Alex O'Brien

3541@3541.website | 3541 | EN, ES

0421 914 822 | a03541 | MELBOURNE, VICTORIA

Systems programmer and Computer Science student fascinated by robust and well-engineered software.

Education

2020–2022

FINAL YEAR

Bachelor of Science/Master of Data Science (Graduate Degree Package), The University of Melbourne

Majoring in Computing and Software Systems. Maintaining an average grade of *First-Class Honours*. Relevant subjects include *Algorithms and Data Structures*, *Computer Systems*, *Probability*, and *Statistics*.

Skills

Technical

Fundamentals Deep understanding of hardware and software architecture — all the way from CPU architecture to operating systems implementation.

Programming Experience in a variety of languages and paradigms, including object-oriented and functional practices. Most fluent with Rust, C, and C++, with working knowledge of Java, Haskell, Go, JavaScript, x86_64 assembly, and others. Very capable of learning on the fly and picking up new technologies quickly.

Security Understanding of security best-practices and fundamentals with common services (e.g., Apache and nginx), with experience in implementation of real systems.

Operations Experience both in a personal and professional context with the administration and maintenance of common enterprise technologies, including Active Directory, Open Directory, Profile Manager, and Unix-descended systems (Linux and FreeBSD in particular).

Communication

Written Strong written communication skills both in an informative and argumentative style.

Spoken Experience and success in competitive speaking and debate.

Instruction Served as a teaching assistant/mentor in mathematics for a year 10 mathematics class in 2019. Coached debate team and assisted with year 10 programming classes throughout 2021.

Experience

References available upon request.

2021-2022

CONCURRENCY
NETWORKING
PERFORMANCE

Software Engineer Intern, IMC Trading

Worked on microsecond-sensitive concurrent infrastructure, delivering a project which provides monitoring of a high-performance IPC system with near-zero overhead.

2021

INSTRUCTION
COMMUNICATION

Debate Instructor, John Monash Science School

Helped coach the JMSS debate team throughout the 2021 school year. Provided instruction in argumentation technique, personal feedback, and supervision at competitions and events.

2020

Software Developer, Embedthis Software

Embedded
Web
Security

Delivered version 2.2 of the GoAhead embedded web server:

- ▷ Modernized and cleaned up the codebase, backporting and implementing significant security fixes, including seven CVEs.
- ▷ Maintained API compatibility and stability for existing applications.

2016–2017

IT Technician, University Prep

IT
Technical Support
Systems Administration

Worked two summers full-time in general IT and support, involving a variety of tasks in multiple areas of expertise, including:

- ▷ Maintained images across multiple platforms and OS versions. Also delivered brand-new images for the organization's upgrade to Windows 10.
- ▷ Delivered a new library checkout system using Raspberry Pis as thin clients.
- ▷ Managed device setup for new staff and equipment – testing, imaging, and communication with end-users.

Personal Projects

Names are hyperlinked to the relevant repository.

2020-present

Short Circuit, Web Server

Servers
Networking
HTTP

A high-performance web server for Linux using the new io_uring asynchronous I/O interface.

- ▷ Unix network programming.
- ▷ HTTP parsing and implementation.
- ▷ Performance profiling and optimization.

2020-present

liba3, Utility Library

C
Library design
Data structures

A lightweight utility library in C implementing useful features and data structures absent from the standard library.

- ▷ More ergonomic strings.
- ▷ Type-generic hash table, cache, list, and priority queue.
- ▷ Growable buffer.
- ▷ Object pool allocator.

2019-2020

3cc, C Compiler

Compilers
Parsing

A C compiler in Rust. Features a lexer and hand-written recursive descent parser. Currently implements most unary and binary operators and has early support for local variables. On the backburner, pending a better re-implementation.

2017–present

Syzygy, Kernel

Operating Systems
Computer Architecture

A kernel implemented in Rust, currently featuring physical and virtual memory management, an initramfs, and interrupt handling. Presently working on multitasking.

- ▷ Kernel and bare-metal programming.
- ▷ `no-std` Rust programming.
- ▷ x86_64 architecture and assembly.