

JASMINE TANG

CA, USA (Open to Remote and Relocation)

badumbatish.github.io – jjasmine@berkeley.edu – github.com/badumbatish

EDUCATION & SKILLS

Electrical Engineering & Computer Science - UC Berkeley with a Compiler/Systems focus.

Skills: Assembly, C, C++, LLVM, MLIR, Makefile, CMake, GDB, LLDB, Rust, Git, GitHub, Python, Bash, Unix, Linux, Bazel, Docker, Testing, Multithreading, Operating Systems, Compiler.

EXPERIENCE

Toolchain Compiler Engineer | Igalia Feb 2026 - Present

- Proprietary project...

LLVM Compiler Engineer | Igalia | LLVM, C++ Jun 2025 - Present

- Upstreamed contributions to the LLVM WebAssembly backend: implementing missing relaxed SIMD floating-point constructions, DAGCombiner peephole optimizations and TableGen instruction selection patterns over SelectionDAG, and memcmp lowering optimizations.
- Upstreamed contributions to ClangIR: porting `__builtin*` intrinsic codegen from legacy Clang to ClangIR's MLIR pipeline, defining new ops in the ClangIR MLIR dialect via TableGen, and establishing formatting CI/CD infrastructure.

GCCRS Compiler Intern | Google Summer Of Code | C++, Rust, GCC March 2024 - Aug 2024

- Initiated development of inline assembly for Rust in GCC. In C++, programmed the parser, set up the code infrastructure for TREE IR generation in the backend, along with AST, HIR lowering and typechecking, resulting in usable inline assembly code in Rust, with test cases in both ARM and x86.
- Opened 40+ issues and PRs of new features and bug fixes. Developed ARM-based gccrs Docker environment via docker compose, maintaining CI/CD pipeline and set up 32-bit Alpine CI/CD for new bugs and warnings.

Software Engineering Intern | Fermilab | Python, Matplotlib, NumPy Jan 2023 - Aug 2023

- Developed data processing pipeline for roughly 20000 HDF5 files, reducing processing time by 85%. Performed data analysis for the beamline optimization project with Matplotlib and NumPy.
- Authored an in-memory DSL interpreter in Python to extend an existing particle beam simulation language with programmatic control, enabling automated search over the simulation parameter space.

OPEN SOURCE CONTRIBUTIONS AND PROJECTS

Sammine Compiler | C++, Compiler, MLIR, LLVM, GPU Oct 2023 - Present

- Authoring a statically-typed compiled language featuring compile-time monomorphized generic types and type-classes, first class functions and ADTs with pattern matching, targeting native code via MLIR/LLVM with C FFI support.
- Implementing a recursive descent parser, bidirectional type inference engine and linear type system.
- TODO: Implementing a package manager with recursive transitive dependency resolution for generic typeclasses and functions.
- TODO: Extending the language with native GPU targeting via SPIR-V/NVPTX lowering, without requiring separate kernel syntax.

Technical writer June 2024 - Present

- Provides accessible and technical compiler blog posts, advocating for new grads and junior compiler engineers.
- Topics covered includes: [DEX-to-MLIR recompilation](#), [linear register allocation](#), and [LLVM codegen](#).

PintOS | C, Operating Systems, Class Project

- Collaborated with other students to extend a bare-bone OS with ELF executable loading, a small syscall interface for process control and file system methods
- added a strict priority scheduler with priority donation for deadlock avoidance and multi-threading capabilities, alongside a conditional variable based buffer cache for speeding up I/O and an extensible file system with sub-directories constructed with inodes.