

УДК 004.051

СРАВНЕНИЕ МОДЕЛЕЙ МНОГОПОТОЧНОГО ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ C++20

Доронин О.В., Дергун К.И (Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики (ИТМО) г.Санкт-Петербург)

Научный руководитель – к.т.н, доцент Дергачев А.М.

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики (ИТМО), г.Санкт-Петербург

Стандарт языка C++ в последнее время активно развивается. В первых стандартах языка отсутствовала поддержка даже потоков. Но уже сейчас, помимо потоков, добавлены модели `promise-future` и `coroutines`. В работе приводится сравнительный анализ каждой модели и задачи для их применения.

Введение. В многопоточном программировании на языке C++ существуют различные модели исполнения кода, к таким моделям относятся: POSIX потоки, потоки стандартной библиотеки, инструментирование кода на этапе компиляции (Open MP), использование модели `promise-future`, `coroutines`, `actors` и другие модели. Но остается открытым вопрос, какую модель и для каких задач использовать. В данной работе приведен сравнительный анализ моделей, представленных в стандартной библиотеке языка C++: `std::thread`, `promise-future`, `coroutines`.

Основная часть. Исторически стандарт языка C++ не включал в себя поддержку многопоточности. Со временем ситуация изменилась. В начале, в стандарт языка был добавлен класс `std::thread`, после этого модель `promise-future`, а уже последний стандарт включает в себя `coroutines`. Класс стандартной библиотеки `std::thread` является оберткой над нативными потоками, которые один к одному соотносятся с потоками операционной системы. Модель `promise-future` направлена на минимизацию количества потоков, используемых в системе. И последние новшества в стандарте это `coroutines`, которые направлены на упрощение работы с асинхронными операциями. В данной работе приведен сравнительный анализ описанных моделей, а также проведены эксперименты для конкретных задач.

Выводы. В ходе работы был проведен сравнительный анализ каждой модели многопоточного программирования, выделены преимущества и недостатки, определены классы решаемых задач. Результаты данного анализа можно использовать для построения высоконагруженных приложений в зависимости от поставленных задач.

Доронин О.В. (автор)

Подпись

Дергун К.И. (соавтор)

Подпись

Дергачев А.М. (научный руководитель)

Подпись