

**Name - Shaan Alam**

---

**Roll no - 20/63027**

---

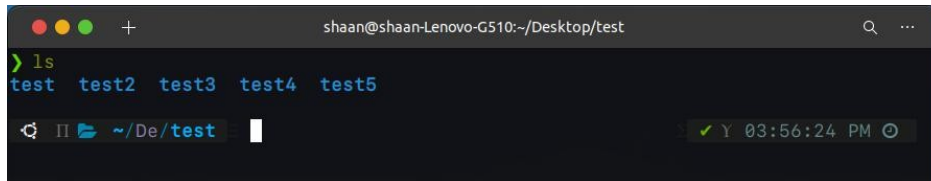
**Subject - Operating System**

---

**Sem - III**

## ls

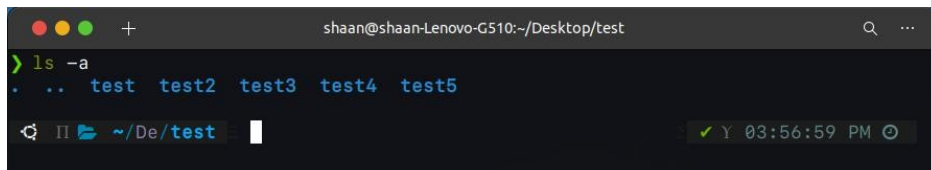
This command is used to list the files in a directory

A terminal window titled 'shaan@shaan-Lenovo-G510: ~/Desktop/test' showing the command 'ls' and its output: 'test test2 test3 test4 test5'. The terminal has a dark background with light blue text for the command and output. The output files are listed in a single line.

```
shaan@shaan-Lenovo-G510: ~/Desktop/test
> ls
test test2 test3 test4 test5
```

## ls -a

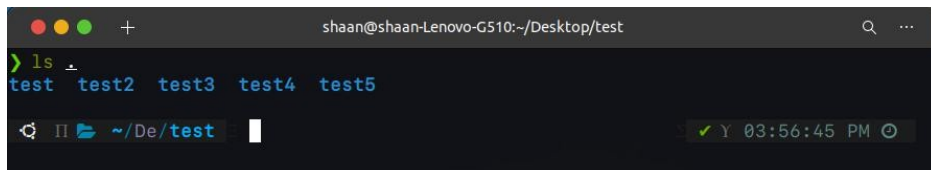
This command is used to list all the files

A terminal window titled 'shaan@shaan-Lenovo-G510: ~/Desktop/test' showing the command 'ls -a' and its output: '. .. test test2 test3 test4 test5'. The output includes hidden files represented by dots.

```
shaan@shaan-Lenovo-G510: ~/Desktop/test
> ls -a
. .. test test2 test3 test4 test5
```

## ls .

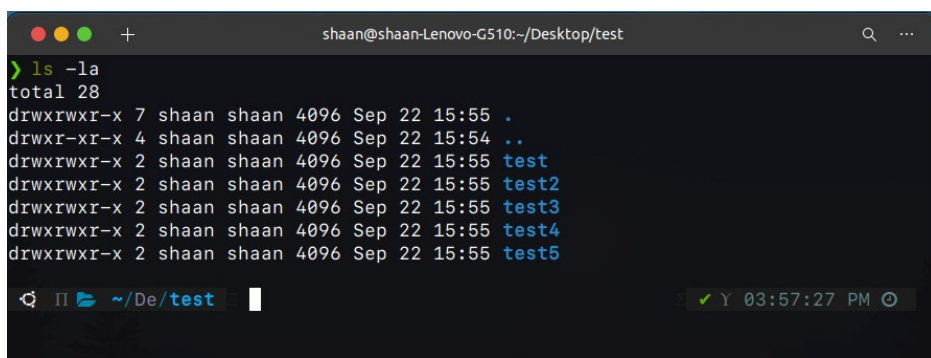
This command is used to list all the files in the current working directory

A terminal window titled 'shaan@shaan-Lenovo-G510: ~/Desktop/test' showing the command 'ls .' and its output: 'test test2 test3 test4 test5'. The output is identical to the 'ls' command.

```
shaan@shaan-Lenovo-G510: ~/Desktop/test
> ls .
test test2 test3 test4 test5
```

## ls -la

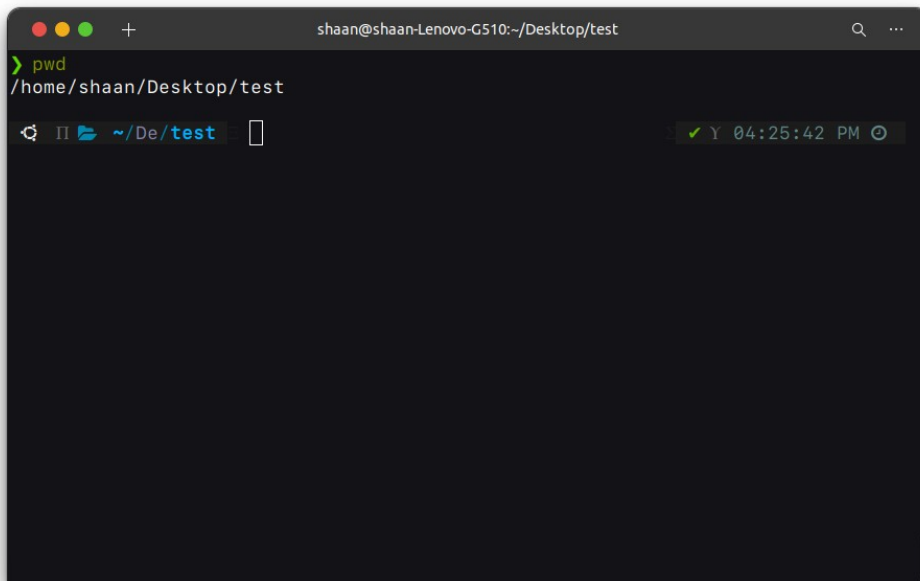
This command is used to display the long listing directories

A terminal window titled 'shaan@shaan-Lenovo-G510: ~/Desktop/test' showing the command 'ls -la' and its output. The output is a long listing showing permissions, owner, group, size, date, and file name for each item in the directory.

```
shaan@shaan-Lenovo-G510: ~/Desktop/test
> 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
```

## pwd

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

A terminal window titled 'shaan@shaan-Lenovo-G510: ~/Desktop/test' with search and menu icons. The prompt is a green prompt character followed by 'pwd'. The output is '/home/shaan/Desktop/test'. The terminal has a dark background and a status bar at the bottom showing a green checkmark, a battery icon, the path '~ / Desktop / test', and a timestamp '04:25:42 PM'.

---

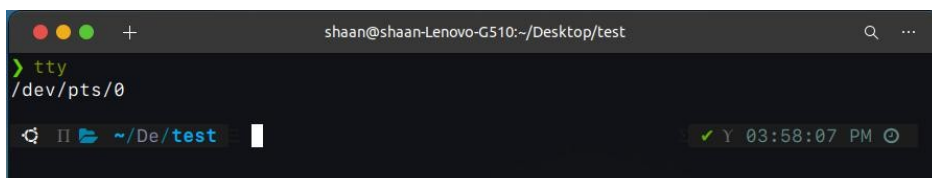
## tty

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

---

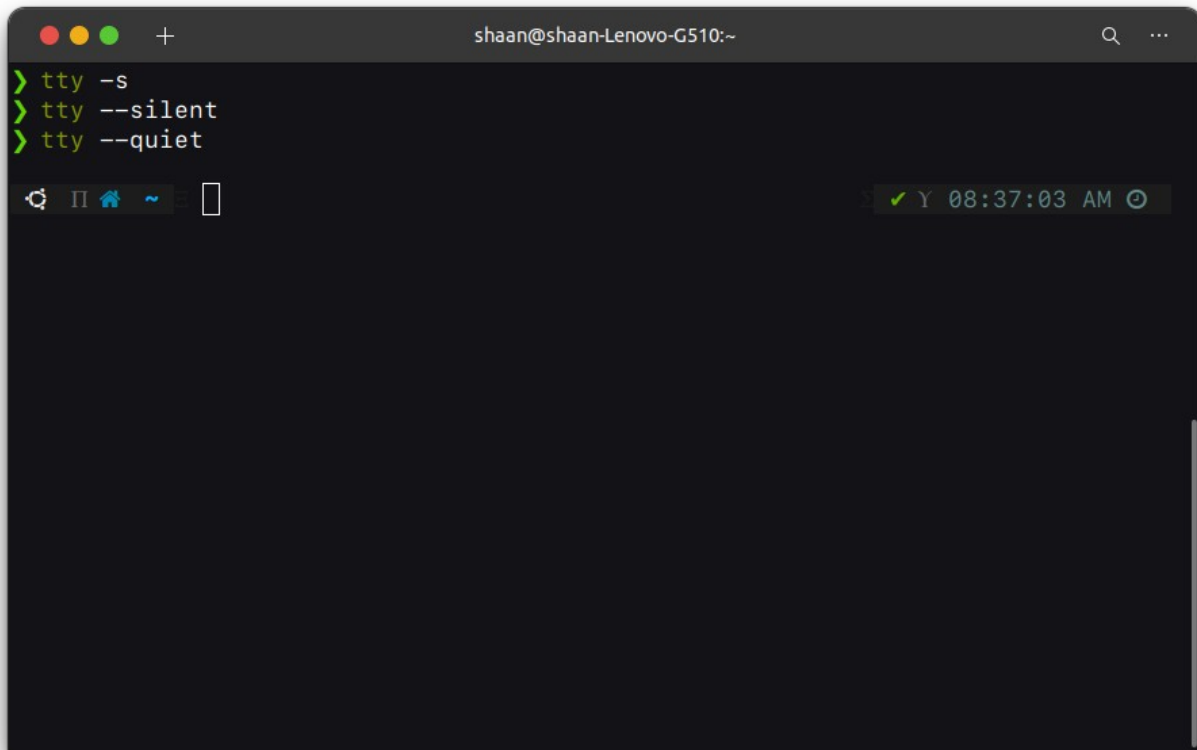
## tty

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

A terminal window titled 'shaan@shaan-Lenovo-G510: ~/Desktop/test' with search and menu icons. The prompt is a green prompt character followed by 'tty'. The output is '/dev/pts/0'. The terminal has a dark background and a status bar at the bottom showing a green checkmark, a battery icon, the path '~ / Desktop / test', and a timestamp '03:58:07 PM'.

