

Types of Synchronization & Ordering Constraints

This lesson introduces the types of synchronization and ordering constraints in C++.

WE'LL COVER THE FOLLOWING ^

- Modification Order Consistency

There are, roughly speaking, three different types of synchronization and ordering constraints in C++:

- Sequential consistency: `memory_order_seq_cst`
- Acquire-release: `memory_order_consume`, `memory_order_acquire`, `memory_order_release` and `memory_order_acq_rel`
- Relaxed: `memory_order_relaxed`

Modification Order Consistency

While the sequential consistency establishes a global order between threads, the acquire-release semantic establishes an ordering between **read** and **write** operations on the same atomic variable with different threads. The relaxed semantic only guarantees that operations on one specific data type in the same thread cannot be reordered. This guarantee is called modification order consistency, but other threads can see this operation in a different order.

The different memory models and their effects on atomic and non-atomic operations make the C++ memory model an interesting and challenging topic. In the next lessons, let us discuss the synchronization and ordering constraints of the sequential consistency, the acquire-release semantic, and the relaxed semantics.

