Dhruv Chawla

Compiler Engineer

dhruv263.dc@gmail.com
+91 9910299843
https://dc03.github.io
Vellore, Tamil Nadu

A capable programmer learning compiler design and the LLVM compiler infrastructure. Experienced with interpreters, and familiar with systems and assembly programming. 4 years of C++ experience.

Languages

English	-	Fluent
Hindi	-	Intermediate

Skills

- C++
- C
- Python
- Rust
- Compilers
- Interpreters
- Operating Systems
- Linux
- Git

Education

B.Tech in Information Technology
VIT, Vellore 2020 - 2024
Current CGPA: 9.28

XII (Senior Secondary), CBSE
Navy Children School, Mumbai 2020
Percentage: 96.4%

X (Secondary), CBSE
Navy Children School, Mumbai 2018
Percentage: 93.6%

Experience

The LLVM Compiler Infrastructure
January 2023 - Present
Examples of Patches Contributed

- `[clang][alias|ifunc]: Add a diagnostic for mangled names`
- `[NFC][TLI] Replace std::lower_bound call in getLibFunc with DenseMap lookup`
- `[NFC][ValueTracking]: Remove redundant computeKnownBits call for LoadInst in isKnownNonZero`
- `[InstCombine] Fold icmps comparing uadd_sat with a constant`
- `[SetVector] Improve performance for small sizes`

Google Summer of Code Contributor
(The ENIGMA Team)
June 13, 2022 - September 12, 2022
Project name: Data Buffers / Serialization

- Worked on rewriting most of the frontend of the ENIGMA Development Language compiler, a scripting language based on GML
- Rewrote most of the Binary Buffer system which deals with storing and reading data from byte streams
- Made a serialization and deserialization system which uses template metaprogramming for static polymorphism
- `https://summerofcode.withgoogle.com/programs/2022/projects/BrXiUNA2`

Projects

nyx `https://github.com/dc03/nyx` *September 2020*

- A simple, interpreted language implemented in C++
- Features classes with constructors and destructors, lists, tuples
- Static type system
- Copy, reference and move semantics
- Bytecode virtual machine
- Code formatter, bytecode dumper, VM execution tracing

rispy `https://github.com/dc03/rispy` *February 2022*

- Interpreter for a lispy-inspired lisp
- Implemented in Rust
- Tree-walk interpreter
- Testing for lexical analyzer and parser

tictactoe-arduino `https://github.com/dc03/tictactoe-arduino` *February 2023*

- Tic-tac-toe implemented on an Arduino Uno
- Multiplexing of outputs (LEDs) and inputs (buttons) to reduce pin usage
- Compact layout of game state to reduce memory usage
- Part of a university course project

Certifications

Introduction to Haskell Programming (NPTEL) *Issued Sep 2022*

Percentage: 85% Credential ID NPTEL22CS69S2318078809012045

Compiler Design (NPTEL) *Issued Apr 2022*

Percentage: 90% Credential ID NPTEL22CS14S2446142802071248

Design and Analysis of Algorithms (NPTEL) *Issued Oct 2021*

Percentage: 85% Credential ID NPTEL21CS68S4332059403122958