Михаил Разаков

SOFTWARE ENGINEER · COMPILERS, RUNTIME & PERFORMANCE

Москва / удалённо · razakov.mikhail@outlook.com · Telegram · GitHub

ПРОФИЛЬ

Разрабатываю компиляторные и runtime-компоненты на C#/.NET и C++/LLVM: от проверки допустимости преобразований до генерации и проверки исполняемого кода. В SharpChecker (ИСП РАН) работаю с символьным анализом C#, в МЦСТ занимался LLVM/LICM. Для экспериментальных оптимизаций отделяю проверку корректности от измерений производительности.

ТЕХНОЛОГИИ

- C#/.NET, CIL/CLR, DynamicMethod, Reflection.Emit, Roslyn, BenchmarkDotNet; interpreter/CIL backends
- C++23/LLVM 22, IR/CFG/SSA, LICM, анализ эффектов, Rust/x86-64 (SysV ABI), differential testing, Git/Linux

ОПЫТ

ИСП РАН · SharpChecker октябрь 2026 — н. в.

Символьные вычисления и структурная метаинформация C#-кода/деклараций для статического анализа и задач обнаружения ошибок.

МЦСТ · стажировка по компиляторам июль — август 2026

C++23/LLVM 22: экспериментальный LICM с консервативными условиями допустимости; Python-проверки поведения до/после преобразования.

Коммерческая разработка · C#/.NET и смежный стек суммарно около 1 года

Прикладные и серверные задачи на C#/.NET и Unity: ASP.NET Core, SQL/PostgreSQL, файловая система и серверное окружение. Продолжительность — по данным кандидата.

ПРОЕКТЫ

UniversalToolchain · compiler/runtime-инфраструктура

Контракты модулей и зависимости проверяются в LanguagePlan до исполнения; раздельные interpreter/CIL backends позволяют сравнивать результаты одного языка. Есть отдельные сценарии проверки паритета.

LLVM Global IV / LICM · допустимость оптимизаций

Анализ межпроцедурных эффектов отделён от PromotionLegality и IR-преобразования. Volatile/atomic доступы, неизвестные вызовы и неподдерживаемый CFG приводят к отказу от оптимизации; есть негативные регрессии.

x86-64 codegen · Rust

Пайплайн SSA/CFG → liveness → register allocation → x86-64. Отдельный verifier проверяет назначения регистров и spills; native execution сравнивается с эталонным IR-интерпретатором.

Quark2 · измерение вызовов сгенерированного кода

DynamicMethod/Reflection.Emit и RuntimeMethodHandle; BenchmarkDotNet сравнивает reflection Invoke, delegate и вызов по указателю для сгенерированного метода.

ОБРАЗОВАНИЕ И ДОСТИЖЕНИЯ

НИУ ВШЭ — ФКН, Программная инженерия.

«Юниор» НИЯУ МИФИ, диплом I степени (2026); доклад LangDev 2026.