Miguel Young de la Sota

(787) 548-0156 / mcyoung@mit.edu

I'm Miguel, an engineering leader with a focus on compilers, optimization, and education. I believe that systems programming should be accessible, welcoming, and useful, and I bring this philosophy to every aspect of my work. This means I am passionate about mentorship, leadership, and growing the technical skills of others, and building software tools that meet users where they are.

Career Experience

Senior Software Engineer, *Google* (2018–2023)

- **Protobuf**, *Tech Lead*. I joined the Protobuf team to lead the *Editions* project, an ambitious ecosystems-wide effort to bring incremental language evolution to Protobuf. I was responsible for the design and execution strategy, as well as the org-level OKRs.

  I was simultaneously the lead for *Protobuf Rust*, a first-party implementation of Protobuf for emerging Rust users at Google. I was responsible for the delivery of an implementation that met our evolvability and performance-critical needs, as well as the technical growth of senior- and staff-level contributors.

  I also contributed optimizations and code health cleanups to our open-source C++ project as spare time allowed.

  Key Accomplishments

  - Led design of Protobuf Editions, which was reviewed and approved by all ecosystem stakeholders, including Sanjay Ghemawat.
  - Ramped up a staff engineer to handle tactical execution of the Editions roadmap.
  - Defined norms and expectations for design contributions on Rust Protobuf, with a focus on psychological safety.
  - Held daily Rust seminars for new senior contributors to learn advanced Rust concepts and techniques.
  - Implemented and deployed a new JSON codec[1] for Protobuf C++, including extensive tests and bug-compatibility. This codec is used by all Google production services that process JSON input in C++.
  - Refactored the core codegen utilities of `protoc`, improving compiler developers' velocity.
  - Designed and implemented Protoscope[2] as a 20% time project. It is a simple, human-editable format for inspecting and modifying potentially invalid Protobuf wire format blobs. This tool is used extensively by optimization engineers to debug wire format codecs.
  - Participated in the on-duty rotation, keeping CI green and answering user questions.

- **OpenTitan**, *Software Engineer*. The OpenTitan project is an open-source root of trust SoC, consisting of both silicon designs and firmware to run on the device. I responsible for our low-level support libraries and contributed significantly to our build system and cryptography libraries.

  I was also responsible for the *Manticore* project, an implementation of Microsoft's Cerberus attestation protocol in Rust.

  Key Accomplishments

  - Migrated the entire project from ad-hoc Make scripts to Meson; two years later, contributed a significant portion of the migration from Meson to Bazel.
  - Established a long-term relationship with Microsoft senior staff engineers working on Cerberus, including a mutually agreed-upon RFC process for Cerberus.
  - Unified our linker scripts and assembly files under one project style.
  - Implemented and maintained core libraries, such as C runtime support and optimized math utilities.
  - Developed and implemented methodology for unit-testing driver code off-device.

[1] `https://github.com/protocolbuffers/protobuf/tree/main/src/google/protobuf/json`
[2] `https://github.com/protocolbuffers/protoscope`

  - Designed fault-injection and power analysis attack mitigation strategies for use in high-assurance privileged code; multiple patents were filed based on this work.
  - Engaged with customer teams on cryptographic primitive requirements, balancing legacy use-cases with modern security practice.
  - Provided technical mentorship for multiple new hires, including our cryptographer.

- **20% Work**. In addition to my assigned work, I dedicated significant time to community efforts, primarily focused around the theme of education.

  Key Accomplishments

  - Joined the admin group of *C++ Readability*, a mentorship program that teaches C++ best practices to engineers via randomized code review. I also performed approximately 500 such mentorship reviews.
  - Taught multiple sessions of *Comprehensive Rust*[3], a multi-day Rust 101 course; student feedback was overwhelmingly positive. I also organized initial development of a revised edition of the course materials.
  - Wrote entries for the *C++ Tip of the Week* best practices publication.
  - Kicked off the Rust style guide working group, contributing strategic vision for Rust best pratices at Google.
  - Designed *moveit*[4] a novel move semantics library for Rust, bringing custom move operations to the language. This design was adopted by Google's Rust/C++ interop tooling for bridging C++ move constructors to Rust.

Personal Projects

I maintain a personal blog at *mcyoung.xyz*, where I post long-form explainers on advanced systems-programming topics, written for an intermediate-level audience, with a focus on making the material accessible.

Most of my larger presonal projects are some kind of toolchain or programming language. *Alkyne*[5] is a pure scripting language inspires by Starlark and Jsonnet, intended as a Starlark alternative; *snasm*[6] is a full Super Nintendo (MOS 65816) toolchain, including an assembler, a disassembler, and a linker; *jas*[7] is a JVM bytecode assembler.

Other smaller projects include *0x*[8], a binary dumper with colorization capabilities and an RPN calculator, and *voltorb*[9], a Picross-like game with graphics and animations that runs inside of a terminal.

[3] `https://github.com/google/comprehensive-rust`
[4] `https://github.com/mcy/moveit`
[5] `https://github.com/mcy/alkyne`
[6] `https://github.com/mcy/snasm`
[7] `https://github.com/mcy/jas`
[8] `https://github.com/mcy/0x`
[9] `https://github.com/mcy/voltorb`