Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'strptime.3p' command

$ man strptime.3p

STRPTIME(3P) POSIX Programmer's Manual STRPTIME(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

strptime ? date and time conversion

SYNOPSIS

```
#include <time.h>
char *strptime(const char *restrict buf, const char *restrict format,
    struct tm *restrict tm);
```

DESCRIPTION

The strptime() function shall convert 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.

The format is composed of zero or more directives. Each directive is composed of one of the following: one or more white-space characters (as specified by isspace()); an ordinary character (neither '%' nor a white-space character); or a conversion specification.

Each conversion specification is introduced by the '%' character after which the following appear in sequence:

* An optional flag, the zero character ('0') or the <plus-sign> char?

acter ('+'), which is ignored.

* An optional field width. If a field width is specified, it shall be interpreted as a string of decimal digits that will determine the maximum number of bytes converted for the conversion rather than the number of bytes specified below in the description of the con? version specifiers.
* An optional E or O modifier.
* A terminating conversion specifier character that indicates the type of conversion to be applied.

The conversions are determined using the LC_TIME category of the cur? rent locale. The application shall ensure that there is white-space or other non-alphanumeric characters between any two conversion specifica? tions unless all of the adjacent conversion specifications convert a known, fixed number of characters. In the following list, the maximum number of characters scanned (excluding the one matching the next di? rective) is as follows:

* If a maximum field width is specified, then that number
* Otherwise, the pattern "{x}" indicates that the maximum is x
* Otherwise, the pattern "[x,y]" indicates that the value shall fall within the range given (both bounds being inclusive), and the maxi? mum number of characters scanned shall be the maximum required to represent any value in the range without leading zeros and without a leading <plus-sign>

The following conversion specifiers are supported.

The results are unspecified if a modifier is specified with a flag or with a minimum field width, or if a field width is specified for any conversion specifier other than C or Y.

a The day of the week, using the locale's weekday names; either the abbreviated or full name may be specified.

A Equivalent to %a.

b The month, using the locale's month names; either the abbrevi? ated or full name may be specified.

B Equivalent to %b.

c Replaced by the locale's appropriate date and time representa? tion.

C All but the last two digits of the year {2}; leading zeros shall be permitted but shall not be required. A leading '+' or '-' character shall be permitted before any leading zeros but shall not be required.

d The day of the month [01,31]; leading zeros shall be permitted but shall not be required.

D The date as %m/%d/%y.

e Equivalent to %d.

h Equivalent to %b.

H The hour (24-hour clock) [00,23]; leading zeros shall be per? mitted but shall not be required.

I The hour (12-hour clock) [01,12]; leading zeros shall be per? mitted but shall not be required.

j The day number of the year [001,366]; leading zeros shall be permitted but shall not be required.

m The month number [01,12]; leading zeros shall be permitted but shall not be required.

M The minute [00,59]; leading zeros shall be permitted but shall not be required.

n Any white space.

p The locale's equivalent of a.m. or p.m.

r 12-hour clock time using the AM/PM notation if t_fmt_ampm is not an empty string in the LC_TIME portion of the current lo? cale; in the POSIX locale, this shall be equivalent to %I:%M:%S %p.

R The time as %H:%M.

S The seconds [00,60]; leading zeros shall be permitted but shall not be required.

t Any white space.

T The time as %H:%M:%S.

U The week number of the year (Sunday as the first day of the

week) as a decimal number [00,53]; leading zeros shall be per?
mitted but shall not be required.

w The weekday as a decimal number [0,6], with 0 representing Sun?
day.

W The week number of the year (Monday as the first day of the
week) as a decimal number [00,53]; leading zeros shall be per?
mitted but shall not be required.

x The date, using the locale's date format.

X The time, using the locale's time format.

y The last two digits of the year. When format contains neither a
C conversion specifier nor a Y conversion specifier, values in
the range [69,99] shall refer to years 1969 to 1999 inclusive
and values in the range [00,68] shall refer to years 2000 to
2068 inclusive; leading zeros shall be permitted but shall not
be required. A leading '+' or '-' character shall be permitted
before any leading zeros but shall not be required.

Note: It is expected that in a future version of this stan?
dard the default century inferred from a 2-digit year
will change. (This would apply to all commands ac?
cepting a 2-digit year as input.)

