Fork me on GitHub

- Tutorials
    - MPI
    - POSIX
-
-

Home / Posix / Creating and Destroying Mutexes

Creating and Destroying Mutexes

Routines:

`pthread_mutex_init(mutex, attr)`

`pthread_mutex_destroy(mutex)`

`pthread_mutexattr_init(attr)`

`pthread_mutexattr_destroy(attr)`

Usage:

Mutex variables must be declared with type `pthread_mutex_t`, and must be initialized before they can be used. There are two ways to initialize a mutex variable:

1. Statically, when it is declared. For example:

```
pthread_mutex_t mymutex = PTHREAD_MUTEX_INITIALIZER;
```

2. Dinamically, using `pthread_mutex_init()`. For example:

```
pthread_mutex_t mymutex;
pthread_mutex_init(&mymutex, NULL);
```

The mutex is initially unlocked.

The `attr` object is used to establish properties for the mutex object, and must be of type `pthread_mutexattr_t` if used (may be specified as NULL to accept defaults). The Pthreads standard defines three optional mutex attributes:

- Protocol: Specifies the protocol used to prevent priority inversions for a mutex.
- Prioceiling: Specifies the priority ceiling of a mutex.
- Process-shared: Specifies the process sharing of a mutex.

Note that not all implementations may provide the three optional mutex attributes.

The `pthread_mutexattr_init()` and `pthread_mutexattr_destroy()` routines are used to create and destroy mutex attribute objects respectively.

`pthread_mutex_destroy()` should be used to free a mutex object which is no longer needed.

Lawrence Livermore National Laboratory
| 7000 East Avenue • Livermore, CA 94550 | LLNL-WEB-458451
Operated by the Lawrence Livermore National Security, LLC for the Department of Energy's National Nuclear Security Administration Learn about the Department of Energy's Vulnerability Disclosure Program

Home Privacy & Legal Notice