Curtis McEnroe

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

I'm interested in learning the intricate details of systems and building well-crafted tools. I want to make software that works.

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

- C, make, mdoc, zlib, curses
- Rust, Protobuf, American Fuzzy Lop
- Ruby, TypeScript/JavaScript, PostgreSQL, SQLite
- FreeBSD, NetBSD, macOS, Arch Linux

Untether AI, April 2018 –

Using a modified QEMU, I created an emulated device for the PCIe interface of a neural network accelerator. The QEMU device communicates with a software simulation of the accelerator over a socket, using the same protocol that will be used in hardware. This allowed for rapid development of kernel- and user-space drivers as well as end-to-end testing of the proposed design.

Funemployed, August 2017 – April 2018

Away from work, I focused on creativity and communication. I created a collaborative ASCII art project as a public SSH service. I started a terminal IRC client and worked on easy but significant UX improvements. I designed a colour scheme, wrote a PNG optimizer, a bitmap font editor and a syntax highlighter.

Hopper, May 2017 – August 2017

I learned a little bit of Scala, HBase and Elasticsearch. I didn't stay long due to lasting negative mental health effects of my previous job.

AdGear, October 2015 – April 2017

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

concurrency issues in the legacy system. I vectorized a hash table implementation in a specialized key-value store to take advantage of SIMD. I set up fuzzing jobs for many Erlang NIFs written in C, for which crashes can bring down the BEAM. I also designed schemas and database constraints to ensure data integrity.

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.