Race Conditions

This lesson gives an overview of race condition problems that might occur during the implementation of concurrency in C++.

A race condition is a situation in which the result of an operation depends on the interleaving of certain individual operations.

Race conditions are quite difficult to spot; It depends on the interleaving of the threads whether they occur. That means the number of cores, the utilization of your system, or the optimization level of your executable may all be reasons why a race condition appears or does not.

Race conditions are not bad per se. It is the nature of threads that they will interleave in different ways, but this can often cause serious problems. In this case, I will call them malign race conditions. Typical effects of malign race conditions are data races, breaking of program invariants, blocking issues of threads, or lifetime issues of variables.