

NAME

pthread_barrier_destroy, **pthread_barrier_init**, **pthread_barrier_wait** - destroy, initialize or wait on a barrier object

LIBRARY

POSIX Threads Library (libpthread, -lpthread)

SYNOPSIS

```
#include <pthread.h>
```

int

```
pthread_barrier_destroy(pthread_barrier_t *barrier);
```

int

```
pthread_barrier_init(pthread_barrier_t *restrict barrier, const pthread_barrierattr_t *attr,  
    unsigned count);
```

int

```
pthread_barrier_wait(pthread_barrier_t *barrier);
```

DESCRIPTION

The **pthread_barrier_init**() function will initialize *barrier* with attributes specified in *attr*, or if it is NULL, with default attributes. The number of threads that must call **pthread_barrier_wait**() before any of the waiting threads can be released is specified by *count*. The **pthread_barrier_destroy**() function will destroy *barrier* and release any resources that may have been allocated on its behalf.

The **pthread_barrier_wait**() function will synchronize calling threads at *barrier*. The threads will be blocked from making further progress until a sufficient number of threads calls this function. The number of threads that must call it before any of them will be released is determined by the *count* argument to **pthread_barrier_init**(). Once the threads have been released the barrier will be reset.

IMPLEMENTATION NOTES

In 1:1 Threading Library (libthr, -lthr) the PTHREAD_BARRIER_SERIAL_THREAD return value will always be returned by the last thread to reach the barrier.

RETURN VALUES

If successful, both **pthread_barrier_destroy**() and **pthread_barrier_init**() will return zero. Otherwise, an error number will be returned to indicate the error. If the call to **pthread_barrier_wait**() is successful, all but one of the threads will return zero. That one thread will return PTHREAD_BARRIER_SERIAL_THREAD. Otherwise, an error number will be returned to indicate

the error.

None of these functions will return EINTR.

ERRORS

The **pthread_barrier_destroy()** function will fail if:

[EBUSY] An attempt was made to destroy *barrier* while it was in use.

The **pthread_barrier_destroy()** and **pthread_barrier_wait()** functions may fail if:

[EINVAL] The value specified by *barrier* is invalid.

The **pthread_barrier_init()** function will fail if:

[EAGAIN] The system lacks resources, other than memory, to initialize *barrier*.

[EINVAL] The *count* argument is less than 1.

[ENOMEM] Insufficient memory to initialize *barrier*.

SEE ALSO

pthread_barrierattr(3)

HISTORY

The **pthread_barrier_destroy()**, **pthread_barrier_init()** and **pthread_barrier_wait()** functions first appeared in N:M Threading Library (libkse, -lkse) in FreeBSD 5.2, and in 1:1 Threading Library (libthr, -lthr) in FreeBSD 5.3.