



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

\$ man aio_fsync.3p

AIO_FSYNC(3P) POSIX Programmer's Manual AIO_FSYNC(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_fsync ? asynchronous file synchronization

SYNOPSIS

```
#include <aio.h>

int aio_fsync(int op, struct aiocb *aiocbp);
```

DESCRIPTION

The `aio_fsync()` function shall asynchronously perform a file synchronization operation, as specified by the `op` argument, for I/O operations associated with the file indicated by the file descriptor `aio_fildes` member of the `aiocb` structure referenced by the `aiocbp` argument and queued at the time of the call to `aio_fsync()`. The function call shall return when the synchronization request has been initiated or queued to the file or device (even when the data cannot be synchronized immediately).

If `op` is `O_DSYNC`, all currently queued I/O operations shall be completed as if by a call to `fdatasync()`; that is, as defined for synchronized I/O data integrity completion.

If `op` is `O_SYNC`, all currently queued I/O operations shall be completed as if by a call to `fsync()`; that is, as defined for synchronized I/O file integrity completion. If the `aio_fsync()` function fails, or if the operation queued by `aio_fsync()` fails, then outstanding I/O operations are not guaranteed to have been completed.

If `aio_fsync()` succeeds, then it is only the I/O that was queued at the time of the call to `aio_fsync()` that is guaranteed to be forced to the relevant completion state. The completion of subsequent I/O on the file descriptor is not guaranteed to be completed in a synchronized fashion. The `aio_cbp` argument refers to an asynchronous I/O control block. The `aio_cbp` value may be used as an argument to `aio_error()` and `aio_return()` in order to determine the error status and return status, respectively, of the asynchronous operation while it is proceeding. When the request is queued, the error status for the operation is `[EINPROGRESS]`. When all data has been successfully transferred, the error status shall be reset to reflect the success or failure of the operation. If the operation does not complete successfully, the error status for the operation shall be set to indicate the error. The `aio_sigevent` member determines the asynchronous notification to occur as specified in Section 2.4.1, Signal Generation and Delivery when all operations have achieved synchronized I/O completion. All other members of the structure referenced by `aio_cbp` are ignored. If the control block referenced by `aio_cbp` becomes an illegal address prior to asynchronous I/O completion, then the behavior is undefined.

If the `aio_fsync()` function fails or `aio_cbp` indicates an error condition, data is not guaranteed to have been successfully transferred.

RETURN VALUE

The `aio_fsync()` function shall return the value 0 if the I/O operation is successfully queued; otherwise, the function shall return the value -1 and set `errno` to indicate the error.

ERRORS

The `aio_fsync()` function shall fail if:

EAGAIN The requested asynchronous operation was not queued due to tem?

porary resource limitations.

EBADF The `aio_fildes` member of the `aio_cb` structure referenced by the `aio_cbp` argument is not a valid file descriptor.

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

EINVAL The `aio_fildes` member of the `aio_cb` structure refers to a file on which an `fsync()` operation is not possible.

EINVAL A value of `op` other than `O_DSYNC` or `O_SYNC` was specified, or `O_DSYNC` was specified and the implementation does not provide runtime support for the Synchronized Input and Output option, or `O_SYNC` was specified and the implementation does not provide runtime support for the File Synchronization option.

In the event that any of the queued I/O operations fail, `aio_fsync()` shall return the error condition defined for `read()` and `write()`. The error is returned in the error status for the asynchronous operation, which can be retrieved using `aio_error()`.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

Note that even if the file descriptor is not open for writing, if there are any pending write requests on the underlying file, then that I/O will be completed prior to the return of a call to `aio_error()` or `aio_return()` indicating that the operation has completed.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

`aio_error()`, `aio_return()`, `fcntl()`, `fdatasync()`, `fsync()`, `open()`, `read()`, `write()`

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

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_FSYNC(3P)