NAME

threads_into - introduction to the Adelsbach Enhanced Threading Library

SYNOPSIS

library "threads"
#include <threads.h>
#include <threadse.h>

DESCRIPTION

The Adelsbach Enhanced Threading Library aims to implement a conform C11 (ISO/IEC 9899:2011) threading interface on top the standard POSIX threading API (IEEE Std 1003.1c-1995). It furthermore implements extensions to the former for spinlocks, read/write locks and thread barriers.

The C11 standard functions are declared in the *threads.h* header file, whereas all extensions are declared in the *threadse.h* header file.

The following functionality is supported:

Threading
: These implement the standard threading functions.

  thrd_create(3) thrd_exit(3) thrd_current(3) thrd_detach(3) thrd_join(3) thrd_equal(3) thrd_sleep(3) thrd_yield(3)

Mutexes
: Mutexes will stall a thread until an exclusive lock on the mutex can be acquired.

  mtx_init(3) mtx_destroy(3) mtx_lock(3) mtx_timedlock(3) mtx_trylock(3) mtx_unlock(3)

Condition variables
: Condition variables will stall a single or multiple threads until a condition has been send out by an other thread. By comparison to a mutex multiple threads can continue execution after the condition has been given.

  cnd_init(3) cnd_destroy(3) cnd_broadcast(3) cnd_signal(3) cnd_wait(3) cnd_timedwait(3)

Thread-local storage
: Thread-local storage allocates global variables with data specific to each thread.

  tss_create(3) tss_delete(3) tss_get(3) tss_set(3)

Call once
: This synchronization primitive allows only a single thread to execute a specified function

exactly once.

call_once(3)

Read/Write Locks

Read/Write locks allow synchronization for read/write operations. Threads can hold either read or write locks, multiple read locks can be held by multiple threads but only one write lock can be held at a time. The write lock can only be acquired if no read locks are acquired, the read lock can only be acquired once there is no write lock acquired.

rwmtx_init(3) rwmtx_destroy(3) rwmtx_rdlock(3) rwmtx_wrlock(3) rwmtx_tryrdlock(3) rwmtx_trywrlock(3) rwmtx_timedrdlock(3) rwmtx_timedwrlock(3) rwmtx_unlock(3)

Spinlocks

Spinlocks are fast mutexes, that loop on an atomic variable rather than yielding the thread. They are more lightweight than a normal mutex and are suitable for short synchronizations that will not require much wait.

smtx_init(3) smtx_destroy(3) smtx_lock(3) smtx_unlock(3) smtx_trylock(3)

Barriers

Barriers allow for synchronization of thread groups. A barrier will stop all threads upon entering the barrier until a specified amount of threads is waiting, after which execution continues.

bar_init(3) bar_destroy(3) bar_wait(3)

SEE ALSO

pthreads(3)

AUTHORS

Jan Adelsbach <jan@jadelsbach.de>