 
```

-d → Removes empty directories

Example: `rm -d emptydir`



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ rm a.txt
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ ls
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/move$ 
```

--preserve-root → Prevents deletion of the root directory (default safety feature)

Example: `rm --preserve-root -rf /`

--no-preserve-root → Allows deletion of root directory (very dangerous)

Example: `rm --no-preserve-root -rf /`

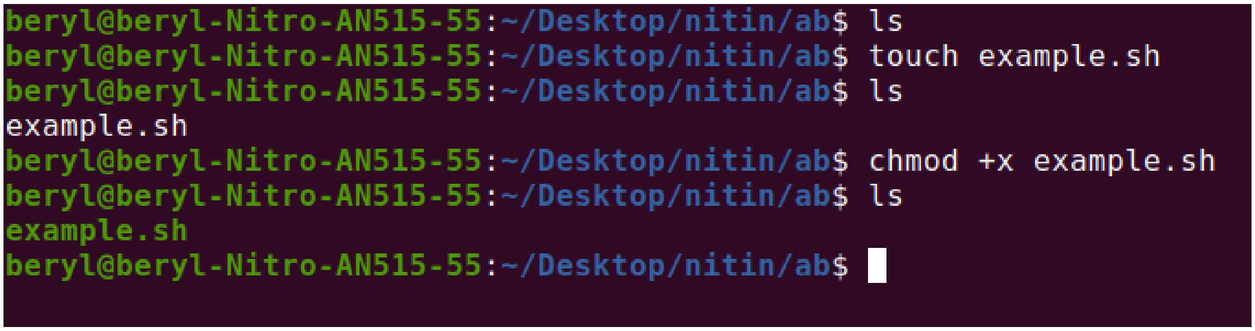
Command name: chmod – Change file permissions

Purpose : Changes the permissions (read, write, execute) of files or directories.

Syntax: chmod [options] mode file

Examples: chmod 755 script.sh

chmod +x script.sh



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ ls
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ touch example.sh
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ ls
example.sh
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ chmod +x example.sh
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ ls
example.sh
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ 
```

chmod Permission Modes (important to know)

Numeric (Octal) mode

- 4 → read (r)
- 2 → write (w)
- 1 → execute (x)

Examples:

```
chmod 777 file   # rwxrwxrwx

chmod 755 file   # rwxr-xr-x

chmod 644 file   # rw-r--r--
```

Symbolic mode

- u → user (owner)
- g → group

- o → others

- a → all

Examples:

```
chmod u+x file
chmod g-w file
chmod a+r file
```

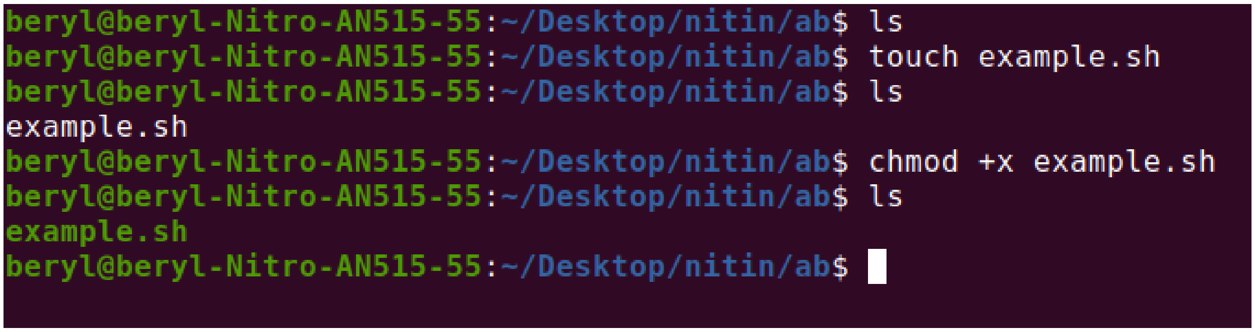
chmod Options

-R → Changes permissions recursively for directories and their contents

Examples: `chmod -R 755 project`

-v → Displays a message for each file whose permissions are changed

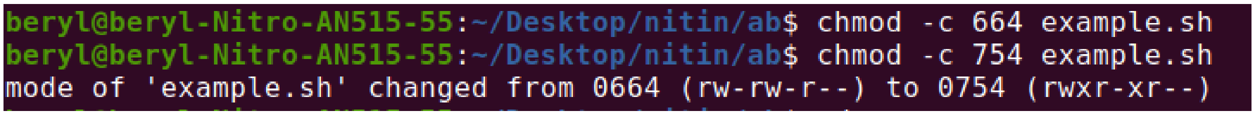
Examples: `chmod -v 644 file.txt`



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ ls
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ touch example.sh
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ ls
example.sh
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ chmod +x example.sh
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ ls
example.sh
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ 
```

-c → Shows output only when a change is made

Examples: `chmod -c 755 script.sh`



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ chmod -c 664 example.sh
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ chmod -c 754 example.sh
mode of 'example.sh' changed from 0664 (rw-rw-r--) to 0754 (rwxr-xr--)
```

--reference=FILE → Sets permissions same as another file

Examples: `chmod --reference=ref.txt target.txt`

Command name: `chown` – Change file owner and group

Purpose : Changes the ownership (user and/or group) of files or directories.

Syntax: `chown [options] user[:group] file`

Examples:

```
chown user file.txt
chown user:group file.txt
```

`chown` Options

`-R` → Changes ownership recursively for directories and their contents

Examples:

```
chown -R user:group project
```

-v → Displays a message for each file whose ownership is changed

Examples:

```
chown -v user file.txt
```

-c → Shows output only when a change is made

Examples:

```
chown -c user file.txt
```

--reference=FILE → Sets ownership same as another file

Examples:

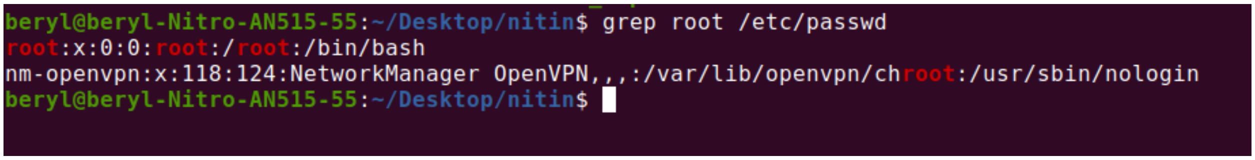
```
chown --reference=ref.txt target.txt
```

Command name: grep – Search text patterns

Purpose : Searches for a specific pattern or word inside files.

Syntax: `grep [options] pattern file`

Example: `grep root /etc/passwd`

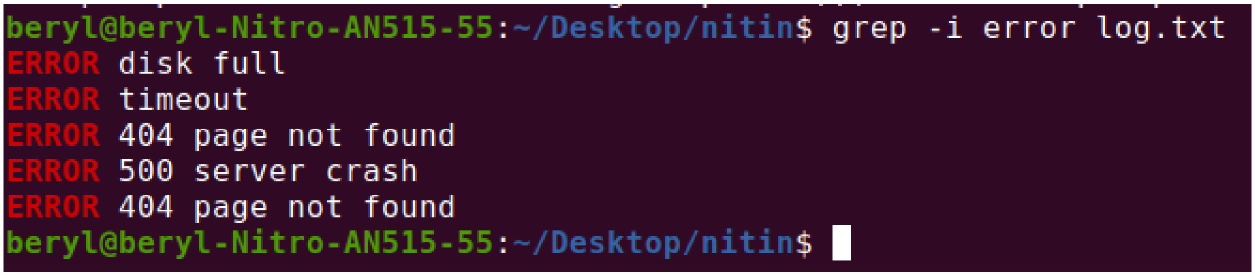