Y The full year {4}; leading zeros shall be permitted but shall
not be required. A leading '+' or '-' character shall be per?
mitted before any leading zeros but shall not be required.

% Replaced by %.

Modified Conversion Specifiers

Some conversion specifiers can be modified by the E and O modifier
characters to indicate that an alternative format or specification
should be used rather than the one normally used by the unmodified con?
version specifier. If the alternative format or specification does not
exist in the current locale, the behavior shall be as if the unmodified
conversion specification were used.

%Ec The locale's alternative appropriate date and time representa?
tion.

%EC The name of the base year (period) in the locale's alternative representation.

%Ex The locale's alternative date representation.

%EX The locale's alternative time representation.

%Ey The offset from %EC (year only) in the locale's alternative representation.

%EY The full alternative year representation.

%Od The day of the month using the locale's alternative numeric symbols; leading zeros shall be permitted but shall not be required.

%Oe Equivalent to %Od.

%OH The hour (24-hour clock) using the locale's alternative numeric symbols.

%OI The hour (12-hour clock) using the locale's alternative numeric symbols.

%Om The month using the locale's alternative numeric symbols.

%OM The minutes using the locale's alternative numeric symbols.

%OS The seconds using the locale's alternative numeric symbols.

%OU The week number of the year (Sunday as the first day of the week) using the locale's alternative numeric symbols.

%Ow The number of the weekday (Sunday=0) using the locale's alternative numeric symbols.

%OW The week number of the year (Monday as the first day of the week) using the locale's alternative numeric symbols.

%Oy The year (offset from %C) using the locale's alternative numeric symbols.

A conversion specification composed of white-space characters is executed by scanning input up to the first character that is not white-space (which remains unscanned), or until no more characters can be scanned.

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 directive, the di?

rective fails, and the differing and subsequent characters remain un?
scanned.

A series of conversion 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.

Any other conversion specification is executed by scanning characters until a character matching the next directive is scanned, or until no more characters can be scanned. These characters, except the one match? ing the next directive, are then compared to the locale values associ? ated 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. Case is ignored when matching items in buf such as month or weekday names. If no match is found, strptime() fails and no more characters are scanned.

RETURN VALUE

Upon successful completion, strptime() shall return a pointer to the character following the last character parsed. Otherwise, a null pointer shall be returned.

ERRORS

No errors are defined.

The following sections are informative.

EXAMPLES

Convert a Date-Plus-Time String to Broken-Down Time and Then into Seconds

The following example demonstrates the use of strptime() to convert a string into broken-down time. The broken-down time is then converted into seconds since the Epoch using mktime().

```
#include <time.h>
...
struct tm tm;
time_t t;
if (strptime("6 Dec 2001 12:33:45", "%d %b %Y %H:%M:%S", &tm) == NULL)
    /* Handle error */;
```

```
printf("year: %d; month: %d; day: %d;\n",
        tm.tm_year, tm.tm_mon, tm.tm_mday);
printf("hour: %d; minute: %d; second: %d\n",
        tm.tm_hour, tm.tm_min, tm.tm_sec);
printf("week day: %d; year day: %d\n", tm.tm_wday, tm.tm_yday);
tm.tm_isdst = -1;      /* Not set by strptime(); tells mktime()
                          to determine whether daylight saving time
                          is in effect */
t = mktime(&tm);
if (t == -1)
    /* Handle error */;
printf("seconds since the Epoch: %ld\n", (long) t);"
```

APPLICATION USAGE

Several ``equivalent to'' 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().

It should be noted that dates constructed by the strftime() function with the %Y or %C%y conversion specifiers may have values larger than 9999. If the strptime() function is used to read such values using %C%y or %Y, the year values will be truncated to four digits. Applications should use %+w%y or %+xY with w and x set large enough to contain the full value of any years that will be printed or scanned.

See also the APPLICATION USAGE section in strftime().

It is unspecified whether multiple calls to strptime() using the same tm structure will update the current contents of the structure or over? write all contents of the structure. Conforming applications should make a single call to strptime() with a format and all data needed to completely specify the date and time being converted.

RATIONALE

See the RATIONALE section for strftime().

FUTURE DIRECTIONS

None.

SEE ALSO

fprintf(), fscanf(), strftime(), time()

The Base Definitions volume of POSIX.1?2017, <time.h>

COPYRIGHT