



Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'aio_cancel.3p' command

\$ man aio_cancel.3p

AIO_CANCEL(3P) POSIX Programmer's Manual AIO_CANCEL(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

aio_cancel ? cancel an asynchronous I/O request

SYNOPSIS

```
#include <aio.h>

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

DESCRIPTION

The aio_cancel() function shall attempt to cancel one or more asynchronous I/O requests currently outstanding against file descriptor fildes.

The aiocbp argument points to the asynchronous I/O control block for a particular request to be canceled. If aiocbp is NULL, then all outstanding cancelable asynchronous I/O requests against fildes shall be canceled.

Normal asynchronous notification shall occur for asynchronous I/O operations that are successfully canceled. If there are requests that cannot be canceled, then the normal asynchronous completion process shall take place for those requests when they are completed.

For requested operations that are successfully canceled, the associated

error status shall be set to [ECANCELED] and the return status shall be -1. For requested operations that are not successfully canceled, the aiocbp shall not be modified by aio_cancel().

If aiocbp is not NULL, then if fildes does not have the same value as the file descriptor with which the asynchronous operation was initiated, unspecified results occur.

Which operations are cancelable is implementation-defined.

RETURN VALUE

The aio_cancel() function shall return the value AIO_CANCELED if the requested operation(s) were canceled. The value AIO_NOTCANCELED shall be returned if at least one of the requested operation(s) cannot be canceled because it is in progress. In this case, the state of the other operations, if any, referenced in the call to aio_cancel() is not indicated by the return value of aio_cancel(). The application may determine the state of affairs for these operations by using aio_error().

The value AIO_ALLDONE is returned if all of the operations have already completed. Otherwise, the function shall return -1 and set errno to indicate the error.

ERRORS

The aio_cancel() function shall fail if:

EBADF The fildes argument is not a valid file descriptor.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

aio_read(), aio_write()

The Base Definitions volume of POSIX.1?2017, <aio.h>

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html .

IEEE/The Open Group

2017

AIO_CANCEL(3P)