NAME

aio_read, **aio_readv** - asynchronous read from a file (REALTIME)

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <aio.h>

int
aio_read(*struct aiocb *iocb*);

#include <sys/uio.h>

int
aio_readv(*struct aiocb *iocb*);

DESCRIPTION

The **aio_read**() and **aio_readv**() system calls allow the calling process to read from the descriptor *iocb->aio_fildes* beginning at the offset *iocb->aio_offset*. **aio_read**() will read *iocb->aio_nbytes* from the buffer pointed to by *iocb->aio_buf*, whereas **aio_readv**() reads the data into the *iocb->aio_iovcnt* buffers specified by the members of the *iocb->aio_iov* array. Both syscalls return immediately after the read request has been enqueued to the descriptor; the read may or may not have completed at the time the call returns.

For **aio_readv**() the *iovec* structure is defined in readv(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->aio_lio_opcode* argument is ignored by the **aio_read**() and **aio_readv**() system calls.

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), then the call returns without having enqueued the request.

If the request is successfully enqueued, the value of *iocb->aio_offset* can be modified during the request as context, so this value must not be referenced after the request is enqueued.

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* and the buffer that the *iocb->aio_buf* member of that structure references must remain valid until the operation has completed.

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

Modifications of the Asynchronous I/O Control Block structure or the buffer contents are not allowed while the request is queued.

If the file offset in *iocb->aio_offset* is past the offset maximum for *iocb->aio_fildes*, no I/O will occur.

RETURN VALUES

The **aio_read**() and **aio_readv**() functions return the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

DIAGNOSTICS

None.

ERRORS

The **aio_read**() and **aio_readv**() system calls will fail if:

[EAGAIN]	The request was not queued because of system resource limitations.
[EFAULT]	Part of *aio_iov* points outside the process's allocated address space.
[EINVAL]	The asynchronous notification method in *iocb->aio_sigevent.sigev_notify* is invalid or not supported.
[EOPNOTSUPP]	Asynchronous read operations on the file descriptor *iocb->aio_fildes* are unsafe and unsafe asynchronous I/O operations are disabled.

The following conditions may be synchronously detected when the **aio_read**() or **aio_readv**() system call is made, or asynchronously, at any time thereafter. If they are detected at call time, **aio_read**() or **aio_readv**() 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 invalid.
[EINVAL]	The offset *iocb->aio_offset* is not valid, the priority specified by *iocb->aio_reqprio* is not a valid priority, or the number of bytes specified by *iocb->aio_nbytes* is not valid.
[EOVERFLOW]	The file is a regular file, *iocb->aio_nbytes* is greater than zero, the starting offset in *iocb->aio_offset* is before the end of the file, but is at or beyond the *iocb->aio_fildes* offset maximum.

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) system call, and the value returned by the **aio_error**() system call is either one of the error returns from the read(2) system call, or one of:

[EBADF]	The *iocb->aio_fildes* argument is invalid for reading.
[ECANCELED]	The request was explicitly cancelled via a call to **aio_cancel**().
[EINVAL]	The offset *iocb->aio_offset* would be invalid.

SEE ALSO

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

STANDARDS

The **aio_read**() system call is expected to conform to the IEEE Std 1003.1 ("POSIX.1") standard. The **aio_readv**() system call is a FreeBSD extension, and should not be used in portable code.

HISTORY

The **aio_read**() system call first appeared in FreeBSD 3.0. The **aio_readv**() system call first appeared in FreeBSD 13.0.

AUTHORS

This manual page was written by Terry Lambert <*terry@whistle.com*>.

BUGS

Invalid information in *iocb->_aiocb_private* may confuse the kernel.