

Introduction to `std::async`

This lesson gives an introduction to `std::async` which is used in C++ for multithreading.

`std::async` behaves like an asynchronous function call. This function call takes a `callable` together with its arguments. `std::async` is a variadic template and can, therefore, take an arbitrary number of arguments. The call to `std::async` returns a future object `fut`. That's your handle for getting the result via `fut.get()`.



`std::async` should be your first choice

The C++ runtime decides if `std::async` is executed in a separate thread. The decision of the C++ runtime may depend on the number of CPU cores available, the utilization of your system, or the size of your work package. By using `std::async`, you only specify the task that should run; the C++ runtime automatically manages the creation and also the lifetime of the thread.

Optionally, you can specify a start policy for `std::async`.