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

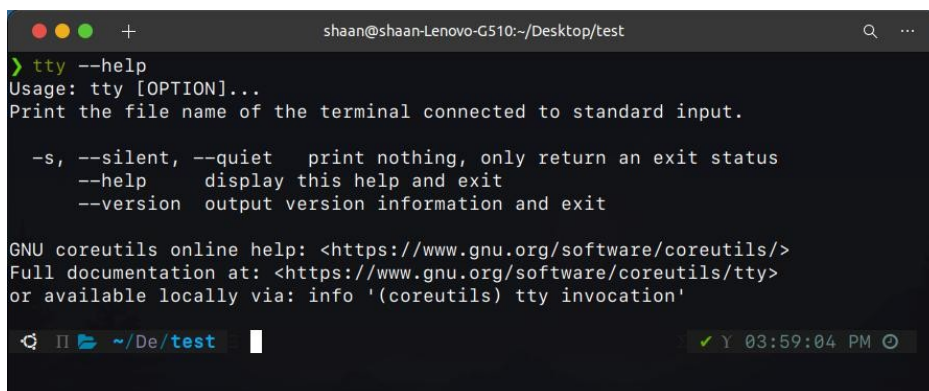
These command will print nothing but return only exit status.

A terminal window titled 'shaan@shaan-Lenovo-G510:~' with standard macOS window controls. The terminal shows three commands: 'tty -s', 'tty --silent', and 'tty --quiet'. At the bottom, there is a status bar with icons for settings, window management, and a clock showing '08:37:03 AM' with a green checkmark and a refresh icon.

```
> tty -s
> tty --silent
> tty --quiet
```

## tty --help

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

A terminal window titled 'shaan@shaan-Lenovo-G510:~/Desktop/test' showing the output of the 'tty --help' command. The output includes usage instructions, a list of options (-s, --silent, --quiet, --help, --version), and links to GNU coreutils online help and full documentation. The status bar at the bottom shows the time '03:59:04 PM' with a green checkmark and a refresh icon.

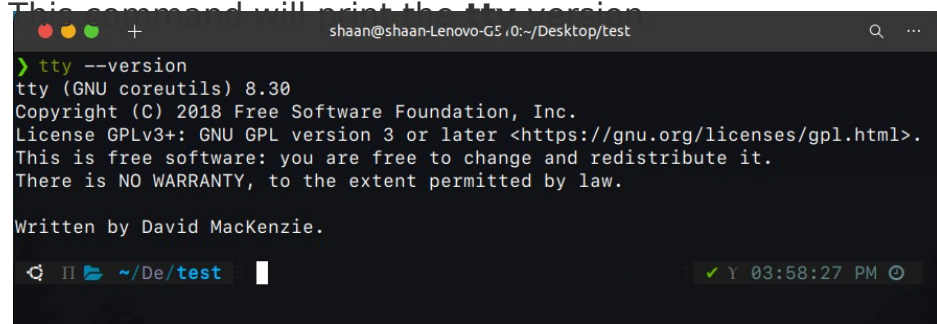
```
> tty --help
Usage: tty [OPTION]...
Print the file name of the terminal connected to standard input.

  -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'
```

## tty --version

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

A terminal window titled 'shaan@shaan-Lenovo-G510:~/Desktop/test' showing the output of the 'tty --version' command. The output displays the version number (8.30), copyright information (2018 Free Software Foundation, Inc.), the license (GPLv3+), and the author (David MacKenzie). The status bar at the bottom shows the time '03:58:27 PM' with a green checkmark and a refresh icon.

```
> 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.
```

## cat

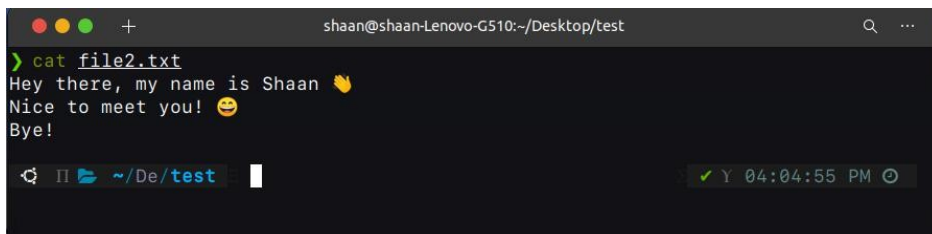
---

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

---

### cat file\_name

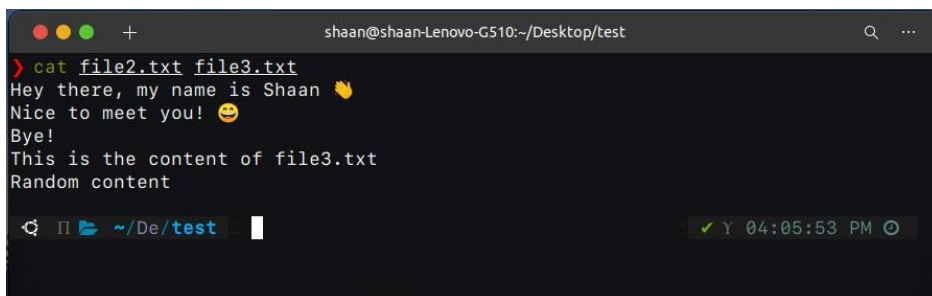
This command will print the content of a single file in the terminal.

A terminal window titled 'shaan@shaan-Lenovo-G510: ~/Desktop/test' showing the command 'cat file2.txt' and its output: 'Hey there, my name is Shaan 🙋', 'Nice to meet you! 😊', and 'Bye!'. The terminal has a dark background with green and blue accents. The prompt is '>'.

```
> cat file2.txt
Hey there, my name is Shaan 🙋
Nice to meet you! 😊
Bye!
```

### cat file\_name\_1 file\_name\_2

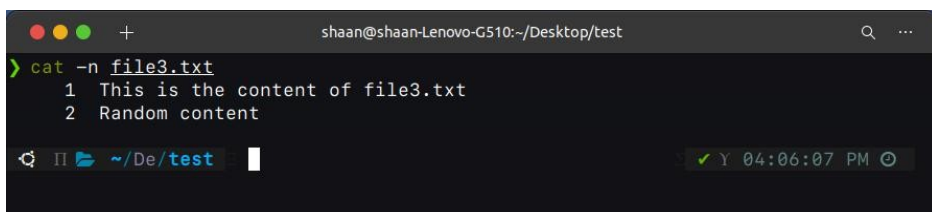
This command will print the content of multiple files in the terminal

A terminal window titled 'shaan@shaan-Lenovo-G510: ~/Desktop/test' showing the command 'cat file2.txt file3.txt' and its output: 'Hey there, my name is Shaan 🙋', 'Nice to meet you! 😊', 'Bye!', 'This is the content of file3.txt', and 'Random content'. The terminal has a dark background with green and blue accents. The prompt is '>'.

```
> 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
```

### cat -n file\_name

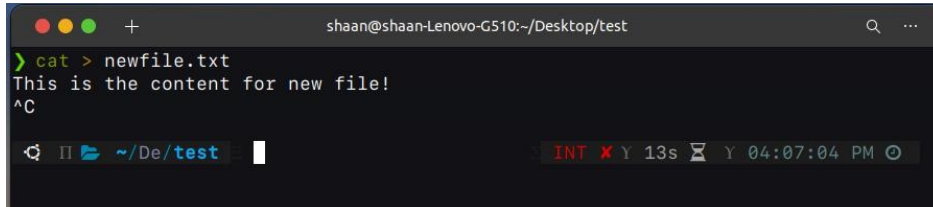
This command will print the content of a file along with line numbers.

A terminal window titled 'shaan@shaan-Lenovo-G510: ~/Desktop/test' showing the command 'cat -n file3.txt' and its output: '1 This is the content of file3.txt' and '2 Random content'. The terminal has a dark background with green and blue accents. The prompt is '>'.

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

## **cat > file\_name**

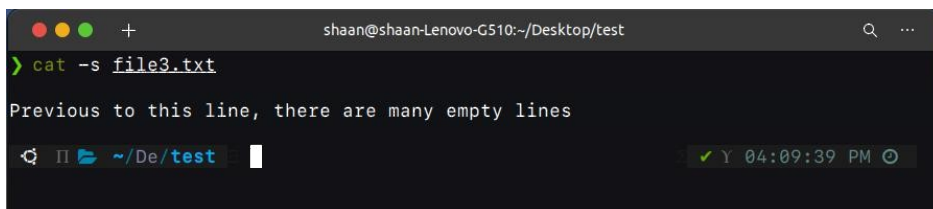
This command will create a new file.

