Name - Shaan Alam

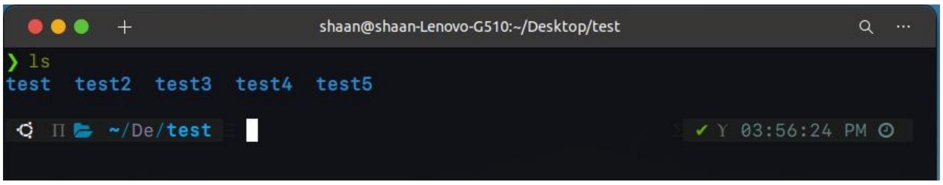
Roll no - 20/63027

Subject - Operating System

Sem – III

ls

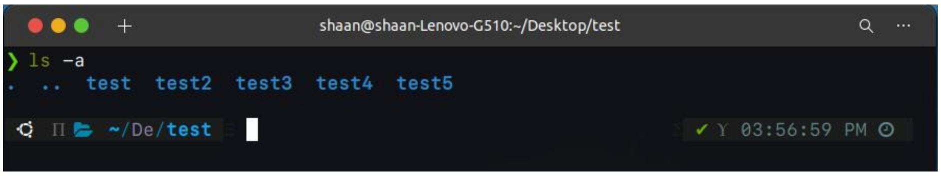
This comand is used to list the files in a directory



```
❯ ls
test  test2  test3  test4  test5

  ~/De/test                                   ✔  03:56:24 PM
```

ls -a

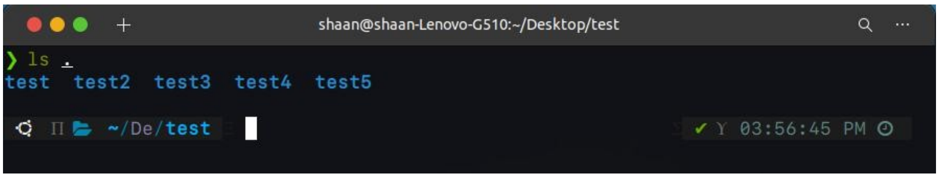
This command is used to list all the files



```
❯ ls -a
.  ..  test  test2  test3  test4  test5

  ~/De/test                                   ✔  03:56:59 PM
```

ls .

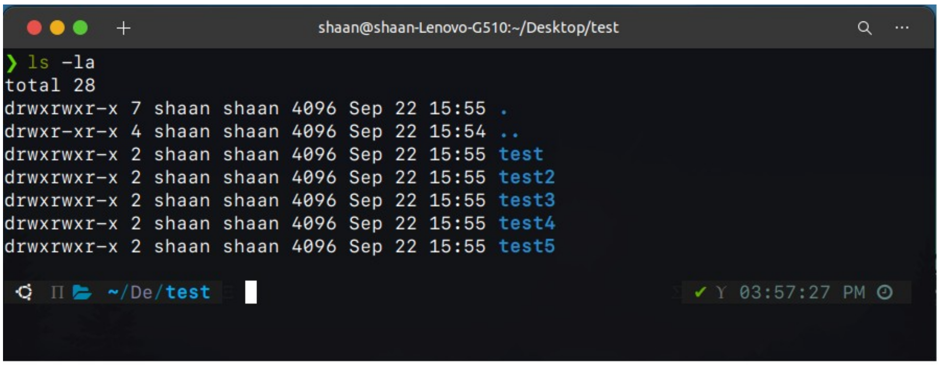
This command is used to list all the files in the current working directory



```
❯ ls .
test  test2  test3  test4  test5

  ~/De/test                                   ✔  03:56:45 PM
```

ls -la

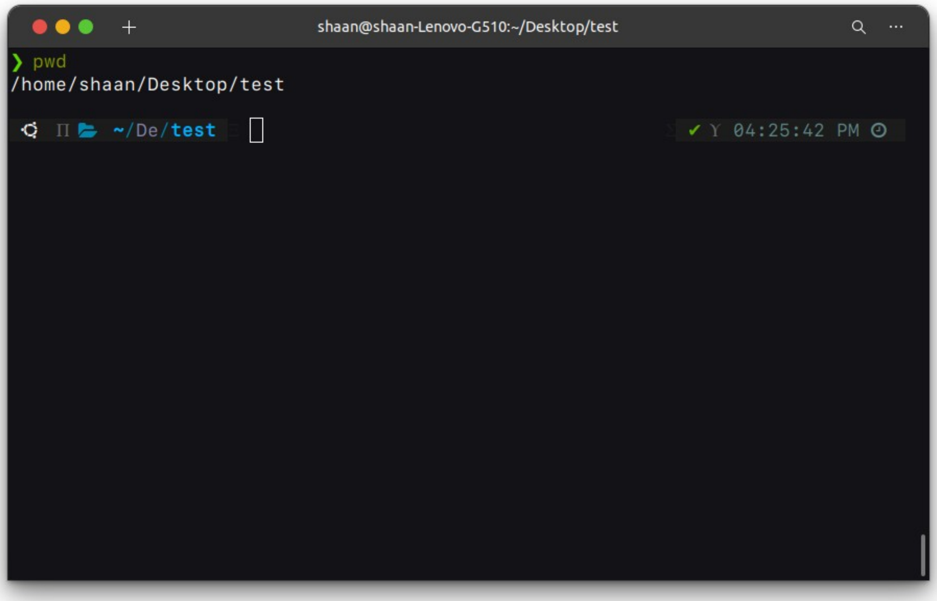
This command is used to display the long listing directories



```
❯ ls -la
total 28
drwxrwxr-x 7 shaan shaan 4096 Sep 22 15:55 .
drwxr-xr-x 4 shaan shaan 4096 Sep 22 15:54 ..
drwxrwxr-x 2 shaan shaan 4096 Sep 22 15:55 test
drwxrwxr-x 2 shaan shaan 4096 Sep 22 15:55 test2
drwxrwxr-x 2 shaan shaan 4096 Sep 22 15:55 test3
drwxrwxr-x 2 shaan shaan 4096 Sep 22 15:55 test4
drwxrwxr-x 2 shaan shaan 4096 Sep 22 15:55 test5

  ~/De/test                                   ✔  03:57:27 PM
```

pwd

pwd stands for **print working directory**. When invoked the command prints the complete path of the current working directory



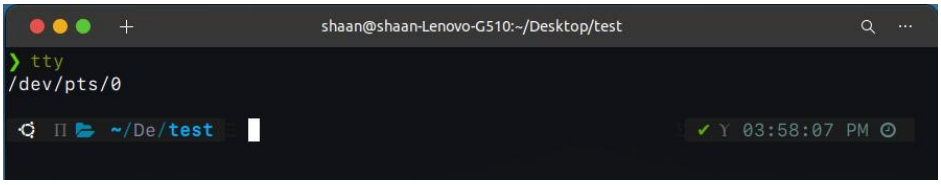
```
❯ pwd
/home/shaan/Desktop/test
```

tty

tty stands for **Teletype** command. **tty** is a command to print the file name of the terminal connected to standard input.

tty

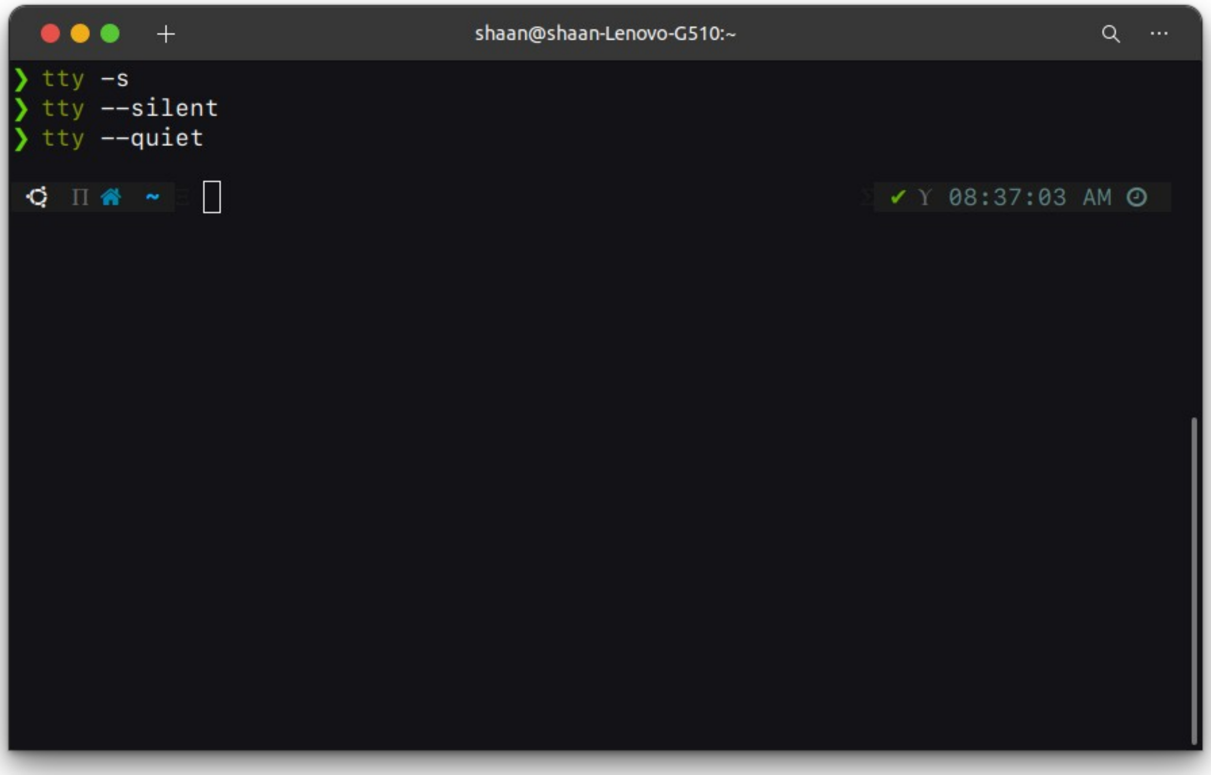
This command prints the filename of the terminal connected to standard input teletype.



```
❯ tty
/dev/pts/0
```

tty -s, tty --silent, tty --quiet

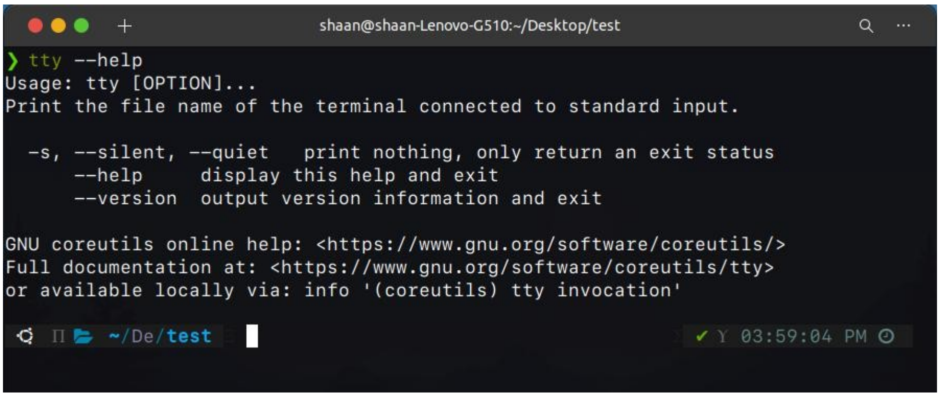
These command will print nothing but return only exit status.



```
❯ tty -s
❯ tty --silent
❯ tty --quiet

  ~                                          ✔ 08:37:03 AM
```

tty --help

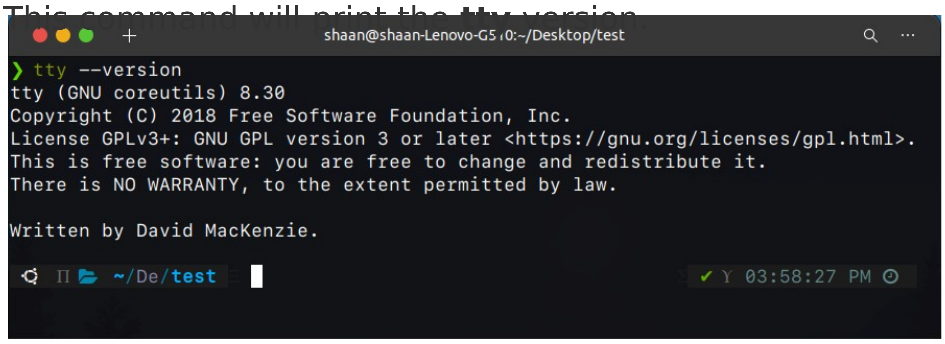
This command will print the help message for **tty** command.



```
❯ tty --help
Usage: tty [OPTION]...
Print the file name of the terminal connected to standard input.

  -s, --silent, --quiet   print nothing, only return an exit status
      --help     display this help and exit
      --version  output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/tty>
or available locally via: info '(coreutils) tty invocation'
  ~/De/test                                  ✔ 03:59:04 PM
```

tty --version

This command will print the **tty** version.



