The Ls command is used to get a list of files and directories. Options can be used to get additional information about the files.

Know ls command syntax and options with practical examples and output.

ls Command in Unix with Examples

ls Syntax:

ls [options] [paths]

The ls command supports the following options:

ls -a: list all files including hidden files. These are files that start with “.”.

ls -A: list all files including hidden files except for “.” and “..” – these refer to the entries for the current directory, and for the parent directory.

ls -R: list all files recursively, descending down the directory tree from the given path.

ls -l: list the files in long format i.e. with an index number, owner name, group name, size, and permissions.

ls – o: list the files in long format but without the group name.

ls -g: list the files in long format but without the owner name.

ls -i: list the files along with their index number.

ls -s: list the files along with their size.

ls -t: sort the list by time of modification, with the newest at the top.

ls -S: sort the list by size, with the largest at the top.

ls -r: reverse the sorting order.

Examples:

List all non-hidden files in the current directory

$ ls

E.g:

dir1 dir2 file1 file2

List all the files including hidden files in the current directory

$ ls -a

E.g:

.. ... .... .hfile dir1 dir2 file1 file2

List all the files including hidden files in the current directory

$ ls -al

E.g:

total 24

drwxr-xr-x 7 user staff 224 Jun 21 15:04 .

drwxrwxrwx 18 user staff 576 Jun 21 15: 02.

-rw-r--r-- 1 user staff 6 Jun 21 15:04 .hfile

drwxr-xr-x 3 user staff 96 Jun 21 15:08 dir1

drwxr-xr-x 2 user staff 64 Jun 21 15:04 dir2

-rw-r--r-- 1 user staff 6 Jun 21 15:04 file1

-rw-r--r-- 1 user staff 4 Jun 21 15:08 file2

List all the files in the current directory in long format, sorted by modification time, oldest first

$ ls -lrt

E.g:

total 16

-rw-r--r-- 1 user staff 6 Jun 21 15:04 file1

drwxr-xr-x 2 user staff 64 Jun 21 15:04 dir2

-rw-r--r-- 1 user staff 4 Jun 21 15:08 file2

drwxr-xr-x 3 user staff 96 Jun 21 15:08 dir1

List all the files in the current directory in long format, sorted by size, smallest first

$ ls -lrS

E.g:

total 16

-rw-r--r-- 1 user staff 4 Jun 21 15:08 file2

-rw-r--r-- 1 user staff 6 Jun 21 15:04 file1

drwxr-xr-x 2 user staff 64 Jun 21 15:04 dir2

drwxr-xr-x 3 user staff 96 Jun 21 15:08 dir1

List all the files recursively from the current directory

$ ls -R

E.g:

dir1 dir2 file1 file2

./dir1:

file3

./dir2:

date command in Linux with examples

**date**command is used to display the system date and time. date command is also used to set date and time of the system. By default the date command displays the date in the time zone on which unix/linux operating system is configured.You must be the super-user (root) to change the date and time.

**Syntax:**

date [OPTION]... [+FORMAT]

date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

**Options with Examples**  
1: **date (no option) :**With no options, the date command displays the current date and time, including the abbreviated day name, abbreviated month name, day of the month, the time separated by colons, the time zone name, and the year.

**Command:**

$date

**Output:**

Tue Oct 10 22:55:01 PDT 2017

**Note :**Here unix system is configured in pacific daylight time.

2:**-u Option:**Displays the time in GMT(Greenwich Mean Time)/UTC(Coordinated Universal Time )time zone.

**Command:**

$date -u

**Output :**

Wed Oct 11 06:11:31 UTC 2017

3: **–date or -d Option:**Displays the given date string in the format of date. But this will not affect the system’s actual date and time value.Rather it uses the date and time given in the form of string.  
**Syntax:**

**$date --date=" string "**

**Command:**

$date --date="2/02/2010"

$date --date="Feb 2 2010"

**Output:**

Tue Feb 2 00:00:00 PST 2010

Tue Feb 2 00:00:00 PST 2010

4:**Using –date option for displaying past dates:**

 Date and time of 2 years ago.

