

# MARVIN JIRAPONGSUWAN

✉ [marvincs@umich.edu](mailto:marvincs@umich.edu)  [marvincs.com](https://marvincs.com)  [linkedin.com/in/marvin-jirapongsuwan](https://linkedin.com/in/marvin-jirapongsuwan)  [github.com/enternal-L](https://github.com/enternal-L)

## Education

---

### University of Michigan

Ann Arbor, MI

*B.S.E. in Computer Science, Minor in Business*

*May 2027*

- **GPA:** 3.78/4.00
- **Awards:** Walter G. Mitchell Memorial, Oskar and Elsie R. Loosme, Dean's List, University Honors
- **Relevant Courses:** Data Structures & Algorithms, Operating Systems, Distributed Systems, Computer Organization, Web Systems, Computer Security

## Technical Skills

---

**Languages:** C++, C, Go, Python, JavaScript, Bash, SQL, Lua, MATLAB

**Systems:** Linux, Unix, POSIX, ZFS, ext4

**Technologies:** React, Next.js, FastAPI, Node.js, MongoDB, PostgreSQL, Git, Make, Apache Kafka, GDB

## Experience

---

### Nutanix

San Jose, CA

*Software Engineer Intern*

*May 2025 – August 2025*

- Spearheaded async I/O integration into core network file server (Ganesha NFS) using shared ring buffers (io\_uring), cutting userspace threads by 68%, improving bandwidth (+14.6%) and latency (-31%) on batched, high latency workloads
- Modified internal file system (ZFS) for correct io\_uring integration, enabling ARC-cache-aware, io\_uring-driven kernel thread dispatch that achieved 27x parallel I/O scaling (32-core cluster) and reduced redundant thread creation by 61%
- Evaluated io\_uring performance & behavior on Linux file systems using Fio across I/O patterns, tracing with bpfttrace/kprobes
- Built custom benchmarking suite for io\_uring designs, measuring I/O bandwidth and latency to guide NFS design integration

### Joy of Coding (University of Michigan)

Remote

*Teaching Assistant (TA)*

*May 2025 – August 2025*

- Supported ~400 students in an online Python course in weekly 1-on-1 Zoom and email support
- Mentored students on programming basics (functions, conditions, loops), AI fundamentals, and data visualization

### UM Direct Brain Interface Lab

University of Michigan

*Undergraduate Research Assistant*

*September 2024 – May 2025*

- Developed EEG-based BCI applications in C++ for individuals with motor impairments to communicate via brain signals
- Improved AAC-BCI keyboard reliability with core bug fixes and error checks, reducing visual and hardware failures by 30%
- Enhanced Choice-Making module with stimuli-skipping, and refactored algorithms to improve response time and flexibility

## Projects

---

### Nutanix Pinsir Security | *React, Flask, ClamAV, PostgreSQL, Nutanix Objects, Apache Kafka*

- Built a real-time malware scanner for S3-compatible storage with a Flask REST API and Kafka-backed pipeline, enabling async file scanning with ClamAV, securing 98% of stored objects and reducing processing time by 90%
- Designed security features including scan-gating, S3 tag-based access control, and audit logging with PostgreSQL
- Led React frontend integration with backend APIs for upload, retrieve, and download workflows

### Manual EQ Tool (Parrot Producer) | *C++, JUCE, Digital Signal Processing*

- Built DAW plugin using JUCE that facilitates EQ for music producers, user-friendly functionality for beginners
- Instant audio processing with GUI, allowing manual tuning of Frequency, Gain, Quality and EQ

### Reddit Classifier (AITAH) | *React, Python, FastAPI*

- Built Reddit-based classifier using a bag-of-words model, achieving 84% accuracy on 3,000+ posts
- Created single-word probability feature, evaluating 24,000+ words to highlight model biases

### Thread Library | *C++, Multi-threading, Mutexes, Condition Variables, Semaphores, Unix*

- Developed a kernel C++ thread library on Unix, handling CPU booting, thread management, management of 50+ CPUs, interrupts, atomicity, and FIFO scheduling order
- Designed spin-locks, mutexes, conditional variables utilizing advanced Unix context management

### Map Reduce | *Python, Threads, Sockets*

- Implemented Hadoop-like MapReduce Distributed System to process large datasets through TCP and UDP communication
- Devised Manager-Worker model with dynamic task assignment, fault tolerance, and support of partitioned clustered data

## Other

---

**Interests:** Filmography/Video Editing, Volleyball, Basketball