The syntax is as follows for the GNU/date and BSD/date command:

```
date +FORMAT
date +"%FORMAT"
date +"%FORMAT%FORMAT"
date +"%FORMAT-%FORMAT"
```

An operand with a leading plus (`+`) sign signals a user-defined format string which specifies the format in which to display the date and time. The following examples are tested on GNU/Linux, Apple OS X Unix, and FreeBSD unix operating system.

Task: Display date in mm-dd-yy format

Open a terminal and type the following date command:

```
$ date +"%m-%d-%y"
```

Sample outputs:

```
02-27-07
```

To turn on 4 digit year display:

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

Just display date as mm/dd/yy format:

```
$ date +"%D"
```

Task: Display time only

Type the following command:

```
$ date +"%T"
```

Sample outputs:

```
19:55:04
```

To display locale’s 12-hour clock time, enter:

```
$ date +"%r"
```

Sample outputs:

```
07:56:05 PM
```

To display time in HH:MM format, type:

```
$ date +"%H-%M"
```

Sample outputs:

```
00-50
```

How do I save time/date format to the shell variable?

Simply type the following command at the shell prompt:

```
$ NOW=$(date +"%m-%d-%Y")
```

To display a variable use echo / printf command:

```
$ echo $NOW
```

A sample shell script

```
#!/bin/bash
NOW=$(date +"%m-%d-%Y")
FILE="backup.$NOW.tar.gz"
echo "Backing up data to /nas42/backup.$NOW.tar.gz file, 
# rest of script
# tar xcvf /nas42/backup.$NOW.tar.gz /home/ /etc/ /var
```

A complete list of FORMAT control characters supported by the GNU/date command

FORMAT controls the output. It can be the combination of any one of the following:

%FORMAT String	Description
%%	a literal %
%a	locale's abbreviated weekday name (e.g., Sun)
%A	locale's full weekday name (e.g., Sunday)
%b	locale's abbreviated month name (e.g., Jan)
%B	locale's full month name (e.g., January)
%c	locale's date and time (e.g., Thu Mar 3 23:05:25 2005)
%C	century; like %Y, except omit last two digits (e.g., 21)
%d	day of month (e.g, 01)
%D	date; same as %m/%d/%y
%e	day of month, space padded; same as %_d
%F	full date; same as %Y-%m-%d
%g	last two digits of year of ISO week number (see %G)
%G	year of ISO week number (see %V); normally useful only with %V
%h	same as %b
%H	hour (00..23)
%I	hour (01..12)
%j	day of year (001..366)
%k	hour (0..23)

%l	hour (1..12)
%m	month (01..12)
%M	minute (00..59)
%n	a newline
%N	nanoseconds (000000000..999999999)
%p	locale’s equivalent of either AM or PM; blank if not known
%P	like %p, but lower case
%r	locale’s 12-hour clock time (e.g., 11:11:04 PM)
%R	24-hour hour and minute; same as %H:%M
%s	seconds since 1970-01-01 00:00:00 UTC
%S	second (00..60)
%t	a tab
%T	time; same as %H:%M:%S
%u	day of week (1..7); 1 is Monday
%U	week number of year, with Sunday as first day of week (00..53)
%V	ISO week number, with Monday as first day of week (01..53)
%w	day of week (0..6); 0 is Sunday
%W	week number of year, with Monday as first day of week (00..53)
%x	locale’s date representation (e.g., 12/31/99)
%X	locale’s time representation (e.g., 23:13:48)
%y	last two digits of year (00..99)
%Y	year
%z	+hhmm numeric timezone (e.g., **-0400**)
%:z	+hh:mm numeric timezone

	(e.g., **-04**:00)
%::z	+hh:mm:ss numeric time zone (e.g., **-04**:00:00)
%:::z	numeric time zone with : to necessary precision (e.g., **-04**, +05:30)
%Z	alphabetic time zone abbreviation (e.g., EDT)

A complete list of FORMAT control characters supported by the BSD/date command

The following works on Apple OS X, FreeBSD and BSD version of the date command:

%A	is replaced by national representation of the full weekday name.
%a	is replaced by national representation of the abbreviated weekday name.
%B	is replaced by national representation of the full month name.
%b	is replaced by national representation of the abbreviated month name.
%C	is replaced by (year / 100) as decimal number; single digits are preceded by a zero.
%c	is replaced by national representation of time and date.
%D	is equivalent to “%m/%d/%y”.
%d	is replaced by the day of the month as a decimal number (01-

	31).
%E* %O*	POSIX locale extensions. The sequences %Ec %EC %Ex %EX %Ey %EY %Od %Oe %OH %OI %Om %OM %OS %Ou %OU %OV %Ow %OW %Oy are supposed to provide alternate representations. Additionally %OB implemented to represent alternative months names (used standalone, without day mentioned).
%e	is replaced by the day of the month as a decimal number (1-31); single digits are preceded by a blank.
%G	is replaced by a year as a decimal number with century. This year is the one that contains the greater part of the week (Monday as the first day of the week).
%g	is replaced by the same year as in "%G", but as a decimal number without century (00-99).
%H	is replaced by the hour (24-hour clock) as a decimal number (00-23).
%h	the same as %b.
%I	is replaced by the hour (12-hour clock) as a decimal number (01-12).
%j	is replaced by the day of the year as a decimal number (001-366).
%k	is replaced by the hour (24-hour clock) as a decimal number (0-23); single digits are preceded by a blank.
%l	is replaced by the hour (12-hour

	clock) as a decimal number (1-12); single digits are preceded by a blank.
%M	is replaced by the minute as a decimal number (00-59).
%m	is replaced by the month as a decimal number (01-12).
%n	is replaced by a newline.
%O*	the same as %E*.
%p	is replaced by national representation of either “ante meridiem” (a.m.) or “post meridiem” (p.m.) as appropriate.
%R	is equivalent to “%H:%M”.
%r	is equivalent to “%I:%M:%S %p”.
%S	is replaced by the second as a decimal number (00-60).
%s	is replaced by the number of seconds since the Epoch, UTC (see mktime(3)).
%T	is equivalent to “%H:%M:%S”.
%t	is replaced by a tab.
%U	is replaced by the week number of the year (Sunday as the first day of the week) as a decimal number (00-53).
%u	is replaced by the weekday (Monday as the first day of the week) as a decimal number (1-7).
%V	is replaced by the week number of the year (Monday as the first day of the week) as a decimal number (01-53). If the week containing January 1 has four or more days in the new year, then it is week 1; otherwise it is the last week of the

	previous year, and the next week is week 1.
%v	is equivalent to “%e-%b-%Y”.
%W	is replaced by the week number of the year (Monday as the first day of the week) as a decimal number (00-53).
%w	is replaced by the weekday (Sunday as the first day of the week) as a decimal number (0-6).
%X	is replaced by national representation of the time.
%x	is replaced by national representation of the date.
%Y	is replaced by the year with century as a decimal number.
%y	is replaced by the year without century as a decimal number (00-99).
%Z	is replaced by the time zone name.
%z	is replaced by the time zone offset from UTC; a leading plus sign stands for east of UTC, a minus sign for west of UTC, hours and minutes follow with two digits each and no delimiter between them (common form for RFC 822 date headers).
%+	is replaced by national representation of the date and time (the format is similar to that produced by date(1)).
%-*	GNU libc extension. Do not do any padding when performing numerical outputs.
%_*	GNU libc extension. Explicitly specify space for padding.

%0*	GNU libc extension. Explicitly specify zero for padding.
%%	is replaced by %.

A sample date session