**Command:**

$date --date="2 year ago"

**Output:**

Sat Oct 10 23:42:15 PDT 2015

 Date and time of 5 seconds ago.

**Command:**

$date --date="5 sec ago"

**Output:**

Tue Oct 10 23:45:02 PDT 2017

 Date and time of previous day.

**Command:**

$date --date="yesterday"

**Output:**

Mon Oct 9 23:48:00 PDT 2017

 Date and time of 2 months ago.

**Command:**

$date --date="2 month ago"

**Output:**

Thu Aug 10 23:54:51 PDT 2017

 Date and time of 10 days ago.

**Command:**

$date --date="10 day ago"

**Output:**

Sat Sep 30 23:56:55 PDT 2017

5:**Using –date option for displaying future date:**

 Date and time of upcoming particular week day.

**Command:**

$date --date="next tue"

**Output:**

Tue Oct 17 00:00:00 PDT 2017

 Date and time after two days.

**Command:**

$date --date="2 day"

**Output:**

Fri Oct 13 00:05:52 PDT 2017

 Date and time of next day.

**Command:**

$date --date="tomorrow"

**Output:**

Thu Oct 12 00:08:47 PDT 2017

 Date and time after 1 year on the current day.

**Command:**

$date --date="1 year"

**Output:**

Thu Oct 11 00:11:38 PDT 2018

6:**-s or –set Option:**To set the system date and time -s or –set option is used.  
**Syntax:**

**$date --set="date to be set"**

**Command:**

$date

**Output:**

Wed Oct 11 15:23:26 PDT 2017

**Command:**

$date --set="Tue Nov 13 15:23:34 PDT 2018"

$date

**Output:**

Tue Nov 13 15:23:34 PDT 2018

7:**–file or -f Option:**This is used to display the date string present at each line of file in the date and time format.This option is similar to –date option but the only difference is that in –date we can only give one date string but in a file we can give multiple date strings at each line.  
**Syntax:**

**$date --file=file.txt**

$cat >> datefile

Sep 23 2018

Nov 03 2019

**Command:**

$date --file=datefile

**Output:**

Sun Sep 23 00:00:00 PDT 2018

Sun Nov 3 00:00:00 PDT 2019

8:**-r Option:**This is used to display the last modified timestamp of a datefile .  
**Syntax:**

**$date -r file.txt**

We can modify the timestamp of a datefile by using touch command.

$touch datefile

$date

Wed Oct 11 15:54:18 PDT 2017

//this is the current date and time

$touch datefile

//The timestamp of datefile is changed using touch command.

This was done few seconds after the above date command’s output.

$date

Wed Oct 11 15:56:23 PDT 2017

//display last modified time of datefile

9: **List of Format specifiers used with date command:**

**%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.

**Syntax:**

**$date +%[format-option]**

**Examples:**

**Command:**

$date "+%D"

**Output:**

10/11/17

**Command:**

$date "+%D %T"

**Output:**

10/11/17 16:13:27

**Command:**

$date "+%Y-%m-%d"

**Output:**

2017-10-11

**Command:**

$date "+%Y/%m/%d"

**Output:**

2017/10/11

**Command:**

$date "+%A %B %d %T %y"

**Output:**

Thursday October 07:54:29 17

# Head command in Linux with examples

It is the complementary of [Tail](https://www.geeksforgeeks.org/tail-command-linux-examples/) command. The head command, as the name implies, print the top N number of data of the given input. By default, it prints the first 10 lines of the specified files. If more than one file name is provided then data from each file is preceded by its file name.

**Syntax:**

head [OPTION]... [FILE]...

Let us consider two files having name **state.txt** and **capital.txt** contains all the names of the Indian states and capitals respectively.

