NAME

aio_fsync - asynchronous file synchronization (REALTIME)

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <aio.h>

int
aio_fsync(*int op*, *struct aiocb *iocb*);

DESCRIPTION

The **aio_fsync**() system call allows the calling process to move all modified data associated with the descriptor *iocb->aio_fildes* to a permanent storage device. The call returns immediately after the synchronization request has been enqueued to the descriptor; the synchronization may or may not have completed at the time the call returns.

The *op* argument can be set to O_SYNC to cause all currently queued I/O operations to be completed as if by a call to fsync(2), or O_DSYNC for the behavior of fdatasync(2).

If _POSIX_PRIORITIZED_IO is defined, and the descriptor supports it, then the enqueued operation is submitted at a priority equal to that of the calling process minus *iocb->aio_reqprio*.

The *iocb* pointer may be subsequently used as an argument to **aio_return**() and **aio_error**() in order to determine return or error status for the enqueued operation while it is in progress.

If the request could not be enqueued (generally due to invalid arguments), the call returns without having enqueued the request.

The *iocb->aio_sigevent* structure can be used to request notification of the operation's completion as described in aio(4).

RESTRICTIONS

The Asynchronous I/O Control Block structure pointed to by *iocb* must remain valid until the operation has completed.

The asynchronous I/O control buffer *iocb* should be zeroed before the **aio_fsync**() call to avoid passing bogus context information to the kernel.

Modification of the Asynchronous I/O Control Block structure is not allowed while the request is queued.

RETURN VALUES

The **aio_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 **aio_fsync**() system call will fail if:

[EAGAIN] The request was not queued because of system resource limitations.

[EINVAL] The asynchronous notification method in *iocb->aio_sigevent.sigev_notify* is invalid or not supported.

[EOPNOTSUPP] Asynchronous file synchronization operations on the file descriptor *iocb->aio_fildes* are unsafe and unsafe asynchronous I/O operations are disabled.

[EINVAL] A value of the *op* argument is not set to O_SYNC or O_DSYNC.

The following conditions may be synchronously detected when the **aio_fsync**() system call is made, or asynchronously, at any time thereafter. If they are detected at call time, **aio_fsync**() returns -1 and sets *errno* appropriately; otherwise the **aio_return**() system call must be called, and will return -1, and **aio_error**() must be called to determine the actual value that would have been returned in *errno*.

[EBADF] The *iocb->aio_fildes* argument is not a valid descriptor.

[EINVAL] This implementation does not support synchronized I/O for this file.

If the request is successfully enqueued, but subsequently cancelled or an error occurs, the value returned by the **aio_return**() system call is per the read(2) and write(2) system calls, and the value returned by the **aio_error**() system call is one of the error returns from the read(2) or write(2) system calls.

SEE ALSO

aio_cancel(2), aio_error(2), aio_read(2), aio_return(2), aio_suspend(2), aio_waitcomplete(2), aio_write(2), fsync(2), sigevent(3), siginfo(3), aio(4)

STANDARDS

The **aio_fsync**() system call is expected to conform to the IEEE Std 1003.1 ("POSIX.1") standard.

HISTORY

The **aio_fsync**() system call first appeared in FreeBSD 7.0. The O_DSYNC option appeared in FreeBSD 13.0.