```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ grep root /etc/passwd
root:x:0:0:root:/root:/bin/bash
nm-openvpn:x:118:124:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ 
```

grep Options

-i → Ignores case sensitivity while searching

Example: `grep -i error log.txt`



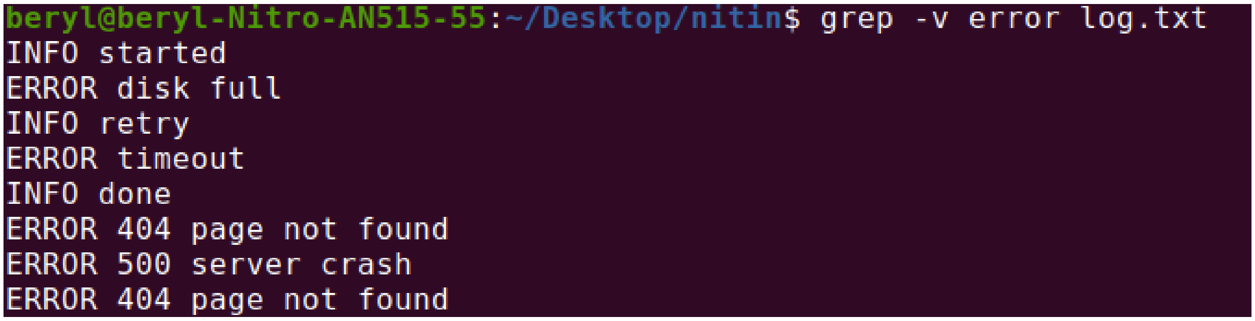
```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ grep -i error log.txt
ERROR disk full
ERROR timeout
ERROR 404 page not found
ERROR 500 server crash
ERROR 404 page not found
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ 
```

-n → Displays line numbers with matching lines

Example: **grep -n main program.c**

-v → Displays lines that do NOT match the pattern

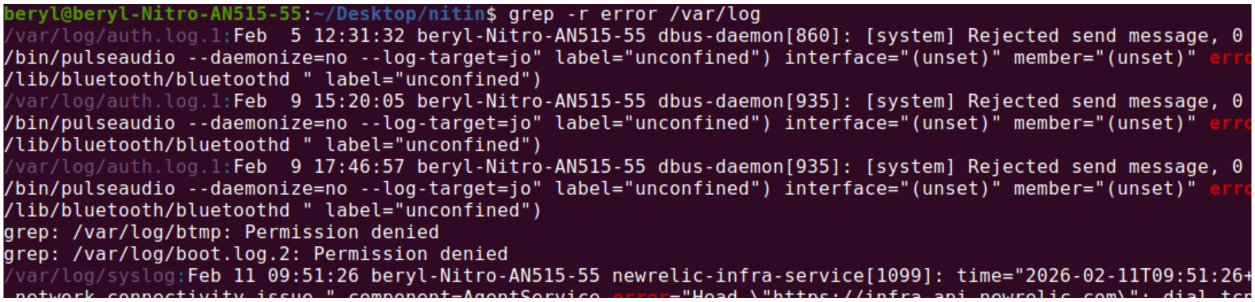
Example: **grep -v root /etc/passwd**



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ grep -v error log.txt
INFO started
ERROR disk full
INFO retry
ERROR timeout
INFO done
ERROR 404 page not found
ERROR 500 server crash
ERROR 404 page not found
```

-r / -R → Searches recursively in directories

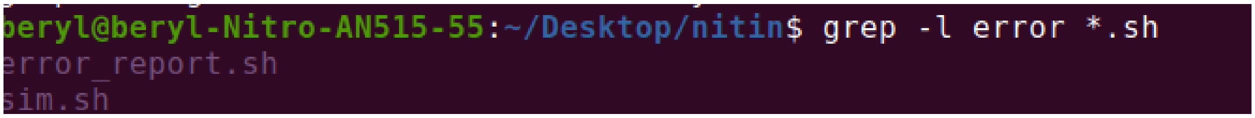
Example: **grep -r error /var/log**



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ grep -r error /var/log
/var/log/auth.log.1:Feb  5 12:31:32 beryl-Nitro-AN515-55 dbus-daemon[860]: [system] Rejected send message, 0
/bin/pulseaudio --daemonize=no --log-target=jo" label="unconfined") interface="(unset)" member="(unset)" erro
/lib/bluetooth/bluetoothd " label="unconfined")
/var/log/auth.log.1:Feb  9 15:20:05 beryl-Nitro-AN515-55 dbus-daemon[935]: [system] Rejected send message, 0
/bin/pulseaudio --daemonize=no --log-target=jo" label="unconfined") interface="(unset)" member="(unset)" erro
/lib/bluetooth/bluetoothd " label="unconfined")
/var/log/auth.log.1:Feb  9 17:46:57 beryl-Nitro-AN515-55 dbus-daemon[935]: [system] Rejected send message, 0
/bin/pulseaudio --daemonize=no --log-target=jo" label="unconfined") interface="(unset)" member="(unset)" erro
/lib/bluetooth/bluetoothd " label="unconfined")
grep: /var/log/btmp: Permission denied
grep: /var/log/boot.log.2: Permission denied
/var/log/syslog:Feb 11 09:51:26 beryl-Nitro-AN515-55 newrelic-infra-service[1099]: time="2026-02-11T09:51:26+
 network connectivity issue " component=AgentService error="Head \"https://infra-api.newrelic.com\": dial tcp
```

-l → Displays only file names containing the match

Example: **grep -l error *.log**



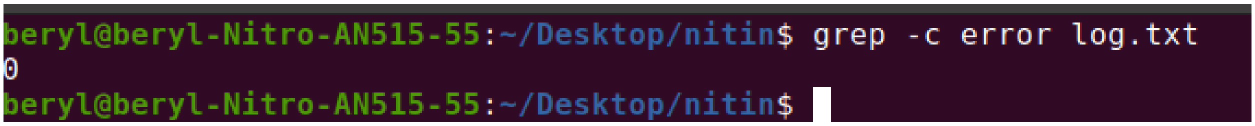
```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ grep -l error *.sh
error_report.sh
sim.sh
```

-w → Matches whole words only

Example: **grep -w root file.txt**

-c → Displays the count of matching lines

Example: grep -c error log.txt



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ grep -c error log.txt
0
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ 
```

--color → Highlights the matched text

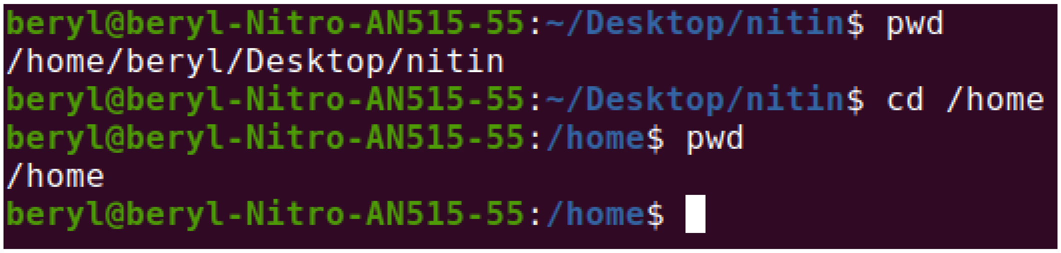
Example: grep --color error log.txt

Command name: cd – Change directory

Purpose : Changes the current working directory.

