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_read.3p' command

$ man aio_read.3p

AIO_READ(3P) POSIX Programmer's Manual AIO_READ(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_read ? asynchronous read from a file

SYNOPSIS

#include <aio.h>

int aio_read(struct aiocb *aiocbp);

DESCRIPTION

The aio_read() function shall read aiocbp->aio_nbytes from the file as? sociated with aiocbp->aio_fildes into the buffer pointed to by aiocbp->aio_buf. The function call shall return when the read request has been initiated or queued to the file or device (even when the data cannot be delivered immediately).

If prioritized I/O is supported for this file, then the asynchronous operation shall be submitted at a priority equal to a base scheduling priority minus aiocbp->aio_reqprio. If Thread Execution Scheduling is not supported, then the base scheduling priority is that of the calling process;

otherwise, the base scheduling priority is that of the calling thread.

The aiocbp value may be used as an argument to aio_error() and aio_re?
turn() in order to determine the error status and return status, re?
spectively, of the asynchronous operation while it is proceeding. If an
error condition is encountered during queuing, the function call shall
return without having initiated or queued the request. The requested
operation takes place at the absolute position in the file as given by
aio_offset, as if lseek() were called immediately prior to the opera?
tion with an offset equal to aio_offset and a whence equal to SEEK_SET.
After a successful call to enqueue an asynchronous I/O operation, the
value of the file offset for the file is unspecified.
The aio_sigevent member specifies the notification which occurs when
the request is completed.
The aiocbp->aio_lio_opcode field shall be ignored by aio_read().
The aiocbp argument points to an aiocb structure. If the buffer pointed
to by aiocbp->aio_buf or the control block pointed to by aiocbp becomes
an illegal address prior to asynchronous I/O completion, then the be?
havior is undefined.
Simultaneous asynchronous operations using the same aiocbp produce un?
defined results.
If synchronized I/O is enabled on the file associated with
aiocbp->aio_fildes, the behavior of this function shall be according to
the definitions of synchronized I/O data integrity completion and syn?
chronized I/O file integrity completion.
For any system action that changes the process memory space while an
asynchronous I/O is outstanding to the address range being changed, the
result of that action is undefined.
For regular files, no data transfer shall occur past the offset maximum
established in the open file description associated with
aiocbp->aio_fildes.

RETURN VALUE

The aio_read() function shall return the value zero if the I/O opera?
tion is successfully queued; otherwise, the function shall return the
value -1 and set errno to indicate the error.

ERRORS

The aio_read() function shall fail if:

EAGAIN The requested asynchronous I/O operation was not queued due to system resource limitations.

Each of the following conditions may be detected synchronously at the time of the call to aio_read(), or asynchronously. If any of the condi? tions below are detected synchronously, the aio_read() function shall return -1 and set errno to the corresponding value. If any of the con? ditions below are detected asynchronously, the return status of the asynchronous operation is set to -1, and the error status of the asyn? chronous operation is set to the corresponding value.

EBADF The aiocbp->aio_fildes argument is not a valid file descriptor open for reading.

EINVAL The file offset value implied by aiocbp->aio_offset would be in? valid,

aiocbp->aio_reqprio is not a valid value, or aiocbp->aio_nbytes is an invalid value.

In the case that the aio_read() successfully queues the I/O operation but the operation is subsequently canceled or encounters an error, the return status of the asynchronous operation is one of the values nor? mally returned by the read() function call. In addition, the error sta? tus of the asynchronous operation is set to one of the error statuses normally set by the read() function call, or one of the following val? ues:

EBADF The aiocbp->aio_fildes argument is not a valid file descriptor open for reading.

ECANCELED

The requested I/O was canceled before the I/O completed due to an explicit aio_cancel() request.

EINVAL The file offset value implied by aiocbp->aio_offset would be in? valid.

The following condition may be detected synchronously or asyn? chronously:

EOVERFLOW

The file is a regular file, aiobcp->aio_nbytes is greater than 0, and the starting offset in aiobcp->aio_offset is before the end-of-file and is at or beyond the offset maximum in the open file description associated with aiocbp->aio_fildes.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

aio_cancel(), aio_error(), lio_listio(), aio_return(), aio_write(), close(), exec, exit(), fork(), lseek(), read()

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

COPYRIGHT