

**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.