Syntax: cd [directory]

Example: cd /home/user

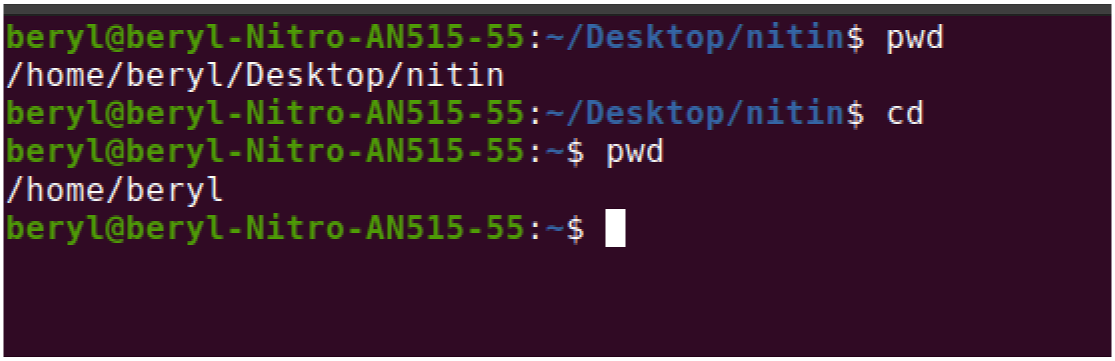


```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ pwd
/home/beryl/Desktop/nitin
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ cd /home
beryl@beryl-Nitro-AN515-55:/home$ pwd
/home
beryl@beryl-Nitro-AN515-55:/home$ 
```

cd Options / Special Forms

cd (no argument) → Changes to the home directory

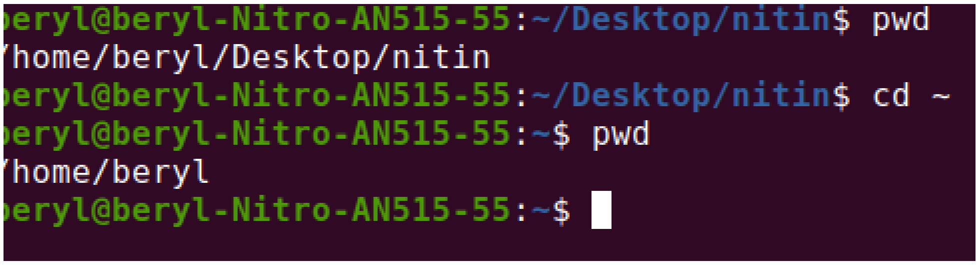
Example: cd



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ pwd
/home/beryl/Desktop/nitin
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ cd
beryl@beryl-Nitro-AN515-55:~$ pwd
/home/beryl
beryl@beryl-Nitro-AN515-55:~$ 
```

cd ~ → Changes to the home directory

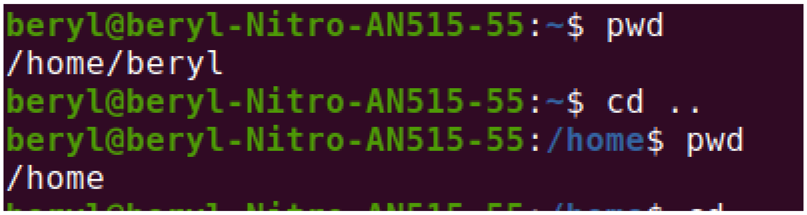
Example: cd ~



```
eryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ pwd
/home/beryl/Desktop/nitin
eryl@beryl-Nitro-AN515-55:~/Desktop/nitin$ cd ~
eryl@beryl-Nitro-AN515-55:~$ pwd
/home/beryl
eryl@beryl-Nitro-AN515-55:~$ 
```

cd .. → Moves to the parent directory

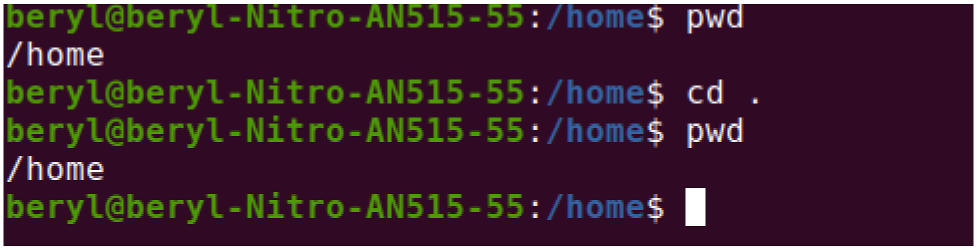
Example: cd ..



```
beryl@beryl-Nitro-AN515-55:~$ pwd
/home/beryl
beryl@beryl-Nitro-AN515-55:~$ cd ..
beryl@beryl-Nitro-AN515-55:/home$ pwd
/home
beryl@beryl-Nitro-AN515-55:/home$ cd
```

cd . → Stays in the current directory

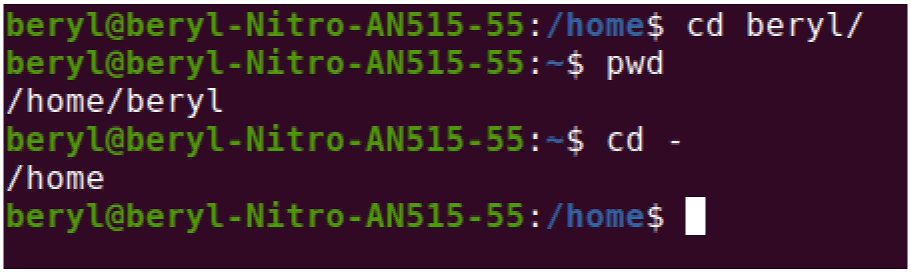
Example: **cd .**



```
beryl@beryl-Nitro-AN515-55:/home$ pwd
/home
beryl@beryl-Nitro-AN515-55:/home$ cd .
beryl@beryl-Nitro-AN515-55:/home$ pwd
/home
beryl@beryl-Nitro-AN515-55:/home$ 
```

cd - → Switches to the previous directory

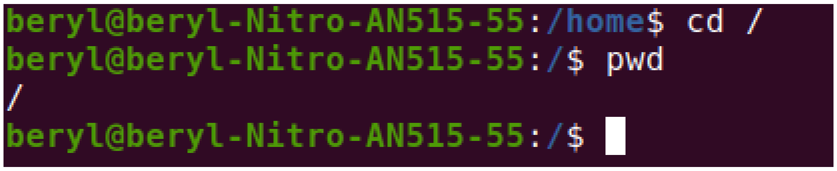
Example: **cd -**



```
beryl@beryl-Nitro-AN515-55:/home$ cd beryl/
beryl@beryl-Nitro-AN515-55:~$ pwd
/home/beryl
beryl@beryl-Nitro-AN515-55:~$ cd -
/home
beryl@beryl-Nitro-AN515-55:/home$ 
```

cd / → Changes to the root directory

Example: **cd /**



