NAME

io_cancel − cancel an outstanding asynchronous I/O operation

SYNOPSIS

#include <libaio.h>

int io_cancel(aio_context_t *ctx_id***, struct iocb ***iocb***,**
struct io_event ***result*);**

Link with *−laio*.

DESCRIPTION

io_cancel() attempts to cancel an asynchronous I/O operation previously submitted with **io_submit**(2). *ctx_id* is the AIO context ID of the operation to be canceled. If the AIO context is found, the event will be canceled and then copied into the memory pointed to by *result* without being placed into the completion queue.

RETURN VALUE

On success, **io_cancel**() returns 0. For the failure return, see NOTES.

ERRORS

EAGAIN
The *iocb* specified was not canceled.

EFAULT
One of the data structures points to invalid data.

EINVAL
The AIO context specified by *ctx_id* is invalid.

ENOSYS
io_cancel() is not implemented on this architecture.

VERSIONS

The asynchronous I/O system calls first appeared in Linux 2.5, August 2002.

CONFORMING TO

io_cancel() is Linux-specific and should not be used in programs that are intended to be portable.

NOTES

Glibc does not provide a wrapper function for this system call.

The wrapper provided in *libaio* for **io_cancel**() does not follow the usual C library conventions for indicating error: on error it returns a negated error number (the negative of one of the values listed in ERRORS). If the system call is invoked via **syscall**(2), then the return value follows the usual conventions for indicating an error: −1, with *errno* set to a (positive) value that indicates the error.

SEE ALSO

io_destroy(2), **io_getevents**(2), **io_setup**(2), **io_submit**(2)

COLOPHON

This page is part of release 3.22 of the Linux *man-pages* project. A description of the project, and information about reporting bugs, can be found at http://www.kernel.org/doc/man-pages/.