

NAME

pthread_spin_init, **pthread_spin_destroy** - initialize or destroy a spin lock

LIBRARY

POSIX Threads Library (libpthread, -lpthread)

SYNOPSIS

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

int

```
pthread_spin_init(pthread_spinlock_t *lock, int pshared);
```

int

```
pthread_spin_destroy(pthread_spinlock_t *lock);
```

DESCRIPTION

The **pthread_spin_init()** function will initialize *lock* to an unlocked state and allocate any resources necessary to begin using it. If *pshared* is set to `PTHREAD_PROCESS_SHARED`, any thread, whether belonging to the process in which the spinlock was created or not, that has access to the memory area where *lock* resides, can use *lock*. If it is set to `PTHREAD_PROCESS_PRIVATE`, it can only be used by threads within the same process.

The **pthread_spin_destroy()** function will destroy *lock* and release any resources that may have been allocated on its behalf.

RETURN VALUES

If successful, both **pthread_spin_init()** and **pthread_spin_destroy()** will return zero. Otherwise, an error number will be returned to indicate the error.

Neither of these functions will return `EINTR`.

ERRORS

The **pthread_spin_init()** and **pthread_spin_destroy()** functions will fail if:

[EBUSY] An attempt to initialize or destroy *lock* while it is in use.

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

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

[EAGAIN] Insufficient resources, other than memory, to initialize *lock*.

[ENOMEM] Insufficient memory to initialize *lock*.

SEE ALSO

pthread_spin_lock(3), pthread_spin_unlock(3)

HISTORY

The **pthread_spin_init()** and **pthread_spin_destroy()** 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. Support for process-shared spinlocks appeared in FreeBSD 11.0.