A terminal window titled 'shaan@shaan-Lenovo-G510: ~/Desktop/test' showing the command 'cat > newfile.txt' being executed. The output is 'This is the content for new file!' followed by a carriage return '^C'. The terminal status bar at the bottom shows 'INT X Y 13s' and a clock icon.

```
shaan@shaan-Lenovo-G510: ~/Desktop/test
> cat > newfile.txt
This is the content for new file!
^C
```

## **cat -s file\_name**

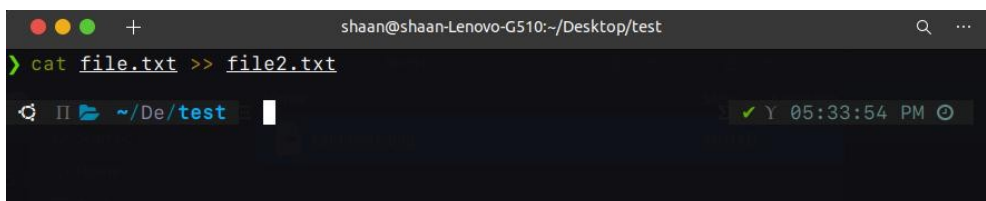
This command will suppress repeated empty lines in the output

A terminal window titled 'shaan@shaan-Lenovo-G510: ~/Desktop/test' showing the command 'cat -s file3.txt' being executed. The output is 'Previous to this line, there are many empty lines'. The terminal status bar at the bottom shows a green checkmark, 'Y 04:09:39 PM', and a clock icon.

```
shaan@shaan-Lenovo-G510: ~/Desktop/test
> cat -s file3.txt
Previous to this line, there are many empty lines
```

## **cat file1 >> file2**

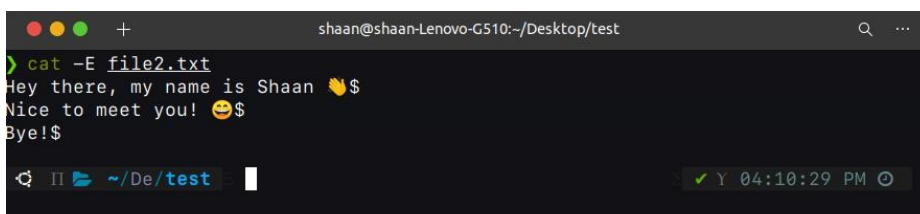
This command will append the content of one file to another

A terminal window titled 'shaan@shaan-Lenovo-G510: ~/Desktop/test' showing the command 'cat file.txt >> file2.txt' being executed. The terminal status bar at the bottom shows a green checkmark, 'Y 05:33:54 PM', and a clock icon.

```
shaan@shaan-Lenovo-G510: ~/Desktop/test
> cat file.txt >> file2.txt
```

## **cat -E file\_name**

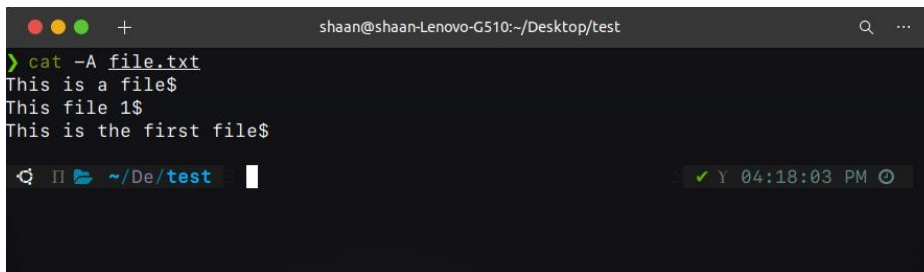
This command will highlight the end of line

A terminal window titled 'shaan@shaan-Lenovo-G510: ~/Desktop/test' showing the command 'cat -E file2.txt' being executed. The output shows three lines of text, each ending with a '\$' symbol to indicate the end of the line. The terminal status bar at the bottom shows a green checkmark, 'Y 04:10:29 PM', and a clock icon.

```
shaan@shaan-Lenovo-G510: ~/Desktop/test
> cat -E file2.txt
Hey there, my name is Shaan 🙋$
Nice to meet you! 😊$
Bye!$
```

## **cat -A file\_name**

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

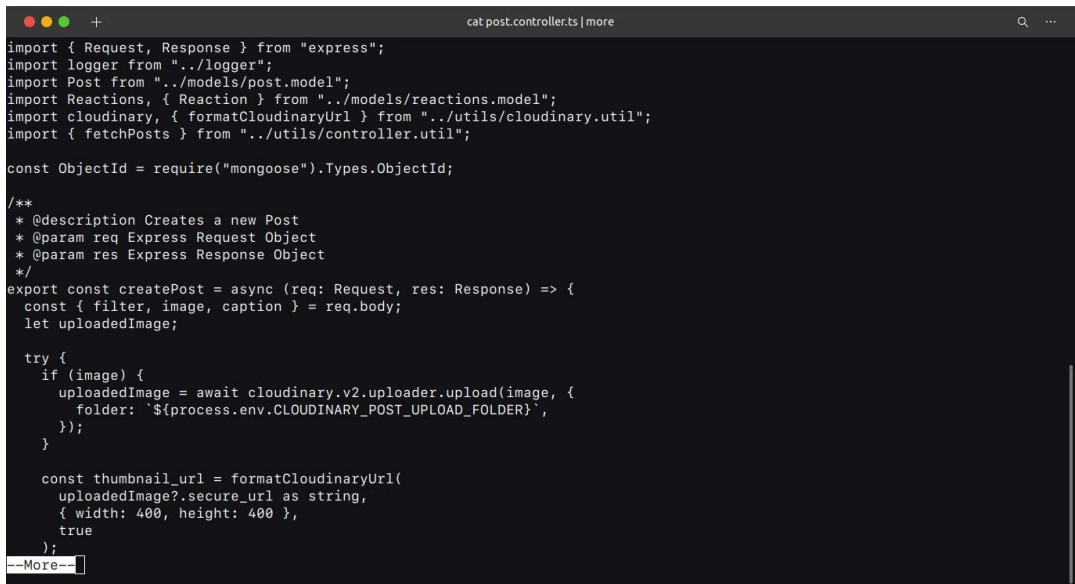
A terminal window titled 'shaan@shaan-Lenovo-G510: ~/Desktop/test'. The command 'cat -A file.txt' has been executed, resulting in the output: 'This is a file\$', 'This file 1\$', and 'This is the first file\$'. The prompt is '~ /De/test'.

```
shaan@shaan-Lenovo-G510: ~/Desktop/test
> cat -A file.txt
This is a file$
This file 1$
This is the first file$
~ /De/test
```

## 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.

A terminal window titled 'cat post.controller.ts | more'. It displays the content of 'post.controller.ts' in chunks, with a 'More' prompt at the bottom of each chunk. The visible code includes imports for express, mongoose, cloudinary, and various utility functions, followed by a 'createPost' function.

```
cat post.controller.ts | more
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.

```
shaan@shaan-Lenovo-G510: ~/Desktop/test
> 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
```

## 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.

