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_rwlock_destroy.3p' command

$ man pthread_rwlock_destroy.3p

PTHREAD_RWLOCK_DESTROY(3P) POSIX Programmer's ManualPTHREAD_RWLOCK_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_rwlock_destroy, pthread_rwlock_init ? destroy and initialize a read-write lock object

SYNOPSIS

```
#include <pthread.h>
int pthread_rwlock_destroy(pthread_rwlock_t *rwlock);
int pthread_rwlock_init(pthread_rwlock_t *restrict rwlock,
    const pthread_rwlockattr_t *restrict attr);
pthread_rwlock_t rwlock = PTHREAD_RWLOCK_INITIALIZER;
```

DESCRIPTION

The pthread_rwlock_destroy() function shall destroy the read-write lock object referenced by rwlock 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_rwlock_init(). An implementa? tion may cause pthread_rwlock_destroy() to set the object referenced by rwlock to an invalid value. Results are undefined if pthread_rwlock_de? stroy() is called when any thread holds rwlock. Attempting to destroy

an uninitialized read-write lock results in undefined behavior.

The pthread_rwlock_init() function shall allocate any resources re? quired to use the read-write lock referenced by rwlock and initializes the lock to an unlocked state with attributes referenced by attr. If attr is NULL, the default read-write lock attributes shall be used; the effect is the same as passing the address of a default read-write lock attributes object. Once initialized, the lock can be used any number of times without being reinitialized. Results are undefined if pthread_rwlock_init() is called specifying an already initialized read-write lock. Results are undefined if a read-write lock is used without first being initialized.

If the pthread_rwlock_init() function fails, rwlock shall not be ini? tialized and the contents of rwlock are undefined.

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

In cases where default read-write lock attributes are appropriate, the macro PTHREAD_RWLOCK_INITIALIZER can be used to initialize read-write locks. The effect shall be equivalent to dynamic initialization by a call to pthread_rwlock_init() with the attr parameter specified as NULL, except that no error checks are performed.

The behavior is undefined if the value specified by the attr argument to pthread_rwlock_init() does not refer to an initialized read-write lock attributes object.

RETURN VALUE

If successful, the pthread_rwlock_destroy() and pthread_rwlock_init() functions shall return zero; otherwise, an error number shall be re? turned to indicate the error.

ERRORS

The pthread_rwlock_init() function shall fail if:

EAGAIN The system lacked the necessary resources (other than memory) to initialize another read-write lock.

ENOMEM Insufficient memory exists to initialize the read-write lock.

EPERM The caller does not have the privilege to perform the operation.

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

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

Applications using these and related read-write lock functions may be subject to priority inversion, as discussed in the Base Definitions volume of POSIX.1?2017, Section 3.291, Priority Inversion.

RATIONALE

If an implementation detects that the value specified by the rwlock ar? gument to pthread_rwlock_destroy() does not refer to an initialized read-write 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 attr argu? ment to pthread_rwlock_init() does not refer to an initialized read-write lock attributes object, it is recommended that the function should fail and report an [EINVAL] error.

If an implementation detects that the value specified by the rwlock ar? gument to pthread_rwlock_destroy() or pthread_rwlock_init() refers to a locked read-write lock object, or detects that the value specified by the rwlock argument to pthread_rwlock_init() refers to an already ini? tialized read-write lock object, it is recommended that the function should fail and report an [EBUSY] error.

FUTURE DIRECTIONS

None.

SEE ALSO

pthread_rwlock_rdlock(), pthread_rwlock_timedrdlock(), pthread_rwlock_timedwrlock(), pthread_rwlock_trywrlock(), pthread_rwlock_unlock()

The Base Definitions volume of POSIX.1?2017, Section 3.291, Priority Inversion, <pthread.h>

COPYRIGHT