NAME

strptime – date and time conversion

SYNOPSIS

#include <time.h>

char ∗**strptime(const char** ∗*buf,* **const char** ∗*format,* **struct tm** ∗*tm***);**

MT-LEVEL

MT-Safe

DESCRIPTION

The **strptime( )** function converts the character string pointed to by *buf* to values which are stored in the **tm** structure pointed to by *tm*, using the format specified by *format*.

format is composed of zero or more conversion specifications. Each conversion specification is composed of a '%' (percent) character followed by one or two conversion characters which specify the replacement required. One or more white space characters (as specified by **isspace**(3C)) may precede or follow a conversion specification. There must be white-space or other non-alphanumeric characters between any two conversion specifications.

The following conversion specifications are supported:

%%	same as %
%a	day of week, using the locale's weekday names; either the abbreviated or full name may be specified
%A	same as **%a**
%b	month, using the locale's month names; either the abbreviated or full name may be specified
%B	same as **%b**
%c	locale's appropriate date and time representation
%C	century number [0,99]; leading zero is permitted but not required
%d	day of month [1,31]; leading zero is permitted but not required
%D	date as **%m/%d/%y**
%e	same as **%d**
%h	same as **%b**
%H	hour (24-hour clock) [0,23]; leading zero is permitted but not required
%I	hour (12-hour clock) [1,12]; leading zero is permitted but not required
%j	day number of the year [1,366]; leading zeros are permitted but not required
%m	month number [1,12]; leading zero is permitted but not required
%M	minute [0-59]; leading zero is permitted but not required
%n	any white space
%p	locale's equivalent of either a.m. or p.m.
%r	appropriate time representation in the 12-hour clock format with **%p**
%R	time as **%H**:**%M**
%S	seconds [0,61]; leading zero is permitted but not required
%t	any white space
%T	time as **%H**:**%M**:**%S**
%U	week number of the year as a decimal number [0,53], with Sunday as the first day of the week; leading zeros are permitted but not required
%w	weekday as a decimal number [0,6], with 0 representing Sunday;

%W week number of the year as a decimal number [0,53], with Monday as the first day of the week; leading zero is permitted but not required
%x locale's appropriate date representation
%X locale's appropriate time representation
%y year within the century [0,99]; leading zero is permitted but not required
%Y year, including the century (for example, 1993)
%Z timezone name or no characters if no time zone information exists

Modified Conversion Specifications

Some conversion specifications can be modified by the **E** and **O** modifier characters to indicate that an alternate format or specification should be used rather than the one normally used by the unmodified specification. If the alternate format or specification does not exist in the current locale, the behaviour will be as if the unmodified conversion specification were used.

%Ec locale's alternate appropriate date and time representation
%EC name of the base year (era) in the locale's alternate representation
%Ex locale's alternate date representation
%EX locale's alternate time representation
%Ey offset from **%EC** (year only) in the locale's alternate representation
%EY full alternate year representation
%Od day of the month using the locale's alternate numeric symbols
%Oe same as **%Od**
%OH hour (24-hour clock) using the locale's alternate numeric symbols
%OI hour (12-hour clock) using the locale's alternate numeric symbols
%Om month using the locale's alternate numeric symbols
%OM minutes using the locale's alternate numeric symbols
%OS seconds using the locale's alternate numeric symbols
%OU week number of the year (Sunday as the first day of the week) using the locale's alternate numeric symbols
%Ow number of the weekday (Sunday=0) using the locale's alternate numeric symbols
%OW week number of the year (Monday as the first day of the week) using the locale's alternate numeric symbols
%Oy year (offset from **%C**) in the locale's alternate representation and using the locale's alternate numeric symbols

General Specifications

A conversion specification that is an ordinary character is executed by scanning the next character from the buffer. If the character scanned from the buffer differs from the one comprising the specification, the specification fails, and the differing and subsequent characters remain unscanned.

A series of specifications composed of **%n**, **%t**, white-space characters or any combination is executed by scanning up to the first character that is not white space (which remains unscanned), or until no more characters can be scanned. White space is defined by **isspace**(3C).

Any other conversion specification is executed by scanning characters until a character matching the next specification is scanned, or until no more characters can be scanned. These characters, except the one matching the next specification, are then compared to the locale values associated with the conversion specifier. If a match is found, values for the appropriate *tm* structure members are set to values corresponding to the locale information. If no match is found, **strptime( )** fails and no more characters are scanned.

The month names, weekday names, era names, and alternate numeric symbols can consist of any combination of upper and lower case letters. The user can request that the input date or time specification be in a specific language by setting the **LC_TIME** category using **setlocale**(3C).

RETURN VALUES

Upon successful completion, **strptime( )** returns a pointer to the character following the last character parsed. Otherwise, a null pointer is returned.

FILES

/usr/lib/locale/*locale***/LC_TIME/time**
locale specific date and time information
/usr/lib/locale/*locale***/LC_CTYPE/ctype**
character characterization information

SEE ALSO

isspace(3C), **getdate**(3C), **setlocale**(3C), **strftime**(3C)

NOTES

Several "same as" formats, and the special processing of white-space characters are provided in order to ease the use of identical *format* strings for **strftime( )** and **strptime( )**.

The range of values for **%S** is [00,61] rather than [00,59] to allow for the occasional leap second and even more occasional double leap second.