Alberto Damo | Kernel & Systems Software Developer

Oderzo, Italy
alberto.damo@proton.me • github.com/mrmonopoly-cyber
alberto-damo-62b81b23a • Master's Student in Computer Science

Profile

Systems and embedded software developer specializing in **low-level C programming**, **operating system internals**, and **bare-metal development**. Currently developing a **UEFI-compliant OS from scratch**, including a custom bootloader, kernel, and disk image generation utility for QEMU (Tianocore). Passionate about Linux, kernel programming, and open-source collaboration. Seeking to contribute to IBM's Linux kernel and virtualization initiatives while deepening expertise in systems architecture and performance.

Technical Skills

Programming: C (expert), Rust, C++, Java, Haskell, Golang, F#, Assembly (x86_64 GAS basics), Bash

Systems: Linux internals, bootloaders, UEFI, memory management, kernel architecture

Embedded: AURIX TC375 (TriCore), tricore-elf-gcc, linker scripts, atomic operations, GPIO & CAN control

Tools: Git, GCC, GDB, QEMU, OVMF (Tianocore), libgpiod, SocketCAN

Concepts: Concurrency, interrupts, synchronization, atomic operations, virtualization fundamentals

Education

University of Padua
Master's Degree in Computer Science — *Expected 2026*
Focus areas: Operating Systems, Embedded Systems, Computer Architecture.

University of Udine
Bachelor's Degree in Computer Science — *2019–2023*
Graduated with a focus on Software Engineering, Systems Programming, and Distributed Systems.

Selected Projects

UEFI-Compliant Operating System from Scratch
Custom OS — *2025–Present*

- Designed and implemented a **UEFI bootloader** and **monolithic kernel** entirely in Rust and Assembly.
- Built a UEFI-compliant **disk image generation utility** compatible with QEMU (OVMF / Tianocore).
- Developed early boot stages including **memory initialization**, console I/O, and process management foundations.

Linux-Based Multi-Component System Simulator
ControlUnitLogicOperator — *2024–2025*

- Built a modular simulation environment of 46 emulated components communicating via shared CAN buses and GPIOs.
- Implemented a **runtime trace stack** for error diagnostics using __FILE__ and __LINE__.
- Designed a **Safety Critical System (SCS)** with atomic synchronization across threads.
- **Repository:** `https://github.com/mrmonopoly-cyber/ControlUnitLogicOperator`

Command-Line Debugging Interface
DPS (Debug Peripheral System) — *2024–2025*

- Created a **CLI-based debugging tool** supporting live monitoring and introspection of system state.
- Provides modular runtime telemetry for GPIO and CAN buses.
- **Repository:** `https://github.com/mrmonopoly-cyber/DPS`
- **Repository:** `https://github.com/mrmonopoly-cyber/dps_cli`

Proof-of-Concept for Expression Evaluation via Interaction Combinators

Arithmetic Combinators Interpreter *2024*

- Designed and implemented an interpreter for arithmetic expressions based on the theory of Interaction Combinators
- Written in Rust, supports parsing of arithmetic expressions and reduction via Interaction combinators graph theory.
- Demonstrates theoretical computation models (graph rewriting, combinators) applied to practical expression evaluation.
- **Repository:** `https://github.com/mrmonopoly-cyber/arithmetic_combinators`

Distributed Computing Cluster

Raft Project Runtime *2024*

- Designed and implemented a **Raft-based distributed runtime** in Go for orchestrating a self-hosted VM cluster.
- Built a fully automated image generation pipeline using **ArchISO** and **libvirt** for deployment.
- Enabled reproducible multi-node boot environments and network coordination through a lightweight cluster manager.
- **Paper link:** `https://www.linkedin.com/feed/update/urn:li:activity:7298456277242806272/`
- **Repository:** `https://github.com/mrmonopoly-cyber/raft_project_runtime`

Extended Gentoo's qmerge utility for DevOps automation — Nucleus S.r.l.

Portage-Utils (qmerge fork) *2023*

- Forked and maintained a customized version of Gentoo's `qmerge` utility to add functionality for internal deployment workflows.
- Integrated new features for dependency resolution and build pipeline optimizations within a large-scale DevOps environment.
- Project available at: `https://github.com/mrmonopoly-cyber/portage-utils`

Experience

RaceUp Formula SAE Team

Software Department Member *2023–2024*

- Member of the software department of **RaceUp**, the official Formula SAE team of the University of Padua.
- Designed and implemented the **ControlUnitLogicOperator** project: a modular Linux-based system simulator for embedded control units.
- Developed a **Safety Critical System (SCS)** with atomic synchronization across threads and real-time exception handling.
- Created the **DPS (Debug Peripheral System)** and **DPS CLI** to provide live debugging and telemetry of GPIO and CAN signals.
- **Repositories:** `https://github.com/mrmonopoly-cyber/ControlUnitLogicOperator`, `https://github.com/mrmonopoly-cyber/DPS`, `https://github.com/mrmonopoly-cyber/dps_cli`

Nucleus S.r.l.

DevOps Developer *2023–2023*

- Contributed to infrastructure automation and software deployment pipelines.
- Forked and extended **Qmerge (Gentoo portage-utils)** to enhance dependency handling and CI behavior.
- Worked with system-level Linux tools, package management, and scripting for system provisioning.
- **Repository:** `https://github.com/mrmonopoly-cyber/portage-utils`

Nucleus S.r.l. (Udine, Italy)

Cloud Engineer Intern *2023-2024*

- Worked on distributed storage and orchestration solutions using **Ceph** and **Kubernetes**.
- Assisted in configuring and maintaining on-premise and hybrid cloud clusters.
- Focused on performance optimization and deployment automation.

Achievements

- Currently building a fully bootable UEFI-compliant OS from scratch.
- Published open-source systems demonstrating atomic synchronization and debugging architecture.
- Built a custom 32-bit Linux environment for emulation and testing as test environment for the **ControlUnitLogicOperator**.

Additional Information

Languages: **English:** C1 (Advanced)
Italian: C2 (Native)

Interests: Kernel engineering, virtualization, systems security, open-source development

Goal: To expand my expertise in kernel and systems programming by contributing to IBM's Linux kernel, virtualization, and confidential computing initiatives.