```
beryl@beryl-Nitro-AN515-55:/home$ cd /
beryl@beryl-Nitro-AN515-55:/$ pwd
/
beryl@beryl-Nitro-AN515-55:/$ 
```

cd Logical Options

-L → Uses the logical path (symbolic links, default)

Example: `cd -L symlink_dir`

-P → Uses the physical path, resolving symbolic links

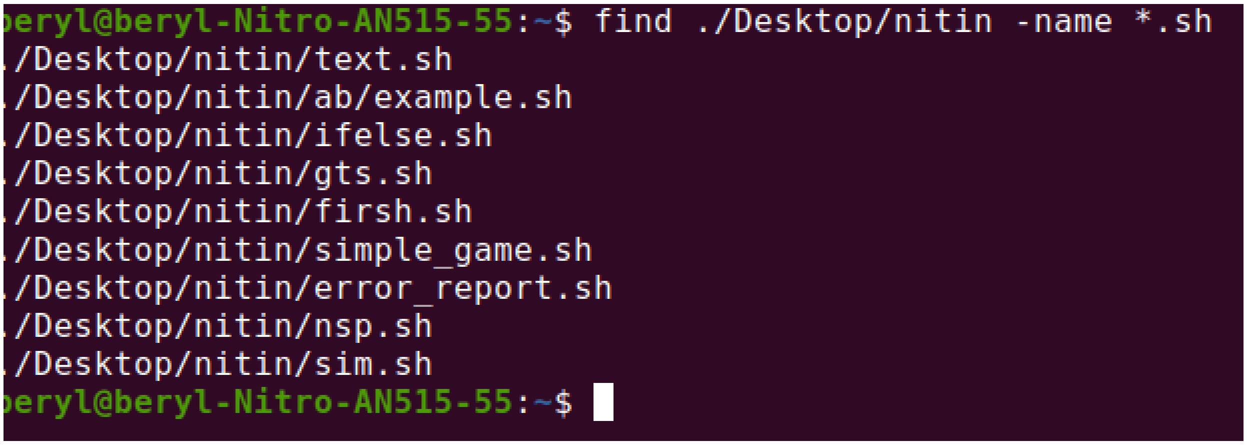
Example: `cd -P symlink_dir`

Command name: find – Search files and directories

Purpose : Searches for files and directories based on name, type, size, time, permissions, and more.

Syntax: `find [path] [options]`

Example: `find /home -name file.txt`



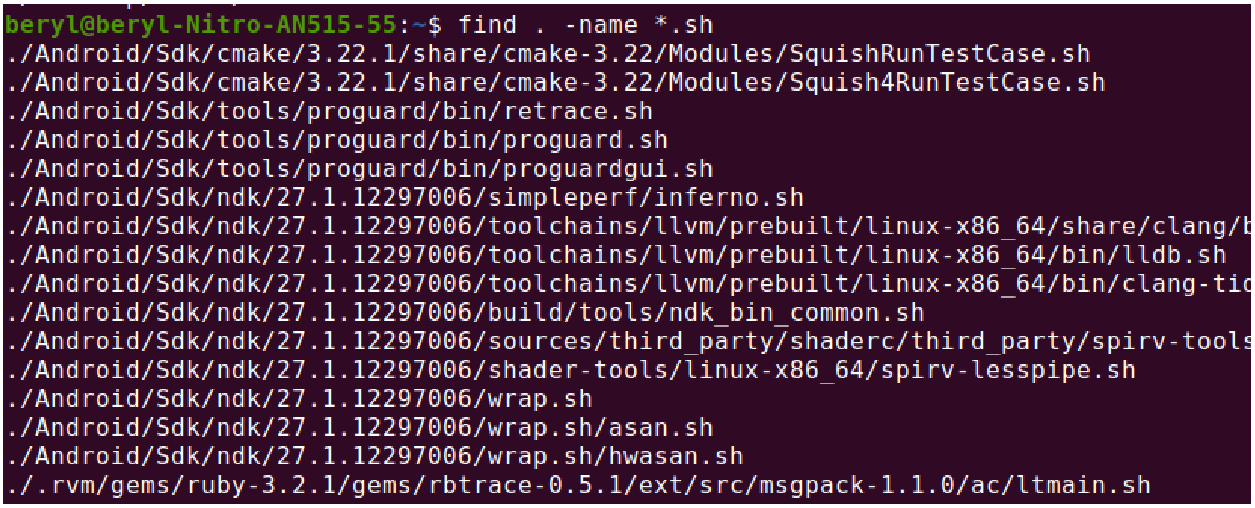
```
beryl@beryl-Nitro-AN515-55:~$ find ./Desktop/nitin -name *.sh
./Desktop/nitin/text.sh
./Desktop/nitin/ab/example.sh
./Desktop/nitin/ifelse.sh
./Desktop/nitin/gts.sh
./Desktop/nitin/firsh.sh
./Desktop/nitin/simple_game.sh
./Desktop/nitin/error_report.sh
./Desktop/nitin/nsp.sh
./Desktop/nitin/sim.sh
beryl@beryl-Nitro-AN515-55:~$
```

find Options

Name & Type Options

-name → Searches by file name (case-sensitive)

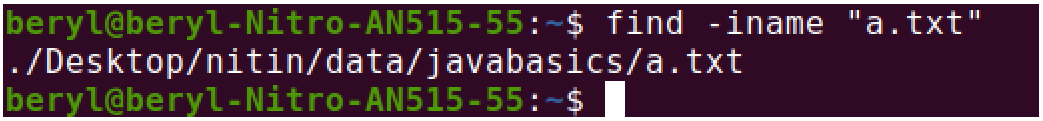
Example: `find . -name "test.txt"`



