


Guhan Iyer

linkedin.com/in/guhansiyer  | github.com/guhansiyer 

Email: 4guhaniyer@gmail.com

Phone: +1 (226) 505-7658

EDUCATION

University of Waterloo

Expected Graduation: **April 2028**

Bachelor of Applied Science in Computer Engineering

Waterloo, Ontario

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

EXPERIENCE

Nokia

Sept. 2025 – Dec. 2025

Firmware Engineering Intern

Ottawa, Ontario

- Implemented a device initialization framework in **C++** for a new optical transceiver ASIC, optimizing boot sequence control across hardware modules.
- Eliminated packet corruption in a critical **C++** message-passing service by fixing race conditions with 128-bit atomic operations, ensuring data integrity.
- Resolved **10+** major defects in the ASIC SDK by analyzing firmware trace logs, significantly reducing crash frequency.

Ford Motor Company

Jan. 2025 – Apr. 2025

Software Development Intern

Waterloo, Ontario

- Created a modular library in **Python** to simplify and scale testing for an in-vehicle security daemon, reducing reliance on external tooling.
- Refactored a deprecated generator utility to integrate with new test infrastructure via **Python** and **Slash**, enhancing and enabling **multiple testing workflows**.
- Reworked **30+** legacy tests to utilize the new library, standardizing test structure for **all downstream testers**.

NCR Voyix

May 2024 – Aug. 2024

Software Engineering Intern

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.

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 information.
- 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