```
shaan@shaan-Lenovo-G510: ~/Desktop/test
> 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
```

## who and whoami

### who

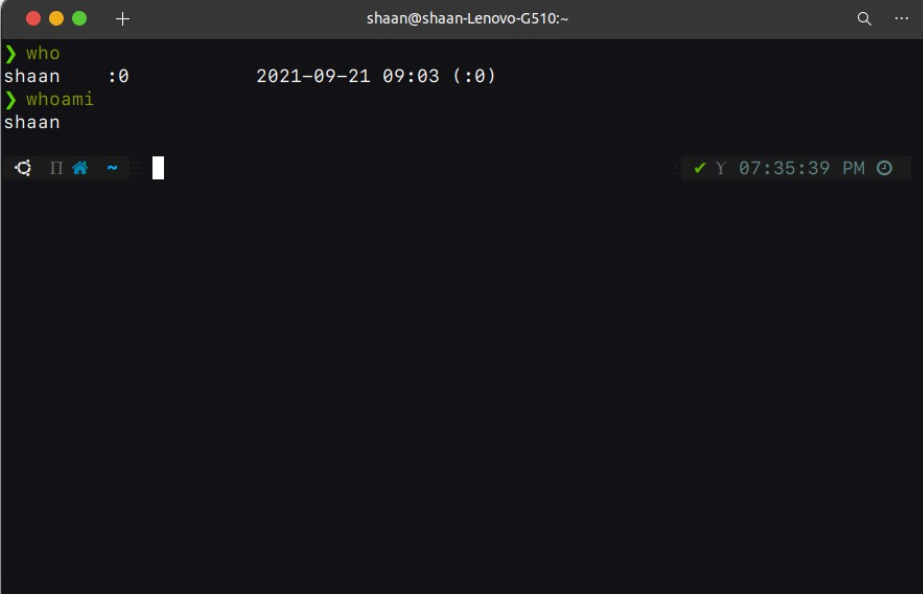
**who** command will print the name of currently logged in user along with some other informations.



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

## **whoami**

**whoami** command prints the user name.

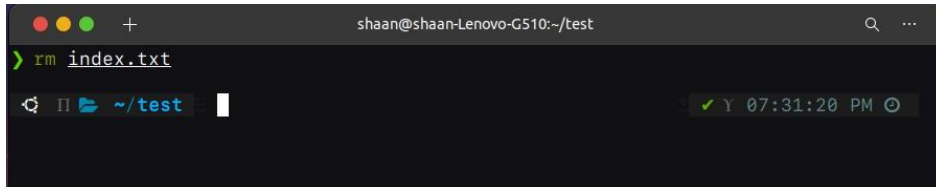
A screenshot of a Linux terminal window. The window title is 'shaan@shaan-Lenovo-G510:~'. The terminal shows the following commands and output:

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

The terminal has a dark background with light green text. At the bottom, there is a status bar showing system icons on the left and a green checkmark, 'Y', the time '07:35:39 PM', and a refresh icon on the right.

## ***rm [filename]***

**rm** command will delete the file specified.

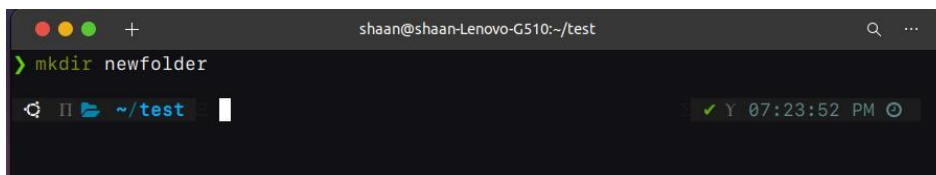
A terminal window with a dark background. The title bar shows 'shaan@shaan-Lenovo-G510:~/test'. The command prompt is '>'. The user has entered 'rm index.txt'. The prompt is now '~/.test'. A status bar at the bottom right shows a green checkmark, 'Y', and the time '07:31:20 PM'.

## **mkdir**

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

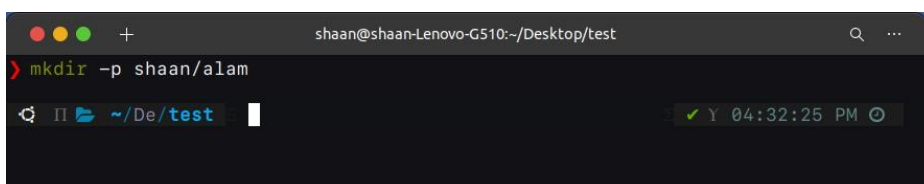
## ***mkdir [dirname]***

This command is used to create a new directory.

A terminal window with a dark background. The title bar shows 'shaan@shaan-Lenovo-G510:~/test'. The command prompt is '>'. The user has entered 'mkdir newfolder'. The prompt is now '~/.test'. A status bar at the bottom right shows a green checkmark, 'Y', and the time '07:23:52 PM'.

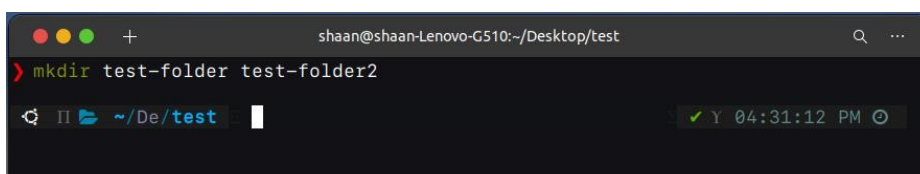
## ***mkdir -p /shaan/alam***

This command will create all required parent directories

A terminal window with a dark background. The title bar shows 'shaan@shaan-Lenovo-G510:~/Desktop/test'. The command prompt is '>'. The user has entered 'mkdir -p shaan/alam'. The prompt is now '~/.test'. A status bar at the bottom right shows a green checkmark, 'Y', and the time '04:32:25 PM'.

## ***mkdir [dirname1] [dirname2]***

This command will create multiple directories at the same time.

A terminal window with a dark background. The title bar shows 'shaan@shaan-Lenovo-G510:~/Desktop/test'. The command prompt is '>'. The user has entered 'mkdir test-folder test-folder2'. The prompt is now '~/.test'. A status bar at the bottom right shows a green checkmark, 'Y', and the time '04:31:12 PM'.

---

## **rmdir**

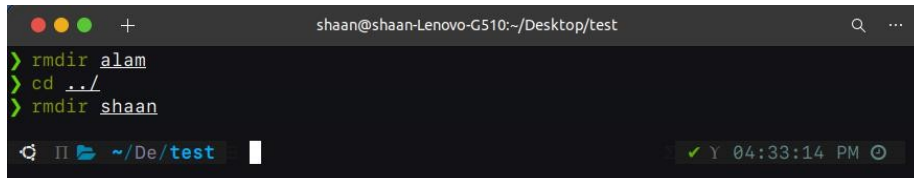
---

This command is used to delete directories specified.

### **rmdir [dirname]**

---

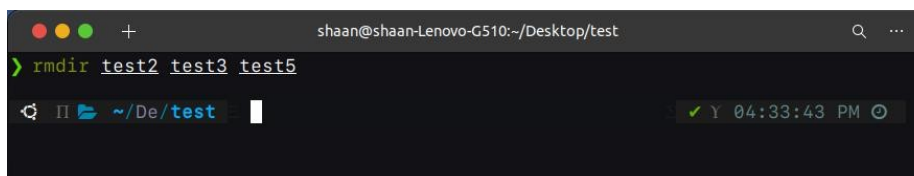
This comand will delete a single directory specified.

A terminal window titled 'shaan@shaan-Lenovo-G510:~/Desktop/test' showing the execution of the 'rmdir' command. The user enters 'rmdir alam', then 'cd ../', and finally 'rmdir shaan'. The prompt changes to '~/.De/test' after the second command. A status bar at the bottom right shows a green checkmark, 'Y', and the time '04:33:14 PM'.

```
shaan@shaan-Lenovo-G510:~/Desktop/test
> rmdir alam
> cd ../
> rmdir shaan
~/.De/test
```

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

This command will delete multiple directories specified.

A terminal window titled 'shaan@shaan-Lenovo-G510:~/Desktop/test' showing the execution of the 'rmdir' command with multiple arguments. The user enters 'rmdir test2 test3 test5'. The prompt is '~/.De/test'. A status bar at the bottom right shows a green checkmark, 'Y', and the time '04:33:43 PM'.

```
shaan@shaan-Lenovo-G510:~/Desktop/test
> rmdir test2 test3 test5
~/.De/test
```

---

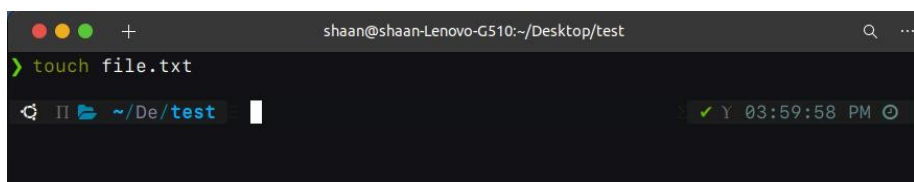
## **touch**

---

**touch** command is used to create new files.

### **touch [filename]**

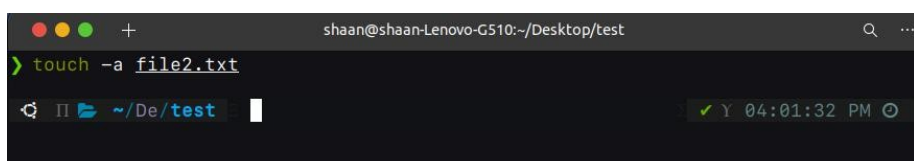
This command will create a new file.

A terminal window titled 'shaan@shaan-Lenovo-G510:~/Desktop/test' showing the execution of the 'touch' command. The user enters 'touch file.txt'. The prompt is '~/.De/test'. A status bar at the bottom right shows a green checkmark, 'Y', and the time '03:59:58 PM'.

```
shaan@shaan-Lenovo-G510:~/Desktop/test
> touch file.txt
~/.De/test
```

### **touch -a [filename]**

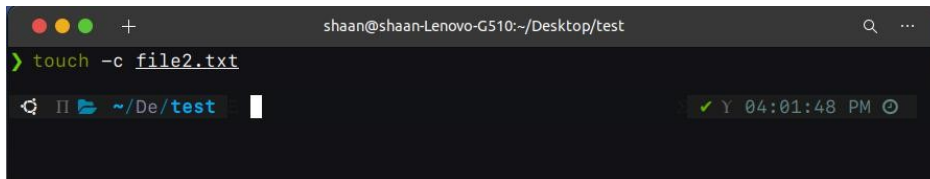
This command is used to change access time only.

A terminal window titled 'shaan@shaan-Lenovo-G510:~/Desktop/test' showing the execution of the 'touch -a' command. The user enters 'touch -a file2.txt'. The prompt is '~/.De/test'. A status bar at the bottom right shows a green checkmark, 'Y', and the time '04:01:32 PM'.

```
shaan@shaan-Lenovo-G510:~/Desktop/test
> touch -a file2.txt
~/.De/test
```

### **touch -c [filename]**

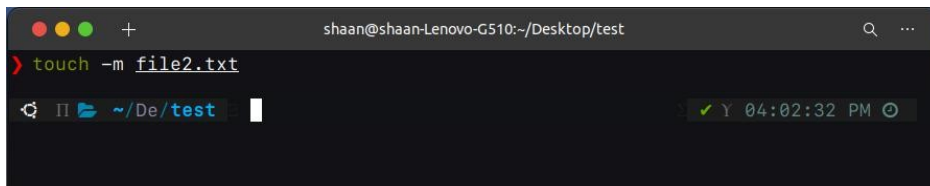
This command is used to check whether a file is created or not. If not created then don't create it.



```
shaan@shaan-Lenovo-G510:~/Desktop/test
> touch -c file2.txt
```

## **touch -m [filename]**

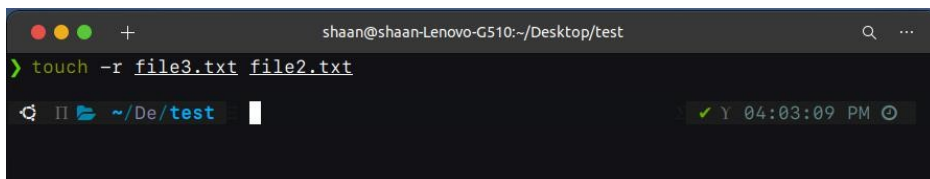
This is used to change the modification time only. It only updates last modification time.