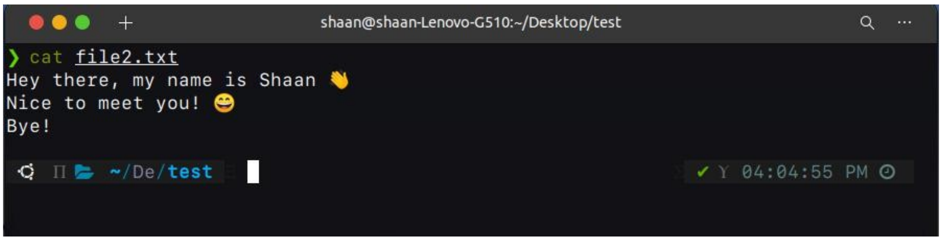
```
❯ tty --version
tty (GNU coreutils) 8.30
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by David MacKenzie.
  ~/De/test                                  ✔ 03:58:27 PM
```

cat

cat command reads the data from the file and and gives their content as output.

cat file_name

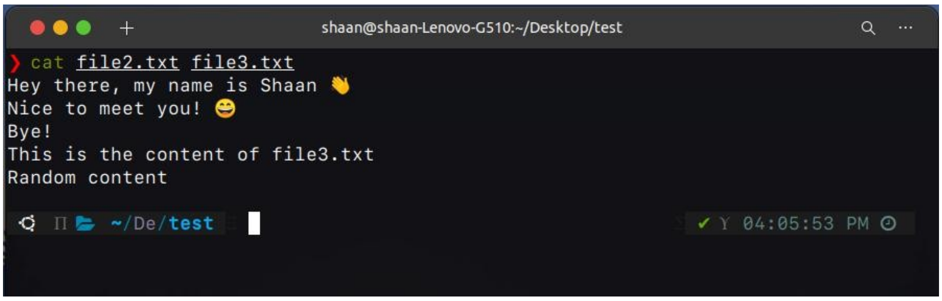
This command will print the content of a single file in the terminal.



```
❯ cat file2.txt
Hey there, my name is Shaan 👋
Nice to meet you! 😄
Bye!

 ~/De/test                                    ✔ 04:04:55 PM
```

cat file_name_1 file_name_2

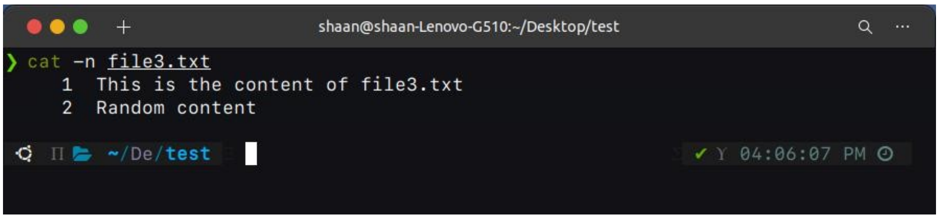
This command will print the content of multiple files in the terminal



```
❯ cat file2.txt file3.txt
Hey there, my name is Shaan 👋
Nice to meet you! 😄
Bye!
This is the content of file3.txt
Random content

 ~/De/test                                    ✔ 04:05:53 PM
```

cat -n file_name

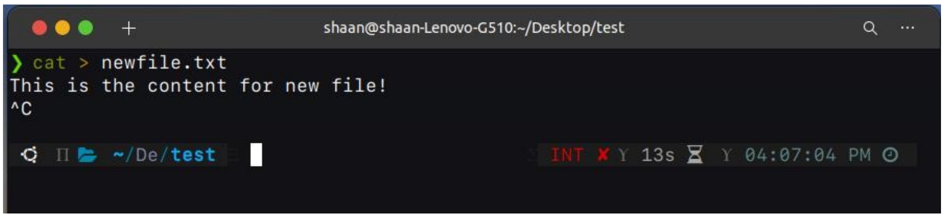
This command will print the content of a file along with line numbers.



```
❯ cat -n file3.txt
     1  This is the content of file3.txt
     2  Random content

 ~/De/test                                    ✔ 04:06:07 PM
```

cat > file_name

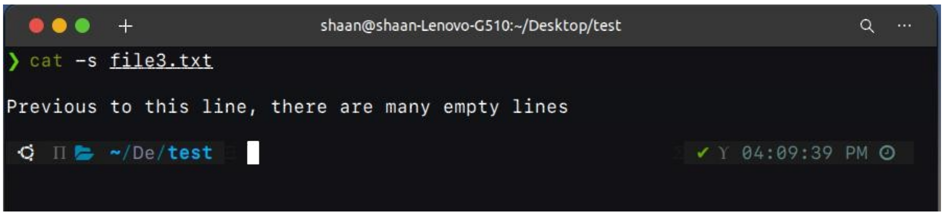
This command will create a new file.



```
❯ cat > newfile.txt
This is the content for new file!
^C
  ~/De/test                                       INT ✘  13s ⌛  04:07:04 PM
```

cat -s file_name

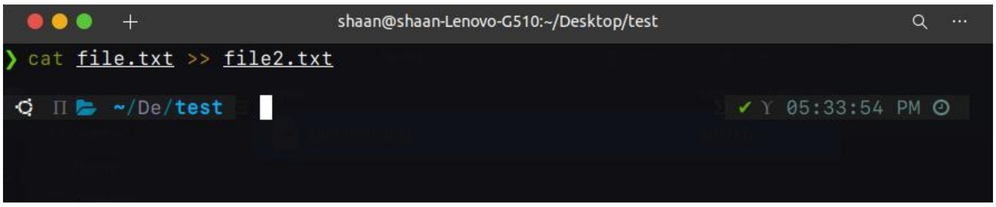
This command will suppress repeated empty lines in the output



```
❯ cat -s file3.txt

Previous to this line, there are many empty lines
  ~/De/test                                              ✔  04:09:39 PM
```

cat file1 >> file2

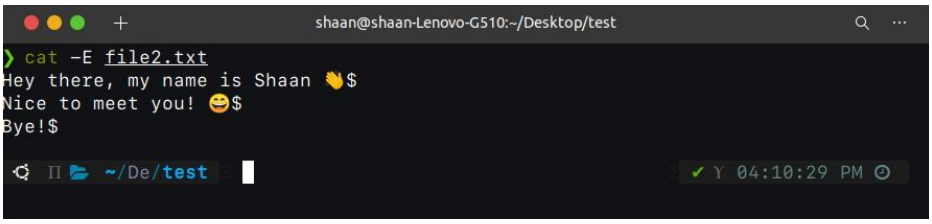
This command will append the content of one file to another



```
❯ cat file.txt >> file2.txt
  ~/De/test                                              ✔  05:33:54 PM
```

cat -E file_name

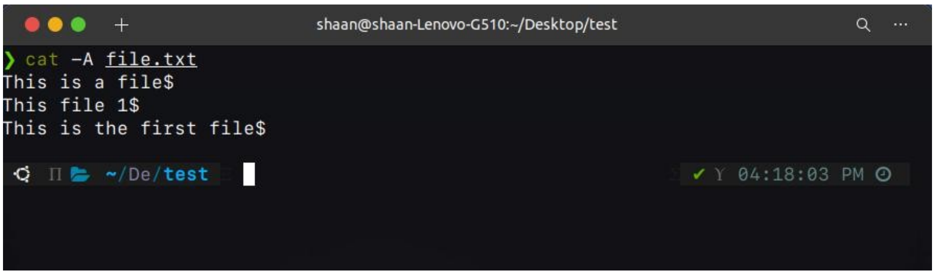
This command will highlight the end of line



```
❯ cat -E file2.txt
Hey there, my name is Shaan 👋$
Nice to meet you! 😄$
Bye!$
  ~/De/test                                              ✔  04:10:29 PM
```

cat -A file_name

Instead of using -vET command, we can use -A command.

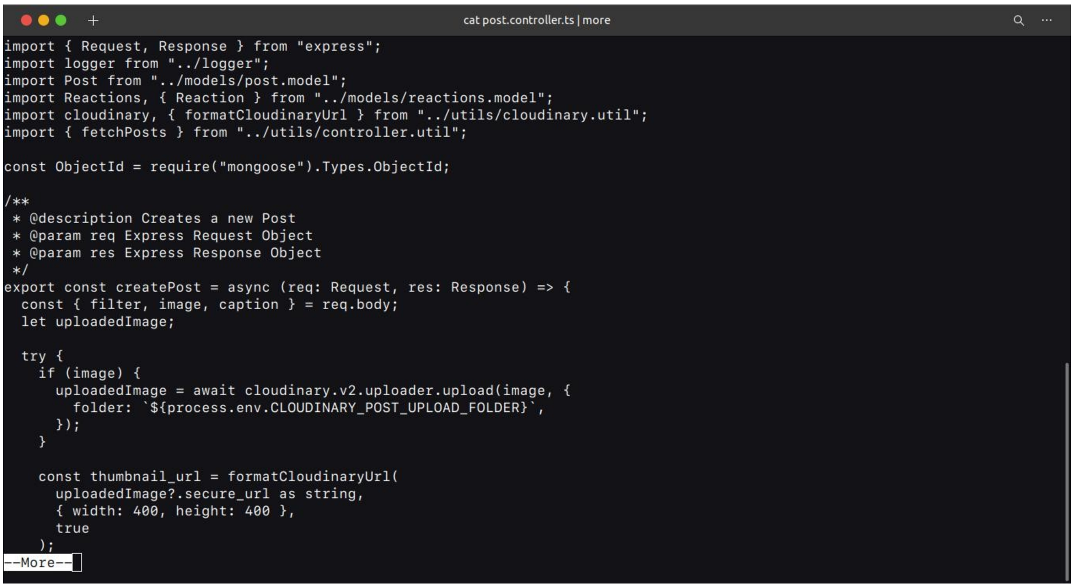


```
❯ cat -A file.txt
This is a file$
This file 1$
This is the first file$

 ~/De/test                                   ✔  04:18:03 PM
```

cat file_name | more

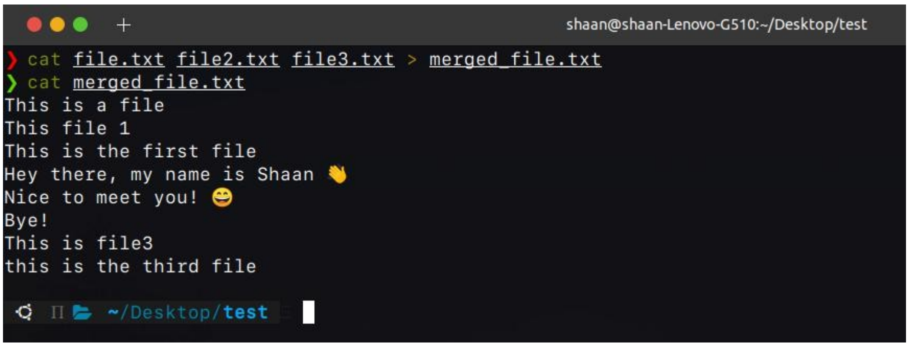
If the file has a lot of content and it can't fit in the terminal window, you can use | more parameter along with the cat command.

