

Curtis McEnroe

curtis@cmcenroe.me

github.com/programble

I am looking to learn more about systems programming and databases, and have a passion for creating well-crafted tools.

Technical skills

- Rust, C, x86 assembly, Ruby (Rails), TypeScript/JavaScript (Node.js, Knockout.js)
- PostgreSQL, Redis, MongoDB
- FreeBSD, Ubuntu, Arch Linux, macOS
- Protobuf, American Fuzzy Lop, Git, Heroku, Digital Ocean

Volunteer experience

I've written many, and contributed to some, open-source projects on GitHub. For example, a friend and I created a Clojure-like Lisp with Ruby interoperation on the Rubinius VM. I've also made web API clients, stylesheets, development tools, data structure libraries, hobby operating systems, and many Befunge interpreters. I publish short technical articles on my website and have spent time helping others in Rust IRC channels.

Work experience

Software Developer, Hopper (Montréal)

May 2017 – present

Learning new things...

Software Developer, AdGear (Montréal)

October 2015 – April 2017

I designed and implemented a replacement for the primary ETL providing time-sensitive state to a global real-time bidding system. The design solved performance,

correctness, and concurrency issues in the legacy system, and served as a strong example of Rust in the company. I vectorized a hash table implementation in a specialized key-value database service to take advantage of SIMD instruction sets and measured the performance increase with microbenchmarks. I set up fuzzing jobs for many Erlang NIFs written in C to try to find edge case crashes, which can bring down the BEAM.

Software Developer, Busbud (Montréal)

December 2013 – April 2015

I migrated geographic data from CouchDB to PostgreSQL, and replaced Elasticsearch with Postgres materialized views and trigram indexes. This centralized data and improved performance. I introduced promises and functional programming to the Node.js server, which facilitated writing organized, asynchronous code. I helped redesign the interfaces and data models for integrating partner APIs into Busbud's, which were previously too restricted and lossy.