NAME

strptime - parse date and time string

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <time.h>

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

#include <time.h>
#include <xlocale.h>

*char **
strptime_l(*const char * restrict buf, const char * restrict format, struct tm * restrict timeptr, locale_t loc*);

DESCRIPTION

The **strptime**() function parses the string in the buffer *buf* according to the string pointed to by *format*, and fills in the elements of the structure pointed to by *timeptr*. The resulting values will be relative to the local time zone. Thus, it can be considered the reverse operation of strftime(3). The **strptime_l**() function does the same as **strptime**(), 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 matched exactly with the buffer, where white space in the format string will match any amount of white space in the buffer. All conversion specifications are identical to those described in strftime(3).

Two-digit year values, including formats *%y* and *%D*, are now interpreted as beginning at 1969 per POSIX requirements. Years 69-00 are interpreted in the 20th century (1969-2000), years 01-68 in the 21st century (2001-2068). The *%U* and *%W* format specifiers accept any value within the range 00 to 53.

If the *format* string does not contain enough conversion specifications to completely specify the resulting *struct tm*, the unspecified members of *timeptr* are left untouched. For example, if *format* is "%H:%M:%S", only *tm_hour*, *tm_sec* and *tm_min* will be modified. If time relative to today is desired, initialize the *timeptr* structure with today’s date before passing it to **strptime**().

RETURN VALUES

Upon successful completion, **strptime**() returns the pointer to the first character in *buf* that has not been required to satisfy the specified conversions in *format*. It returns NULL if one of the conversions failed. **strptime_l**() returns the same values as **strptime**().

SEE ALSO

date(1), scanf(3), strftime(3)

HISTORY

The **strptime**() function appeared in FreeBSD 3.0.

AUTHORS

The **strptime**() function has been contributed by Powerdog Industries.

This man page was written by Jörg Wunsch.

BUGS

Both the *%e* and *%l* format specifiers may incorrectly scan one too many digits if the intended values comprise only a single digit and that digit is followed immediately by another digit. Both specifiers accept zero-padded values, even though they are both defined as taking unpadded values.

The *%p* format specifier has no effect unless it is parsed *after* hour-related specifiers. Specifying *%l* without *%p* will produce undefined results. Note that 12AM (ante meridiem) is taken as midnight and 12PM (post meridiem) is taken as noon.

The *%Z* format specifier only accepts time zone abbreviations of the local time zone, or the value "GMT". This limitation is because of ambiguity due to of the over loading of time zone abbreviations. One such example is *EST* which is both Eastern Standard Time and Eastern Australia Summer Time.

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