Curtis McEnroe

cmcenroe.me curtis@causal.agency github.com/programble

I'm interested in learning more systems programming and really getting into the details of things. I have a passion for well-crafted tools and making computers suck less.

Here are some of the technologies I've enjoyed working with:

- Rust, C, x86 assembly, Ruby, TypeScript/JavaScript
- PostgreSQL, SQLite, Redis
- FreeBSD, macOS, Arch Linux
- Protobuf, American Fuzzy Lop, ncurses, zlib

I write a lot of code because I enjoy solving problems and learning how to do things myself. I've made web API clients, stylesheets, development tools, data structure libraries, hobby operating systems, and too many Befunge interpreters. I try to help out in Rust IRC channels and sometimes write for my technical blog. My posts have even been on Hacker News twice, so you could say I'm kind of a big deal.

Funemployed at home, August 2017 –

Away from work, I tried to focus on creativity. I created a public, collaborative ASCII art SSH service using memory-mapped files, local sockets and ncurses. I wrote an interactive graphics data explorer with framebuffer, X11 and Cocoa interfaces. Along with that, I wrote a PNG encoder and used bugs in my own code to create interesting images.

Software developer at Hopper, May 2017 – August 2017

I learned a little Scala, HBase and Elasticsearch. I couldn't stay long due to burnout from my previous job.

Software developer at AdGear, 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 at Busbud, 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.