

Кривошеков Александр Александрович

C++ Developer (Системное программирование | Графика | Backend)
Новосибирск | Email: alex.krivoshecov@gmail.com Телефон: +7 913 417 84 75

Резюме

C++ разработчик с опытом в системном программировании, компьютерной графике и backend-разработке. Реализовал высокопроизводительные сервисы и игровые движки с использованием C++17/20/23, OpenGL/Vulkan, многопоточности и низкоуровневых оптимизаций. Опыт командной работы и руководства проектами (до 5 человек). Цель — разработка масштабируемых, оптимизированных систем.

Опыт работы

Turing Pi | Удалённо — C++ Developer *Июль 2024 — Апрель 2025 (контракт)*

- Разработал микросервис на C++20 (Drogon), снизив задержки сообщений до <50 мс при 10k+ соединениях.
- Реализовал WebSocket API с низкими задержками и устойчивостью к обрывам соединений.
- Разработал AI-агентов для чат-системы (управление диалогом, контекстом).
- Перевёл проект с GNU Make на CMake, улучшив структуру сборки и сократив время компиляции на 25%.

Проекты

Minecraft Clone (OpenGL 4.6)

- Procedural generation на шуме Перлина, chunk-based мир.
- Реализован pipeline: камера, освещение, текстуры, интерфейс.
- 300 FPS @ 1440p (GTX 1050ti).
- Стек: GLFW, GLEW, GLM, SFML.
- Код: <https://github.com/cpp20120/0ldGithub/tree/main/MineGL> (GitHub: MineGL)

Game Engine (C++23)

- Математическая библиотека с mdspan, TMP и Concepts для type-safe API.
- Архитектура модульного движка; планируется Vulkan-рендерер, ECS, скриптовая система.
- <https://github.com/cpp20120/EngineGithub>: Engine.

Webserver (C++20, io_uring)

- <https://github.com/cpp20120/0ldGithub/tree/main/WebServer> HTTP сервер на корутинах с io_uring; асинхронная обработка запросов.

ThreadPool (C++20, lock-free DAG runtime)

- Мини-рантайм для параллельных задач с DAG-ориентированным планировщиком.
- Собственный work-stealing пул (Chase-Lev) + центральные lock-free MPMC очереди (Michael-Scott) с Hazard Pointers/QSBR.
- Поддержка аффинити, приоритетов, back-pressure, ограничений конкуренции и явного управления переключением (Block/Drop/Fail).
- Высокоуровневый API в стиле TBB: `submit`, `then`, `when_all`, `parallel_for`.
- Документация с архитектурными диаграммами и полным Doxygen-комментированием.
- Специализированный `for_each_ws` для RA-диапазонов (lazy range split + help-first stealing).
- В моих условиях на 1e6 элементов дал ~2.8× быстрее `oneTBB::parallel_for` (RA-кейс). Общий `parallel_for` и микротаски специально не оптимизировались. Бенчмарки:
- <https://github.com/cpp20120/ThreadPool>

Вклад в Open Source

- Эксперимент по переносу hwloc (Hardware Locality) с autotools на CMake.
- Разработка библиотеки метапрограммирования (<https://github.com/cpp20120/MetaUtilsMetaUtils>).

Технические навыки

- **Языки и стандарты:** C++17/20/23, TMP, Concepts, Coroutines
- **Графика:** OpenGL 4.6, Vulkan, GLSL, SPIR-V
- **Backend:** Drogon, Boost.Beast/Asio, REST, WebSocket

- **Метапрограммирование:** CRTP, SFINAE, constexpr вычисления, шаблонные интерпретаторы
- **Многопоточность:** std::thread, TBB, folly
- **Low-latency оптимизации:** io_uring, zero-copy, lock-free (Michael–Scott queue)
- **CI/CD и тесты:** GitHub Actions, CTest, clang-format/tidy, GTest
- **Инструменты:** CMake, Ninja, Git, vcpkg, RenderDoc, Nsight
- **БД:** PostgreSQL (оптимизация запросов)
- **OS:** Linux (low-level), Windows

Образование

Сибирский государственный университет телекоммуникаций и информатики Бакалавриат, «Информатика и вычислительная техника» *2023 — 2027 (ожидаемый выпуск)*

Дополнительно

- Английский — B1 (чтение документации, участие в код-ревью)
- Увлечения: компьютерная графика, алгоритмы

Преподавательская деятельность

- Провёл цикл неформальных занятий по Git для одnogруппников (охват 20+ человек)
- Темы: от базовых команд до продвинутых практик (интерактивный rebase, submodules)
- Результат: 100% группы начали использовать Git в учебных проектах