NAME

aio.h - asynchronous input and output

SYNOPSIS

```
#include <aio.h>
```

DESCRIPTION

The *<aio.h>* header shall define the **aiocb** structure, which shall include at least the following members:

```
int             aio_fildes     File descriptor.
off_t           aio_offset     File offset.
volatile void  *aio_buf        Location of buffer.
size_t          aio_nbytes     Length of transfer.
int             aio_reqprio    Request priority offset.
struct sigevent aio_sigevent   Signal number and value.
int             aio_lio_opcode Operation to be performed.
```

The *<aio.h>* header shall define the **off_t**, **pthread_attr_t**, **size_t**, and **ssize_t** types as described in *<sys/types.h>*.

The *<aio.h>* header shall define the **struct timespec** structure as described in *<time.h>*.

The *<aio.h>* header shall define the **sigevent** structure and **sigval** union as described in *<signal.h>*.

The *<aio.h>* header shall define the following symbolic constants:

AIO_ALLDONE
: A return value indicating that none of the requested operations could be canceled since they are already complete.

AIO_CANCELED
: A return value indicating that all requested operations have been canceled.

AIO_NOTCANCELED
: A return value indicating that some of the requested operations could not be canceled since they are in progress.

LIO_NOP
: A *lio_listio*() element operation option indicating that no transfer is requested.

LIO_NOWAIT
: A *lio_listio*() synchronization operation indicating that the calling thread is to continue execution while the *lio_listio*() operation is being performed, and no notification is given when the operation is complete.

LIO_READ
: A *lio_listio*() element operation option requesting a read.

LIO_WAIT
: A *lio_listio*() synchronization operation indicating that the calling thread is to suspend until the *lio_listio*() operation is complete.

LIO_WRITE
: A *lio_listio*() element operation option requesting a write.

The following shall be declared as functions and may also be defined as macros. Function prototypes shall be provided.

```
int     aio_cancel(int, struct aiocb *);
int     aio_error(const struct aiocb *);
[FSC|SIO]
int     aio_fsync(int, struct aiocb *);

int     aio_read(struct aiocb *);
ssize_t aio_return(struct aiocb *);
int     aio_suspend(const struct aiocb *const [], int,
            const struct timespec *);
int     aio_write(struct aiocb *);
int     lio_listio(int, struct aiocb *restrict const [restrict], int,
            struct sigevent *restrict);
```

Inclusion of the *<aio.h>* header may make visible symbols defined in the headers *<fcntl.h>*, *<signal.h>*, and *<time.h>*.

The following sections are informative.

APPLICATION USAGE

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

<fcntl.h>, *<signal.h>* , *<sys/types.h>*, *<time.h>*

XSH *aio_cancel*, *aio_error* , *aio_fsync*, *aio_read*, *aio_return*, *aio_suspend*, *aio_write*, *fsync*, *lio_listio*, *lseek*, *read*, *write*

CHANGE HISTORY

First released in Issue 5. Included for alignment with the POSIX Realtime Extension.

Issue 6

The *<aio.h>* header is marked as part of the Asynchronous Input and Output option.

The description of the constants is expanded.

The **restrict** keyword is added to the prototype for *lio_listio()*.

Issue 7

The *<aio.h>* header is moved from the Asynchronous Input and Output option to the Base.

This reference page is clarified with respect to macros and symbolic constants, and type and structure declarations are added.

POSIX.1-2008, Technical Corrigendum 1, XBD/TC1-2008/0038 [98] is applied.

POSIX.1-2008, Technical Corrigendum 2, XBD/TC2-2008/0058 [579] is applied.

End of informative text.

---

return to top of page

[Main Index | XBD | XSH | XCU | XRAT]

<<< Previous Home Next >>>