Mikhail Razakov

Compiler Backend / Code Generation Engineer

razakov.mikhail@outlook.com
github.com/Misha1302
LinkedIn

I build compiler-backend components from SSA/CFG validation and liveness to register allocation, spills, phi lowering, and SysV x86-64 emission. I check assignments independently and compare native execution with a reference interpreter; earlier worked with LLVM 22/C++ at MCST.

2022: Compilers / VMs · 2023: CIL + native x86-64 · 2024–2025: AST / IR / backends → toolchain · 2026 →: LLVM + program analysis

PROFESSIONAL EXPERIENCE

October 2026 — present

ISP RAS — SharpChecker development

- Work on symbolic computation in SharpChecker, a C#/.NET static-analysis tool.

July — August 2026

MCST — Compiler Engineering Intern

- Implemented a LICM pass for LLVM 22 in C++; hoisting conditions check loop structure, memory access, side effects, and speculative safety.
- Built a Python harness comparing program behavior before/after optimization and checking transform/no-transform cases.

PROJECTS

x86-64 Compiler Backend & Register Allocation

SSA/CFG validation → liveness/interference → register allocation/spills → phi lowering → SysV x86-64 emission.

Assignment verifier plus native-vs-interpreter differential execution; the current project CI is green; a separate public native-codegen lineage starts with Visk (2023) and Wist4 (2024).

LLVM Interprocedural Global IV Optimization

An LLVM 22 pass localizes supported induction-like globals after interprocedural effect analysis, affine evolution, and a separate applicability check; APInt models fixed-width wraparound.

29 positive/negative regression cases cover LLVM Verifier checks, structural expectations, idempotence, and observable before/after behavior.

UniversalToolchain / Wist2

Dependency/conflict resolution, implementation selection, pass ordering, and typed artifact routes before execution.

1,324/1,324 tests; interpreter/CIL parity and configuration checks before runtime.

SKILLS

Backend / Codegen

x86-64, SysV ABI, stack frames, calling conventions, liveness/interference, register allocation, spills, phi lowering, native execution

Low-level / Assembly

Iced.Intel code emission, executable-memory JIT experiments; NASM IA-32, CDECL, x87

LLVM / IR

C++23, LLVM 22/IR, CFG/SSA, loop/interprocedural analysis, IR transformations

Verification

assignment verifier, LLVM Verifier, structural checks, idempotence, differential execution

EDUCATION

HSE University — Software Engineering

RECOGNITION

MCST LLVM/C++ selection

Organizer correspondence: only 5.0/5.0 score and only solution passing all 104/104 tests among 49 submissions.

LangDev'26 — Build the Language, Then Make the Abstractions Disappear

Delivered the UniversalToolchain architecture talk at LangDev on 9 October 2026.

HSE Faculty of Computer Science — open-source projects competition

Selected as one of the winners with UniversalToolchain / Wist2.

Earlier distinctions

MEPhI Junior: Information Technologies section winner in 2025 and first-degree diploma in 2026; HSE Vysshaya Proba Industrial Programming diploma holder; Baltic Science and Engineering Competition: first-degree diploma and the "Perfection as Hope" prize.

Moscow / remote