Guhan Iyer

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

Email: guhan.iyer@uwaterloo.ca
Phone: (226) 505-7658

Education

University of Waterloo

Candidate for Bachelor of Applied Science in Computer Engineering

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

Experience

Nokia Sept. 2025 – Present

DSP Firmware Engineering Intern *Ottawa, Ontario*

- Implementing device configuration drivers in **C++** for a new optical transceiver module, enabling fine-grained control of hardware modules in the system datapath.
- Re-engineered **C++** message passing service to utilize 128-bit atomic operations, eliminating the likelihood of packet corruption.
- Resolved **10+** developer tooling defects in the digital signal processing SDK by analyzing **Python** test logs.

Ford Motor Company Jan. 2025 – Apr. 2025

Software Development Intern *Waterloo, Ontario*

- Created a modular library in **Python** to simplify and scale testing for a universal security daemon, reducing reliance on external tooling.
- Updated **30+** legacy tests to utilize the new library, standardizing test structure for future development.
- Re-engineered a deprecated generator utility in **Python**, using the **Slash** framework to integrate with existing infrastructure.

NCR Voyix May 2024 – Aug. 2024

Systems Software Engineering Intern *Waterloo, Ontario*

- Utilized **Python** to integrate an internal query utility into a newly-initiated patch management project.
- Instrumented a service to validate per-device network compliance data for use **organization-wide**.
- Developed a patch verification tool to serve over **10,000** devices across **10+** platforms.

Projects

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

- Designed a kernel with basic functionalities such as interrupt handling and inter-process communication.
- Implemented a cooperative multitasking scheduler for efficient and lightweight user process management.
- Built a custom bootloader to initialize hardware peripherals for a stable system boot sequence.

osh: The Open Shell | *C, Linux*

- Created a rudimentary system shell in C for **Linux** systems, with support for various built-in and external commands.
- Utilized **Linux system calls** to implement piping ($|$), redirection ($<, >$) and custom shell built-ins.
- Improved responsiveness by adding a persistent command history with **readline**.

wintop | *C, MSVC, Windows API*

- Developed a command-line based **thread and process inspector** in **C** for **Windows** platforms, providing detailed scheduling information per-unit.
- Leveraged **Win32 functions** to create process snapshots, enumerate active threads and retrieve their metadata.
- Designed a terminal interface to provide **real-time** diagnostic information, emulating **top** and **ps** in *nix systems.

Skills

Languages: C, C++, Rust, Python, Bash, Assembly (ARM, RISC-V)
Libraries & Tools: Valgrind, GDB, CMake, Make, Android Tools (adb, Fastboot), Docker
Technologies & Protocols: Linux, QNX, FreeRTOS, gRPC, HTTP, TCP/IP, UART, I2C