林子篆/Danny Lin

📱 +886(0)975822245 • ✉ dannypsnl@gmail.com • 🌐 dannypsnl.me • dannypsnl

Programming Language Theory ▪ System Programming

Education

Kaohsiung Medical University **Kaohsiung, Taiwan**
Healthcare Administration and Medical Informatics, Bachelor *2015/09–2017/07*
Courses: Operating Systems, Data Structures, Analysis Of Algorithms, Networking, Databases.
Terminated due to family economic issues.

Skills

Languages:	Racket, Zig, Haskell, Agda, Arend, Elixir, Rust
Frameworks:	Phoenix, Svelte, React
Tools:	Kubernetes, PostgreSQL, TimescaleDB
Platforms:	Linux, Web, Arduino, Raspberry, GCP, Azure

Experience

Second State **Tainan, Taiwan(Remote)**
Compiler Developer (Full-time) *2022/08–Present*
Product: WasmEdge is a lightweight, high-performance, and extensible WebAssembly runtime for cloud native, edge, and decentralized applications. It powers serverless apps, embedded functions, microservices, smart contracts, and IoT devices.

- Implementing proposal component-model

Aionic Labs **Tainan, Taiwan(Remote)**
Backend Developer (Full-time) *2021/06–2022/07*
Product: Weever is a game site that provides social media, and online meetings. Users can play with friends, watch events, post, report bugs, etc.
Description: Use Elixir & Phoenix & TimescaleDB stack to develop Weever, based on libcluster on Kubernetes to help Elixir GenServers communicate smoothly. Below are my major contributions.

- Most chat room mechanism.
- Whole voting system.
- Let users can log in to the store via weever, using BigCommerce APIs.
- Maintain DB tables and related APIs.

Glasnostic **Taipei, Taiwan**
Networking Specialist (Full-time) *2018/08–2020/02*
Product: Software router and gatekeeper. A network filter/analyzer only needs four norms, request, bandwidth, concurrency, and latency, but can manage complex infrastructure.
Description: Deeply work with networking fundamentals(including but not limited to libpcap, eBPF, and DPDK) to maintain the product. The following pictures will say more about our router for Kubernetes.
Basic idea: Container network has several concept, the basic is considering layers: machine, bridge, container. When a container sends TCP packages to another container, the full journal will go through virtual pair NIC, container bridge NIC, virtual pair NIC of another container, then backward. To fit different situations, our router can stand at several places for these. If router can control the bridge, eBPF-like model can handle this, else we will create a sidecar bridge to use same logic, this architecture is one of my major contributions. It works well for Flannel, AKS, EKS, and non-Kubernetes environments. And if Kubernetes cross machines, it will depend the platform to ensure how to work.
Details:

- How to handle service IP? Indeed, this one never gets a perfect solution when I'm still here, the major idea is Pod names will similar to the service name. The mapping is based on this assumption. And of course, the router with service IP rewriting is more complicated than described above, tracking the real service Pod is important for the following connection.
- How know whether the TCP connection should be kept? The router maintains a TCP state internally for every connection pair. This problem is originally found by me and I contribute a lot to this.
- How do you know the CIDR of the node? Usually, this is defined in **.spec.podCIDR**, some exceptions like EKS don't have it. In this case, the router will pick a subnet 24 as a guess or get from our configuration if we assign it one.

AndroVideo **Taipei, Taiwan**
Backend & Embedded System Developer (Full-time) *2018/02–2018/08*

Developing cloud web service with container-solution, and maintaining the device(camera) HMI system by communicating with the Android system.

- Detect special motions like stealing or attacking and send signals.
- Detect the human face for the access control system.
- The backend system will notify which camera sent the signal and help human goes to handle it.

Mapacode — **Tainan, Taiwan**
Fullstack & Embedded System Developer (Full-time) — *2017/09–2018/02*
Developing Human Machine Interface to interact with CNC.

- User interface for adjusting parameters.
- Compile high-level parameters and paths to Forth language.
- Send compiled results to CNC machine.

Projects

typed-nanopass — **Racket**
(Work in progress) rebuild nanopass with typed supports. — *2022–Present*

sauron - DrRacket plugin — **Racket**
Make DrRacket become a better IDE. — *2020–Present*

- Refactoring
- File explorer
- Auto formatting
- Jump to definition

racket-llvm - LLVM binding — **Racket**
racket llvm C-API bindings. — *2022–Present*

k - A theorem prover (dependent type) — **Racket**
(Work in progress) Research-oriented, open-source, theorem prover based on dependent type. — *2021–Present*

eikyo - Programming language — **Racket**
(Work in progress) Research-oriented, open-source, polymorphism language with the mark system. — *2022–Present*

racket-langserver (Contribution) — **Racket**
A Language Server Protocol implementation for Racket. — *2021–2022*

- implement cross-file jump to definition
- auto formatting: remove trailing whitespace

formatted-string - Reader macro — **Racket**
The language extends racket string to get formatted string syntax — *2022–Present*

avr-arduino-zig - Arduino using Zig! — **Zig**
Fork and make it easier to install and use. — *2022–Present*

rocket - Light-weight web framework — **Go**
Use metaprogramming(reflection) to help users bind data from HTTP requests easier. — *2017–2021*

scheme-to-arm64 - A Scheme to Arm64 compiler — **Racket, Zig**
Use nanopass to build small step compilers, generate arm64 directly, and write runtime in Zig — *2021–2022*

little-scheme - A little scheme build for playing SICP — **Haskell**
Use megaparsec to parse, build error reporter and interpreter via Haskell — *2019–2020*

Talks

COSCUP - Closure conversion — **Taipei, Taiwan**
Why conversion is important and makes a brief implementation to show how to make one. — *2022/07*

Clojure Taiwan - Clojure isn't Lisp enough — **Taipei, Taiwan**
Why Clojure didn't really use s-expression is a problem, and what are macros expected to do. — *2021/04*

Racket Fest - macro as type — **Online**
How to encode Hindley-Milner type in macros, and what should go further. — *2021/03*