# Mason Miller

+1 (734) 277-3137 masonmil@umich.edu @masonmil @masonmill

## Education

**University of Michigan** **Ann Arbor, MI**

*Bachelor of Science in Computer Science* *August 2022 - May 2026*

- **GPA:** 3.87/4.00 | **Awards:** James B. Angell Scholar, William J. Branstrom Prize, University Honors, Regent's Scholar
- **Relevant Courses:** Data Structures & Algorithms, Operating Systems, *Computer Architecture*, Web Systems, Computer Science Theory, Intro to Computer Architecture, Discrete Math, Probability Theory, Linear Algebra

## Technical Skills

**Languages**: C, C++, ARM Assembly, Bash, Python, JavaScript, HTML, CSS, SQL
**Technologies**: Linux, UNIX, ARMv8, QEMU, Git, GDB, Valgrind, System calls, POSIX APIs

## Experience

**Ordered Systems Lab** **University of Michigan**

*Research Assistant* *May 2024 - Present*

- Developed a program in **C** leveraging the **liburing** API to asynchronously read a file on Linux with **io_uring**.
- Built Meta's **RocksDB** with CMake on a Debian Linux VM using **QEMU**, integrating Meta's **folly** library for advanced **C++20** features, including coroutines and async I/O. Evaluated MultiGet & Scan operations with **db_bench** and **Google Benchmark**. Conducted a comparative analysis of async I/O libraries, including **liburing** and **libaio**.

**The Boyle Lab** **Michigan Medicine**

*Research Assistant* *May 2023 - September 2023*

- Engineered a bioinformatics application using the **NCBI C++ Toolkit**, **SAMtools**, and **minimap2** to identify DNA sequences in large datasets. Implemented a pipeline for Cas9-targeted nanopore sequencing, enhancing the detection of mobile element insertions (MEIs). This work improved computational efficiency in genomic analysis.

**Departmental Computing Organization** **University of Michigan**

*Computer Consultant* *June 2022 - May 2023*

- Installed, configured, and troubleshot Windows (10 & 11), MacOS, and Linux (Red Hat Enterprise & Ubuntu) operating systems, ensuring optimal performance across 100+ machines. Managed network configurations using Active Directory and Domain Name System, maintaining security and connectivity in EECS server rooms.

## Projects

### Thread Library

- Developed a kernel-level C++ thread library on UNIX, managing CPU booting, thread life cycle, and scheduling for 50+ CPUs. Implemented synchronization primitives like spin-locks, mutexes, and condition variables using advanced UNIX context management techniques.

### Virtual Memory Pager

- Designed and implemented a virtual memory pager supporting multiple processes with swap-backed and file-backed memory pages, akin to UNIX mmap. Handled process creation, page faults, memory management unit (MMU) bits, process forking, and destruction with copy-on-write optimization.

### Multi-threaded Network File Server

- Built a concurrent, crash-consistent network file server, supporting multiple users with nested files and directories. Ensured crash consistency using committing writes, and optimized concurrency with Boost threads and reader-writer locks. Implemented network communication using POSIX sockets for client-server interactions.

### MapReduce Framework

- Implemented a fault-tolerant MapReduce framework for distributed processing across a cluster, utilizing threads, processes, and socket-based networking. Enhanced system reliability and efficiency in processing large datasets.

### Instagram Clone

- Created a full-stack web application with React for the frontend, implementing features like authentication, post interactions, and user accounts. Built REST APIs using Flask, processing requests with dynamic data from an SQLite database. Deployed the application on AWS EC2, utilizing Gunicorn and Nginx for scalable and efficient hosting.

## Other

**Interests**: Low-level Programming, Bouldering, Korean Language & Pungmul, League of Legends, Tennis