JASMINE TANG

Berkeley, CA, USA (Open to Remote and Relocation)
badumbatish.github.io – jjasmine@berkeley.edu – github.com/badumbatish

EDUCATION

Electrical Engineering & Computer Science - UC Berkeley Class of 2025
Relevant Coursework: Compiler Optimization and Code Generation (CS265), Operating Systems (CS162), Compilers (CS164), Probability and Random Processes (CS126), Data Structures and Algorithms (CS61B).

EXPERIENCE

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

- Initiated development of inline assembly support 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 ARM and x86.
- Opened 40+ issues and pull requests of new features, bug fixes and code maintenance. Developed ARM-based gccrs Docker environment via docker compose, maintaining CI/CD pipeline and set up 32-bit Alpine CI/CD to monitor new bugs and warnings.

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

- Developed efficient data processing for roughly 19,000 HDF5 files with Python's multiprocessing, and Big O complexity, reducing processing time by 85%. Performed data analysis for the beamline optimization project with Python's Matplotlib and NumPy.
- Wrote a custom interpreter for an in-memory DSL in Python for the physics simulation program to aid in brute-forcing the optimization search space. Automated documentation for continuation of project.

OPEN SOURCE CONTRIBUTION

Shuriken Binary Recompiler | *C++, Compiler, MLIR* Jul 2024 - Current

- Developing IR conversions between Dalvik Executable Instructions (DEI) and MLIR (MjolnIR) using TableGen and Braun's SSA algorithm, with support for lowering MjolnIR to DEI via Smali.
- Enhanced code quality and CI/CD pipelines by enforcing strict C++ warnings and reducing MLIR CI/CD times by 50% through GitHub-hosted LLVM packages.

CXXGraph Library Developer | *C++, Graph Theory* Jan 2024 - Jun 2024

- Debugged and fixed critical bug in comparison operator of edge and node shared pointers in C++. Implemented Welsh-Powell node coloring with the library's API and unit tested via GoogleTest.
- Refined the general CI/CD GitHub Actions and set up CI/CD pipeline for building, testing and benchmarking on MacOS. Update documentation and refactored CMake scripts for more robust testing.

PROJECTS

ChocoPy Compiler | *Java, Python, Compiler* | Develop a statically typed subset of Python compiler with Java's JFlex and CUP's parser generator combined with semantic analysis and code generation via RISC-V architecture. Deploy a multi-core test harness in Python that handles file input and hanging tests for a more robust development cycle. Enable code-size and run-time metrics to benchmark.

Bril Compiler Optimizer | *Rust, Compiler* | Implemented different compiler optimization on an education Bril IR in Rust. Including but not excluding: LVN, GVN (SSA generation), Conditional Constant Propagation, Liveness Analysis, Loop Invariant Code Motion, Generic Dataflow Solver.

PintOS | *C, Operating Systems* | Collaborated with team members to extend a bare-bone OS with ELF executable loading, a small syscall interface for both process control and file system methods, 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.

SKILLS

Assembly, C, C++, LLVM, MLIR, Makefile, CMake, GDB, LLDB, Rust, RISC-V, Git, GitHub, Python (Multiprocessing, Argparse), Bash, Unix, Linux, Docker, Testing, Multithreading, Operating Systems, Compiler, Java.