```
shaan@shaan-Lenovo-G510:~/Desktop/test
> touch -m file2.txt
```

## **touch -r [filename]**

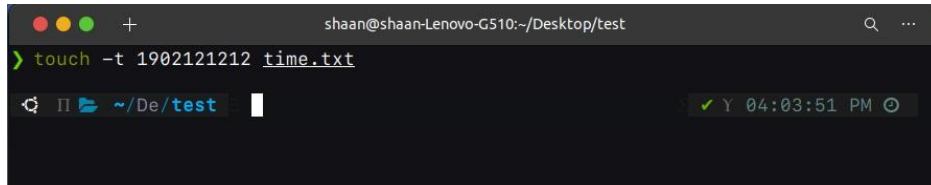
This command is used to use the timestamp of another file.



```
shaan@shaan-Lenovo-G510:~/Desktop/test
> touch -r file3.txt file2.txt
```

## **touch -t [filename]**

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



```
shaan@shaan-Lenovo-G510: ~/Desktop/test
> touch -t 1902121212 time.txt
```

The terminal window shows the command `touch -t 1902121212 time.txt` being executed. The prompt is `shaan@shaan-Lenovo-G510: ~/Desktop/test`. The status bar at the bottom indicates the time is 04:03:51 PM.

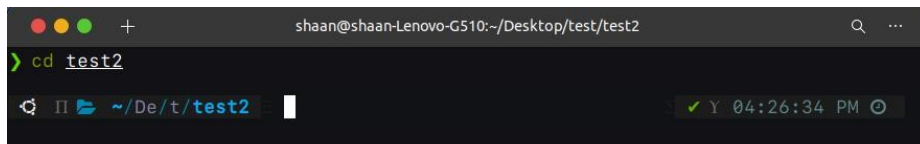
---

## **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 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`)

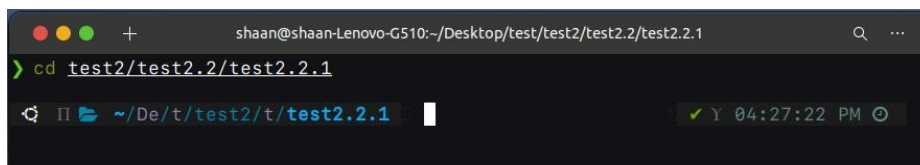


```
shaan@shaan-Lenovo-G510: ~/Desktop/test/test2
> cd test2
```

The terminal window shows the command `cd test2` being executed. The prompt is `shaan@shaan-Lenovo-G510: ~/Desktop/test/test2`. The status bar at the bottom indicates the time is 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`)

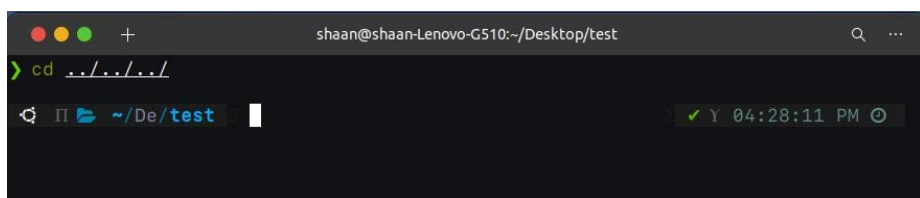


```
shaan@shaan-Lenovo-G510: ~/Desktop/test/test2/test2.2/test2.2.1
> cd test2/test2.2/test2.2.1
```

The terminal window shows the command `cd test2/test2.2/test2.2.1` being executed. The prompt is `shaan@shaan-Lenovo-G510: ~/Desktop/test/test2/test2.2/test2.2.1`. The status bar at the bottom indicates the time is 04:27:22 PM.

### **cd ../**

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



```
shaan@shaan-Lenovo-G510: ~/Desktop/test
> cd ../../../../
```

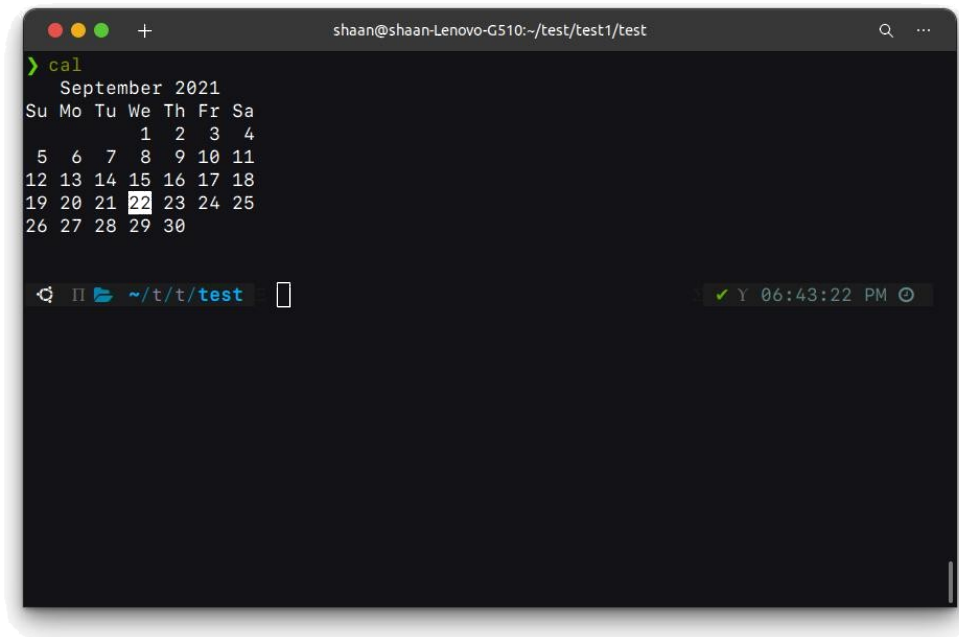
The terminal window shows the command `cd ../../../../` being executed. The prompt is `shaan@shaan-Lenovo-G510: ~/Desktop/test`. The status bar at the bottom indicates the time is 04:28:11 PM.

## 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.



```
shaan@shaan-Lenovo-G510:~/test/test$ 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
```

The screenshot shows a terminal window with the title bar "shaan@shaan-Lenovo-G510:~/test/test". The command "cal" has been executed, resulting in a calendar for September 2021. The calendar is displayed in a grid format with days of the week as headers. The date 22 is highlighted. At the bottom of the terminal, there is a status bar showing a green checkmark, the letter 'Y', the time "06:43:22 PM", and a circular icon.

## **cal [year]**

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

```
shaan@shaan-Lenovo-G510:~/test/test1/test
> 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
 2  3  4  5  6  7  8    1  2  3  4  5    1  2  3  4  5
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   6  7  8  9 10 11 12
16 17 18 19 20 21 22   20 21 22 23 24 25 26   13 14 15 16 17 18 19
23 24 25 26 27 28 29   27 28                20 21 22 23 24 25 26
30 31                27 28 29 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  2    1  2  3  4  5    1  2  3
```

## **cal [month] [year]**

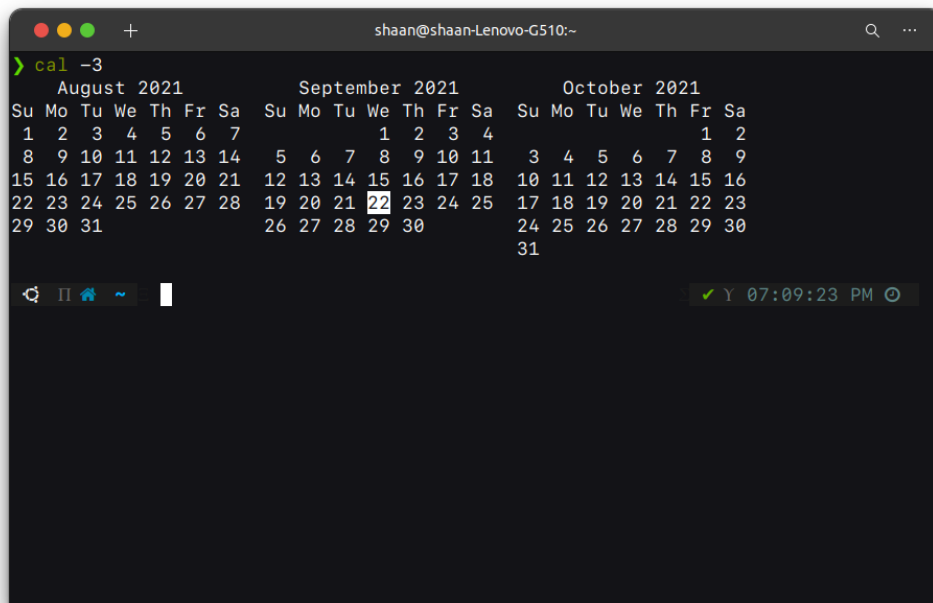
This command will print the calendar for the specified month of the year.