```
beryl@beryl-Nitro-AN515-55:~$ find . -name *.sh
./Android/Sdk/cmake/3.22.1/share/cmake-3.22/Modules/SquishRunTestCase.sh
./Android/Sdk/cmake/3.22.1/share/cmake-3.22/Modules/Squish4RunTestCase.sh
./Android/Sdk/tools/proguard/bin/retrace.sh
./Android/Sdk/tools/proguard/bin/proguard.sh
./Android/Sdk/tools/proguard/bin/proguardgui.sh
./Android/Sdk/ndk/27.1.12297006/simpleperf/inferno.sh
./Android/Sdk/ndk/27.1.12297006/toolchains/llvm/prebuilt/linux-x86_64/share/clang/b
./Android/Sdk/ndk/27.1.12297006/toolchains/llvm/prebuilt/linux-x86_64/bin/lldb.sh
./Android/Sdk/ndk/27.1.12297006/toolchains/llvm/prebuilt/linux-x86_64/bin/clang-tid
./Android/Sdk/ndk/27.1.12297006/build/tools/ndk_bin_common.sh
./Android/Sdk/ndk/27.1.12297006/sources/third_party/shaderc/third_party/spirv-tools
./Android/Sdk/ndk/27.1.12297006/shader-tools/linux-x86_64/spirv-lesspipe.sh
./Android/Sdk/ndk/27.1.12297006/wrap.sh
./Android/Sdk/ndk/27.1.12297006/wrap.sh/asan.sh
./Android/Sdk/ndk/27.1.12297006/wrap.sh/hwasan.sh
./.rvm/gems/ruby-3.2.1/gems/rbtrace-0.5.1/ext/src/msgpack-1.1.0/ac/ltmain.sh
```

-iname → Searches by name ignoring case

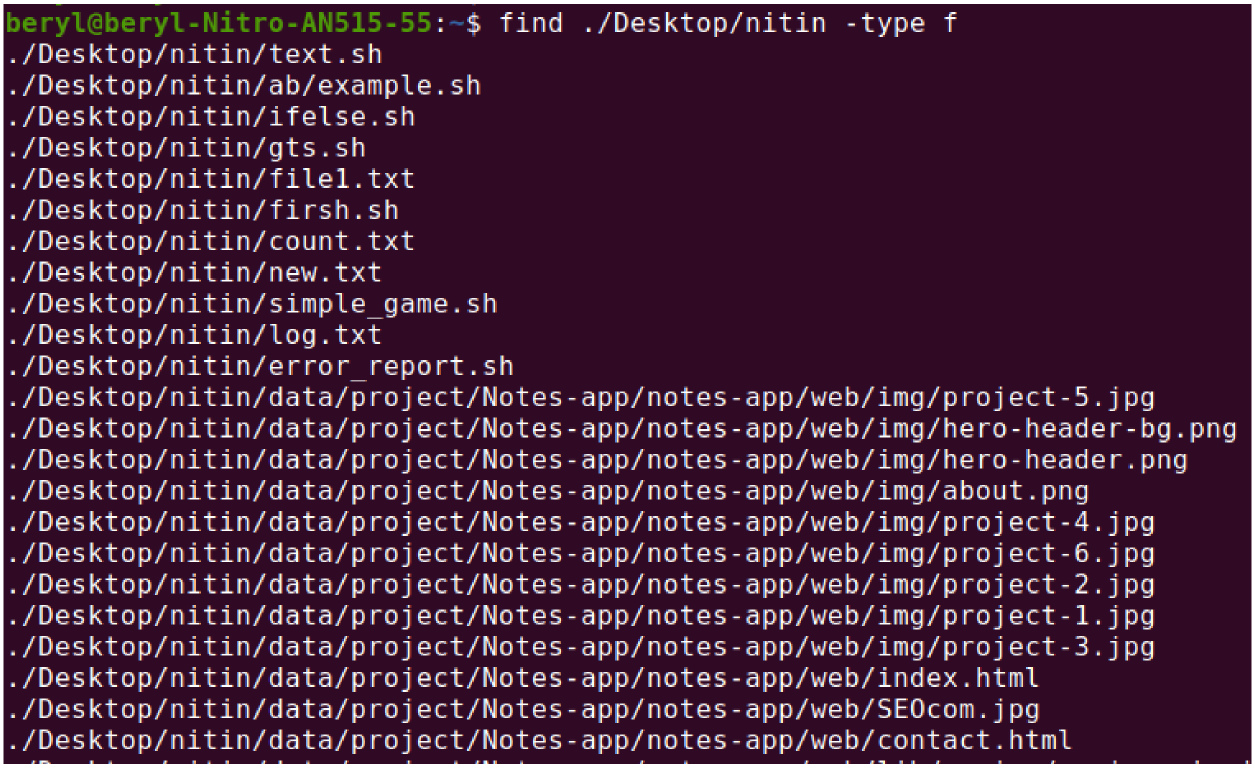
Example: **find . -iname "test.txt"**



```
beryl@beryl-Nitro-AN515-55:~$ find -iname "a.txt"
./Desktop/nitin/data/javabasics/a.txt
beryl@beryl-Nitro-AN515-55:~$ 
```

-type f → Searches for regular files

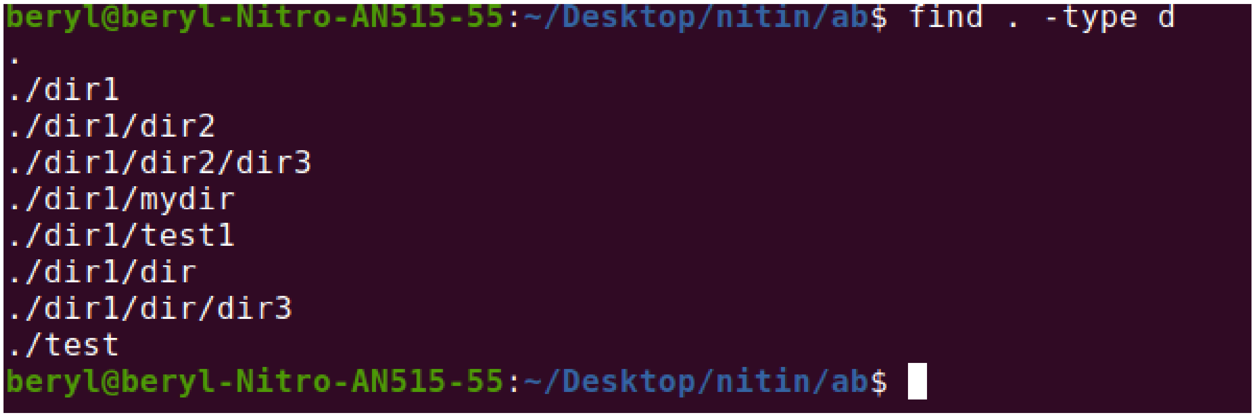
Example: **find . -type f**



```
beryl@beryl-Nitro-AN515-55:~$ find ./Desktop/nitin -type f
./Desktop/nitin/text.sh
./Desktop/nitin/ab/example.sh
./Desktop/nitin/ifelse.sh
./Desktop/nitin/gts.sh
./Desktop/nitin/file1.txt
./Desktop/nitin/firsh.sh
./Desktop/nitin/count.txt
./Desktop/nitin/new.txt
./Desktop/nitin/simple_game.sh
./Desktop/nitin/log.txt
./Desktop/nitin/error_report.sh
./Desktop/nitin/data/project/Notes-app/notes-app/web/img/project-5.jpg
./Desktop/nitin/data/project/Notes-app/notes-app/web/img/hero-header-bg.png
./Desktop/nitin/data/project/Notes-app/notes-app/web/img/hero-header.png
./Desktop/nitin/data/project/Notes-app/notes-app/web/img/about.png
./Desktop/nitin/data/project/Notes-app/notes-app/web/img/project-4.jpg
./Desktop/nitin/data/project/Notes-app/notes-app/web/img/project-6.jpg
./Desktop/nitin/data/project/Notes-app/notes-app/web/img/project-2.jpg
./Desktop/nitin/data/project/Notes-app/notes-app/web/img/project-1.jpg
./Desktop/nitin/data/project/Notes-app/notes-app/web/img/project-3.jpg
./Desktop/nitin/data/project/Notes-app/notes-app/web/index.html
./Desktop/nitin/data/project/Notes-app/notes-app/web/SEOcom.jpg
./Desktop/nitin/data/project/Notes-app/notes-app/web/contact.html
```

-type d → Searches for directories

Example: **find . -type d**



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ find . -type d
.
./dir1
./dir1/dir2
./dir1/dir2/dir3
./dir1/mydir
./dir1/test1
./dir1/dir
./dir1/dir/dir3
./test
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ 
```

