Daniel Prilik

Software Engineer

Contact

- ✉ daniel@prilik.com
- 🌐 prilik.com
- 🐙 daniel5151

Skills

DEVELOPMENT

Rust
C++
C
JavaScript (ES6)
Typescript
HTML5 + CSS
React & React Native

EXPERIENCED WITH

Linux Kernel Patching
Containerization
Virtualization
Hardware Architecture
KVM
CMake
Travis & Appveyor CI
Visual Studio
Unreal Engine 4
SDL2
Wwise

Education

University of Waterloo
BSE Software Engineering
2015 - 2020

Interests

Mechanical Keyboards
Massdrop ALT 65% with MX Blues. Contributor to QMK, the open-source keyboard firmware

Emulator Development
I keep a close eye on /r/emulation, and enjoy writing my own!

Exploring Music
Alternative Rock, Psych Rock, and [Future] Funk in particular

PC Gaming & VR
Mainly PUBG, Rocket League, and Beat Saber. Excited for Cyberpunk 2077!

Employment

Microsoft - Azure IoT
Software Engineering Intern - *Rust, Go*
Redmond, WA
Fall 2019
- Implementing a new lightweight Container Runtime in Rust
- Understanding and Implementing the OCI Image and Distribution specifications

Google - ChromeOS
Software Engineering Intern - *Rust, C*
Mountain View, CA
Winter 2019
- Contributed to *crosvm,* an open-source KVM-based hypervisor written entirely in Rust
- Implemented *virtio-fs*, a paravirtualized shared filesystem device that leverages FUSE and shared-memory to improve filesystem performance by 2x-8x over virtio-9p
- Worked with low-level Linux kernel APIs such as KVM, shared-memory mmaps, and sending file-descriptors over Unix Domain Sockets
- Gained experience backporting experimental kernel patches, and kernel debugging

Apple Inc - Platform Architecture
Software Engineering Intern - *C, C++, Objective-C, Python*
Cupertino, CA
Summer 2018
- Replaced hundreds of large, autogenerated modules with a single runtime-configurable module
- Deleted >100,000 LOC, reducing compile-times by 30%, and improving maintainability
- Coordinated across teams to develop a long-term support plan for the module

Microsoft - The Coalition
Audio Engineering Intern on *Gears 5 - C++, Unreal Engine 4, Wwise*
Vancouver, BC
Winter 2017
- Orchestrated a large-scale audio data migration, reducing asset duplication by a factor of 11x
- Exposed Wwise functionality in the UE4 editor by creating custom UI components, substantially improving audio artist workflow

LCBO|next
Full Stack Web Developer - *React, React Native, node.js, Mapbox, Sass*
Kitchener, ON
Summer 2017
- Spearheaded the design and implementation of a React SPA
- Created graphs, charts, and visualizations using both popular libraries and custom components

Polar.me
Front End Web Developer - *HTML5, Sass, JS, node.js*
Toronto, ON
Summer 2016
- Designed and implemented Ads for clients such as Engadget, Wired, and AOL
- Engineered and deployed a build system to streamline and automate Ad development

Projects

ANESE + wideNES - prilik.com/ANESE
- Created a cross-platform Nintendo Entertainment System (NES) emulator in C++11 + SDL2
- Designed wideNES, a novel technique to automatically map-out worlds and levels in NES games
- Documented the inner-workings of wideNES in a *detailed blog post*, inspiring others to port the technique to other systems (e.g: *WideGB* - Gameboy)

nfinite.space - https://nfinite.space
- Creating a cloud file-storage service to store files without requiring centralized data centers
- Implementing a protocol to securely and fairly rent disk space from connected users to store data
- Developed a proof-of-concept at Hack the North 2016, winning the AWS Sponsor Prize
- Actively improving and expanding the project into an Engineering Capstone Project

mips241 - prilik.com/mips241
- Implemented an emulator for the MIPS system used by CS 241 at Waterloo
- Features an interactive debugger with step-by-step execution, breakpoints, and instruction disassembly
- Experimented with C++ to Javascript compilation via emscripten
- Revisited the project in 2019, adding new instructions, and a time-traveling debugger (see *uwmips*)

Awards

Winner of the Amazon Web Services (AWS) Sponsor Prize
Hack the North 2016, QHacks 2017
For work on *nfinite.space* and *fastify* [github.com/Melinysh/fastify] respectively
2016 and 2017