Synchronization

Synchronization in C

Stefano Quer
Dipartimento di Automatica e Informatica
Politecnico di Torino

This slide introduces the topic of synchronization in C, which is crucial for managing access to shared resources in concurrent programming.

License Information

This work is licensed under the license

CC BY-NC-ND 4.0

Attribution-NonCommercial-NoDerivatives 4.0 International

This license requires that reusers give credit to the creator. It allows reusers to copy and distribute the material in any medium or format in unadapted form and for noncommercial purposes only.

BY: Credit must be given to you, the creator.

NC: Only noncommercial use of your work is permitted.
Noncommercial means not primarily intended for or directed towards commercial advantage or monetary compensation.

ND: No derivatives or adaptations of your work are permitted.

To view a copy of the license, visit:
https://creativecommons.org/licenses/by-nc-nd/4.0/?ref=chooser-v1

Semaphore implementations

- Mutexes in C are
  - Represented by object of type **mtx_t**
  - Defined in threads.h, i.e., insert
    - #include <threads.h>

Representation: Mutexes are represented by objects of type mtx_t.
Header File: Defined in threads.h, so you need to include this header file using #include <threads.h>.

- See documentation for
  - Atomic operation and fences (barriers) in C

Barriers are introduced in unit 06 section 07

Mutual exclusion

For more operations see the reference documentation

Table of Functions:
int mtx_init(mtx_t *mtx, int mtxtype);
Meaning: Create a mutex (mtx) with some properties (mtxtype).
void mtx_destroy(mtx_t *mtx);
Meaning: Destroy the mutex pointed to by mtx.
int mtx_lock(mtx_t *mtx);
Meaning: Blocks the calling thread until it obtains the mutex referenced by mtx.
int mtx_trylock(mtx_t *mtx);
Meaning: Try to obtain the mutex referenced by mtx without blocking the thread.
int mtx_timedlock(mtx_t *mtx, const struct timespec *ts);
Meaning: Try to obtain the mutex referenced by mtx but blocks the thread only for a specific time.
int mtx_unlock(mtx_t *mtx);
Meaning: Releases the mutex referenced by mtx.

Type	Meaning
int mtx_init(mtx_t *mtx, int muxtype);	Create a mutex (mtx) with some properties (muxtype).
void mtx_destroy(mtx *mtx);	Destrpy the mutex pointed by mtx.
int mtx_lock(mtx_t *mtx);	Blocks the calling thread until it obtain the mutex referenced by mtx.
int mtx_trylock(mtx_t *mtx);	Try to obtain the mutex referenced by mtx but it does not block the thread.
int mtx_timedlock(mtx_t *mtx, cont struct timespec *ts);	Try to obtain the mutex referenced by mtx but it blocks the thread only for a specific time.
int mtx_unlock(mtx_x *mtx);	Releases the mutex referred by mtx.

This slide provides a summary of the functions used to manage mutexes in C. Here's a detailed explanation of
1. mtx_init:
Purpose: Initializes a mutex with specified properties.
Parameters:
mtx_t *mtx: Pointer to the mutex to be initialized.
int mtxtype: Properties of the mutex (e.g., whether it is recursive).

```
mtx_t myMutex;
mtx_init(&myMutex, mtx_plain); // Initializes a plain mutex
```

mtx_destroy:
Purpose: Destroys a mutex, freeing any resources it may be using.
Parameters:
mtx_t *mtx: Pointer to the mutex to be destroyed.
Usage: