

# Guhan Iyer

[linkedin.com/in/guhansiyer](https://linkedin.com/in/guhansiyer) | [github.com/guhansiyer](https://github.com/guhansiyer)

Email: 4guhaniyer@gmail.com

Phone: (226) 505-7658

## EDUCATION

### University of Waterloo

Bachelor of Applied Science in Computer Engineering

Expected Graduation: April 2027

Waterloo, Ontario

- Relevant Coursework: Systems Programming & Concurrency, Algorithms & Data Structures, Digital Computers

## EXPERIENCE

### Nokia

Firmware Engineering Intern

Sept. 2025 – Dec. 2025

Ottawa, Ontario

- Implemented a device initialization framework in **C++** for a new optical transceiver ASIC, optimizing boot sequence control across hardware modules.
- Re-engineered critical **C++** message passing service by introducing 128-bit atomic operations, eliminating data corruption in **100% of packets sent**.
- Resolved **10+** major defects in ASIC SDK by analyzing firmware trace logs, significantly reducing crash frequency.

### Ford Motor Company

Software Development Intern

Jan. 2025 – Apr. 2025

Waterloo, Ontario

- Created a modular library in **Python** to simplify and scale testing for an in-vehicle security daemon, reducing reliance on external tooling.
- Updated **30+** legacy tests to utilize the new library, standardizing test structure and reducing maintenance effort.
- Refactored a deprecated generator utility in **Python** with the **Slash** framework, integrating directly with existing infrastructure.

### NCR Voyix

Software Engineering Intern

May 2024 – Aug. 2024

Waterloo, Ontario

- Integrated an internal **Python** query utility into a new patch management system, enabling automatic device data retrieval.
- Instrumented a service to validate per-device network compliance data for use **organization-wide**.
- Developed a cross-platform patch verification tool serving **10,000+** devices across **10+** platforms.

## TECHNICAL PROJECTS

### ARM Microkernel | C, Raspberry Pi, ARM Assembly

November 2025

- Designed **core kernel functionality** including interrupt vectors, inter-process communication, and memory-mapped device I/O.
- Implemented a cooperative multitasking scheduler with low-overhead context switching for lightweight user-process management.
- Built a custom **ARM** bootloader to initialize hardware peripherals for a stable system boot sequence.

### osh: The Open Shell | C, Linux

April 2025

- Created a lightweight **Linux** shell in C supporting built-ins, external programs, and custom commands.
- Utilized Linux syscalls to implement piping, redirection (<, >), and custom built-ins.
- Improved responsiveness by adding a persistent command history with **readline**.

### wintop | C, MSVC, Windows API

March 2025

- Developed a CLI-based **thread and process inspector** in C for **Windows** platforms, exposing detailed per-thread scheduling and runtime metadata.
- Leveraged **Win32 APIs** to enumerate active threads and retrieve metadata with minimal runtime overhead.
- Designed a terminal interface to provide **real-time** diagnostic information, emulating **top** and **ps**.

## SKILLS

**Languages:** C, C++, Python, Java, Bash, Assembly (ARM, RISC-V)

**Libraries & Tools:** Valgrind, GDB, CMake, Make, Android Tools (adb, Fastboot), Docker

**Technologies & Protocols:** Linux, QNX, FreeRTOS, CAN, TCP/IP, UART, I2C, gRPC, protobuf