NAME

futimens, **utimensat** - set file access and modification times

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <sys/stat.h>

int
futimens(*int fd*, *const struct timespec times[2]*);

int
utimensat(*int fd*, *const char *path*, *const struct timespec times[2]*, *int flag*);

DESCRIPTION

The access and modification times of the file named by *path* or referenced by *fd* are changed as specified by the argument *times*. The inode-change-time of the file is set to the current time.

If *path* specifies a relative path, it is relative to the current working directory if *fd* is AT_FDCWD and otherwise relative to the directory associated with the file descriptor *fd*.

The *tv_nsec* field of a *timespec* structure can be set to the special value UTIME_NOW to set the current time, or to UTIME_OMIT to leave the time unchanged. In either case, the *tv_sec* field is ignored.

If *times* is non-NULL, it is assumed to point to an array of two timespec structures. The access time is set to the value of the first element, and the modification time is set to the value of the second element. For file systems that support file birth (creation) times (such as UFS2), the birth time will be set to the value of the second element if the second element is older than the currently set birth time. To set both a birth time and a modification time, two calls are required; the first to set the birth time and the second to set the (presumably newer) modification time. Ideally a new system call will be added that allows the setting of all three times at once. If *times* is NULL, this is equivalent to passing a pointer to an array of two timespec structures with both *tv_nsec* fields set to UTIME_NOW.

If both *tv_nsec* fields are UTIME_OMIT, the timestamps remain unchanged and no permissions are needed for the file itself, although search permissions may be required for the path prefix. The call may or may not succeed if the named file does not exist.

If both *tv_nsec* fields are UTIME_NOW, the caller must be the owner of the file, have permission to write the file, or be the super-user.

For all other values of the timestamps, the caller must be the owner of the file or be the super-user.

The values for the *flag* argument of the **utimensat**() system call are constructed by a bitwise-inclusive OR of flags from the following list, defined in <*fcntl.h*>:

AT_SYMLINK_NOFOLLOW
: If *path* names a symbolic link, the symbolic link's times are changed. By default, **utimensat**() changes the times of the file referenced by the symbolic link.

AT_RESOLVE_BENEATH
: Only walk paths below the directory specified by the *fd* descriptor. See the description of the O_RESOLVE_BENEATH flag in the open(2) manual page.

AT_EMPTY_PATH
: If the *path* argument is an empty string, operate on the file or directory referenced by the descriptor *fd*. If *fd* is equal to AT_FDCWD, operate on the current working directory.

RETURN VALUES

Upon successful completion, the value 0 is returned; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

These system calls will fail if:

[EACCES]	The *times* argument is NULL, or both *tv_nsec* values are UTIME_NOW, and the effective user ID of the process does not match the owner of the file, and is not the super-user, and write access is denied.
[EFAULT]	The *times* argument points outside the process's allocated address space.
[EINVAL]	The *tv_nsec* component of at least one of the values specified by the *times* argument has a value less than 0 or greater than 999999999 and is not equal to UTIME_NOW or UTIME_OMIT.
[EIO]	An I/O error occurred while reading or writing the affected inode.
[EINTEGRITY]	Corrupted data was detected while reading from the file system.
[EPERM]	The *times* argument is not NULL nor are both *tv_nsec* values UTIME_NOW, nor are both *tv_nsec* values UTIME_OMIT and the calling process's effective user ID

does not match the owner of the file and is not the super-user.

[EPERM] The named file has its immutable or append-only flag set, see the chflags(2) manual page for more information.

[EROFS] The file system containing the file is mounted read-only.

The **futimens**() system call will fail if:

[EBADF] The *fd* argument does not refer to a valid descriptor.

The **utimensat**() system call will fail if:

[EACCES] Search permission is denied for a component of the path prefix.

[EBADF] The *path* argument does not specify an absolute path and the *fd* argument is neither AT_FDCWD nor a valid file descriptor.

[EFAULT] The *path* argument points outside the process's allocated address space.

[ELOOP] Too many symbolic links were encountered in translating the pathname.

[ENAMETOOLONG]
A component of a pathname exceeded NAME_MAX characters, or an entire path name exceeded PATH_MAX characters.

[ENOENT] The named file does not exist.

[ENOTDIR] A component of the path prefix is not a directory.

[ENOTDIR] The *path* argument is not an absolute path and *fd* is neither AT_FDCWD nor a file descriptor associated with a directory.

[ENOTCAPABLE] *path* is an absolute path, or contained a ".." component leading to a directory outside of the directory hierarchy specified by *fd*, and the process is in capability mode or the AT_RESOLVE_BENEATH flag was specified.

SEE ALSO

chflags(2), stat(2), symlink(2), utimes(2), utime(3), symlink(7)

STANDARDS

The **futimens**() and **utimensat**() system calls are expected to conform to IEEE Std 1003.1-2008 ("POSIX.1").

HISTORY

The **futimens**() and **utimensat**() system calls appeared in FreeBSD 10.3.