```
shaan@shaan-Lenovo-G510:~
> 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

✓ Y 07:10:30 PM
```

## cal -3

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



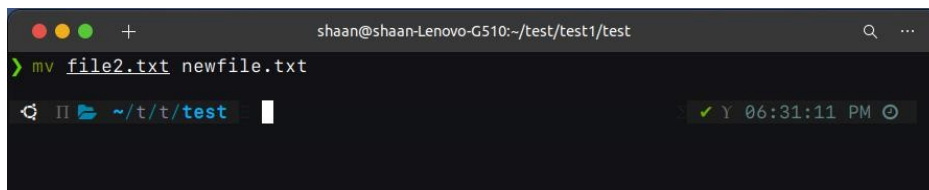
```
> cal -3
      August 2021          September 2021          October 2021
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7    1  2  3  4          1  2
 8  9 10 11 12 13 14    5  6  7  8  9 10 11    3  4  5  6  7  8  9
15 16 17 18 19 20 21   12 13 14 15 16 17 18   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
```

## 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*



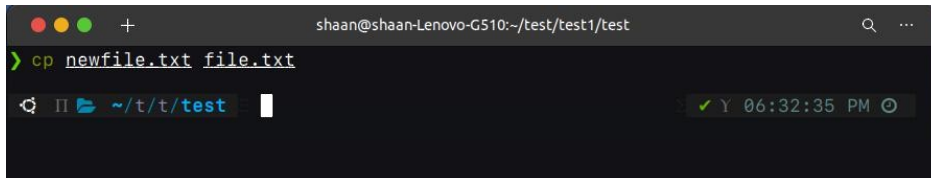
```
shaan@shaan-Lenovo-G510:~/test/test1/test
> mv file2.txt newfile.txt
```



## cp

---

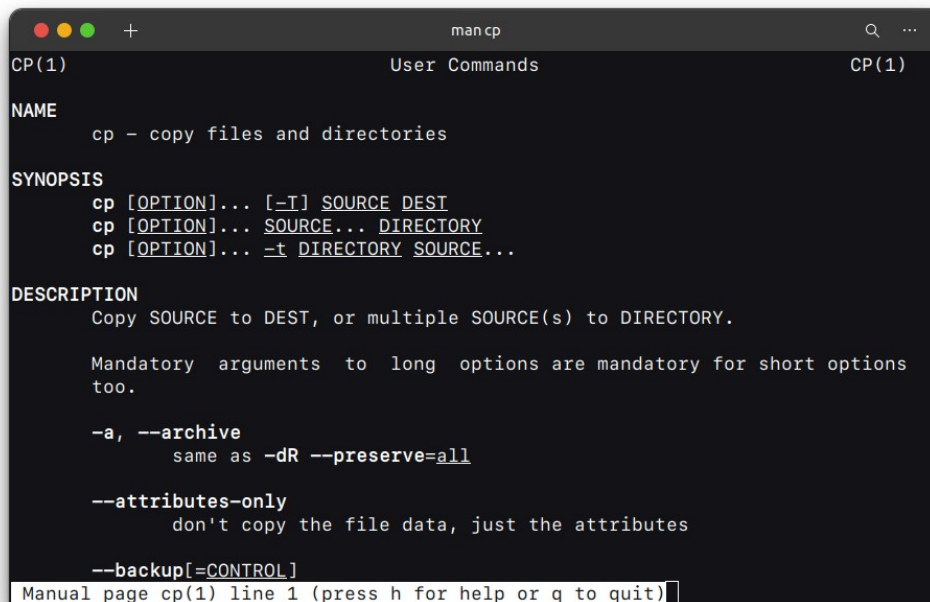
This command is used to copy a file. The syntax is *cp file\_to\_be\_copied new\_file\_name*

A terminal window with a dark background. The title bar shows 'shaan@shaan-Lenovo-G510:~/test/test1/test'. The command prompt is '>'. The command 'cp newfile.txt file.txt' has been entered and executed. The status bar at the bottom shows '~/t/t/test' and a green checkmark with the time '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.

A terminal window showing the output of the 'man cp' command. The title bar is 'man cp'. The output is as follows:  
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

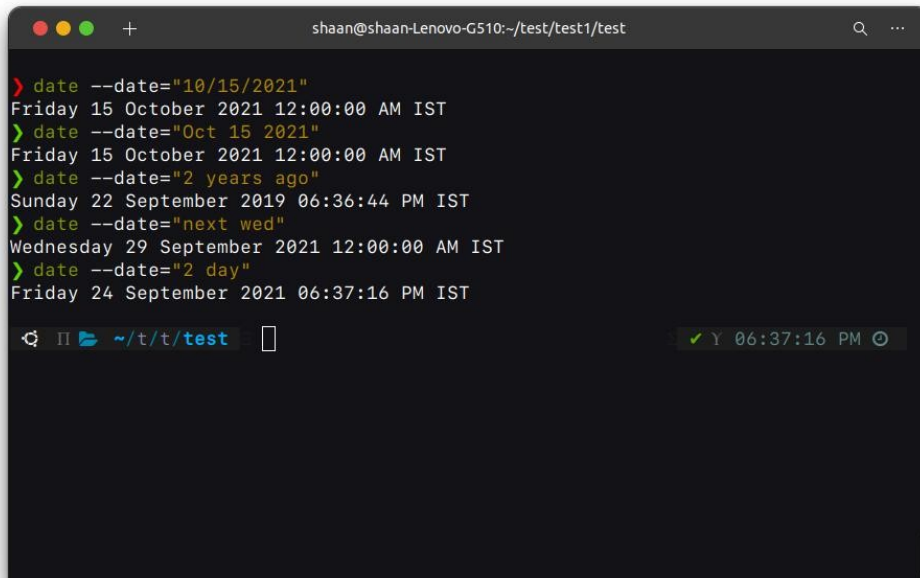
---

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

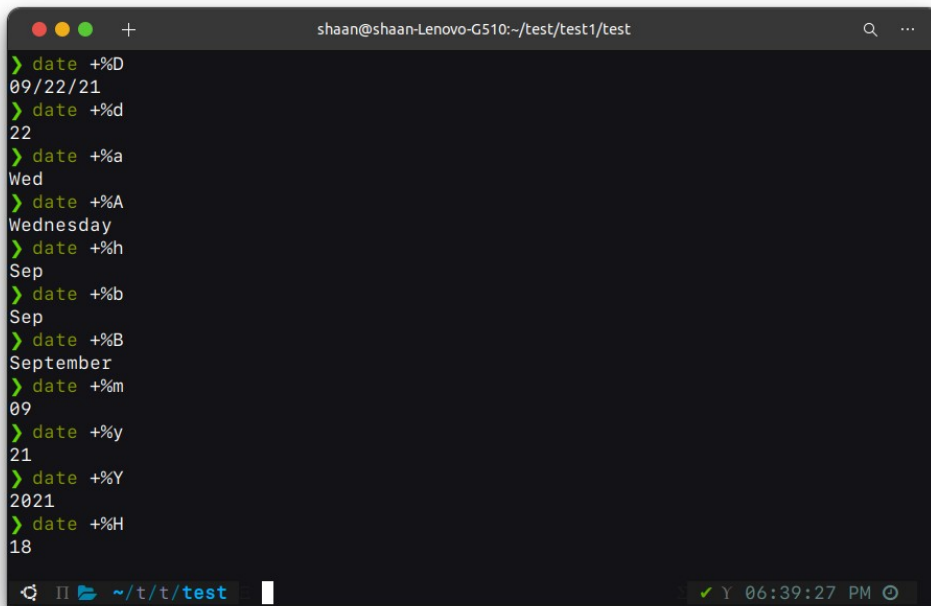
---

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

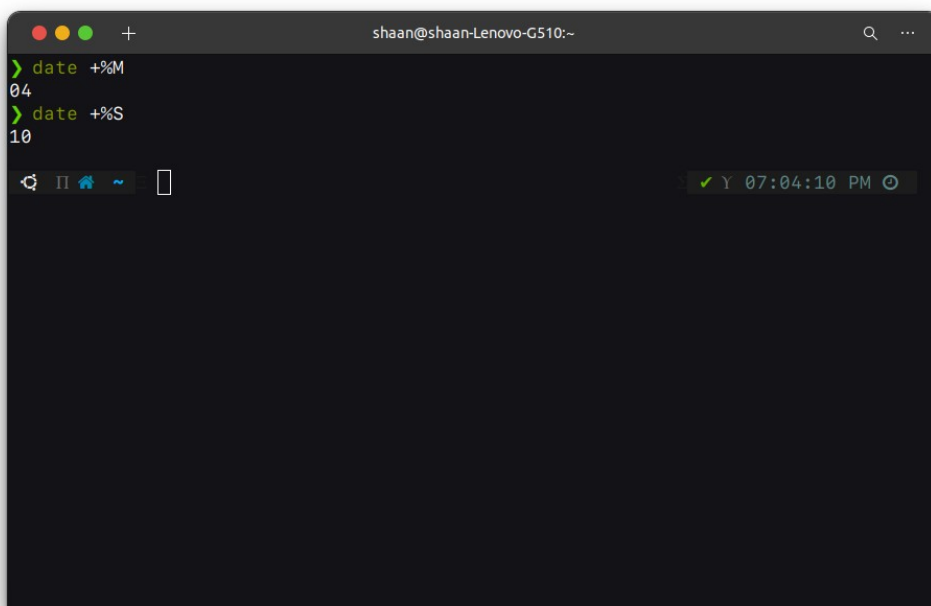
A terminal window titled 'shaan@shaan-Lenovo-G510:~/test/test1/test' showing the execution of the 'date' command with various flags. The output shows dates for '10/15/2021', 'Oct 15 2021', '2 years ago', 'next wed', and '2 day'. The terminal has a dark background with green and yellow text. The window title bar shows standard Linux window controls and the current directory path. The status bar at the bottom shows a green checkmark, the user 'Y', and the time '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.



```
shaan@shaan-Lenovo-G510: ~/test1/test
> 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
```



```
shaan@shaan-Lenovo-G510: ~
> 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)		

+

shaan@shaan-Lenovo-G510:~/test/test1/test

Q ...

> chmod +x file.txt

> ls -la file.txt

-rwxrwxr-x 1 shaan shaan 12 Sep 16 19:07 file.txt

~/t/t/test

Y 08:06:01 PM

+

shaan@shaan-Lenovo-G510:~/test/test1/test

Q ...