**$ cat state.txt**

Andhra Pradesh

Arunachal Pradesh

Assam

Bihar

Chhattisgarh

Goa

Gujarat

Haryana

Himachal Pradesh

Jammu and Kashmir

Jharkhand

Karnataka

Kerala

Madhya Pradesh

Maharashtra

Manipur

Meghalaya

Mizoram

Nagaland

Odisha

Punjab

Rajasthan

Sikkim

Tamil Nadu

Telangana

Tripura

Uttar Pradesh

Uttarakhand

West Bengal

Without any option, it displays only the first 10 lines of the file specified.  
Example:

**$ head state.txt**

Andhra Pradesh

Arunachal Pradesh

Assam

Bihar

Chhattisgarh

Goa

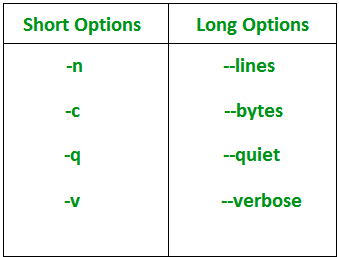
Gujarat

Haryana

Himachal Pradesh

Jammu and Kashmir

**Options**



**1. -n num:**Prints the first ‘num’ lines instead of first 10 lines. **num** is mandatory to be specified in command otherwise it displays an error.

**$ head -n 5 state.txt**

Andhra Pradesh

Arunachal Pradesh

Assam

Bihar

Chhattisgarh

**2. -c num:**Prints the first ‘num’ bytes from the file specified. Newline count as a single character, so if head prints out a newline, it will count it as a byte. **num** is mandatory to be specified in command otherwise displays an error.

**$ head -c 6 state.txt**

Andhra

**3. -q:**It is used if more than 1 file is given. Because of this command, data from each file is not precedes by its file name.

**Without using -q option**

==> state.txt capital.txt <==

Hyderabad

Itanagar

Dispur

Patna

Raipur

Panaji

Gandhinagar

Chandigarh

Shimla

Srinagar

**With using -q option**

**$ head -q state.txt capital.txt**

Andhra Pradesh

Arunachal Pradesh

Assam

Bihar

Chhattisgarh

Goa

Gujarat

Haryana

Himachal Pradesh

Jammu and Kashmir

Hyderabad

Itanagar

Dispur

Patna

Raipur

Panaji

Gandhinagar

Chandigarh

Shimla

Srinagar

**4. -v:**By using this option, data from the specified file is always preceded by its file name.

**$ head -v state.txt**

==> state.txt <==

Andhra Pradesh

Arunachal Pradesh

Assam

Bihar

Chhattisgarh

Goa

Gujarat

Haryana

Himachal Pradesh

Jammu and Kashmir

**Applications of head Command**

1. **Print line between M and N lines:** For this purpose we use head, tail and pipeline(|) commands. Command is: **head -M file\_name | tail -(M-N)**, since the first line takes first M lines and tail command cuts (M-N)Lines starting from the end. Let say from state.txt file we have to print lines between 10 and 20.
2. **$ head -n 20 state.txt | tail -10**
3. Jharkhand
4. Karnataka
5. Kerala
6. Madhya Pradesh
7. Maharashtra
8. Manipur
9. Meghalaya
10. Mizoram
11. Nagaland
12. Odisha
13. **How to use the head with pipeline(|):** The head command can be piped with other commands. In the following example, the output of the ls command is piped to head to show only the three most recently modified files or folders.
14. Display all recently modified or recently used files.
15. **$ ls -t**
16. e.txt
17. d.txt
18. c.txt
19. b.txt
20. a.txt
21. Cut three most recently used file.
22. **$ ls -t | head -n 3**
23. e.txt
24. d.txt
25. c.txt

It can also be piped with one or more filters for additional processing. For example, the sort filter could be used to sort the three most recently used files or folders in the alphabetic order.

**$ ls -t | head -n 3 | sort**

c.txt

d.txt

e.txt

There are number of other filters or commands along which we use head command. Mainly, it can be used for viewing huge log files in Unix.