If the file has a lot of content and it can't fit in the terminal window, you can use | more parameter along with the cat command.



```
import { Request, Response } from "express";
import logger from "../logger";
import Post from "../models/post.model";
import Reactions, { Reaction } from "../models/reactions.model";
import cloudinary, { formatCloudinaryUrl } from "../utils/cloudinary.util";
import { fetchPosts } from "../utils/controller.util";

const ObjectId = require("mongoose").Types.ObjectId;

/**
 * @description Creates a new Post
 * @param req Express Request Object
 * @param res Express Response Object
 */
export const createPost = async (req: Request, res: Response) => {
  const { filter, image, caption } = req.body;
  let uploadedImage;

  try {
    if (image) {
      uploadedImage = await cloudinary.v2.uploader.upload(image, {
        folder: `${process.env.CLOUDINARY_POST_UPLOAD_FOLDER}`,
      });
    }

    const thumbnail_url = formatCloudinaryUrl(
      uploadedImage?.secure_url as string,
      { width: 400, height: 400 },
      true
    );
--More--
```

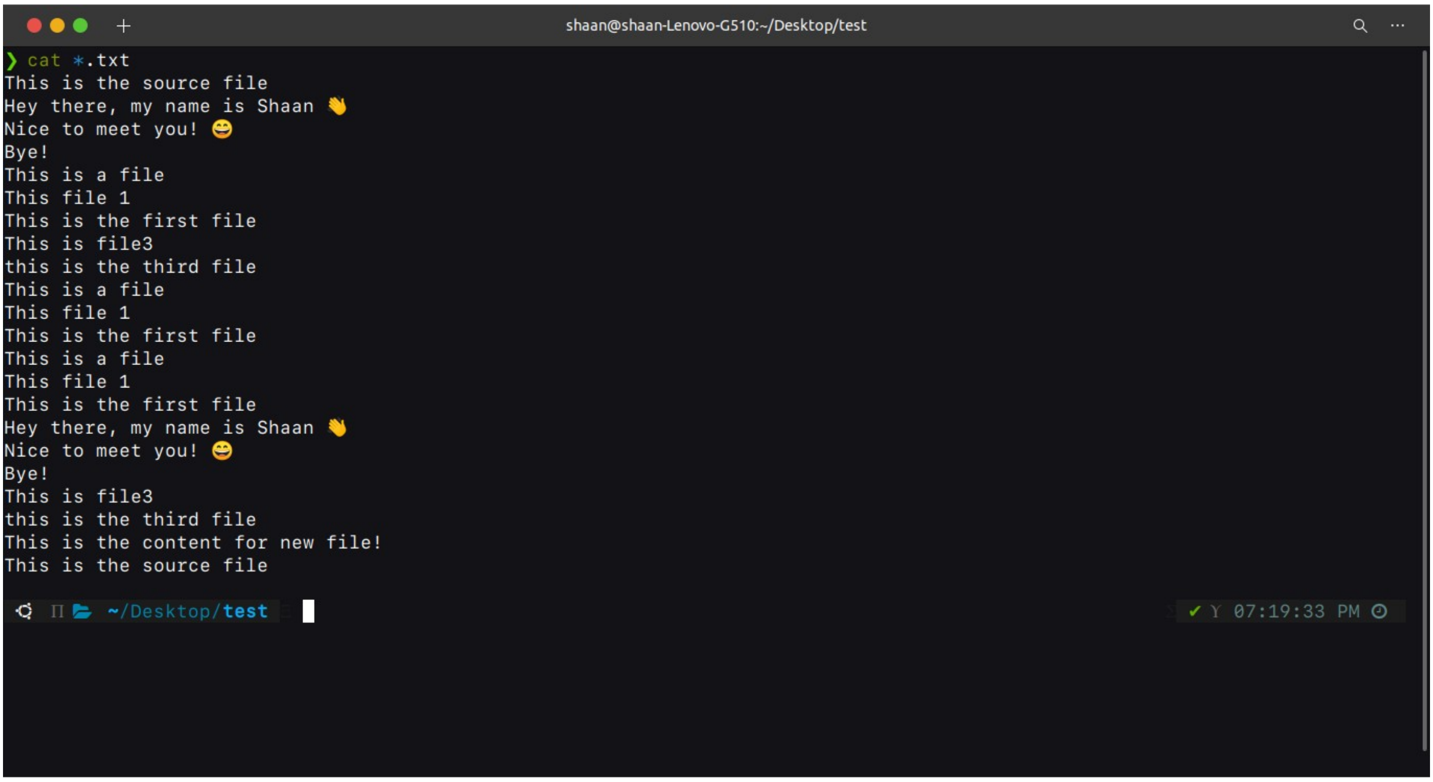
cat file_name_1 file_name_2 file_name_3 > merged_file_name
This command will merge the mentioned files into a single file.



```
❯ cat file.txt file2.txt file3.txt > merged_file.txt
❯ cat merged_file.txt
This is a file
This file 1
This is the first file
Hey there, my name is Shaan 👋
Nice to meet you! 😄
Bye!
This is file3
this is the third file

 ~/Desktop/test
```

cat *.txt

This command will print all the .txt files present in the current folder. **Note-** You can also use this command to print other type of files like *html*, *js*, *ts* etc. For ex - cat *.html will print all the *.html* files present in the current folder.



```
❯ cat *.txt
This is the source file
Hey there, my name is Shaan 👋
Nice to meet you! 😄
Bye!
This is a file
This file 1
This is the first file
This is file3
this is the third file
This is a file
This file 1
This is the first file
This is a file
This file 1
This is the first file
Hey there, my name is Shaan 👋
Nice to meet you! 😄
Bye!
This is file3
this is the third file
This is the content for new file!
This is the source file

 ~/Desktop/test                                                          ✔  07:19:33 PM
```

---

who and whoami

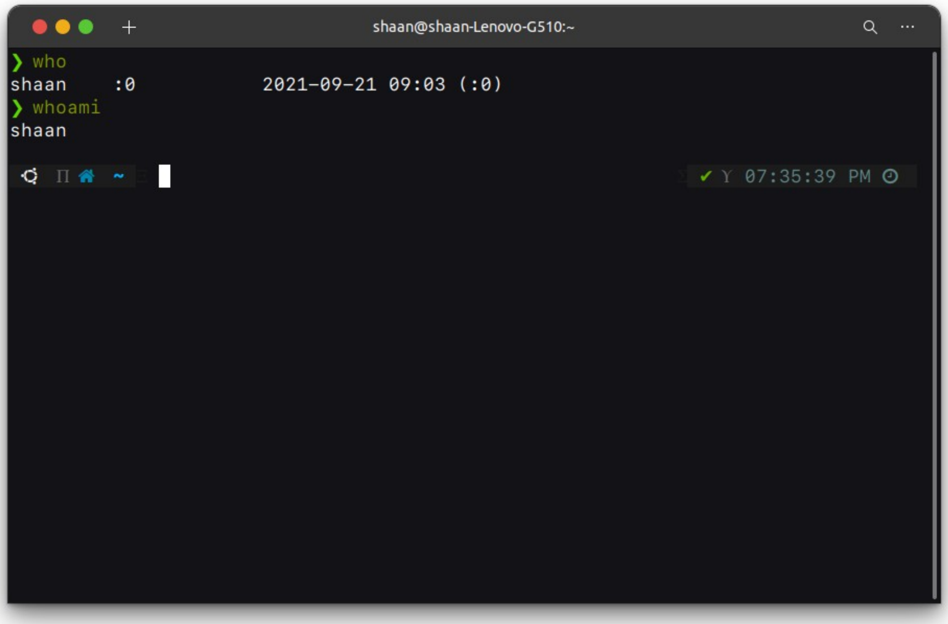
who

who command will print the name of currently logged in user along with some other informations.

- Login name of the users
- Terminal line numbers
- Login time of the users in to system

whoami

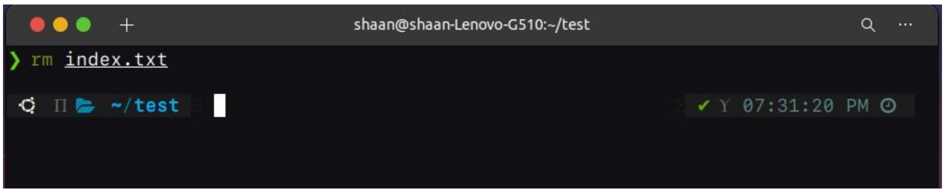
whoami command prints the user name.



```
❯ who
shaan    :0           2021-09-21 09:03 (:0)
❯ whoami
shaan
```

rm *[filename]*

rm command will delete the file specified.



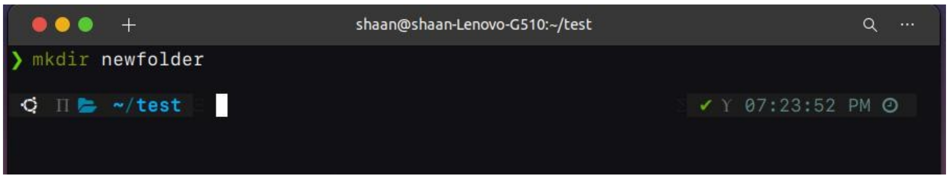
```
❯ rm index.txt

  ~/test                                         ✔ 07:31:20 PM
```

mkdir

This command is used to create a new directories in the current working directory.

mkdir *[dirname]*

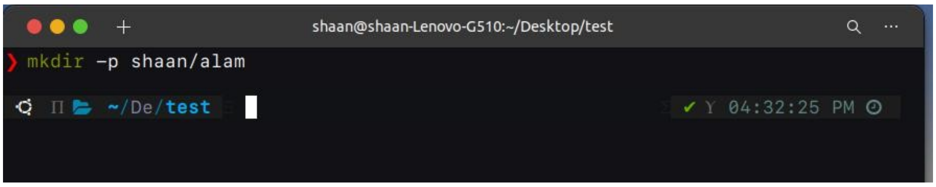
This command is used to create a new directory.



```
❯ mkdir newfolder

  ~/test                                         ✔ 07:23:52 PM
```

mkdir -p /shaan/alam

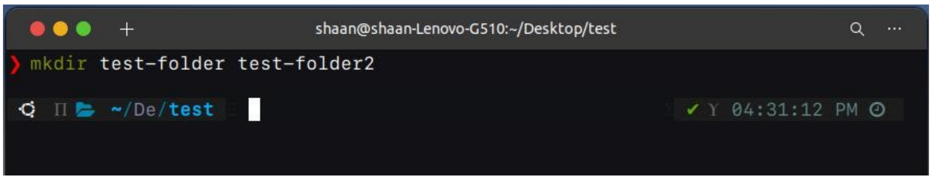
This command will create all required parent directories



```
❯ mkdir -p shaan/alam

  ~/De/test                                      ✔ 04:32:25 PM
```

mkdir *[dirname1] [dirname2]*

This command will create multiple directories at the same time.



