

C++ Advanced

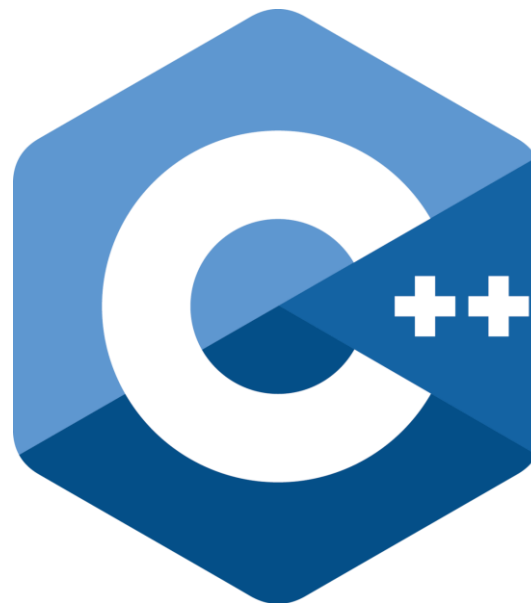
Параллельное программирование. Задачи и параллельные алгоритмы.

C++ Advanced

Автор курса



Кирилл Чернега



C++ Advanced

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал на [TestProvider.com](http://testprovider.com)

Параллельное программирование. Задачи и параллельные алгоритмы.

C++ Advanced

Содержание урока

1. Параллельное программирование на основе задач.
2. `async`.
3. `future`, `promise`.
4. Стратегии запуска.
5. Параллельные алгоритмы стандартной библиотеки.

Параллельное программирование на основе задач.

«Предпочитайте программирование на основе задач программированию на основе потоков» (Скотт Мейерс).

В `std::async` передаваемый функциональный объект рассматривается как задача.

Преимущества:

- Имеем возвращаемое значение.
- Возможность получить доступ к вызову через функцию `get()` у возвращаемого значения.
- Доступ к сгенерированному исключению через функцию `get()`.
- Ответственность за управление потоками лежит на плечах стандартной библиотеки.

Параллельное программирование на основе задач.

Случаи, когда подход на основе потоков предпочтительнее:

- Вам нужен доступ к API, лежащей в основе реализации потоков. (Для предоставления доступа к API реализации потоков `std::thread` обычно предлагает функцию-член `native_handle`.)
- Вам требуется возможность оптимизации потоков в вашем приложении.
- Вам требуется реализовать поточную технологию, выходящую за рамки API параллельных вычислений в C++, например пулы потоков на платформах, на которых ваши реализации C++ их не предоставляют.

C++ Advanced

Стратегии запуска

Применение `std::async` со стратегией запуска по умолчанию имеет некоторые интересные последствия. Для потока `t`, выполняющего приведенную ниже инструкцию, справедливы следующие утверждения.

auto fut = std::async (f) ;

- Невозможно предсказать, будет ли `f` выполняться параллельно с `t`, поскольку выполнение `f` может быть отложено планировщиком.
- Невозможно предсказать, будет ли `f` выполняться потоком, отличным от того, в котором вызываются функции-члены `get` или `wait` объекта `fut`. Если этот поток - `t`, отсюда вытекает невозможность предсказать, будет ли `f` выполняться потоком, отличным от `t`.
- Может быть невозможно предсказать, будет ли `f` выполнена вообще, поскольку может оказаться невозможно гарантировать, что функции-члены `get` или `wait` объекта `fut` будут вызваны на всех путях выполнения программы.

C++ Advanced

Параллельное программирование на основе задач.

