Implementation: A worker function is spawned by a thread, we have a barrier which makes sure all the threads are spawned before any work is done. This will come handy in the work-stealing, as if the worker is trying to steal from the thread that has not spawned it will segfault .

At the beginning of the worker function the mutex is locked so no other thread can edit the data, while the thread is working. The mutex is unlocked when the task is about to be executed, and locked right after the task has finished. Worker function exits with a unlocked mutex.
Future to synchronize data we used condition variable which is called in the future submit function to the let the threads know that a future is available.

Worker function also implements work-stealing, it first checks if there are any futures in its queue, then it moves on to check the global queue for futures. If there are no future on either queues, it then iterates over the workers list looking for a future that has not been started and steals it from the first worker with that condition.