
*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 'utime.3p' command*

***$ man utime.3p***

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

utime ? set file access and modification times

SYNOPSIS

```
#include <utime.h>

int utime(const char *path, const struct utimbuf *times);
```

DESCRIPTION

The utime() function shall set the access and modification times of the file named by the path argument.

If times is a null pointer, the access and modification times of the file shall be set to the current time. The effective user ID of the process shall match the owner of the file, or the process has write permission to the file or has appropriate privileges, to use utime() in this manner.

If times is not a null pointer, times shall be interpreted as a pointer to a utimbuf structure and the access and modification times shall be set to the values contained in the designated structure. Only a process with the effective user ID equal to the user ID of the file or a

process with appropriate privileges may use utime() this way.

The utimbuf structure is defined in the <utime.h> header. The times in the structure utimbuf are measured in seconds since the Epoch.

Upon successful completion, the utime() function shall mark the last file status change timestamp for update; see <sys/stat.h>.

RETURN VALUE

Upon successful completion, 0 shall be returned. Otherwise, -1 shall be returned and errno shall be set to indicate the error, and the file times shall not be affected.

ERRORS

The utime() function shall fail if:

EACCES Search permission is denied by a component of the path prefix; or the times argument is a null pointer and the effective user ID of the process does not match the owner of the file, the process does not have write permission for the file, and the process does not have appropriate privileges.

ELOOP A loop exists in symbolic links encountered during resolution of the path argument.

ENAMETOOLONG

The length of a component of a pathname is longer than {NAME_MAX}.

ENOENT A component of path does not name an existing file or path is an empty string.

ENOTDIR

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, or the path argument contains at least one non-<slash> character and ends with one or more trailing <slash> characters and the last pathname component names an existing file that is neither a di? rectory nor a symbolic link to a directory.

EPERM The times argument is not a null pointer and the effective user ID of the calling process does not match the owner of the file and the calling process does not have appropriate privileges.

EROFS The file system containing the file is read-only.

The utime() function may fail if:

ELOOP More than {SYMLOOP_MAX} symbolic links were encountered during resolution of the path argument.

ENAMETOOLONG

The length of a pathname exceeds {PATH_MAX}, or pathname resolu? tion of a symbolic link produced an intermediate result with a length that exceeds {PATH_MAX}.

The following sections are informative.

## EXAMPLES

None.

## APPLICATION USAGE

Since the utimbuf structure only contains time_t variables and is not accurate to fractions of a second, applications should use the utimen? sat() function instead of the obsolescent utime() function.

## RATIONALE

The actime structure member must be present so that an application may set it, even though an implementation may ignore it and not change the last data access timestamp on the file. If an application intends to leave one of the times of a file unchanged while changing the other, it should use stat() or fstat() to retrieve the file's st_atim and st_mtim parameters, set actime and modtime in the buffer, and change one of them before making the utime() call.

## FUTURE DIRECTIONS

The utime() function may be removed in a future version.

## SEE ALSO

fstat(), fstatat(), futimens()

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

## COPYRIGHT

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.ker? nel.org/doc/man-pages/reporting_bugs.html .