

NAME

strftime - format date and time

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <time.h>
```

size_t

```
strftime(char * restrict buf, size_t maxsize, const char * restrict format,  
         const struct tm * restrict timeptr);
```

size_t

```
strftime_l(char * restrict buf, size_t maxsize, const char * restrict format, const struct tm * restrict timeptr,  
          locale_t loc);
```

DESCRIPTION

The **strftime()** function formats the information from *timeptr* into the buffer *buf* according to the string pointed to by *format*. The function **strftime_l()** does the same as **strftime()** but takes an explicit locale rather than using the current locale.

The *format* string consists of zero or more conversion specifications and ordinary characters. All ordinary characters are copied directly into the buffer. A conversion specification consists of a percent sign "%" and one other character.

No more than *maxsize* characters will be placed into the array. If the total number of resulting characters, including the terminating NUL character, is not more than *maxsize*, **strftime()** returns the number of characters in the array, not counting the terminating NUL. Otherwise, zero is returned and the buffer contents are indeterminate.

The conversion specifications are copied to the buffer after expansion as follows:

%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.

%F is equivalent to "%Y-%m-%d".

%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 `'%'`.

SEE ALSO

`date(1)`, `printf(1)`, `ctime(3)`, `printf(3)`, `strptime(3)`, `wcsftime(3)`

STANDARDS

The `strftime()` function conforms to ISO/IEC 9899:1990 ("ISO C90") with a lot of extensions including `'%C'`, `'%D'`, `'%E*'`, `'%e'`, `'%G'`, `'%g'`, `'%h'`, `'%k'`, `'%l'`, `'%n'`, `'%O*'`, `'%R'`, `'%r'`, `'%s'`, `'%T'`, `'%t'`, `'%u'`, `'%V'`, `'%z'`, `'%+'`.

The peculiar week number and year in the replacements of `'%G'`, `'%g'` and `'%V'` are defined in ISO 8601. The `strptime_I()` function conforms to IEEE Std 1003.1-2008 ("POSIX.1").

BUGS

There is no conversion specification for the phase of the moon.

The `strftime()` function does not correctly handle multibyte characters in the *format* argument.