The Foundation & Challenges

This lesson briefs the foundation and challenges in the C++ memory model.

WE'LL COVER THE FOLLOWING

The Foundation

The C++ memory model has to deal with the following points:

- Atomic operations: operations that can be performed without interruption.
- Partial ordering of operations: sequences of operations that must not be reordered.
- Visible effects of operations: guarantees when operations on shared variables are visible to other threads.

The foundation of the contract are operations on atomics that have two characteristics: They are by definition atomic or indivisible, and they create synchronization and order constraints on the program execution. These synchronization and order constraints will also hold for operations on non-atomics. On one hand, an operation on an atomic is always atomic; on the other hand, you can tailor the synchronizations and order constraints to your needs.

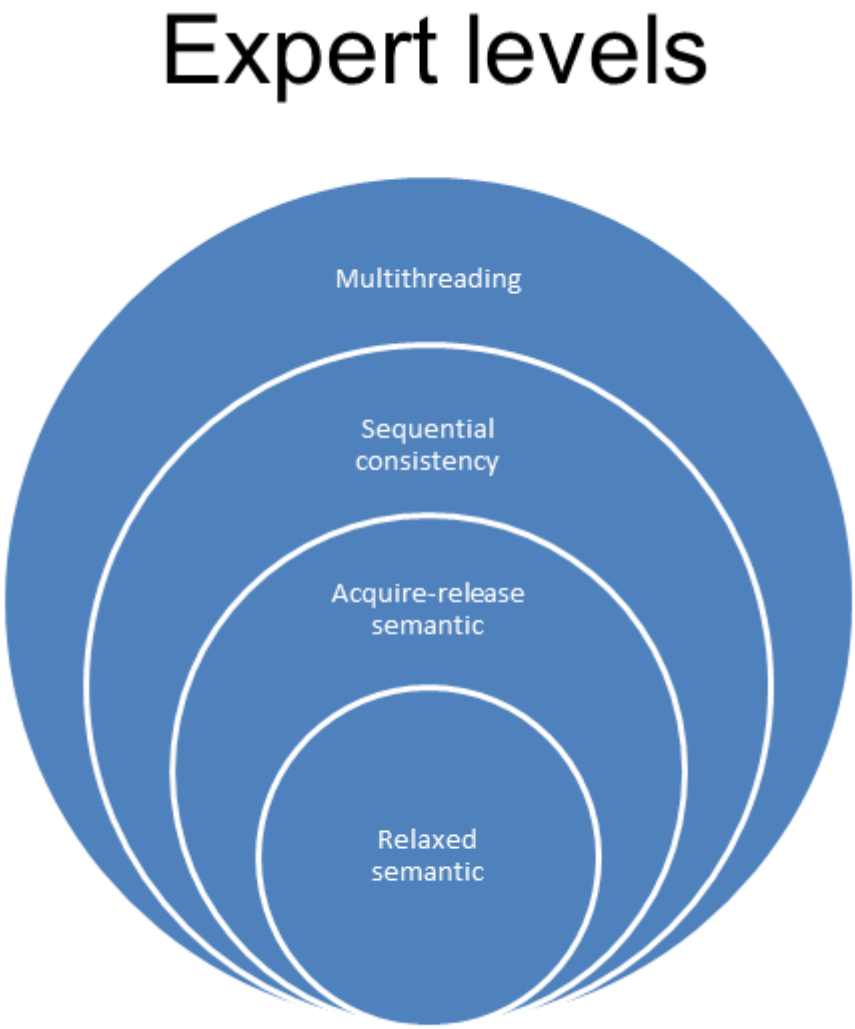
The Challenges

The more we weaken the memory model, the more we will change our focus towards other things:

- More optimization potential for the system

- The possible number of control flows of the program increases exponentially
- Domain for the experts
- Breaks our intuition of the control flow
- Areas for micro-optimization

To deal with multithreading, we should be an expert. In case we want to deal with atomics (sequential consistency), we should open the door to the next level of expertise. What will happen when we talk about the acquire-release semantic or relaxed semantic? We'll advance one step higher to (or deeper into) the next expertise level.



Now, we dive deeper into the C++ memory model and start with lock-free programming. On our journey, I will write about atomics and their operations. Once we are done with the basics, the different levels of the memory model

will follow. The starting point will be the straightforward sequential

consistency, the acquire-release semantic will follow, and the not-so-intuitive relaxed semantic will be the end point of our journey.

Let’s start with atomics in the next lesson. See ya!