NAME

futimesat − change timestamps of a file relative to a directory file descriptor

LIBRARY

Standard C library (*libc*, *−lc*)

SYNOPSIS

```
#include <fcntl.h>          /* Definition of AT_* constants */
#include <sys/time.h>

[[deprecated]] int futimesat(int dirfd, const char *pathname,
                             const struct timeval times[2]);
```

Feature Test Macro Requirements for glibc (see **feature_test_macros**(7)):

futimesat():
_GNU_SOURCE

DESCRIPTION

This system call is obsolete. Use **utimensat**(2) instead.

The **futimesat**() system call operates in exactly the same way as **utimes**(2), except for the differences described in this manual page.

If the pathname given in *pathname* is relative, then it is interpreted relative to the directory referred to by the file descriptor *dirfd* (rather than relative to the current working directory of the calling process, as is done by **utimes**(2) for a relative pathname).

If *pathname* is relative and *dirfd* is the special value **AT_FDCWD**, then *pathname* is interpreted relative to the current working directory of the calling process (like **utimes**(2)).

If *pathname* is absolute, then *dirfd* is ignored. (See **openat**(2) for an explanation of why the *dirfd* argument is useful.)

RETURN VALUE

On success, **futimesat**() returns a 0. On error, −1 is returned and *errno* is set to indicate the error.

ERRORS

The same errors that occur for **utimes**(2) can also occur for **futimesat**(). The following additional errors can occur for **futimesat**():

EBADF
pathname is relative but *dirfd* is neither **AT_FDCWD** nor a valid file descriptor.

ENOTDIR
pathname is relative and *dirfd* is a file descriptor referring to a file other than a directory.

VERSIONS

futimesat() was added in Linux 2.6.16; library support was added in glibc 2.4.

STANDARDS

This system call is nonstandard. It was implemented from a specification that was proposed for POSIX.1, but that specification was replaced by the one for **utimensat**(2).

A similar system call exists on Solaris.

NOTES

glibc notes

If *pathname* is NULL, then the glibc **futimesat**() wrapper function updates the times for the file referred to by *dirfd*.

SEE ALSO

stat(2), **utimensat**(2), **utimes**(2), **futimes**(3), **path_resolution**(7)