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_barrier_destroy.3p' command

$ man pthread_barrier_destroy.3p

PTHREAD_BARRIER_DESTROY(3P)POSIX Programmer's ManuaPTHREAD_BARRIER_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_barrier_destroy, pthread_barrier_init ? destroy and initialize a barrier object

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 *restrict attr, unsigned count);
```

DESCRIPTION

The pthread_barrier_destroy() function shall destroy the barrier refer? enced by barrier and release any resources used by the barrier. The ef? fect of subsequent use of the barrier is undefined until the barrier is reinitialized by another call to pthread_barrier_init(). An implemen? tation may use this function to set barrier to an invalid value. The results are undefined if pthread_barrier_destroy() is called when any thread is blocked on the barrier, or if this function is called with an uninitialized barrier.

The pthread_barrier_init() function shall allocate any resources re? quired to use the barrier referenced by barrier and shall initialize the barrier with attributes referenced by attr. If attr is NULL, the default barrier attributes shall be used; the effect is the same as passing the address of a default barrier attributes object. The results are undefined if pthread_barrier_init() is called when any thread is blocked on the barrier (that is, has not returned from the pthread_bar? rier_wait() call). The results are undefined if a barrier is used with? out first being initialized. The results are undefined if pthread_bar? rier_init() is called specifying an already initialized barrier.

The count argument specifies the number of threads that must call pthread_barrier_wait() before any of them successfully return from the call. The value specified by count must be greater than zero.

If the pthread_barrier_init() function fails, the barrier shall not be initialized and the contents of barrier are undefined.

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

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_barrier_init() function shall fail if:

EAGAIN The system lacks the necessary resources to initialize another barrier.

EINVAL The value specified by count is equal to zero.

ENOMEM Insufficient memory exists to initialize the barrier.

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 barrier argument to pthread_barrier_destroy() does not refer to an initialized barrier object, it is recommended that the function should fail and re? port an [EINVAL] error.

If an implementation detects that the value specified by the attr argu? ment to pthread_barrier_init() does not refer to an initialized barrier 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 barrier argument to pthread_barrier_destroy() or pthread_barrier_init() refers to a barrier that is in use (for example, in a pthread_barrier_wait() call) by another thread, or detects that the value specified by the barrier argument to pthread_barrier_init() refers to an already ini? tialized barrier object, it is recommended that the function should fail and report an [EBUSY] error.

FUTURE DIRECTIONS

None.

SEE ALSO

pthread_barrier_wait()

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

COPYRIGHT

nel.org/doc/man-pages/reporting_bugs.html .