&gt; chmod g+x file.txt

&gt; ls -la file.txt

-rwxrwxr-x 1 shaan shaan 12 Sep 16 19:07 file.txt

~/t/t/test

Y 08:06:41 PM

+

shaan@shaan-Lenovo-G510:~/test/test1/test

Q ...

&gt; chmod o+x file.txt

&gt; ls -la file.txt

-rwxrwxr-x 1 shaan shaan 12 Sep 16 19:07 file.txt

~/t/t/test

Y 08:06:57 PM

+

shaan@shaan-Lenovo-G510:~/test/test1/test

Q ...

&gt; chmod +r file.txt

&gt; ls -la file.txt

-rw-rw-r-- 1 shaan shaan 12 Sep 16 19:07 file.txt

~/t/t/test

Y 08:05:46 PM

+

shaan@shaan-Lenovo-G510:~/test/test1/test

Q ...

&gt; chmod u+x file.txt

&gt; ls -la file.txt

-rwxrwxr-x 1 shaan shaan 12 Sep 16 19:07 file.txt

~/t/t/test

Y 08:06:23 PM

+

shaan@shaan-Lenovo-G510:~/test/test1/test

Q ...

&gt; chmod +w file.txt

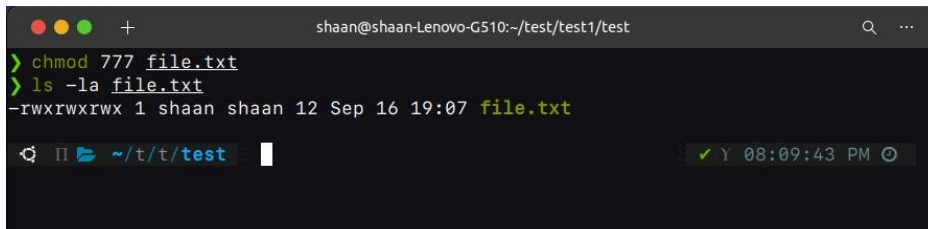
&gt; ls -la file.txt

-rw-rw-r-- 1 shaan shaan 12 Sep 16 19:07 file.txt

~/t/t/test

Y 08:05:29 PM

## Using chmod with Absolute permission

A terminal window with a dark background. The title bar shows 'shaan@shaan-Lenovo-G510: ~/test/test1/test'. The prompt is '>'. The user enters 'chmod 777 file.txt'. The prompt changes to '>'. The user enters 'ls -la file.txt'. The output is '-rwxrwxrwx 1 shaan shaan 12 Sep 16 19:07 file.txt'. The bottom status bar shows a search icon, a window icon, the path '~/t/t/test', and a green checkmark with the time '08:09:43 PM' and a refresh icon.

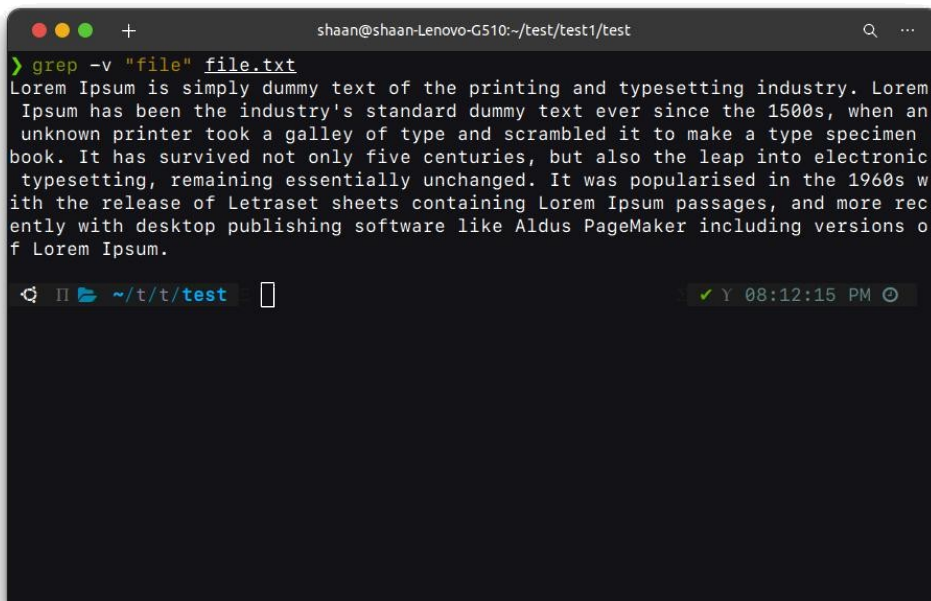
```
> chmod 777 file.txt
> ls -la file.txt
-rwxrwxrwx 1 shaan shaan 12 Sep 16 19:07 file.txt
```

## grep

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

### grep -v [pattern] [filename]

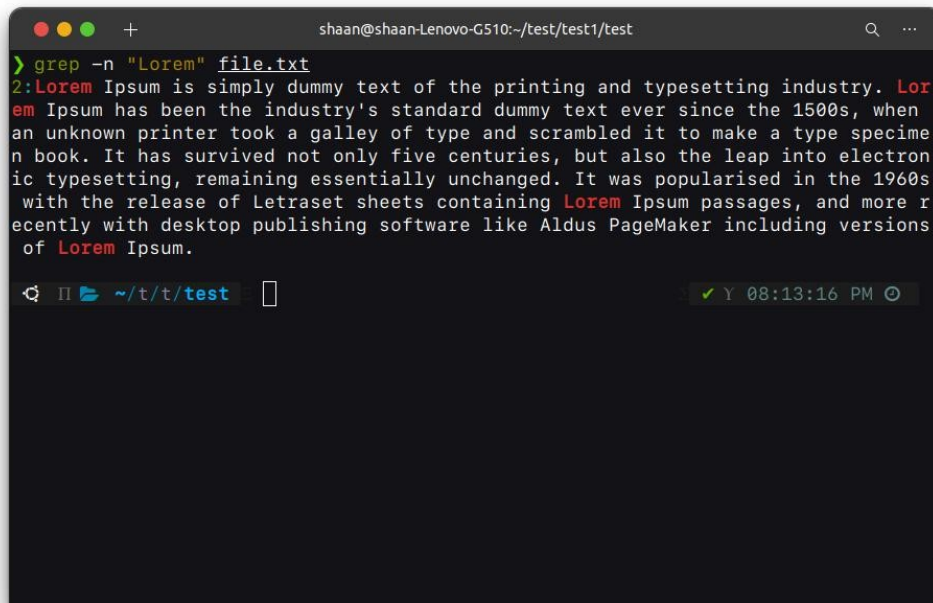
Print all lines that do not match pattern.

A terminal window with a dark background. The title bar shows 'shaan@shaan-Lenovo-G510: ~/test/test1/test'. The prompt is '>'. The user enters 'grep -v "file" file.txt'. The output is a block of Lorem Ipsum text. The bottom status bar shows a search icon, a window icon, the path '~/t/t/test', and a green checkmark with the time '08:12:15 PM' and a refresh icon.

```
> 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.
```

## **grep -n [pattern] [filename]**

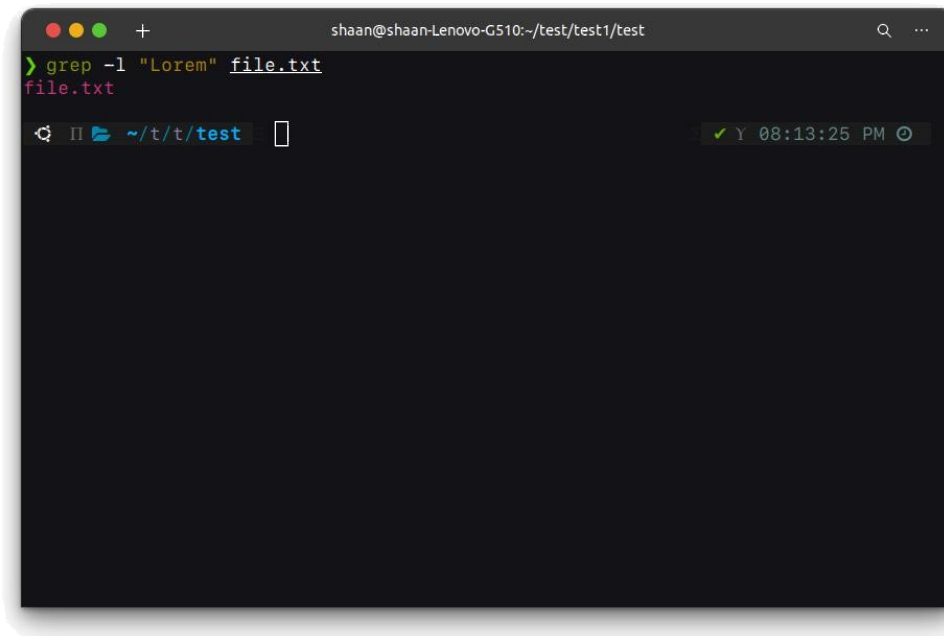
Print the matched line and its line number.

