Para que se usa MUTEX?

Los algoritmos de exclusión mutua (comúnmente abreviada como **mutex** por mutual exclusion) se usan en programación concurrente **para** evitar que entre más de un proceso a la vez en la sección crítica. La sección crítica **es** el fragmento de código donde puede modificarse un recurso compartido (datos a los que se accede de manera concurrente).

Como se usa el MUTEX

* #include <mutex> // incluye libreria
* std::mutex mtx; //declara variable mutex
* mtx.lock()
* mtx.unlock()

Algunas veces ocurre deadlock (comportamientos indefinidos) cuando por ejeplo dos threads esperan por un mutex que es propiedad de otro proceso

**Evitar deadlocks**

Utilizar unique\_lock lock\_guard

A lock\_guard simplifies the usage of mutex in C++. It holds a lock on its mutex from the time it is created to the end of its scope. This simplifies the writing of locking and unlocking code using a mutex.

See this earlier example for usage of a mutex to control concurrent access to a queue from multiple threads. Below is the same written using lock guard:

{

std::mutex m,

std::lock\_guard<std::mutex> lockGuard(m);

sharedVariable= getVar();

}

its lifetime ends when it leaves the critical section. Exactly at that time point, the destructor of std::lock\_guard is called and - I guess, you know it - the mutex is released. It happens automatically, and, in addition, it happens if getVar() in sharedVariable = getVar() throws an exception. Of course, function body scope or loop scope also limit the lifetime of an object.