

date

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 the 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]]

Various Format Options:

- 1) date +%D → To display only date in the form: mm/dd/yy
- 2) date +%T → To display only time in the form: hh:mm:ss
- 3) date +%d → To display only day value
- 4) date +%m → To display only month value
- 5) date +%y → To display only year value in yy form
- 6) date +%Y → To display only year value in yyyy form.
- 7) date +%H → To display only Hours value (in 24 hours scale format)
- 8) date +%M → To display only Minutes value
- 9) date +%S → To display only Seconds value
- 10) date +%A → To display full day name ex:- Sunday
- 11) date +%a → abbreviated weekday name ex:- sun
- 12) date +%B → to full month name ex:- January
- 13) date +%b → abbreviated month name ex :- jan
- 14) date +%w → To display day of the week (0..6), 0 is Sunday
- 15) date +%W → To display week number of year
- 16) date +%q → To display quarter of year
- 17) date +%j → To display day of year (1----> 366)
- 18) date +%I → to display hour in 12 hour format (01..12)
- 19) date +%r → 12-hour clock format ex :- 11:11:04 PM

You can also combine various options such as:

- 1) \$date "+%D %T" → To display date and time
- 2) \$date "+%Y-%m-%d" → yy-mm-dd
- 3) \$date "+%Y/%m/%d" → yy/mm/dd

Date Flags :-

- 1) date -u → to display UTC time (universal standard)
- 2) date -s "string" → to set date as given string.
- 3) date --date="string" / date -d "string" → to display past and future date time
- 4) date -r file.txt → Last modified time.
- 5) TZ = xxxxx date → changes Timezone to xxxxx.

Date String Examples:-

- 1) Date as it is:
 - a) "2/02/2023" → Display date using this as a Base.
 - b) "Feb 2 2023"
- 2) Ago last next
 - a) "2 year ago" → Print today's date 2 years ago.
 - b) "5 sec ago" → time 5 sec ago
 - c) "next tue" → next tuesday date
 - d) "last month" → last tuesday date
- 3) +x and -x
 - a) "+2 day" or "2 day" → date and time after 2 day
 - b) "-2 week" → date and time of 2 weeks ago.
- 4) Change timezone:
 - a) "TZ = xxxxx"
- 5) Can combine all the above options:
 - a) "19761118 -3 days"
 - b) "TZ = xxxxx next Friday"

cal

If a user wants a quick view of the calendar in the Linux terminal, cal is the command for you. By default, the cal command shows the current month calendar as output.

Syntax : cal [options] [[month] year]

Examples:

- 1) cal → To display the current month calendar.
- 2) cal 2020 → To display the total year calendar.
- 3) cal 1 → To display the 1st year calendar.
- 4) cal 9999 → To display the 9999th year calendar.
- 5) cal 10000 cal: year '10000' not in range 1..9999
- 6) cal 08 2019 To display august 2019th calendar
- 7) cal -j ----> show julian calendar
- 8) cal -3 ----> past current next month calendar
- 9) cal -m 5 ----> 5th month calendar
- 10) cal -y 2024 ----> year 2024 calendar
- 11) cal -1 ----> current month calendar (this is default)
- 12) cal -d string -----> string will be a date which support many formats like "July 2023", "07-2023"

uptime

Uptime Command In Linux: It is used to find out how long the system is active (running). This command returns a set of values that involve:

- 1) the current time
- 2) the amount of time the system is in running state
- 3) number of users currently logged into
- 4) the load time for the past 1, 5 and 15 minutes respectively.

Syntax: uptime [options]

Options:

- 1) -p → Makes out pretty
- 2) -s → Since when the system is running

Note: Consider my load averages: 0.05, 0.16, 0.21

That means, over the last minute, on an average, 0.05 processes have been waiting for resources.

Should it be greater than nproc(Number of processors) then there should be a concern.

Hostname :-

hostname command in Linux is used to obtain the DNS (Domain Name System) name and set the system's hostname or NIS (Network Information System) domain name. A hostname is a name given to a computer and attached to the network. Its main purpose is to uniquely identify over a network.

Syntax: hostname [options] [file]

Options:

- 1) -i → show ip address of hostname (only primay network interface)
- 2) -l → list all IP addresses associated with the system
- 3) -s → short hostname. ex:- for myhost.example.com short hostname will be myhost
- 4) -f → to display the fully qualified domain name (FQDN). ex :- myhostname.example.com
- 5) -d → to display the domain name of the system. ex:- example.com
- 6) -a → to display all the alias of current hostname
- 7) -A → This option is used to get all FQDNs (Fully Qualified Domain Name) of the host system.
- 8) -b → Used to always set a hostname.
- 9) -F → This option is used to set the hostname specified in a file. Can be performed by the superuser(root) only.

uname

The command 'uname' displays the information about the system.

Options

- 1) -a → displays all the info about the system.
- 2) -v → current kernel version
- 3) -n → hostname
- 4) -s → kernel name
- 5) -r → release version
- 6) -o → os name
- 7) -i → hardware platform
- 8) -m → machine hardware name
- 9) -p → processor type

which

The which command allows users to search the list of paths in the \$PATH environment variable and outputs the full path of the executable file of the command specified as an argument.

Syntax: which [options] filenames...

Option:

- 1) -a → can be used to display all the occurrences of the file.

bc

basic calculator, will do calculation on basic math expression

it will take standard input and give calculation output on standard output

Syntax:

bc [options] → Shows an interactive Option
echo "expression" | bc [options] → takes input from echo
bc [options] < file → takes input from a file.

Options:

- 1) -h, {- -help } → Print the usage and exit
- 2) -i, {- -interactive } → Force interactive mode
- 3) -l, {- -mathlib } → Define the standard math library
- 4) -w, {- -warn } → Give warnings for extensions to POSIX bc
- 5) -s, {- -standard } → Process exactly the POSIX bc language
- 6) -q, {- -quiet } → Do not print the normal GNU bc welcome

7) -v, {- -version } → Print the version number and copyright and quit

The bc command supports the following features:

- 1) Arithmetic operators:
- 2) Increment or Decrement operators:
- 3) Assignment operators:
- 4) Comparison or Relational operators:
- 5) Logical or Boolean operators:
- 6) Math functions:
- 7) Conditional statements:
- 8) Iterative statements:

6. Mathematical Functions

The built-in math functions supported are :

s (x): The sine of x, x is in radians.

c (x) : The cosine of x, x is in radians.

a (x) : The arctangent of x, arctangent returns radians.

l (x) : The natural logarithm of x.

e (x) : The exponential function of raising e to the value x.

j (n,x) : The bessel function of integer order n of x.

sqrt(x) : Square root of the number x. If the expression is negative, a run time error is generated.

In addition to the math functions, the following functions are also supported :

length(x) : returns the number of digits in x.

read() : Reads the number from the standard input.

scale(expression) : The value of the scale function is the number of digits after the decimal point in the expression.

ibase and obase define the conversion base for input and output numbers. The default for both input and output is base 10.