Luke Rule

Phone : +44 7484 104303 - **Email** : luke.rule@outlook.com - linkedin.com/in/luke-rule - github.com/luke-rule

Personal Profile

Driven Computer Science student passionate about crafting creative solutions to compelling computational problems, reflected by the independent development of complex projects, consistent achievement of results within the top 10 of the near 400 large CS cohort at The University of Manchester, and achieving high impact at internships within industry.

Education

The University of Manchester - *September 2022 - June 2026*

- **BSc Computer Science with Industrial Experience** : **Expected First Class - 86%, 87%** *(1st & 2nd Year Results)*
  - **Key Modules** :
    - **System Architecture** : Studied the various techniques employed to increase the performance, reliability and flexibility of systems in more depth, including pipelining, caching and memory management.
    - **Algorithms and Data Structures** : Explored the design and reasoning behind a range of key algorithms and data structures, and how this is further applicable. Compared solutions with theoretical and practical complexity analysis.
    - **Programming Languages & Paradigms** : Investigated practically the various ways problems can be approached, focusing on C, C++, Rust & Haskell. Studied wider topics such as compilation, concurrency and memory allocation.
  - **Awards** :
    - **Netcraft Prize, *Netcraft*** : Gained one of the 10 highest overall marks, in both first and second year, out of 372 fellow CS students.
    - **Kate Kneebone Acorn Award, *Arm*** : Demonstrated 'commitment, determination, enthusiasm, personal application, promise and academic merit', as the sole awardee out of 372 fellow CS students.

Horsforth High School - *September 2015 - June 2022*

- **A levels** : Maths **(A*)**, Further Maths **(A*)**, Physics **(A*)**, Computer Science **(A*)**.

Experience

Meta : *Software Engineering Intern (Reality Labs)* : **C#, Hack, JavaScript, React, GraphQL** - *June - September 2025*

- Directly responsible for designing and adding a new feature to VR/AR and mobile platforms enabling users to teleport to each other in-world, improving social interactivity and ease of use for Horizon World's 500,000+ users.
- Over-delivered on project brief by extending scope to include transportation for groups of users.
- Worked with cross-functional teams to debug and overcome limitations of the large, rapidly changing engine codebase.
- Completed to a strict development timeline, overcoming late-stage requirement changes by leading resolution discussions between relevant stakeholders.

Sophos : *Software Engineering Intern (Windows System Protection)* : **C++, Python, Lua, Git** - *July 2024 - June 2025*

- Upgraded endpoint functionality to vastly increase information and specificity returned when scanning defective files.
- Furthered the protection of user privacy by implementing stricter IPC security for system scans.
- Delivered and presented multiple significant projects, including new internal debugging and testing tools, to stakeholders, demonstrating the impact made to both the entire Windows endpoint team and 600,000+ customers over millions of endpoints.
- Embraced new responsibility as the Product Acceptance Testing representative for the Windows System Protection team, investigating and discussing all of the wide-ranging issues that occurred.

Projects

Project Demo Videos *(YouTube Playlist)* : Link

Asynchronous seL4 File Server *(Final Year Project)* : **C** - *September 2025 - Present*

- Created a POSIX style, dynamically allocating file server as a user-space module for the seL4 microkernel.
- Utilised an io_uring methodology to implement asynchronous file queues for multiple clients, increasing overall bandwidth to combat the performance drawbacks of microkernels compared with monolithic kernels.

C Compiler *(Personal Project)* : **C** - *July - August 2023*

- Inspired by theory taught in first year, created a compiler for a subset of C to dive practically into how compilers work.
- Utilised C's low-level language capabilities to allow a complete construction from scratch, leading to full flexibility in creation and a better understanding of a compiler's inner workings.
- Introduced support for functions, loops, conditionals, scope, operations and C standard library calls, implemented via a three-phase methodology of a lexer, parser and code generator, realised with a token stream and AST.