
*Full credit is given to the above companies including the OS that this PDF file was generated!*

## *Red Hat Enterprise Linux Release 9.2 Manual Pages on 'pthread_spin_destroy.3p' command*

***$ man pthread_spin_destroy.3p***

PTHREAD_SPIN_DESTROY(3P)  POSIX Programmer's Manual  PTHREAD_SPIN_DESTROY(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

pthread_spin_destroy, pthread_spin_init ? destroy or initialize a spin lock object

SYNOPSIS

```
#include <pthread.h>
int pthread_spin_destroy(pthread_spinlock_t *lock);
int pthread_spin_init(pthread_spinlock_t *lock, int pshared);
```

DESCRIPTION

The pthread_spin_destroy() function shall destroy the spin lock refer? enced by lock and release any resources used by the lock. The effect of subsequent use of the lock is undefined until the lock is reinitialized by another call to pthread_spin_init(). The results are undefined if pthread_spin_destroy() is called when a thread holds the lock, or if this function is called with an uninitialized thread spin lock.

The pthread_spin_init() function shall allocate any resources required to use the spin lock referenced by lock and initialize the lock to an unlocked state.

If the Thread Process-Shared Synchronization option is supported and the value of pshared is PTHREAD_PROCESS_SHARED, the implementation shall permit the spin lock to be operated upon by any thread that has access to the memory where the spin lock is allocated, even if it is allocated in memory that is shared by multiple processes.

See Section 2.9.9, Synchronization Object Copies and Alternative Map? pings for further requirements.

The results are undefined if pthread_spin_init() is called specifying an already initialized spin lock. The results are undefined if a spin lock is used without first being initialized.

If the pthread_spin_init() function fails, the lock is not initialized and the contents of lock are undefined.

Only the object referenced by lock may be used for performing synchro? nization.

The result of referring to copies of that object in calls to pthread_spin_destroy(), pthread_spin_lock(), pthread_spin_trylock(), or pthread_spin_unlock() is undefined.

RETURN VALUE

Upon successful completion, these functions shall return zero; other? wise, an error number shall be returned to indicate the error.

ERRORS

The pthread_spin_init() function shall fail if:

EAGAIN The system lacks the necessary resources to initialize another spin lock.

ENOMEM Insufficient memory exists to initialize the lock.

These functions shall not return an error code of [EINTR].

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

None.

RATIONALE

If an implementation detects that the value specified by the lock argu?

ment to pthread_spin_destroy() does not refer to an initialized spin lock object, it is recommended that the function should fail and report an [EINVAL] error.

If an implementation detects that the value specified by the lock argument to pthread_spin_destroy() or pthread_spin_init() refers to a locked spin lock object, or detects that the value specified by the lock argument to pthread_spin_init() refers to an already initialized spin lock object, it is recommended that the function should fail and report an [EBUSY] error.

## FUTURE DIRECTIONS

None.

## SEE ALSO

pthread_spin_lock(), pthread_spin_unlock()

The Base Definitions volume of POSIX.1?2017, <pthread.h>

## COPYRIGHT