

NAME

aio_mlock - asynchronous mlock(2) operation

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

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

```
int
```

```
aio_mlock(struct aiocb *iocb);
```

DESCRIPTION

The **aio_mlock()** system call allows the calling process to lock into memory the physical pages associated with the virtual address range starting at *iocb->aio_buf* for *iocb->aio_nbytes* bytes. The call returns immediately after the locking request has been enqueued; the operation may or may not have completed at the time the call returns.

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 aio(4) limits), then the call returns without having enqueued the request.

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_mlock()** call to avoid passing bogus context information to the kernel.

Modifications of the Asynchronous I/O Control Block structure or the memory mapping described by the virtual address range are not allowed while the request is queued.

RETURN VALUES

The **aio_mlock()** function returns the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **aio_mlock()** system call will fail if:

- | | |
|----------|--|
| [EAGAIN] | The request was not queued because of system resource limitations. |
| [EINVAL] | The asynchronous notification method in <i>iocb->aio_sigevent.sigev_notify</i> is invalid or not supported. |

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 **mlock(2)** system call, and the value returned by the **aio_error()** system call is one of the error returns from the **mlock(2)** system call, or **ECANCELED** if the request was explicitly cancelled via a call to **aio_cancel()**.

SEE ALSO

aio_cancel(2), **aio_error(2)**, **aio_return(2)**, **mlock(2)**, **sigevent(3)**, **aio(4)**

PORTABILITY

The **aio_mlock()** system call is a FreeBSD extension, and should not be used in portable code.

HISTORY

The **aio_mlock()** system call first appeared in FreeBSD 10.0.

AUTHORS

The system call was introduced by Gleb Smirnoff <glebius@FreeBSD.org>.