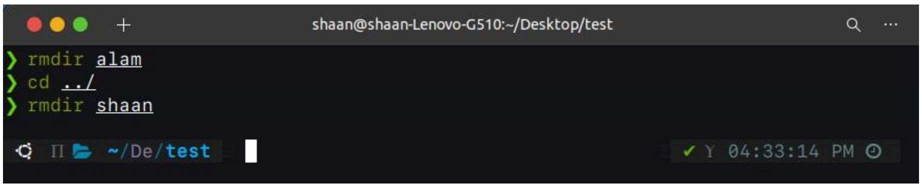
```
❯ mkdir test-folder test-folder2

  ~/De/test                                      ✔ 04:31:12 PM
```

rmdir

This command is used to delete directories specified.

rmdir *[dirname]*

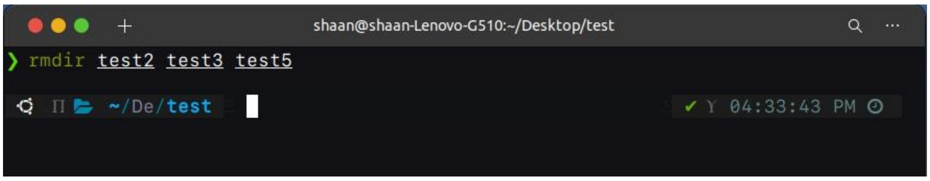
This comand will delete a single directory specified.



```
❯ rmdir alam
❯ cd ../
❯ rmdir shaan

 ~/De/test                              ✔ 04:33:14 PM
```

rmdir *[dirname] [dirname]* ...

This command will delete multiple directories specified.



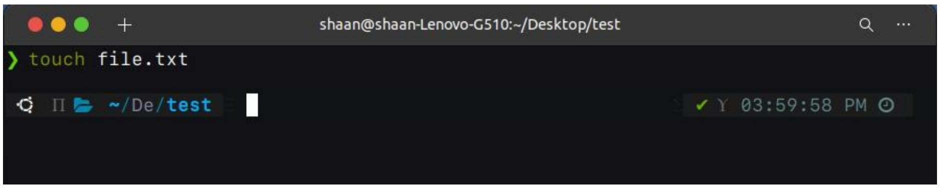
```
❯ rmdir test2 test3 test5

 ~/De/test                              ✔ 04:33:43 PM
```

touch

touch command is used to create new files.

touch *[filename]*

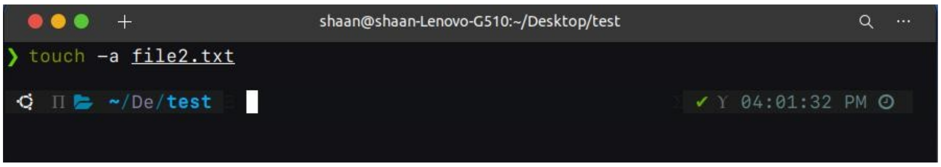
This command will create a new file.



```
❯ touch file.txt

 ~/De/test                              ✔ 03:59:58 PM
```

touch -a *[filename]*

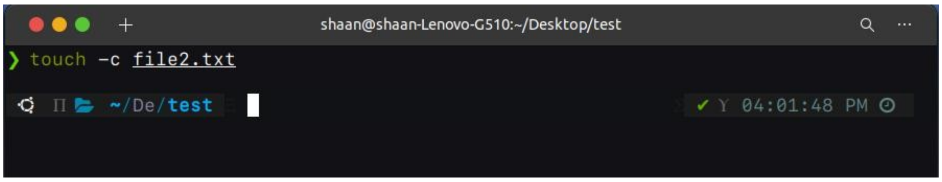
This command is used to change access time only.



```
❯ touch -a file2.txt

 ~/De/test                              ✔ 04:01:32 PM
```

touch -c *[filename]*

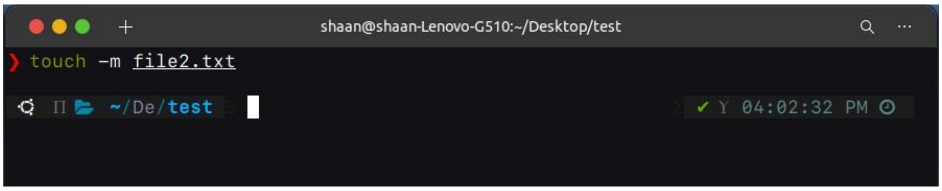
This command is used to check whether a file is created or not. If not created then don’t create it.



```
shaan@shaan-Lenovo-G510:~/Desktop/test
❯ touch -c file2.txt
  ~/De/test                                   ✔ 04:01:48 PM
```

touch -m *[filename]*

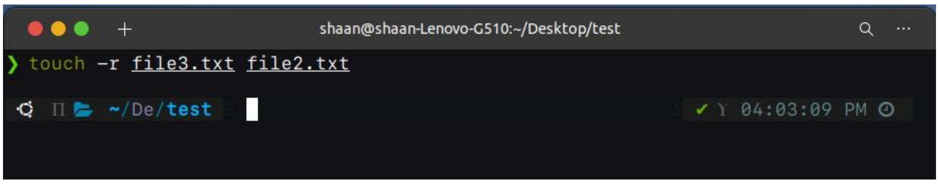
his is used to change the modification time only. It only updates last modification time.



```
shaan@shaan-Lenovo-G510:~/Desktop/test
❯ touch -m file2.txt
  ~/De/test                                   ✔ 04:02:32 PM
```

touch -r *[filename]*

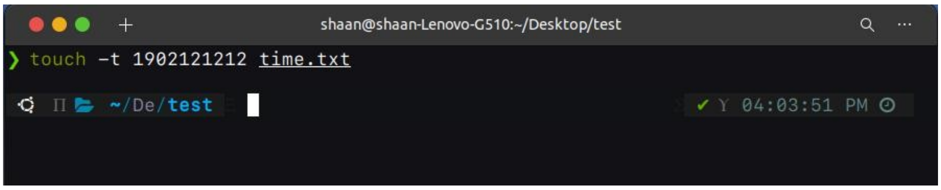
This command is used to use the timestamp of another file.



```
shaan@shaan-Lenovo-G510:~/Desktop/test
❯ touch -r file3.txt file2.txt
  ~/De/test                                   ✔ 04:03:09 PM
```

touch -t *[filename]*

This is used to create a file using a specified time.



```
❯ touch -t 1902121212 time.txt

  ~/De/test                                        ✔ Y 04:03:51 PM
```

---

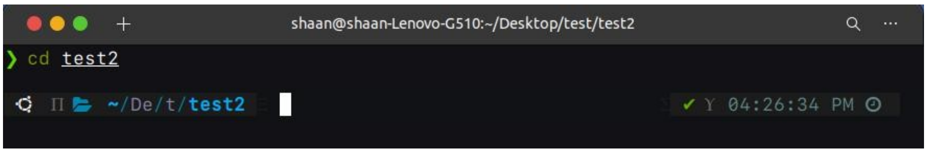
cd

---

cd command stands for **Change Directory**. As the name specifies, it is used to change the directory from the current working directory.

cd *[dirname]*

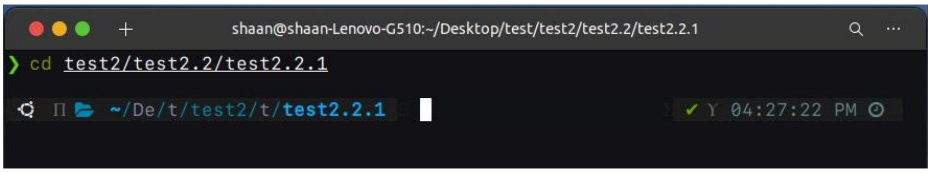
This command will the change the current working directory to [dirname] directory. (Here, *[dirname]* is the name of the directory you want to change to.You can provide your own directory name instead of *[dirname]* for example :- cd shaan)



```
❯ cd test2

  ~/De/t/test2                                     ✔ Y 04:26:34 PM
```

cd *[dir1]/[dir2]/[dir3]*

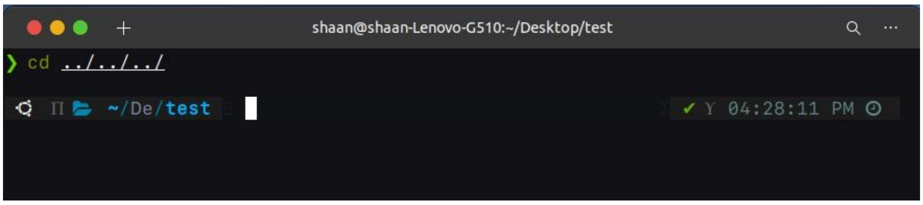
This command will change the current working directory to *dir1/dir2/dir3* directory. (Here, *[dir]* is the name of the directory you want to change to.You can provide your own directory name instead of *[dir]* for example :- cd linux_class/shaan)


shaan@shaan-Lenovo-G510:~/Desktop/test/test2/test2.2/test2.2.1

```
❯ cd test2/test2.2/test2.2.1

  ~/De/t/test2/t/test2.2.1                         ✔ Y 04:27:22 PM
```

cd ../

This command will change the current working directory to the parent directory or previous folder.



```
❯ cd ../../../

  ~/De/test                                        ✔ Y 04:28:11 PM
```

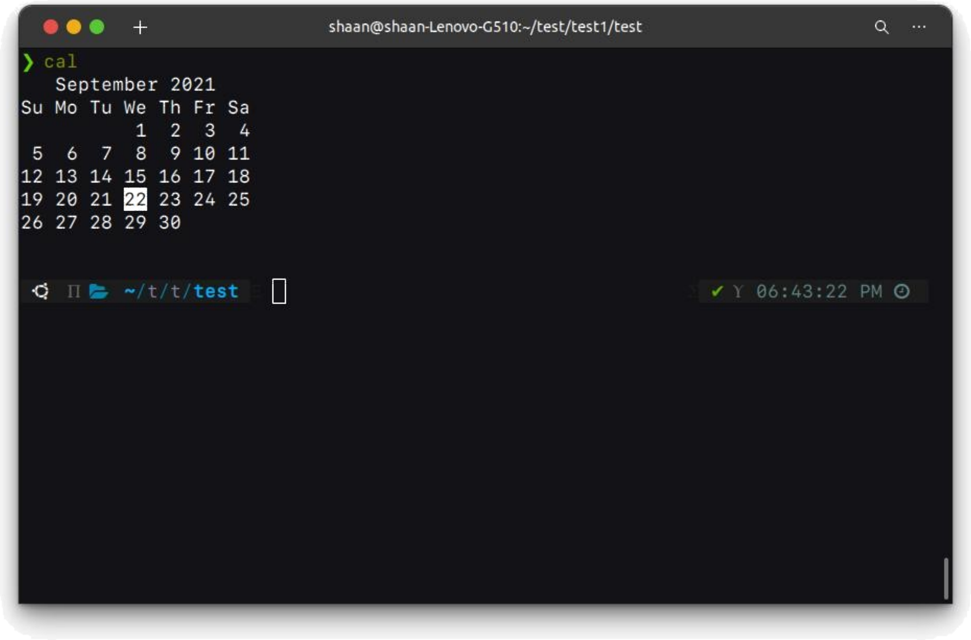
---

cal

cal command will print the calendar in terminal window. By default, cal command will show the current month as output.

