



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 'ctime.3p' command

\$ man ctime.3p

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

ctime, ctime_r ? convert a time value to a date and time string

SYNOPSIS

```
#include <time.h>

char *ctime(const time_t *clock);

char *ctime_r(const time_t *clock, char *buf);
```

DESCRIPTION

For ctime(): The functionality described on this reference page is aligned with the ISO C standard. Any conflict between the requirements described here and the ISO C standard is unintentional. This volume of POSIX.1?2017 defers to the ISO C standard.

The ctime() function shall convert the time pointed to by clock, representing time in seconds since the Epoch, to local time in the form of a string. It shall be equivalent to:

```
asctime(localtime(clock))
```

The asctime(), ctime(), gmtime(), and localtime() functions shall return values in one of two static objects: a broken-down time structure

and an array of char. Execution of any of the functions may overwrite the information returned in either of these objects by any of the other functions.

The `ctime()` function need not be thread-safe.

The `ctime_r()` function shall convert the calendar time pointed to by `clock` to local time in exactly the same form as `ctime()` and put the string into the array pointed to by `buf` (which shall be at least 26 bytes in size) and return `buf`.

Unlike `ctime()`, the `ctime_r()` function is not required to set `tzname`. If `ctime_r()` sets `tzname`, it shall also set `daylight` and `timezone`. If `ctime_r()` does not set `tzname`, it shall not set `daylight` and shall not set `timezone`.

RETURN VALUE

The `ctime()` function shall return the pointer returned by `asctime()` with that broken-down time as an argument.

Upon successful completion, `ctime_r()` shall return a pointer to the string pointed to by `buf`. When an error is encountered, a null pointer shall be returned.

ERRORS

No errors are defined.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

These functions are included only for compatibility with older implementations. They have undefined behavior if the resulting string would be too long, so the use of these functions should be discouraged. On implementations that do not detect output string length overflow, it is possible to overflow the output buffers in such a way as to cause applications to fail, or possible system security violations. Also, these functions do not support localized date and time formats. To avoid these problems, applications should use `strftime()` to generate strings from broken-down times.

Values for the broken-down time structure can be obtained by calling `gmtime()` or `localtime()`.

The `ctime_r()` function is thread-safe and shall return values in a user-supplied buffer instead of possibly using a static data area that may be overwritten by each call.

Attempts to use `ctime()` or `ctime_r()` for times before the Epoch or for times beyond the year 9999 produce undefined results. Refer to `asctime_r()`.

RATIONALE

The standard developers decided to mark the `ctime()` and `ctime_r()` functions obsolescent even though they are in the ISO C standard due to the possibility of buffer overflow. The ISO C standard also provides the `strptime()` function which can be used to avoid these problems.

FUTURE DIRECTIONS

These functions may be removed in a future version.

SEE ALSO

`asctime()`, `clock()`, `difftime()`, `gmtime()`, `localtime()`, `mktime()`, `strptime()`, `time()`, `utime()`

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

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html>.

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html.

