

NAME

aio_suspend – wait for asynchronous I/O operation or timeout

LIBRARY

Real-time library (*librt*, *-lrt*)

SYNOPSIS

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

```
int aio_suspend(int n;
                const struct aiocb *const aiocb_list[n], int n,
                const struct timespec *restrict timeout);
```

DESCRIPTION

The **aio_suspend()** function suspends the calling thread until one of the following occurs:

- One or more of the asynchronous I/O requests in the list *aiocb_list* has completed.
- A signal is delivered.
- *timeout* is not NULL and the specified time interval has passed. (For details of the *timespec* structure, see **nanosleep(2)**.)

The *n* argument specifies the number of items in *aiocb_list*. Each item in the list pointed to by *aiocb_list* must be either NULL (and then is ignored), or a pointer to a control block on which I/O was initiated using **aio_read(3)**, **aio_write(3)**, or **lio_listio(3)**. (See **aio(7)** for a description of the *aiocb* structure.)

If **CLOCK_MONOTONIC** is supported, this clock is used to measure the timeout interval (see **clock_gettime(2)**).

RETURN VALUE

If this function returns after completion of one of the I/O requests specified in *aiocb_list*, 0 is returned. Otherwise, -1 is returned, and *errno* is set to indicate the error.

ERRORS**EAGAIN**

The call timed out before any of the indicated operations had completed.

EINTR

The call was ended by signal (possibly the completion signal of one of the operations we were waiting for); see **signal(7)**.

ENOSYS

aio_suspend() is not implemented.

ATTRIBUTES

For an explanation of the terms used in this section, see **attributes(7)**.

Interface	Attribute	Value
aio_suspend()	Thread safety	MT-Safe

STANDARDS

POSIX.1-2008.

HISTORY

glibc 2.1. POSIX.1-2001.

POSIX doesn't specify the parameters to be *restrict*; that is specific to glibc.

NOTES

One can achieve polling by using a non-NULL *timeout* that specifies a zero time interval.

If one or more of the asynchronous I/O operations specified in *aiocb_list* has already completed at the time of the call to **aio_suspend()**, then the call returns immediately.

To determine which I/O operations have completed after a successful return from **aio_suspend()**, use **aio_error(3)** to scan the list of *aiocb* structures pointed to by *aiocb_list*.

BUGS

The glibc implementation of **aio_suspend()** is not async-signal-safe, in violation of the requirements of POSIX.1.

SEE ALSO

aio_cancel(3), **aio_error(3)**, **aio_fsync(3)**, **aio_read(3)**, **aio_return(3)**, **aio_write(3)**, **lio_listio(3)**, **aio(7)**, **time(7)**