`-type l` → Searches for symbolic links

Example: `find . -type l`

Size Options

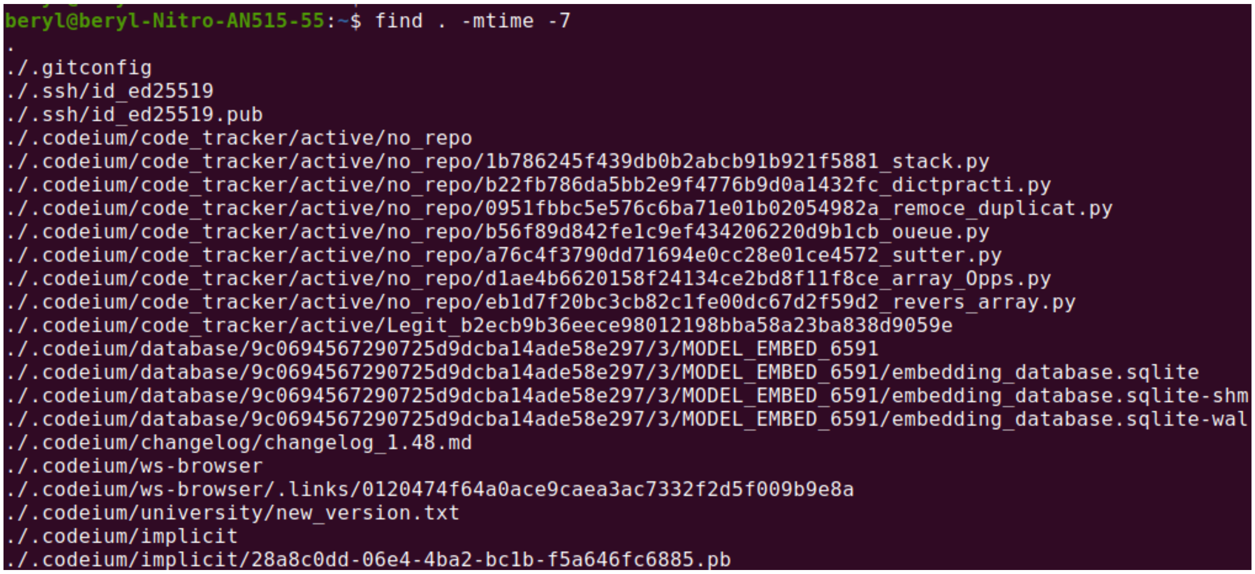
`-size` → Searches by file size

Example:
```
find . -size +10M  # larger than 10MB
find . -size -1M   # smaller than 1MB
```

Time Options

-mtime → Modified time (days)

Example: `find . -mtime -7     # last 7 days`



```
beryl@beryl-Nitro-AN515-55:~$ find . -mtime -7
.
./.gitconfig
./.ssh/id_ed25519
./.ssh/id_ed25519.pub
./.codeium/code_tracker/active/no_repo
./.codeium/code_tracker/active/no_repo/1b786245f439db0b2abcb91b921f5881_stack.py
./.codeium/code_tracker/active/no_repo/b22fb786da5bb2e9f4776b9d0a1432fc_dictpracti.py
./.codeium/code_tracker/active/no_repo/0951fbbc5e576c6ba71e01b02054982a_remoce_duplicat.py
./.codeium/code_tracker/active/no_repo/b56f89d842fe1c9ef434206220d9b1cb_oueue.py
./.codeium/code_tracker/active/no_repo/a76c4f3790dd71694e0cc28e01ce4572_sutter.py
./.codeium/code_tracker/active/no_repo/d1ae4b6620158f24134ce2bd8f11f8ce_array_Opps.py
./.codeium/code_tracker/active/no_repo/eb1d7f20bc3cb82c1fe00dc67d2f59d2_revers_array.py
./.codeium/code_tracker/active/Legit_b2ecb9b36eece98012198bba58a23ba838d9059e
./.codeium/database/9c0694567290725d9dcba14ade58e297/3/MODEL_EMBED_6591
./.codeium/database/9c0694567290725d9dcba14ade58e297/3/MODEL_EMBED_6591/embedding_database.sqlite
./.codeium/database/9c0694567290725d9dcba14ade58e297/3/MODEL_EMBED_6591/embedding_database.sqlite-shm
./.codeium/database/9c0694567290725d9dcba14ade58e297/3/MODEL_EMBED_6591/embedding_database.sqlite-wal
./.codeium/changelog/changelog_1.48.md
./.codeium/ws-browser
./.codeium/ws-browser/.links/0120474f64a0ace9caea3ac7332f2d5f009b9e8a
./.codeium/university/new_version.txt
./.codeium/implicit
./.codeium/implicit/28a8c0dd-06e4-4ba2-bc1b-f5a646fc6885.pb
```

-atime → Access time (days)

Example: `find . -atime 1`

-ctime → Change time (days)

Example: `find . -ctime -2`

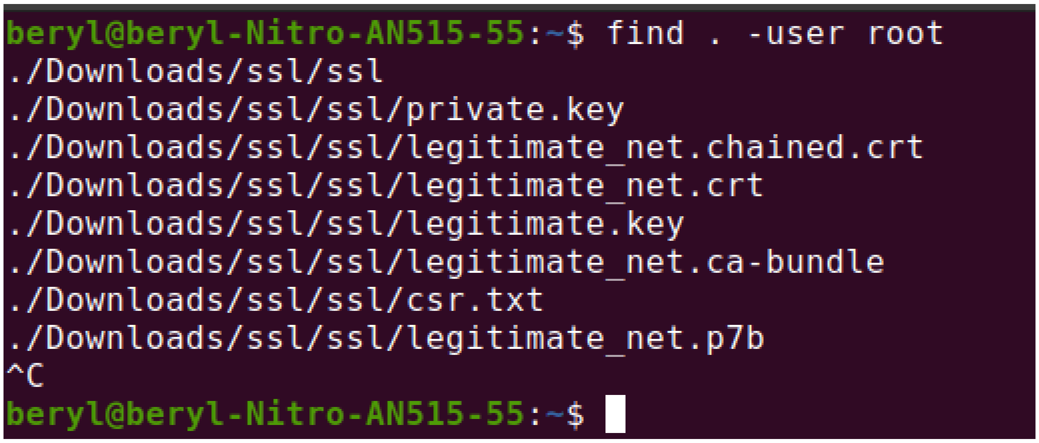
Permission & Ownership

-perm → Search by permissions

Example: `find . -perm 755`

-user → Search files owned by user

Example: `find . -user root`



```
beryl@beryl-Nitro-AN515-55:~$ find . -user root
./Downloads/ssl/ssl
./Downloads/ssl/ssl/private.key
./Downloads/ssl/ssl/legitimate_net.chained.crt
./Downloads/ssl/ssl/legitimate_net.crt
./Downloads/ssl/ssl/legitimate.key
./Downloads/ssl/ssl/legitimate_net.ca-bundle
./Downloads/ssl/ssl/csr.txt
./Downloads/ssl/ssl/legitimate_net.p7b
^C
beryl@beryl-Nitro-AN515-55:~$
```

-group → Search files owned by group

Example: `find . -group admin`

Depth & Path Control

-maxdepth →Limit directory traversal depth

Example:

```
find . -maxdepth 2
```

-mindepth →Skip top levels

Example:

```
find . -mindepth 1
```

-path → Search by path pattern

Example:

```
find . -path "./src/*"
```

Action Options

-exec → Execute a command on each result

Example:

```
find . -name "*.log" -exec rm {} \;
```

-delete → Delete matched files (dangerous)

