Oliver Iliffe

Email | +44 7944 727309 | GitHub | Stack Overflow | LinkedIn | Website

Education

Imperial College London – Advanced Computing MSc — Starting Sep 2024

King's College London – Computer Science BSc — Sep 2021 - May 2024

First-class average (83%, 4.0 GPA).

Modules include: Database Systems, Operating Systems & Concurrency, Discrete Maths, Linear Algebra, Machine Learning, Robotics, Compilers & Formal Languages, Security Engineering, Optimization Methods, and Financial Computing.

Experience & Projects

Teaching Assistant for Operating Systems & Concurrency — **(January 2024 – Present)**

TA for the OSC module at King's College London.

- I teach Operating Systems theory (primarily **x86-64 Linux**) and concurrency in Java
- I create additional content for the students, both module-specific (e.g. revision sessions) and related content.
- For example, an in-depth look at **Linux's CFS** including a visual demonstration of the impact of nice values.
- Or rewriting the assignments to let students do them in **C** with **`pmutex`** et al.
- As well as additional sessions about **ELF**, **x64 page tables** and **interrupt handlers**.

Compiler for a Functional Language — view repository

Implemented a small purely functional language in **Rust**. The entire list of features is documented on the GitHub page.

- CLI build tool for compiling and running programs.
- Hand-written lexer for efficiency.
- Clear and precise error messages with syntax highlighting and exact error location.
- Statically typed – lowers to **LLVM-IR**

Train Departure Board for ESP32-S3 Micro-controller — view repository

Display live departures on a little LCD for TfL services...

- Completely `#[no_std]`, I limit myself to embedded-hal as the highest level of abstraciton
- Submodule `lcdterm` abstracts LCDs to a common interface,
- Allowing lazy updates, scrollable regions and a ported-to-rust **driver** for the ST7789 family of displays.
- Currently working on a from-scratch (partial) **TCP** implementation.

libhopeful – Tracing heap allocations in Rust — view repository

Build inspectable graphs of the allocations active Unix processes. 'Tracing' is used here in the 'tracing GC' sense.

- Consume DWARF debug info, such that we can attempt to link any `T` to a representation.
- **Lock-free** data-structure using `std::atomic` (this is practically identical to **C++ atomics**) to look up metadata for interior pointers.
- Optimized for **cache locality**.
- Currently profiling to produce reports on alternative implementations. E.g. is some lock contention actually desirable.

og2 – 2D Game Engine — view repository

- Allows for sprite loading, animations, basic shaders and particle effects.
- Project uses wgpu-rs, which is similar to Vulkan (but generally safe).
- Completed external linear algebra courses so that I actually understand how my code works.
- Built upon this knowledge to produce a platformer with bevy, with my own player controller.

Small-string Optimisation for Rust — view repository

A C++-style small-string optimization in Rust for little-endian 64-bit machines.

- SSO is not hard, but I use the project to write "item-scoped unsafe code". View the blog post here.
- Has a crates.io page.

Skills

tokio; axum; wgpu; wgsl; bevy; Linux; Windows; C; C++; C#; .NET; Unity; Lua; Python; Django; TypeScript; JavaScript; axios; React; HTML; CSS; Java; Scala; Redux; Prisma; REST API; OOP; Functional Programming; Web-Dev; TCP/IP; Serialization; gdb;