Oliver Iliffe

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

Education

Imperial College London – Advanced Computing MSc **(Sep 2024 - Sep 2025)**

King's College London – Computer Science BSc **(Sep 2021 - Jun 2024)**

Grade: **First-Class Honours**

Experience

Teaching Assistant for Operating Systems & Concurrency **(Sep 2023 – Present)**

TA for the "Operating Systems & Concurrency" module at King's College London.

- 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**.
- Some feedback: *"very engaging, and you could tell that he has a thorough and in-depth knowledge of the content."*

Undergraduate Research Fellow **(Jun 2024 – Sep 2024)**

Participate in an undergraduate research fellowship at King's College London

- Research novel matching algorithms for regular expressions
- i.e. algorithms that do not rely on the construction of a NFA
- Leverage **Haskell** and **Scala** to produce elegant algorithms that could later be formalised in **Isabelle**

Projects

Hash Table Research view repository

Researched a variety of hashtable implementations, in an effort to produce a very fast one (**C++**).

- Create a testing and benchmark suite for hashtables for any language that supports `extern "C"` declarations.
- Implement many different hashtable designs (robin-hood hashing, quadratic-probing, chaining etc. **SIMD-lookup**).
- Optimize a combination of chaining and probing to produce a very fast map!
- Write a **proof** for the complexity of probing hashtable lookups (as well as other articles).

libhopeful – Tracing Heap Allocations 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::sync::atomic` (this is practically identical to **C++ atomics**) to look up meta-data for interior pointers.

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.

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;