Example:

```
find . -name "*.tmp" -delete
```

`-print` → Print matched paths (default)

Example: `find . -print`

Logical Operators

- `-and` / `-a` → Logical AND
- `-or` / `-o` → Logical OR
- `!` → NOT condition

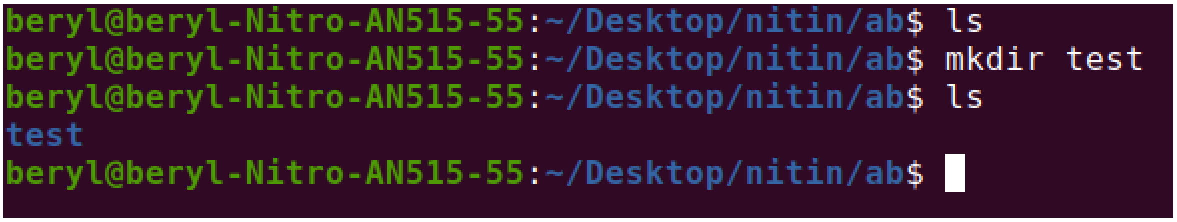
Example:
```
find . -type f -name "*.txt" -and
-size +1
```

Command name: `mkdir` – **Create directories**

Purpose : **Creates one or more directories.**

Syntax: `mkdir [options] directory_name`

Example: `mkdir test`

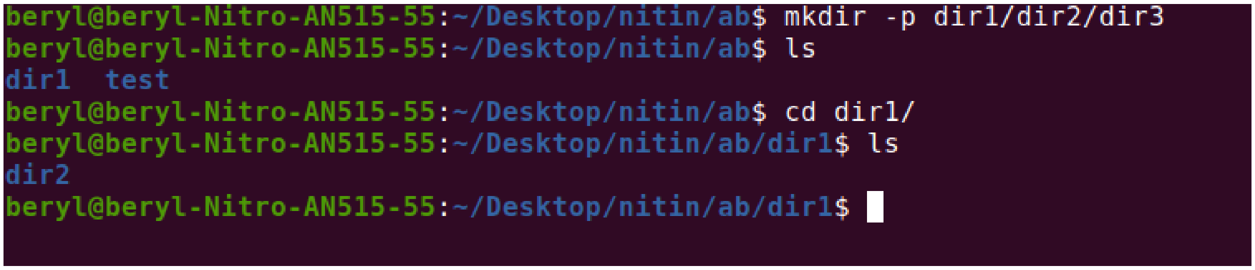


```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ ls
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ mkdir test
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ ls
test
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ 
```

mkdir Options

-p → Creates parent directories as needed and does not show an error if they already exist

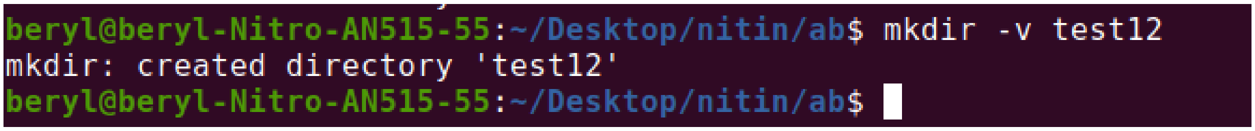
Example: `mkdir -p dir1/dir2/dir3`



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ mkdir -p dir1/dir2/dir3
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ ls
dir1  test
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ cd dir1/
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab/dir1$ ls
dir2
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab/dir1$ 
```

-v → Displays a message for each directory created (verbose output)

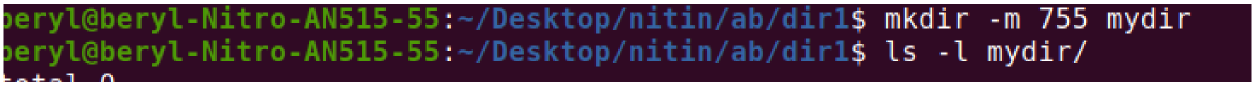
Example: `mkdir -v test`



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ mkdir -v test12
mkdir: created directory 'test12'
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab$ 
```

-m MODE → Sets permissions (mode) for the directory at creation time

Example: `mkdir -m 755 mydir`



```
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab/dir1$ mkdir -m 755 mydir
beryl@beryl-Nitro-AN515-55:~/Desktop/nitin/ab/dir1$ ls -l mydir/
total 0
```

--parents →Same as -p

Example: `mkdir --parents a/b/c`

--mode=MODE → Same as -m

Example: `mkdir --mode=700 secure_dir`

--verbose → Same as -v

Example: `mkdir —-verbose test`

Command name: `cp` **— Copy files and directories**

Purpose : **Copies files or directories from one location to another.**

Syntax: `cp [options] source destination`

Example: `cp file1.txt file2.txt`

`cp` Options

□ Basic & Common Options

`-r` **/** `-R` → Copies directories recursively

Example: `cp -r dir1 dir2`

`-i` → Prompts before overwriting files

Example: `cp -i a.txt b.txt`

`-f` → Forces copy without confirmation

Example: `cp -f a.txt b.txt`

`-v` → Displays verbose output

Example: `cp -v a.txt b.txt`

`-a` → Archive mode (preserves permissions, **ownership, timestamps, links)**

Example: `cp -a /etc /backup`

`-p` → Preserves mode, ownership, timestamps

Example: `cp -p file1 file2`

`-u` → Copies only if source is newer than destination

Example: `cp -u a.txt b.txt`

□ Link & Attribute Options

`-l` → Creates hard links instead of copying

Example: `cp -l file1 file2`

`-s` → Creates symbolic links instead of copying

Example: `cp -s file1 file2`

`-d` → Preserves links instead of copying link targets

Example: `cp -d symlink target`

□ Directory & Structure Options

`-t DIR` → Copies files into a target directory

Example: `cp -t /backup file1 file2`

`--parents` → Preserves directory structure

Example: `cp --parents dir1/file.txt /backup`

□ **Backup & Safety Options**

`--backup` → Creates a backup before overwriting

Example: `cp --backup a.txt b.txt`

`--no-preserve=links` → Does not preserve symbolic links

Example: `cp --no-preserve=links a b`

Command name: `rmdir` **— Remove empty directories**

Purpose : **Deletes empty directories only.**

Syntax: `rmdir [options] directory_name`

Example: `rmdir test`

`rmdir` Options

`-p` → Removes the directory and its parent **directories if they become empty**

Example: `rmdir -p dir1/dir2/dir3`

`-v` → Displays a message for each directory **removed (verbose output)**

Example: `rmdir -v emptydir`

`--ignore-fail-on-non-empty` → Does not show an error if **the directory is not empty**

Example: `rmdir --ignore-fail-on-non-empty dir`

Command name: `cat` **– Concatenate and display file content**

Purpose : **Displays the contents of a file or combines multiple files and shows the output.**

Syntax: `cat [options] file`

Example: `cat file.txt`

`cat` Options

`-n` → Displays line numbers for all lines

Example: `cat -n file.txt`

`-b` → Displays line numbers for non-empty lines only

Example: `cat -b file.txt`

`-s` → Suppresses repeated empty lines (multiple blank lines become one)

Example: `cat -s file.txt`

`-E` → Displays `$` at the end of each line

Example: `cat -E file.txt`

-T → Displays TAB characters as ^I

Example: `cat -T file.txt`

-A → Displays all non-printing characters (same as -vET)

Example: `cat -A file.txt`

-v → Displays non-printing characters (except TAB and newline)

Example: `cat -v file.txt`

Command name: less – View file content page by page

Purpose : Displays file content one screen at a time, allowing scrolling and searching.