cal

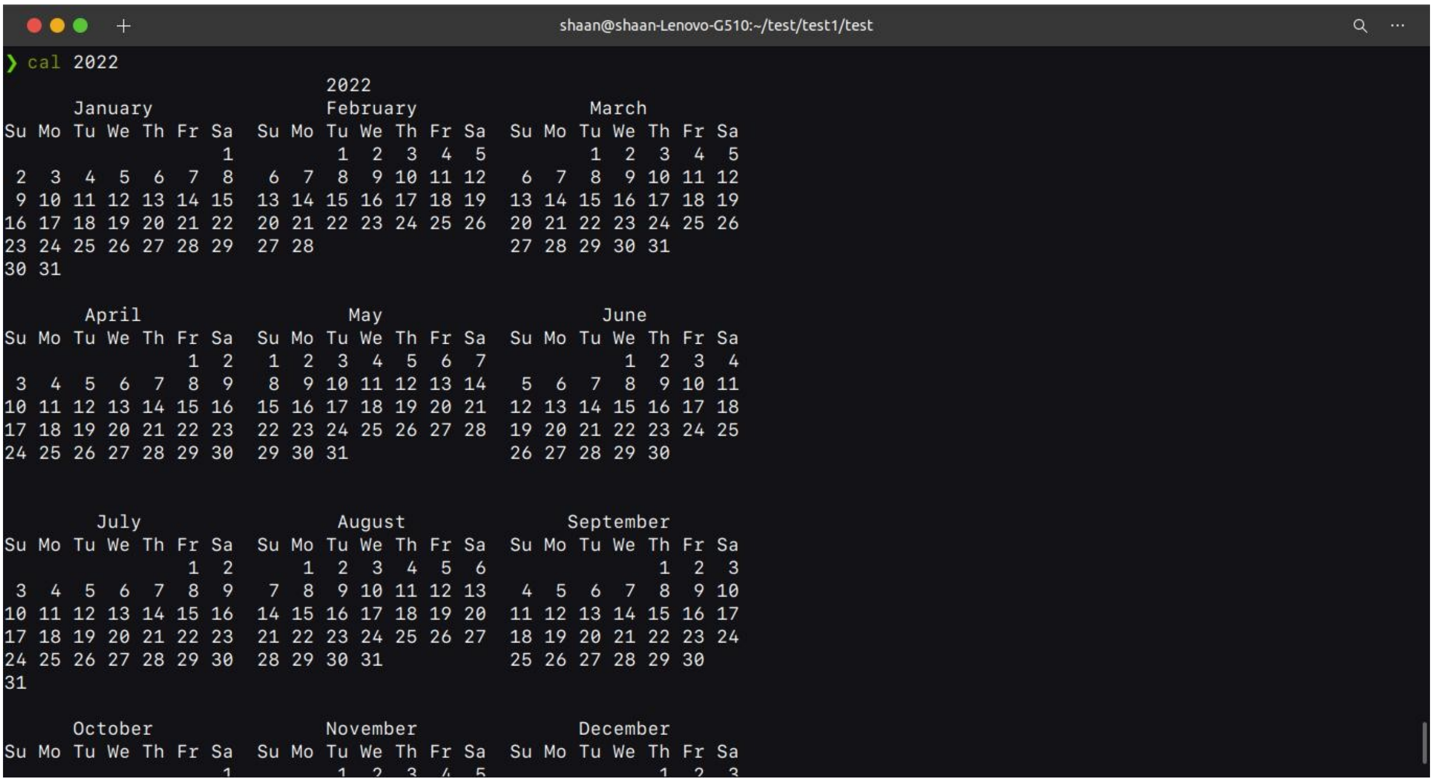
This command will by default show the calendar of current month.



```
❯ cal
   September 2021
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

 ~/t/t/test                                 ✔ 06:43:22 PM
```

cal *[year]*

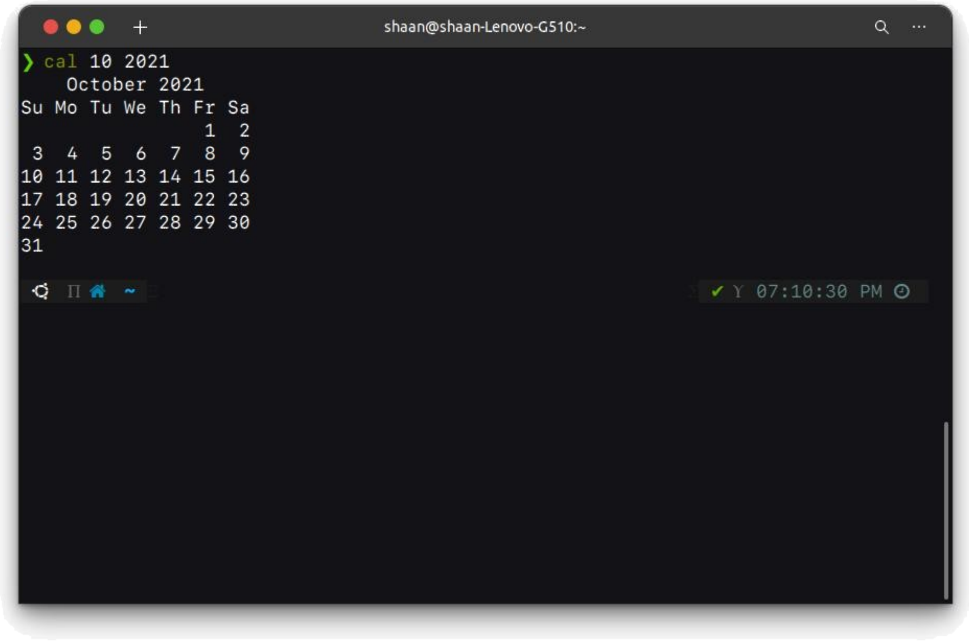
This command will show the entire year's calendar as output.



```
❯ cal 2022
                            2022
      January               February               March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                   1         1  2  3  4  5         1  2  3  4  5
 2  3  4  5  6  7  8   6  7  8  9 10 11 12   6  7  8  9 10 11 12
 9 10 11 12 13 14 15  13 14 15 16 17 18 19  13 14 15 16 17 18 19
16 17 18 19 20 21 22  20 21 22 23 24 25 26  20 21 22 23 24 25 26
23 24 25 26 27 28 29  27 28                 27 28 29 30 31
30 31

       April                  May                   June
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                1  2   1  2  3  4  5  6  7            1  2  3  4
 3  4  5  6  7  8  9   8  9 10 11 12 13 14   5  6  7  8  9 10 11
10 11 12 13 14 15 16  15 16 17 18 19 20 21  12 13 14 15 16 17 18
17 18 19 20 21 22 23  22 23 24 25 26 27 28  19 20 21 22 23 24 25
24 25 26 27 28 29 30  29 30 31              26 27 28 29 30


        July                 August                September
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                1  2      1  2  3  4  5  6               1  2  3
 3  4  5  6  7  8  9   7  8  9 10 11 12 13   4  5  6  7  8  9 10
10 11 12 13 14 15 16  14 15 16 17 18 19 20  11 12 13 14 15 16 17
17 18 19 20 21 22 23  21 22 23 24 25 26 27  18 19 20 21 22 23 24
24 25 26 27 28 29 30  28 29 30 31           25 26 27 28 29 30
31

      October               November              December
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                   1         1  2  3  4  5               1  2  3
```

cal *[month] [year]*

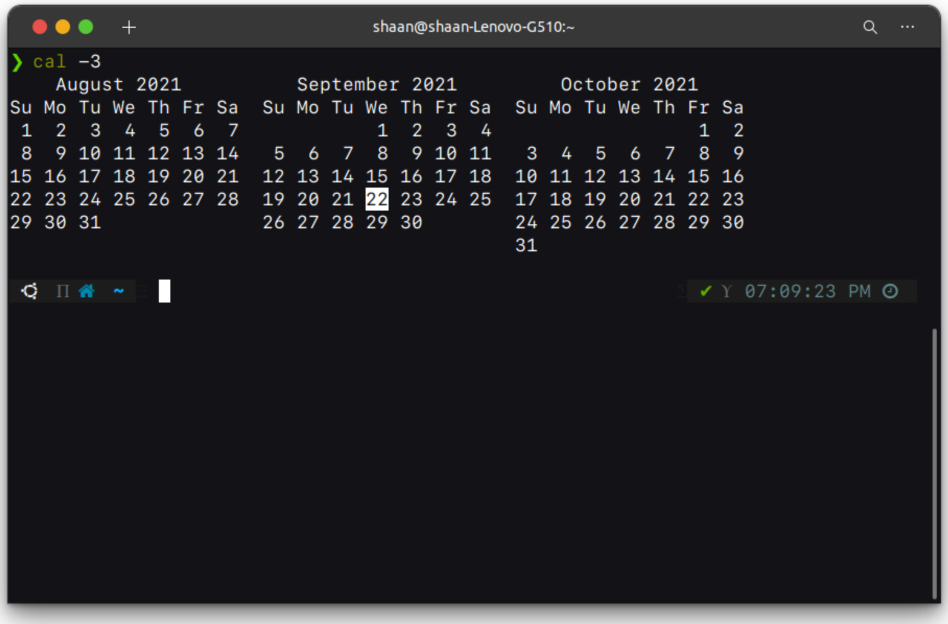
This command will print the calendar for the specified month of the year.



```
❯ cal 10 2021
    October 2021
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
```

cal -3

This command will show the calendar of *previous*, *current*, & *next* month.



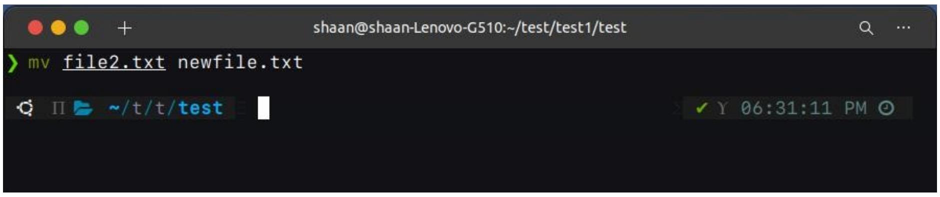
```
❯ cal -3
    August 2021          September 2021         October 2021
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7            1  2  3  4                  1  2
 8  9 10 11 12 13 14   5  6  7  8  9 10 11   3  4  5  6  7  8  9
15 16 17 18 19 20 21  12 13 14 15 16 17 18  10 11 12 13 14 15 16
22 23 24 25 26 27 28  19 20 21 22 23 24 25  17 18 19 20 21 22 23
29 30 31              26 27 28 29 30        24 25 26 27 28 29 30
                                            31

 ~                                                  ✔ 07:09:23 PM
```

mv

This command is used to rename a file or a directory.

mv

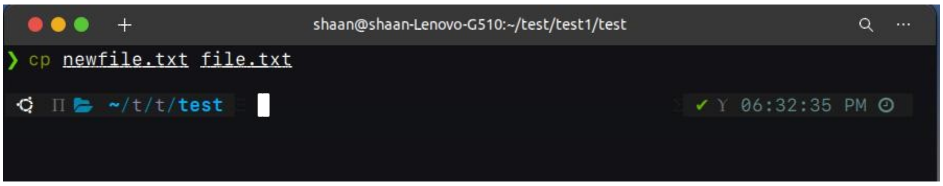
This command can be used to rename a file. The syntax is *mv file_to_be_renamed new_file_name*



```
❯ mv file2.txt newfile.txt

 ~/t/t/test                                         ✔ 06:31:11 PM
```

cp

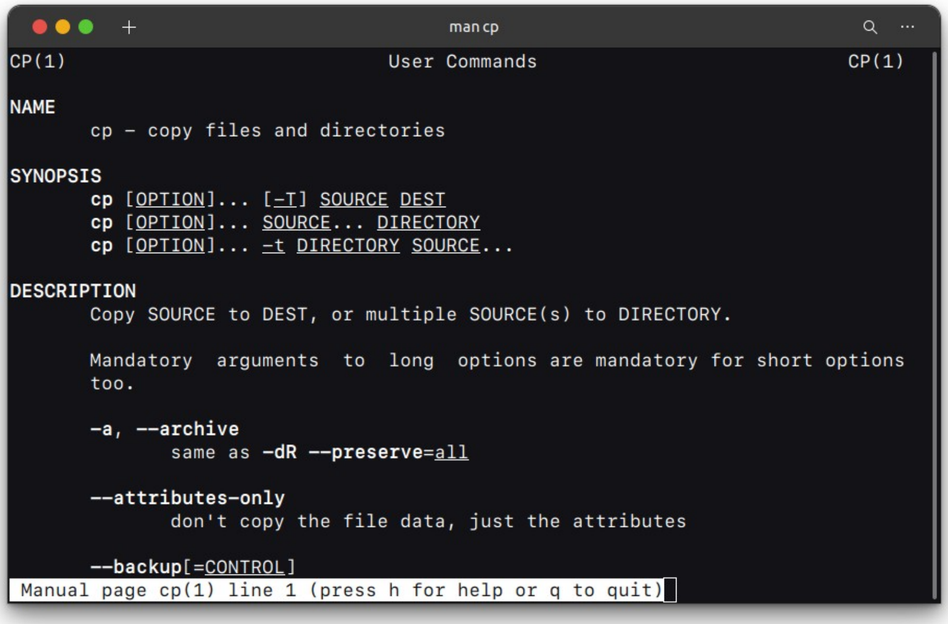
This command is used to copy a file. The syntax is *cp file_to_be_copied new_file_name*



```
❯ cp newfile.txt file.txt

  ~/t/t/test                                      ✔  06:32:35 PM
```

man

This command is used to display a manual for any other shell command. For example, man date will print the **man cp** will print the manual for **cp** command.