A terminal window with a dark background. The title bar shows 'shaan@shaan-Lenovo-G510:~/test/test1/test'. The command prompt is '>'. The user has entered 'grep -n "Lorem" file.txt'. The output shows line 2 with the text '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 with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.'

```
> grep -n "Lorem" file.txt
2: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 with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.
```

## **grep -l [pattern] [filename]**

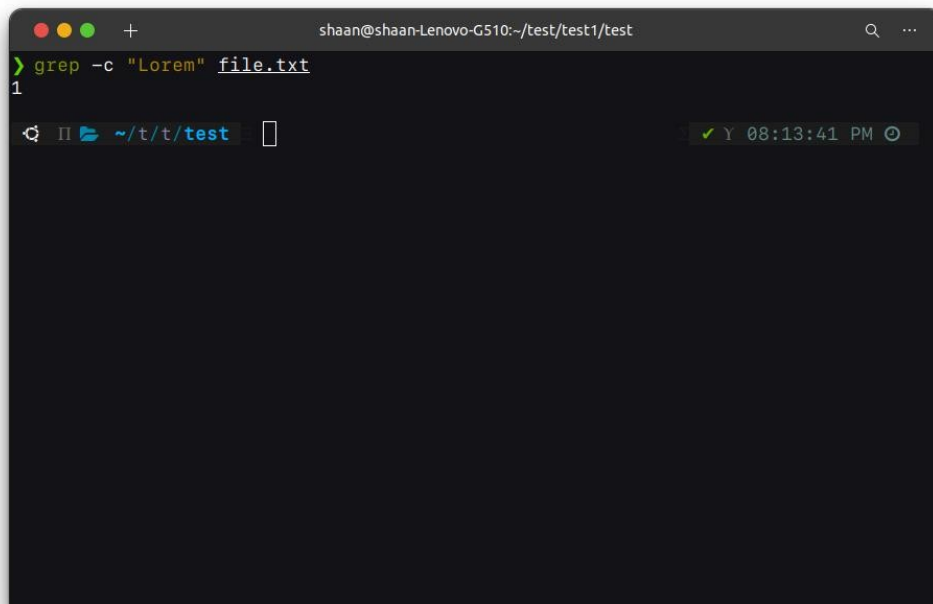
Print only the names of files with matching lines.

A terminal window with a dark background. The title bar shows 'shaan@shaan-Lenovo-G510:~/test/test1/test'. The command prompt is '>'. The user has entered 'grep -l "Lorem" file.txt'. The output shows 'file.txt' in red text.

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

## **grep -c [pattern] [filename]**

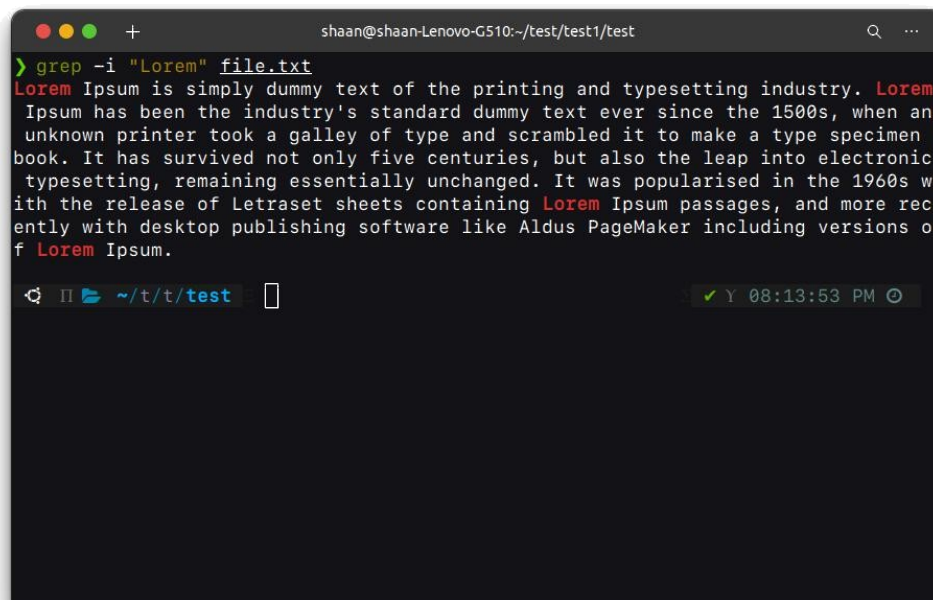
Print only the count of matching lines.

A terminal window with a dark background. The title bar shows 'shaan@shaan-Lenovo-G510: ~/test/test1/test'. The command prompt is '>'. The user has entered 'grep -c "Lorem" file.txt'. The output is '1'. The terminal status bar at the bottom shows '~/t/t/test' and a green checkmark with the time '08:13:41 PM'.

```
> grep -c "Lorem" file.txt
1
```

## **grep -i [pattern] [filename]**

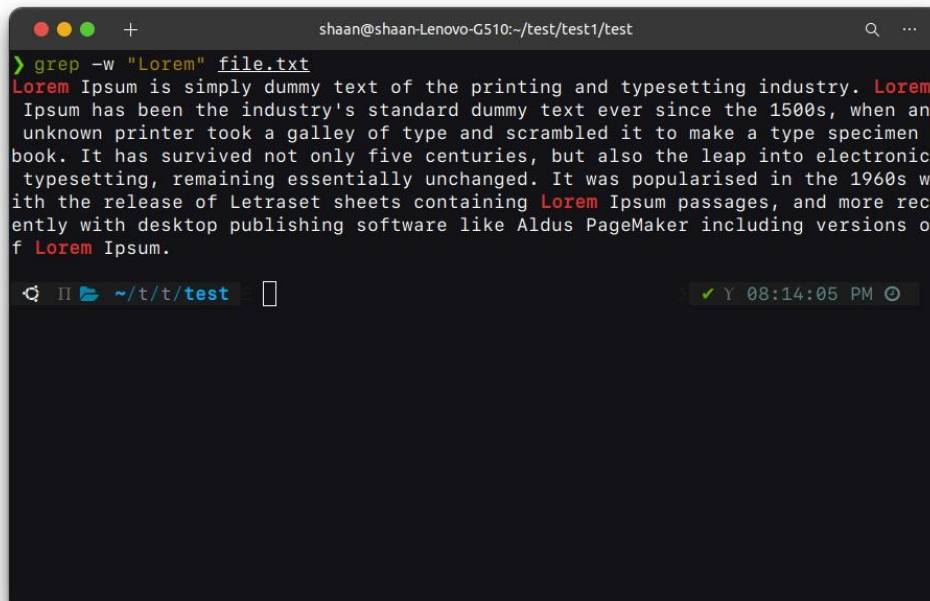
Match either upper- or lowercase.

A terminal window with a dark background. The title bar shows 'shaan@shaan-Lenovo-G510: ~/test/test1/test'. The command prompt is '>'. The user has entered 'grep -i "Lorem" file.txt'. The output is a paragraph of Lorem Ipsum text where the word 'Lorem' is highlighted in red. The terminal status bar at the bottom shows '~/t/t/test' and a green checkmark with the time '08:13:53 PM'.

```
> 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.
```

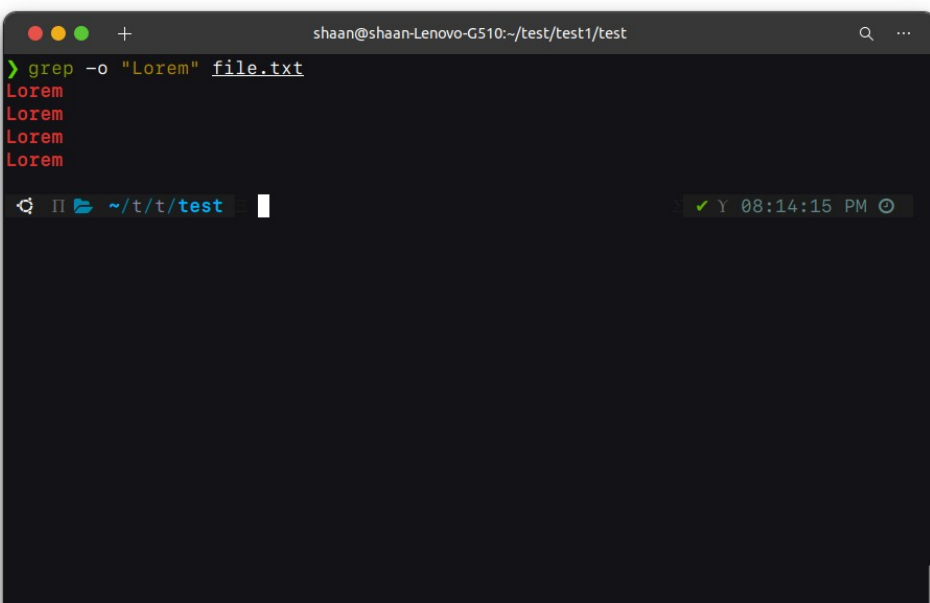
## **grep -w [pattern] [filename]**

Match whole word

A terminal window with a dark background. The title bar shows 'shaan@shaan-Lenovo-G510: ~/test/test1/test'. The command prompt is '> grep -w "Lorem" file.txt'. The output shows the word 'Lorem' in red at the start of two lines in a Lorem Ipsum text block. The terminal has standard window controls and a status bar at the bottom showing 'Y 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.

A terminal window with a dark background. The title bar shows 'shaan@shaan-Lenovo-G510: ~/test/test1/test'. The command prompt is '> grep -o "Lorem" file.txt'. The output shows the word 'Lorem' in red on four separate lines. The terminal has standard window controls and a status bar at the bottom showing 'Y 08:14:15 PM'.



## bc

---

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

A screenshot of a terminal window with a dark background. The window title bar shows standard macOS window controls (red, yellow, green buttons and a plus sign) and the text 'bc'. The terminal content shows the execution of the 'bc' command, displaying version information and copyright details. Subsequent arithmetic operations are entered and their results are shown: '2+2' results in '4', '5+6' results in '11', 'scale=2' followed by '135/4' results in '33.75', and 'scale=1' followed by '25/8' results in '3.1'. A cursor is visible at the end of the last line.

```
> 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
█
```