

# YU WEI WU

✉ wusyong9104@gmail.com · 🌐 wusyong · 📄 wusyong.github.io · in wusyong

## EDUCATION

**National Chiao Tung University**, Taiwan

08/2012 – 07/2016

Bachelor: Computer Science

## WORK EXPERIENCE

**BiiLabs Inc.**, Taiwan

11/2018 – 08/2019

(Cloud Services) Software Engineer

- Improved computation bottleneck significantly with various technics like SIMD and embedded FPGA.
- Developed microservices to accelerate accessing distributed system.
- Maintained both cloud instances and local servers. Learned a lot about Linux programming and its codebase.

**IOTA Foundation**, German Berlin

08/2019 – 12/2020

(Distributed Storage Systems) Senior Software Engineer - Protocol Team

- Developed the protocol of main system service. Built several essential components around the distributed system like async actor runtime, message queue, and database ORM.
- Designed the framework that can be the one source of truth and provides better interoperability through WASM and FFI.
- Mentored and guided other developers in the team become more familiar with Rust programing.
- Learned a lot about distributed system, working remotely, and databases.

## PERSONAL PROJECTS

**Tauri**

<https://github.com/tauri-apps/tauri>

A framework that builds smaller, faster, and more secure desktop applications with a web frontend.

- Compatible with any front-end framework which means developers don't have to change their stack.
- Supports cross-platform compilation to bundle binaries for major desktop platforms.
- Tauri application can have extreme small binary size and memory usage.

**Wry**

<https://github.com/tauri-apps/wry>

Cross-platfrom WebView rendering library in Rust that supports all major desktop platforms like Windows, macOS, and Linux.

**Smol-potat**

<https://github.com/wusyong/smol-potat>

Proc macro for smol async runtime. It provides an ergonomic way to fine tune the runtime system.

**TRPL (Traditional Chinese Version)**

<https://github.com/rust-tw/book-tw>

Traditional Chinese version of The Rust Programming Language, the official book on Rust.

**RCC4r**

<https://github.com/wusyong/RCC4r>

A Rust impl. of toy C Compiler with only 4 Raw functions.

**Honga**

<https://github.com/wusyong/honga>

A 64-bit RISC-V emulator written in Rust that can run xv6, a simple Unix-like OS.

## SKILLS

- **Program Language: multilingual** (not limited to any specific language), especially experienced in Rust C, comfortable with C++ Haskell C# Objective-C Python Swift Racket (in random order).
- **Database:** familiar with various databases such as Rocksdb, Cassandra, mariaDB, SQLite, PostgreSQL, etc. Understand concepts between different SQL/NoSQL databases. Can build protocol implementations without any help of ORM.
- **Rust: 4 years** of experience, have projects collected in Awesome Rust, familiar with Cargo and Bazel, understand procedural macros, Rust asynchronous runtimes, and unsafe usage, contributed to Rust compiler and its tools from time to time.
- **System Programming: 3 years** of experience, familiar with Linux programming interface and also other Unix-like OSes. Understand how to write an operating system from scratch.
- **Ebedded Systems: 1 year** of experience, familiar with several SoCs and Development boards. Understand basic concepts of SPI, I2C, UART, etc.
- **Mobile Development: 1 year** of experience, iOS (Swift), Fuchsia (Flutter)

- **Development Tool:** can adapt to any editors / operating systems, usually use JetBrains IDEs and Vim under Arch Linux, have experience with team collaboration tools like GitLab, Jira, GitHub, ClickUp, Hackmd, Slack, Zenhub.

## MISCELLANEOUS

---

- Crates.io profile: <https://crates.io/users/wusyong>, for publishing interesting Rust libraries
- Languages: English - fluent, Chinese - native, Taiwanese - native
- Member of Tauri, Webview, rust-tw and more, contributed to rust, async-std, veloren, tauri, riscv-opcodes, grcov, windows-rs, gtk-rs, winit, smol, amethyst and other projects