

**NAME**

**aio\_cancel** - cancel an outstanding asynchronous I/O operation (REALTIME)

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

```
#include <aio.h>
```

*int*

```
aio_cancel(int fildes, struct aiocb *iocb);
```

**DESCRIPTION**

The **aio\_cancel**() system call cancels the outstanding asynchronous I/O request for the file descriptor specified in *fildes*. If *iocb* is specified, only that specific asynchronous I/O request is cancelled.

Normal asynchronous notification occurs for cancelled requests. Requests complete with an error result of ECANCELED.

**RESTRICTIONS**

The **aio\_cancel**() system call does not cancel asynchronous I/O requests for raw disk devices. The **aio\_cancel**() system call will always return AIO\_NOTCANCELED for file descriptors associated with raw disk devices.

**RETURN VALUES**

The **aio\_cancel**() system call returns -1 to indicate an error, or one of the following:

[AIO\_CANCELED]

All outstanding requests meeting the criteria specified were cancelled.

[AIO\_NOTCANCELED]

Some requests were not cancelled, status for the requests should be checked with **aio\_error**(2).

[AIO\_ALLDONE]

All of the requests meeting the criteria have finished.

**ERRORS**

An error return from **aio\_cancel**() indicates:

[EBADF]           The *filides* argument is an invalid file descriptor.

## SEE ALSO

aio\_error(2), aio\_read(2), aio\_return(2), aio\_suspend(2), aio\_write(2), aio(4)

## STANDARDS

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

## HISTORY

The **aio\_cancel()** system call first appeared in FreeBSD 3.0. The first functional implementation of **aio\_cancel()** appeared in FreeBSD 4.0.

## AUTHORS

This manual page was originally written by Wes Peters <wes@softweyr.com>. Christopher M Sedore <cmsedore@maxwell.syr.edu> updated it when **aio\_cancel()** was implemented for FreeBSD 4.0.