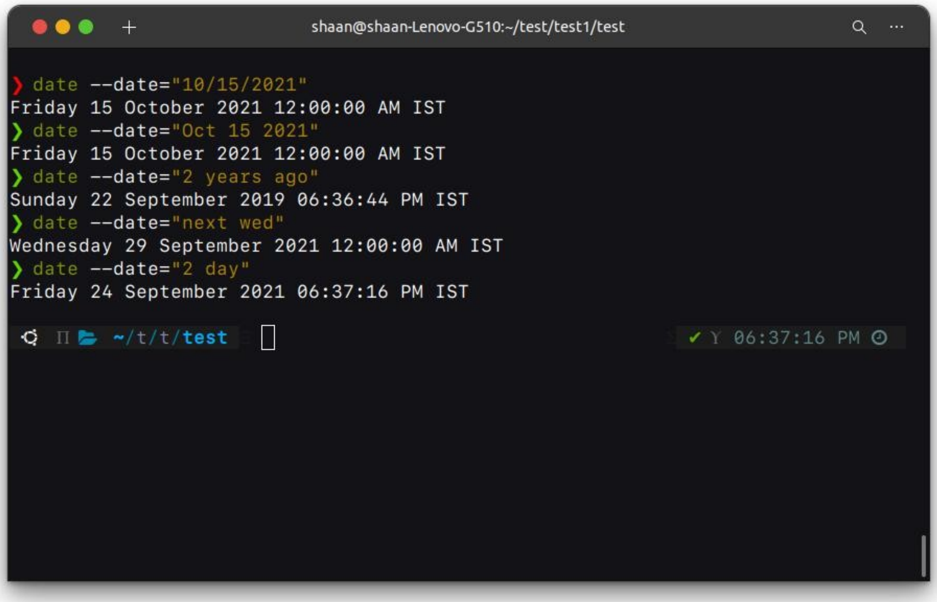
```
CP(1)                        User Commands                        CP(1)

NAME
       cp - copy files and directories

SYNOPSIS
       cp [OPTION]... [-T] SOURCE DEST
       cp [OPTION]... SOURCE... DIRECTORY
       cp [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
       Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

       Mandatory  arguments  to  long  options are mandatory for short options
       too.

       -a, --archive
              same as -dR --preserve=all

       --attributes-only
              don't copy the file data, just the attributes

       --backup[=CONTROL]
 Manual page cp(1) line 1 (press h for help or q to quit)
```

date

Date command is used to display the system date and time. By default **date** command will print the date of the time zone which was configured on the system while installing the operating system.

date commands

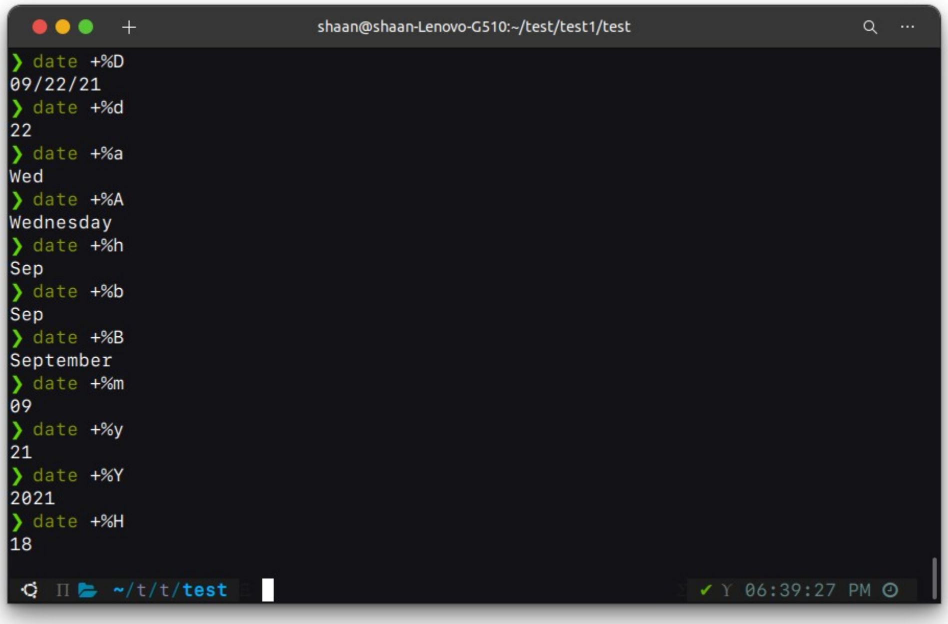
- **--date** - Displays the given date string date format.



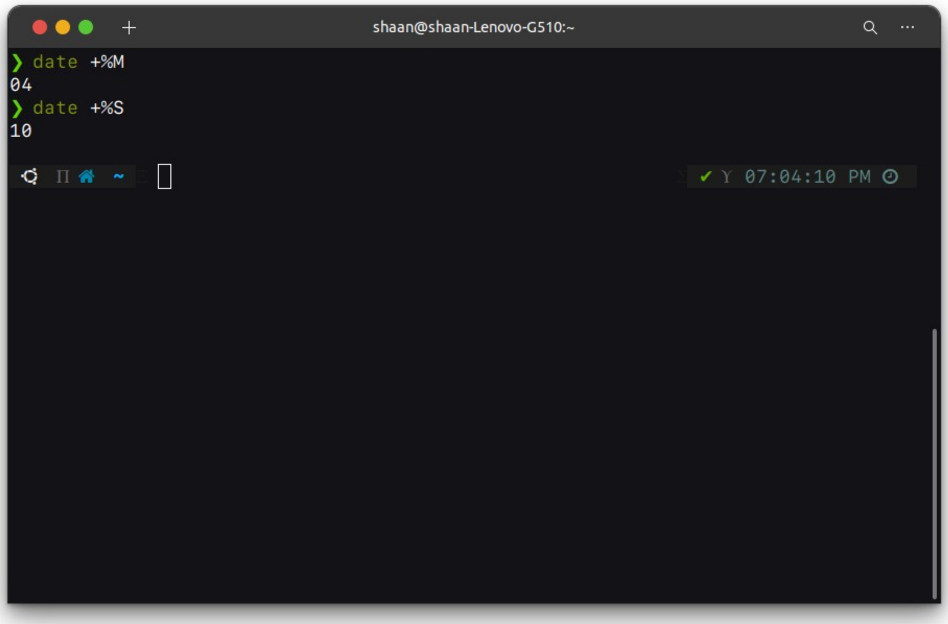
```
❯ date --date="10/15/2021"
Friday 15 October 2021 12:00:00 AM IST
❯ date --date="Oct 15 2021"
Friday 15 October 2021 12:00:00 AM IST
❯ date --date="2 years ago"
Sunday 22 September 2019 06:36:44 PM IST
❯ date --date="next wed"
Wednesday 29 September 2021 12:00:00 AM IST
❯ date --date="2 day"
Friday 24 September 2021 06:37:16 PM IST

 ~/t/t/test                                   ✔ Y 06:37:16 PM
```

-
- **% D** - Display date as mm/dd/yy.
- **%d** - Display the day of the month (01 to 31).
- **%a** - Displays the abbreviated name for weekdays (Sun to Sat).
- **%A** - Displays full weekdays (Sunday to Saturday).
- **%h** - Displays abbreviated month name (Jan to Dec).
- **%b** - Displays abbreviated month name (Jan to Dec).
- **%B** - Displays full month name(January to December).
- **%m** - Displays the month of year (01 to 12).
- **%y** - Displays last two digits of the year(00 to 99).
- **%Y** - Display four-digit year.
- **%T** - Display the time in 24 hour format as - HH:MM:SS.

- **%H** - Display the hour.
- **%M** - Display the minute.
- **%S** - Display the seconds.



```
❯ date +%D
09/22/21
❯ date +%d
22
❯ date +%a
Wed
❯ date +%A
Wednesday
❯ date +%h
Sep
❯ date +%b
Sep
❯ date +%B
September
❯ date +%m
09
❯ date +%y
21
❯ date +%Y
2021
❯ date +%H
18
```



```
❯ date +%M
04
❯ date +%S
10
```

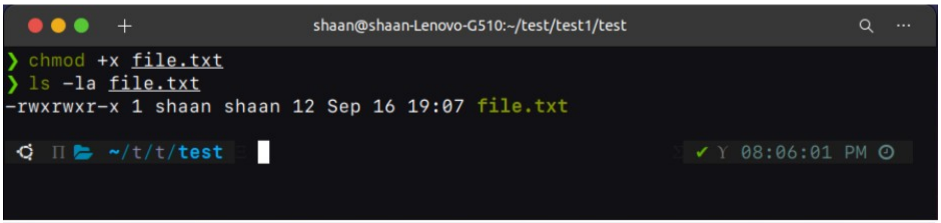
chmod

chmod command stands for **change mode** command. There are 3 modes - - +w (write) - +r (read) - +x (execute)

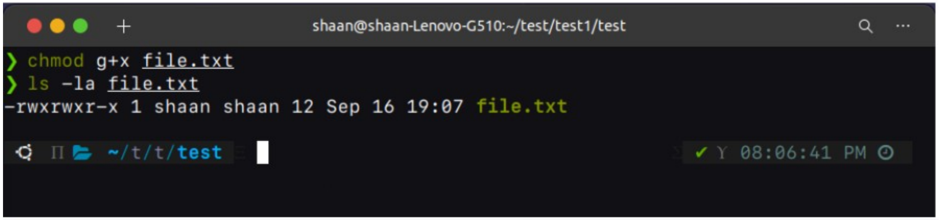
chmod [u/g/o]+[r/w/x]

This command will change the modes for **either user, group, others or all** as **either read, write, execute or all.**

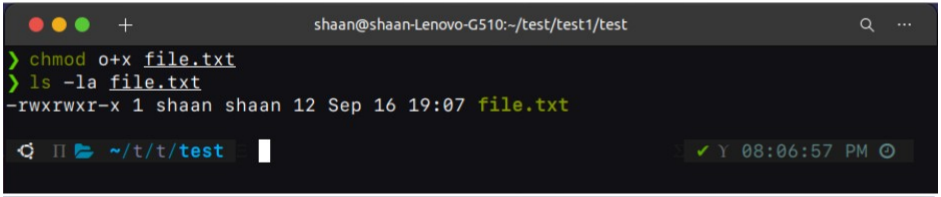
Category	Operation	Permission
u (User)	+ (assign)	r (read)
g (Groups)	- (remove)	w (write)
o (Others)	= (Absolute permission)	x (execute)
a (all)		