Criteria	Thread	Task
Parties Involved	creator thread and child thread	promise and future
Communication	shared variable	communication channel
Thread Creation	obligatory	optional
Synchronisation	<code>join</code> call blocks	<code>get</code> call blocks
Exception in Child	creator thread and child thread die	return value of the promise
Forms of Communication	values	values, notifications, and exceptions

<https://meetingcpp.com/mcpp/slides/2018/BestPractices.pdf>

C++ Advanced

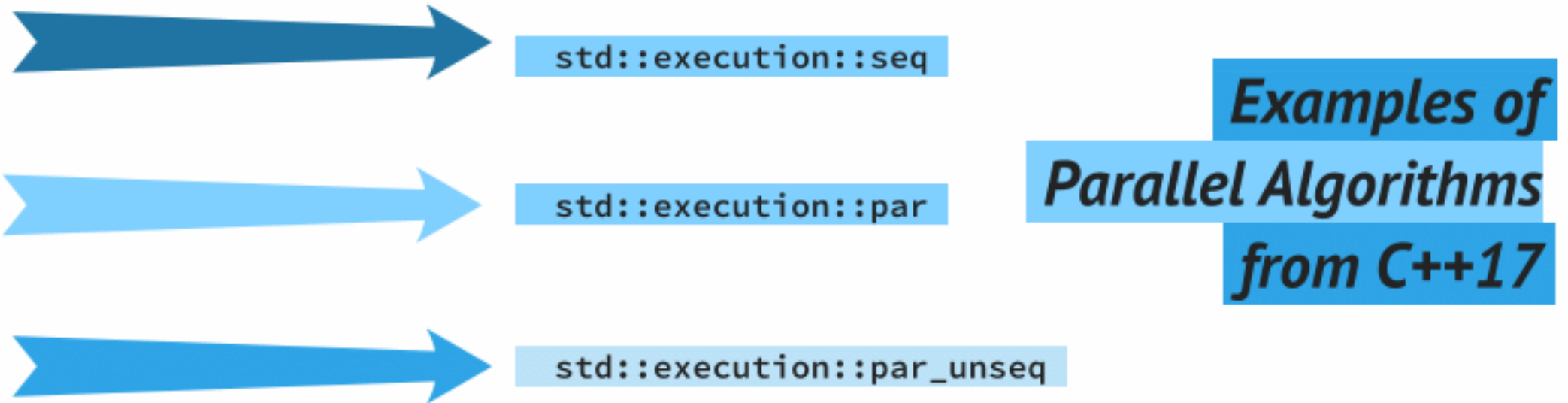
Параллельное программирование на основе задач.

Criteria	Condition Variable	Tasks
Critical Region	Yes	No
Spurious Wakeup	Yes	No
Lost Wakeup	Yes	No
Repeatedly Synchronisation	Yes	No

<https://meetingcpp.com/mcpp/slides/2018/BestPractices.pdf>

C++ Advanced

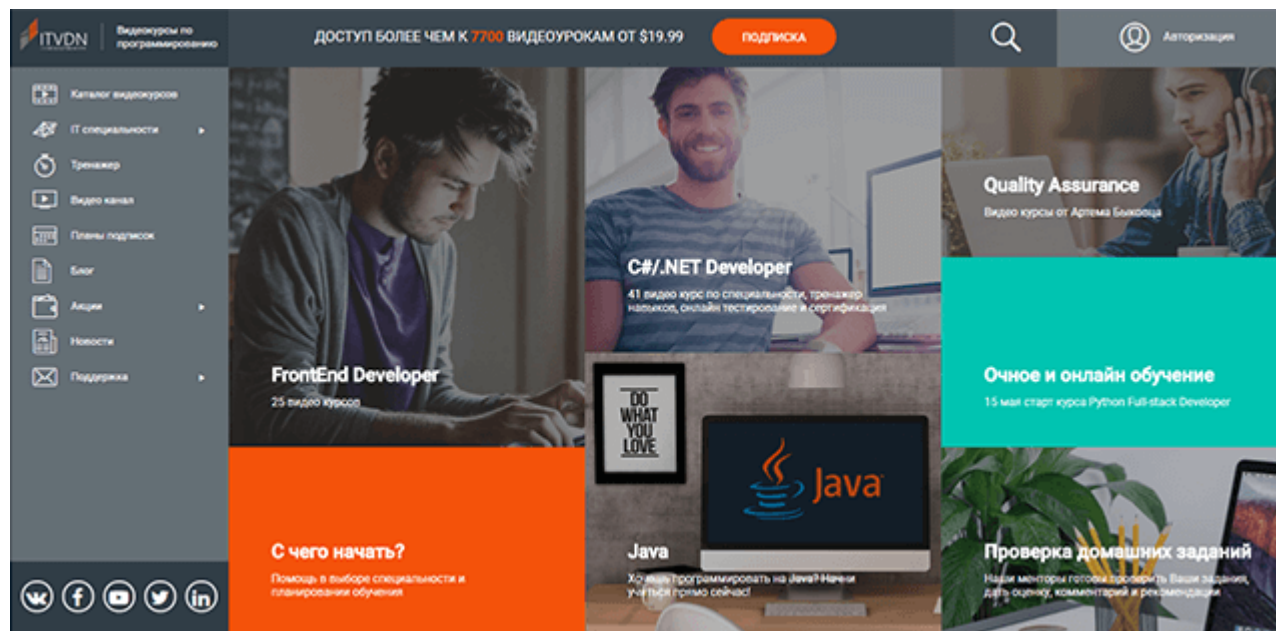
Параллельные алгоритмы STL C++17.



