Jimmy Hartzell: Systems Programmer

Phone: 646-334-9882, **Email**: jah259@cornell.edu, **Website**: https://www.thecodedmessage.com/

Skills

- **Programming languages**: Rust, C++, C, Haskell, Swift, Python, Objective-C, Bash, x86 assembly (32 and 64 bit)
- **Technologies**: Linux systems/low-latency network programming, Tokio, Reflex FRP, Yocto, AWS, Ledger Nano S, Redis, C++ template metaprogramming

Career Experience

- **Two Sigma:** August 2024-Present, *Rust Specialist*
  - **Technologies:** Rust
- **Amtrak:** July 2023-June 2024, *Senior Principal Software Engineer*
  - **Technologies:** C++, HP NonStop
  - Developed simulator for ITCS Positive Train Control protocol
  - Fixed bugs in HP NonStop dispatching codebase
- **Savant Systems:** May 2021-June 2023, *Senior Embedded Linux Software Developer*
  - **Technologies:** Rust (incl. Tokio), Yocto, Swift, Objective-C, Redis
  - Wrote usermode Rust driver for Atmel energy meter
  - Adapted quickly to a decades-old Objective-C codebase
  - Developed and implemented migration plans for core components of system architecture
  - Rewrote Swift microservices and frameworks into Rust
  - Added caching layers around accesses to legacy key-value store, and implemented bidirectional synchronization between it and Redis
- **Obsidian Systems:** March 2018-May 2021, *Software Development Consultant*
  - **Technologies**: Haskell, Reflex FRP, C, Ledger Nano S, Nix, C++
  - Full-stack Haskell application development
  - Worked with a variety of clients, with diverse corporate culture and organizational systems
  - Worked on Incremental View, a database research project for incremental queries on Postgres
  - Wrote apps in embedded C on Ledger Nano S (a platform w/ 4K of RAM)
  - Refactored overengineered client C++ codebases
  - Did trainings and talks on C++, Rust, blockchain, and Haskell
- **Tower Research:** June 2013-March 2018, *Senior Software Developer*
  - **Technologies**: C++ (C++11, C++14), C++ template metaprogramming, Linux systems programming, `clang-format`, `valgrind`, `gdb`, FIX protocol, Intel64 assembly
  - Risk platform, C++ development (2017-2018):
    - Wrote a new high-performance logging system
    - Led a small team to add new trade reconciliation systems to comply with EU regulations
  - Lead training instructor (2016-2018):
    - Developed and taught full-time C++, networking, systems, and low-latency programming programming curriculum for new hires in US and India
    - Trained and mentored other instructors
  - FX trading desk, C++ development (2013-2016):
    - Mentorship: First line of defense for team member questions
    - Continuously made latency improvements for market data handlers
    - Developed new aggregator project to aggregate internal liquidity
    - Owned support for FX "last look" feature
    - Wrote/maintained handlers for many financial protocols
- **Moat:** Feb 2011-March 2013, *Infrastructure Developer*
  - **Technologies**: Python, C++, Bash, AWS, S3
  - Led a 3-member team to develop server discovery and deployment scripts
  - Scalable bloom filter implementation in C++

Education

- **Cornell University**: *Bachelors in Computer Science*