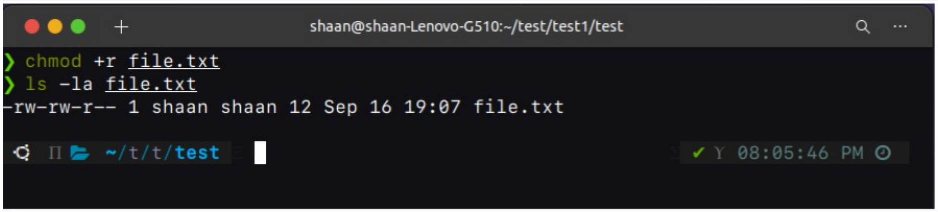
```
❯ chmod +x file.txt
❯ ls -la file.txt
-rwxrwxr-x 1 shaan shaan 12 Sep 16 19:07 file.txt

 ~/t/t/test                                    ✔ 08:06:01 PM
```



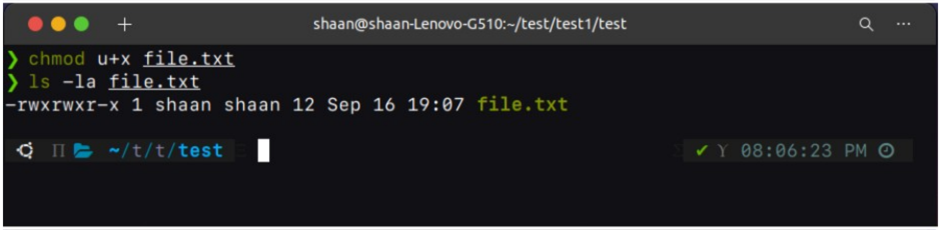
```
❯ chmod g+x file.txt
❯ ls -la file.txt
-rwxrwxr-x 1 shaan shaan 12 Sep 16 19:07 file.txt

 ~/t/t/test                                    ✔ 08:06:41 PM
```



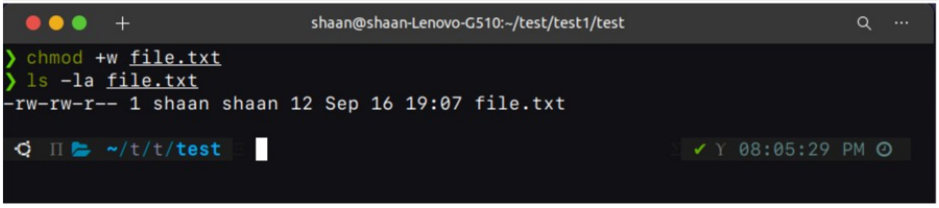
```
❯ chmod o+x file.txt
❯ ls -la file.txt
-rwxrwxr-x 1 shaan shaan 12 Sep 16 19:07 file.txt

 ~/t/t/test                                    ✔ 08:06:57 PM
```



```
❯ chmod +r file.txt
❯ ls -la file.txt
-rw-rw-r-- 1 shaan shaan 12 Sep 16 19:07 file.txt

 ~/t/t/test                                    ✔ 08:05:46 PM
```

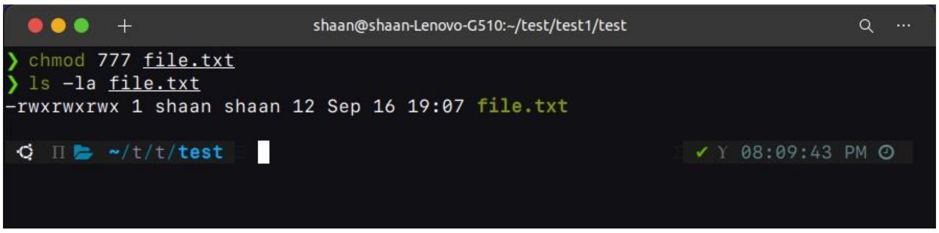


```
❯ chmod u+x file.txt
❯ ls -la file.txt
-rwxrwxr-x 1 shaan shaan 12 Sep 16 19:07 file.txt

 ~/t/t/test                                    ✔ 08:06:23 PM
```



```
❯ chmod +w file.txt
❯ ls -la file.txt
-rw-rw-r-- 1 shaan shaan 12 Sep 16 19:07 file.txt

 ~/t/t/test                                    ✔ 08:05:29 PM
```

Using chmod with Absolute permission



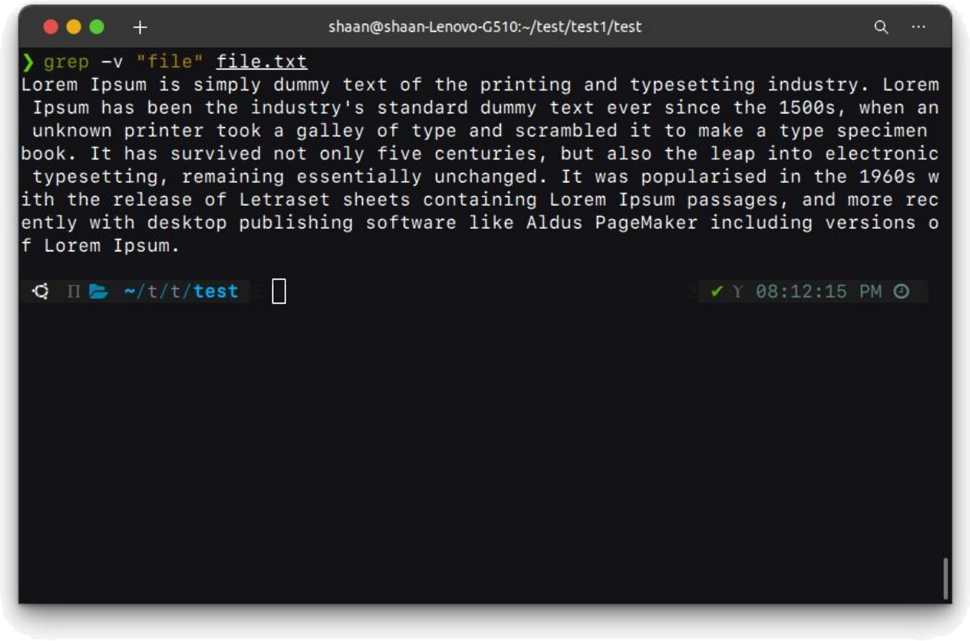
```
❯ chmod 777 file.txt
❯ ls -la file.txt
-rwxrwxrwx 1 shaan shaan 12 Sep 16 19:07 file.txt

 ~/t/t/test                                        ✔ 08:09:43 PM
```

grep

The grep command searched file(s) for lines that match a specified pattern.

grep -v *[pattern] [filename]*

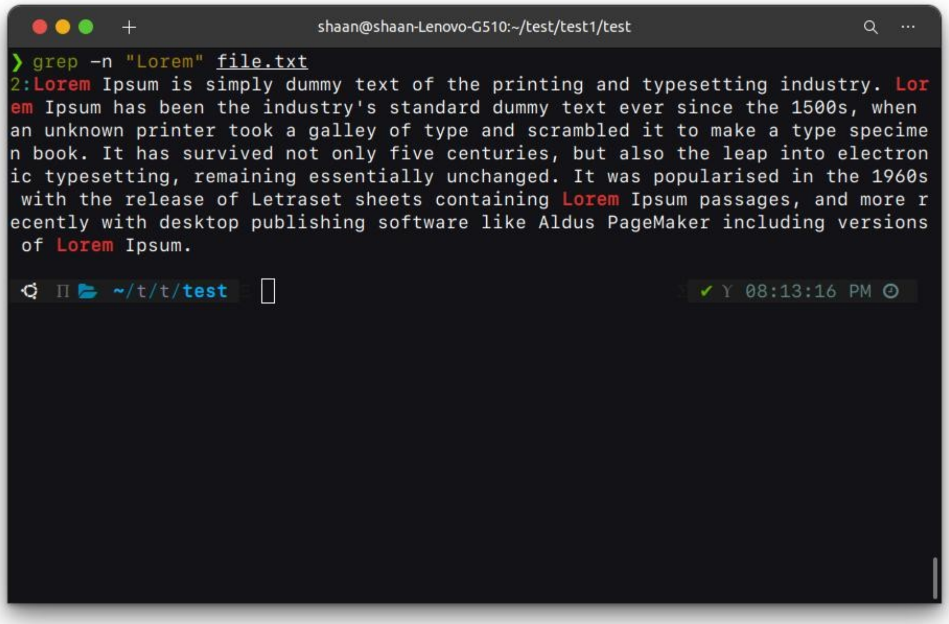
Print all lines that do not match pattern.



```
❯ grep -v "file" file.txt
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem
 Ipsum has been the industry's standard dummy text ever since the 1500s, when an
 unknown printer took a galley of type and scrambled it to make a type specimen
book. It has survived not only five centuries, but also the leap into electronic
 typesetting, remaining essentially unchanged. It was popularised in the 1960s w
ith the release of Letraset sheets containing Lorem Ipsum passages, and more rec
ently with desktop publishing software like Aldus PageMaker including versions o
f Lorem Ipsum.

 ~/t/t/test                                        ✔ 08:12:15 PM
```

grep -n *[pattern] [filename]*

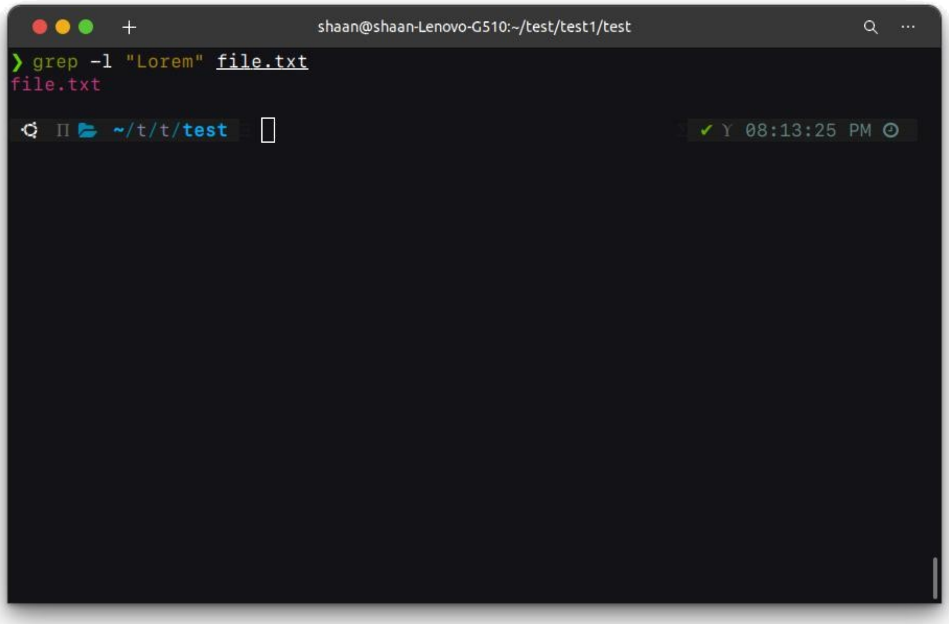
Print the matched line and its line number.



```
❯ grep -n "Lorem" file.txt
2:Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lor
em Ipsum has been the industry's standard dummy text ever since the 1500s, when
an unknown printer took a galley of type and scrambled it to make a type specime
n book. It has survived not only five centuries, but also the leap into electron
ic typesetting, remaining essentially unchanged. It was popularised in the 1960s
 with the release of Letraset sheets containing Lorem Ipsum passages, and more r
ecently with desktop publishing software like Aldus PageMaker including versions
 of Lorem Ipsum.

 ~/t/t/test                                              ✔ 08:13:16 PM
```

