

NAME

aio_init – asynchronous I/O initialization

LIBRARY

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

SYNOPSIS

```
#define _GNU_SOURCE      /* See feature_test_macros(7) */
#include <aio.h>

void aio_init(const struct aiocb *init);
```

DESCRIPTION

The GNU-specific **aio_init()** function allows the caller to provide tuning hints to the glibc POSIX AIO implementation. Use of this function is optional, but to be effective, it must be called before employing any other functions in the POSIX AIO API.

The tuning information is provided in the buffer pointed to by the argument *init*. This buffer is a structure of the following form:

```
struct aiocb {
    int aio_threads;    /* Maximum number of threads */
    int aio_num;        /* Number of expected simultaneous
                        requests */
    int aio_locks;      /* Not used */
    int aio_usedba;     /* Not used */
    int aio_debug;      /* Not used */
    int aio_numusers;   /* Not used */
    int aio_idle_time;  /* Number of seconds before idle thread
                        terminates (since glibc 2.2) */
    int aio_reserved;
```

The following fields are used in the *aiocb* structure:

aio_threads

This field specifies the maximum number of worker threads that may be used by the implementation. If the number of outstanding I/O operations exceeds this limit, then excess operations will be queued until a worker thread becomes free. If this field is specified with a value less than 1, the value 1 is used. The default value is 20.

aio_num

This field should specify the maximum number of simultaneous I/O requests that the caller expects to enqueue. If a value less than 32 is specified for this field, it is rounded up to 32. The default value is 64.

aio_idle_time

This field specifies the amount of time in seconds that a worker thread should wait for further requests before terminating, after having completed a previous request. The default value is 1.

STANDARDS

GNU.

HISTORY

glibc 2.1.

SEE ALSO

aio(7)