NAME

fdatasync, **fsync** - synchronise changes to a file

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <unistd.h>

int
fdatasync(*int fd*);

int
fsync(*int fd*);

DESCRIPTION

The **fsync**() system call causes all modified data and attributes of the file referenced by the file descriptor *fd* to be moved to a permanent storage device. This normally results in all in-core modified copies of buffers for the associated file to be written to a disk.

The **fdatasync**() system call causes all modified data of *fd* to be moved to a permanent storage device. Unlike **fsync**(), the system call does not guarantee that file attributes or metadata necessary to access the file are committed to the permanent storage.

The **fsync**() system call should be used by programs that require a file to be in a known state, for example, in building a simple transaction facility. If the file metadata has already been committed, using **fdatasync**() can be more efficient than **fsync**().

Both **fdatasync**() and **fsync**() calls are cancellation points.

RETURN VALUES

The **fsync**() function returns the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **fsync**() and **fdatasync**() calls fail if:

[EBADF] The *fd* argument is not a valid descriptor.

[EINVAL] The *fd* argument refers to a socket, not to a file.

[EIO] An I/O error occurred while reading from or writing to the file system.

[EINTEGRITY] Corrupted data was detected while reading from the file system.

SEE ALSO

fsync(1), sync(2), syncer(4), sync(8)

HISTORY

The **fsync**() system call appeared in 4.2BSD. The **fdatasync**() system call appeared in FreeBSD 11.1.

BUGS

The **fdatasync**() system call currently does not guarantee that enqueued aio(4) requests for the file referenced by *fd* are completed before the syscall returns.