grep -l *[pattern] [filename]*

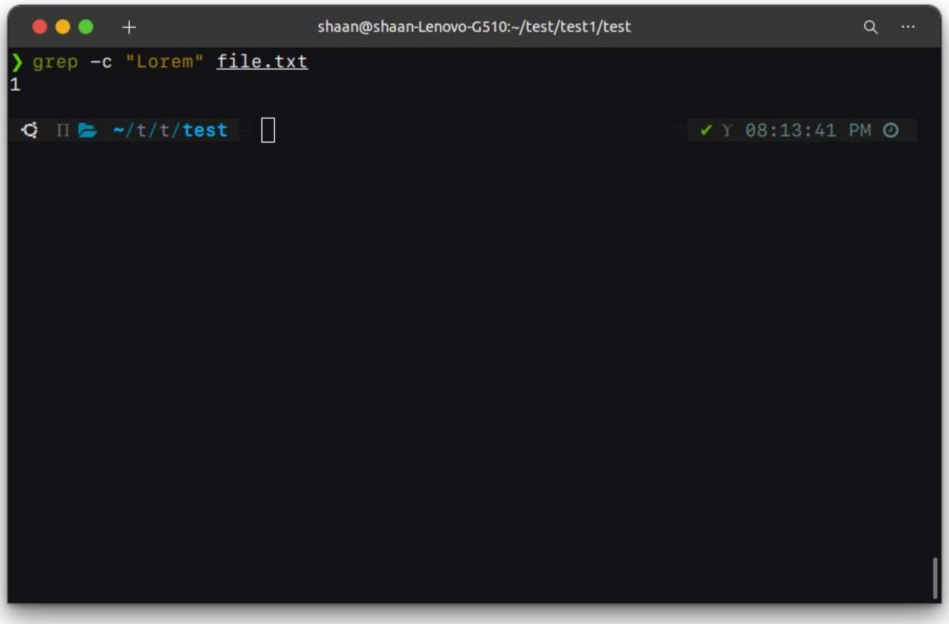
Print only the names of files with matching lines.



```
❯ grep -l "Lorem" file.txt
file.txt

 ~/t/t/test                                              ✔ 08:13:25 PM
```

grep -c *[pattern] [filename]*

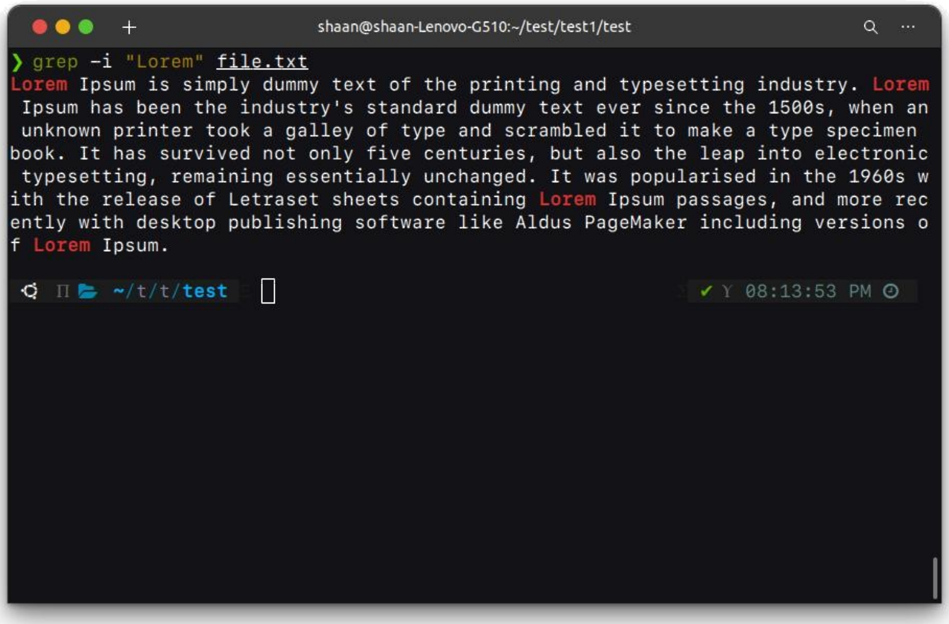
Print only the count of matching lines.



```
❯ grep -c "Lorem" file.txt
1

 ~/t/t/test                                    ✔ 08:13:41 PM
```

grep -i *[pattern] [filename]*

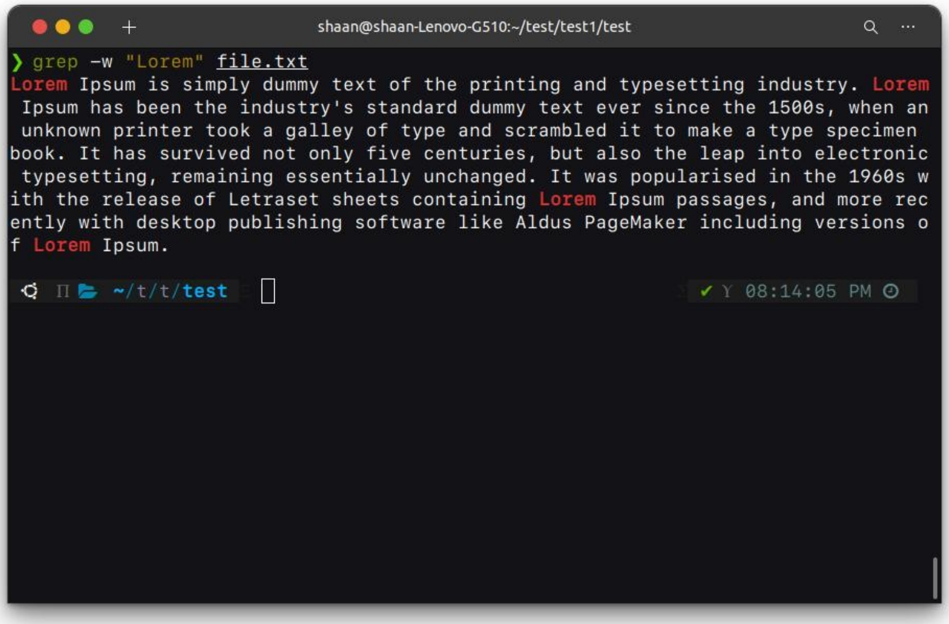
Match either upper- or lowercase.



```
❯ grep -i "Lorem" file.txt
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem
 Ipsum has been the industry's standard dummy text ever since the 1500s, when an
 unknown printer took a galley of type and scrambled it to make a type specimen
book. It has survived not only five centuries, but also the leap into electronic
 typesetting, remaining essentially unchanged. It was popularised in the 1960s w
ith the release of Letraset sheets containing Lorem Ipsum passages, and more rec
ently with desktop publishing software like Aldus PageMaker including versions o
f Lorem Ipsum.

 ~/t/t/test                                    ✔ 08:13:53 PM
```

grep -w *[pattern] [filename]*

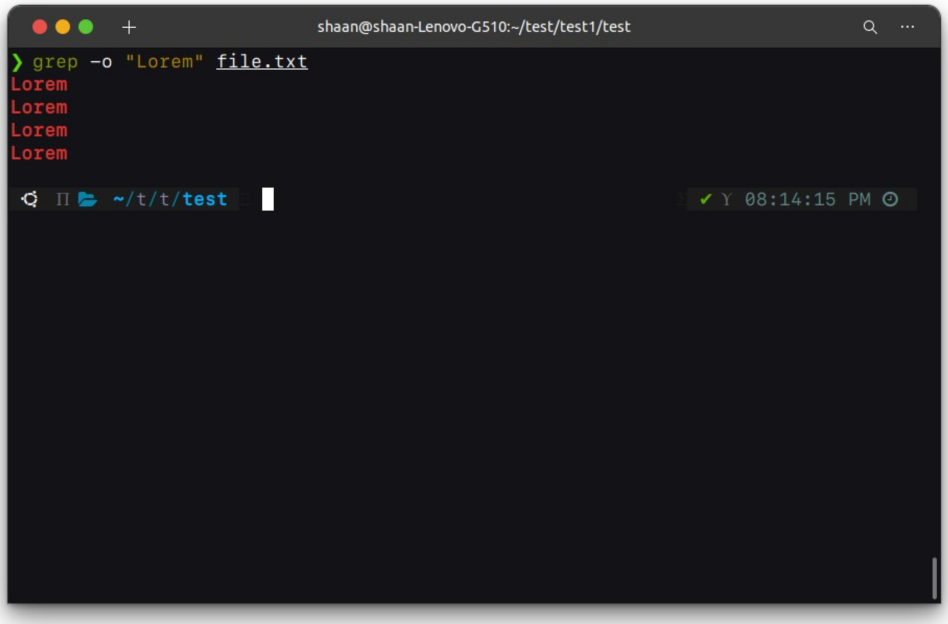
Match whole word



```
❯ grep -w "Lorem" file.txt
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem
 Ipsum has been the industry's standard dummy text ever since the 1500s, when an
 unknown printer took a galley of type and scrambled it to make a type specimen
book. It has survived not only five centuries, but also the leap into electronic
 typesetting, remaining essentially unchanged. It was popularised in the 1960s w
ith the release of Letraset sheets containing Lorem Ipsum passages, and more rec
ently with desktop publishing software like Aldus PageMaker including versions o
f Lorem Ipsum.

 ~/t/t/test                                              ✔  08:14:05 PM
```

grep -o *[pattern] [filename]*

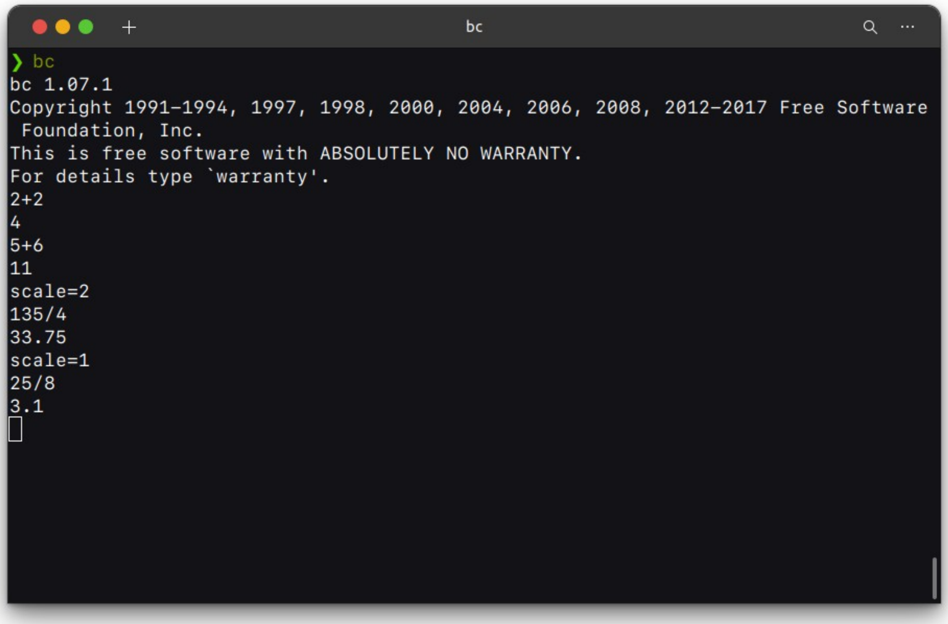
Print only the matched parts of a matching line, with each such part on a separate output line.



```
❯ grep -o "Lorem" file.txt
Lorem
Lorem
Lorem
Lorem

 ~/t/t/test                                              ✔  08:14:15 PM
```

bc

bc stands for *basic calculator* . It is a simple calculator using which we can perform basic arithmatic operations.



```
❯ bc
bc 1.07.1
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2017 Free Software
 Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
2+2
4
5+6
11
scale=2
135/4
33.75
scale=1
25/8
3.1
```