<https://en.cppreference.com/w/cpp/experimental/parallelism>, execution policy

Смотрите наши уроки в видео формате

ITVDN.com



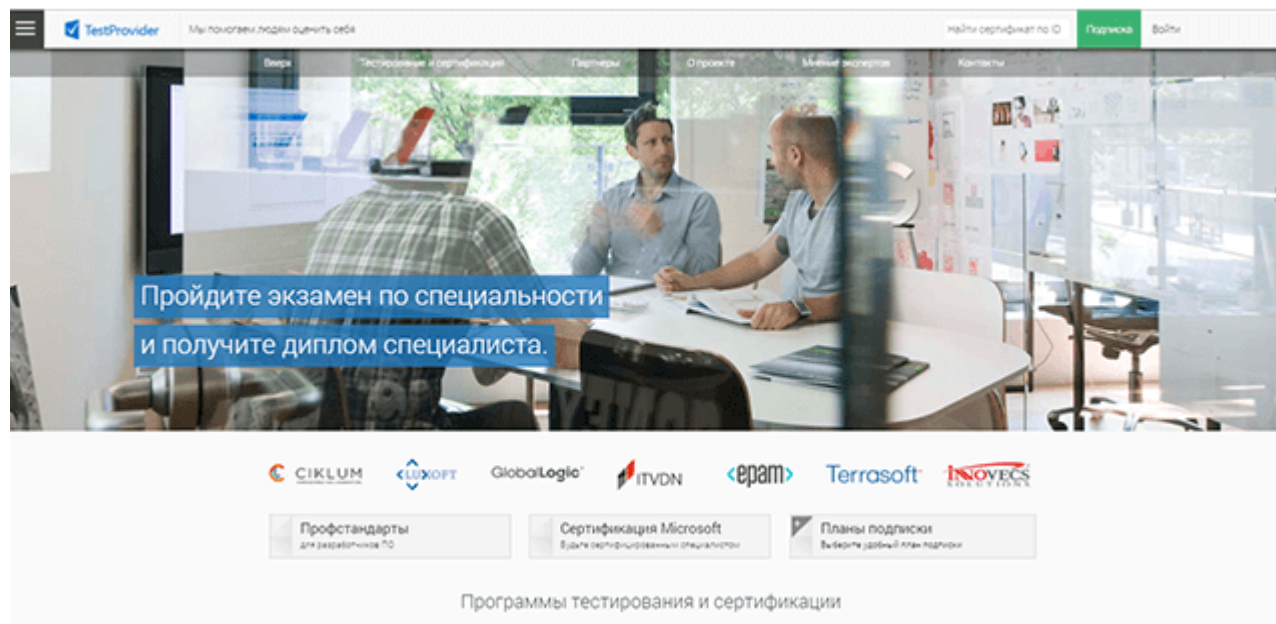
Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://itvdn.com) для закрепления пройденного материала.

Курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics и другими высококвалифицированными разработчиками.



Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](https://testprovider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



Q&A

Информационный видеосервис для разработчиков программного обеспечения

