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_write.3p' command

$ man aio_write.3p

AIO_WRITE(3P) POSIX Programmer's Manual AIO_WRITE(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_write ? asynchronous write to a file

SYNOPSIS

#include <aio.h>

int aio_write(struct aiocb *aiocbp);

DESCRIPTION

The aio_write() function shall write aiocbp->aio_nbytes to the file as? sociated with aiocbp->aio_fildes from the buffer pointed to by aiocbp->aio_buf. The function shall return when the write request has been initiated or, at a minimum, queued to the file or device.

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 argument 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.

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.

If O_APPEND is not set for the file descriptor aio_fildes, then the re? quested operation shall take place at the absolute position in the file as given by aio_offset, as if lseek() were called immediately prior to the operation with an offset equal to aio_offset and a whence equal to SEEK_SET. If O_APPEND is set for the file descriptor, or if aio_fildes is associated with a device that is incapable of seeking, write opera? tions append to the file in the same order as the calls were made, ex? cept under circumstances described in Section 2.8.2, Asynchronous I/O.

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_write().

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_write() 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_write() 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_write(), or asynchronously. If any of the con? ditions below are detected synchronously, the aio_write() function shall return -1 and set errno to the corresponding value. If any of the conditions below are detected asynchronously, the return status of the asynchronous operation shall be set to -1, and the error status of the asynchronous operation is set to the corresponding value.

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

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_write() successfully queues the I/O operation, the return status of the asynchronous operation shall be one of the values normally returned by the write() function call. If the operation is successfully queued but is subsequently canceled or encounters an error, the error status for the asynchronous operation contains one of the values normally set by the write() function call, or one of the following:

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

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

ECANCELED

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

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

EFBIG The file is a regular file, aiobcp->aio_nbytes is greater than 0, and the starting offset in aiobcp->aio_offset 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

Section 2.8.2, Asynchronous I/O, aio_cancel(), aio_error(), aio_read(), aio_return(), close(), exec, exit(), fork(), lio_listio(), lseek(), 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 -- Por? table Operating System Interface (POSIX), The Open Group Base Specifi? cations 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.ker?

nel.org/doc/man-pages/reporting_bugs.html .