Syntax: `less [options] file`

Example: `less logfile.txt`

`less` Options

`-N` → Shows line numbers

Example: `less -N file.txt`

`-S` → Disables line wrapping (long lines stay on one line)

Example: `less -S file.txt`

`-X` → Keeps content on screen after exiting

Example: `less -X file.txt`

`-F` → Automatically exits if file fits on one screen

Example: `less -F file.txt`

-M → Shows detailed prompt (percentage, lines)

Example: `less -M file.txt`

-i → Case-insensitive search (unless uppercase used)

Example: `less -i file.txt`

Command name: head – Display beginning of a file

Purpose : Displays the first few lines of a file (default is 10 lines).

Syntax: `head [options] file`

Example: `head file.txt`

head Options

-n N → Displays the first N lines

Example: `head -n 5 file.txt`

-c N → Displays the first N bytes

Example: `head -c 20 file.txt`

-q → Suppresses headers when multiple files are used

Example: `head -q file1.txt file2.txt`

-v → Always shows file headers

Example: `head -v file1.txt file2.tx`

Command name: tail – Display end of a file

Purpose : Displays the last few lines of a file (default is 10 lines).

Syntax: `tail [options] file`

Example: `tail file.txt`

`tail` Options

`-n N` → Displays the last n lines

Example: `tail -n 5 file.txt`

`-c N` → Displays the last n bytes

Example: `tail -c 50 file.txt`

`-f` → Follows the file and shows new lines as they are added (used for logs)

Example: `tail -f /var/log/syslog`

`-F` → Same as `-f` but reopens the file if it is recreated

Example: `tail -F logfile.log`

`-q` → Suppresses headers when viewing multiple files

Example: `tail -q file1.txt file2.txt`

`-v` → Always shows file headers

Example: `tail -v file1.txt file2.txt`

Command name: `ps` – Display process status

Purpose : Displays information about currently running processes.

Syntax: `ps [options]`

Example: `ps aux`

ps Options

a → Shows processes for all users

Example: ps a

u → Displays output in user-oriented format

Example: ps u

x → Shows processes without a controlling terminal

Example: ps x

aux → Shows all running processes (most commonly used)

Example: ps aux

-e → Displays all processes

Example: ps -e

-f → Displays full-format listing

Example: ps -f

-u USER → Shows processes for a specific user

Example: ps -u root

-p PID → Shows a process by process ID

Example: ps -p 1234

--forest → Displays process hierarchy in tree format

Example: ps -ef --forest

Command name: `top` – Real-time process monitoring

Purpose : Displays real-time information about running processes and system resource usage.

Syntax: `top [options]`

Example: `top`

`top` Options

`-u USER` → Shows processes for a specific user

Example: `top -u root`

`-p PID` → Monitors a specific process ID

Example: `top -p 1234`

`-n N` → Updates display N times and then Exits

Example: `top -n 5`

`-d SECONDS` →Sets refresh delay time

Example: `top -d 2`

`-b` → Runs in batch mode (useful for logging)

Example: `top -b -n 1`

`-H` → Shows threads instead of processes

Example: `top -H`

Command name: `df` – **Display disk space usage**

Purpose : Displays disk space usage of mounted filesystems.

Syntax: `df [options]`

Example: `df -h`

`df` Options

`-h` → Shows sizes in human-readable format (KB, MB, GB)

Example: `df -h`

`-a` → Shows all filesystems, including dummy ones

Example: `df -a`

-T → Displays filesystem type

Example: `df -T`

-i → Shows inode usage instead of block usage

Example: `df -i`

--total → Displays a total summary at the end

Example: `df --total`

Command name: du – Display directory space usage

Purpose : Displays the disk space used by files and directories.

Syntax: `du [options] directory`

Example: `du -h /home`

`du` Options

`-h` → Shows sizes in human-readable format (**KB, MB, GB**)

Example: `du -h folder`

`-s` → Displays only the total size **(summary)**

Example: `du -s folder`

`-sh` → Displays total size in human-readable **format (most used)**

Example: `du -sh folder`

`-a` → Shows disk usage for all files, not just directories

Example: du -a folder

-c → Displays a grand total at the end

Example: du -ch folder

--max-depth=N → Limits output to N directory levels

Example: du -h --max-depth=1

Command name: kill – Terminate a process

Purpose : Sends a signal to terminate or control a running process using its PID.

Syntax: `kill [options] PID`

Example: `kill 1234`

kill Options / Signals

-9 → Forcefully terminates a process (**SIGKILL**)

Example: `kill -9 1234`

-15 → Gracefully terminates a process (**SIGTERM – default**)

Example: `kill -15 1234`

-l → Lists all available signals

Example: `kill -l`

`-SIGTERM` → Sends a specific signal by name

Example: `kill -SIGTERM 1234`

`-SIGKILL` → Immediately kills the process

Example: `kill -SIGKILL 1234`

Command name: `apt` **– Package management**

Purpose : **Used to install, update, upgrade, and remove software packages on Debian-based Linux systems.**

Syntax: `apt [command] [package]`

Example: `apt install vim`

apt Commands / Options

update → Updates the package list from repositories

Example: apt update

upgrade → Upgrades installed packages to latest versions

Example: apt upgrade

full-upgrade →Upgrades packages and handles dependencies

Example: apt full-upgrade

install → Installs a new package

Example: apt install nginx

remove → Removes a package (keeps config files)

Example: `apt remove nginx`

purge → Removes a package including configuration files

Example: `apt purge nginx`

autoremove →Removes unused dependencies

Example: `apt autoremove`

search → Searches for a package

Example: `apt search python`

show → Displays package details

Example: `apt show nginx`

list --installed → Lists installed packages

Example: apt list --installed

Command name: sudo – Execute command as superuser

Purpose : Allows a permitted user to run commands with superuser (root) privileges.

Syntax: sudo [options] command

Example: sudo apt update

sudo Options

-u USER → Runs command as a specific user (default is root)

Example: sudo -u user command

-i → Starts an interactive root shell

Example: `sudo -i`

-s → Runs a shell with root privileges

Example: `sudo -s`

-k → Invalidates cached password (asks again next time)

Example: `sudo -k`

-v → Refreshes cached credentials without running a command

Example: `sudo -v`

-l → Lists commands the user is allowed to run

Example: `sudo -l`

-b → Runs command in the background

Example: sudo -b command

-E → Preserves the user environment variables

Example: sudo -E command

-H → Sets HOME variable to target user's home

Example: sudo -H command

-S → Reads password from standard input

Example: echo password | sudo -S command

Command name awk

Purpose: text processing & pattern scanning tool

Syntax:

```
wk 'pattern { action }' file
```

pattern → condition (optional)

action → what to do

file → input file

Example:

```
awk '{print}' file.txt
```

Optinds

-F fs Specify field separator.

Example:

```
awk -F ":" '{print $1}' /etc/passwd
```

-v var=value Pass variable from shell.

Example:	`awk -v num=10 '{print $1 + num}'file.txt`
`-f programfile`	Use awk script file.
Example:	`awk -f script.awk file.txt`