

DERIC DINU DANIEL

dericdd@umich.edu | deric.dev | github.com/dericdinudaniel | linkedin.com/in/dericdinudaniel

EDUCATION

University of Michigan – Ann Arbor

Computer Science B.S.E.; GPA: 3.64

Dean's List, University Honors

Aug. 2021 – May 2025

Ann Arbor, MI

Relevant Coursework: Operating Systems, Data Structures & Algorithms, Web Systems, Advanced Computer Architecture, Parallel (GPU) Programming, Computer Security, Computer Science Theory, Intro to Circuit Analysis, Music Production

EXPERIENCE

Microsoft

June 2024 – Sept. 2024

Software Engineer Intern

Redmond, WA

- Integrated support for new data source in security/compliance platform to enable data loss prevention on new asset category, furthering protection and security compliance efforts for customers.
- Utilized **C#** cmdlets to fulfill business logic to support new data source, used **React/Typescript** to build intuitive asset picker component, and integrated new asset discovery API to enable end-to-end integration of new data source.
- Supported migration of API to .NET Core to improve cost-effectiveness and scalability through containerization.

Bose – Research Division

Jan. 2024 – June 2024

Systems Software Engineer Intern

Framingham, MA

- Designed real-time streaming system for multiple mics and sensors using LC3 codec, Qualcomm audio system, and **Bluetooth LE** in a large **multithreaded embedded system** in **C**, supporting 5+ high visibility research projects.
- Enabled ability to run larger ML models on host device (vs headphones/earbuds), multichannel data collection, and high-quality audio recording during music playback and voice calls, allowing for new (upcoming) user experiences
- Researched and presented effects of LC3 codec on quality performance of audio deep learning models using SISDR, STOI, and PESQ metrics in an automated **Python** script.
- Prototyped Bluetooth LE GATT receiver system on Infineon Cypress Microcontroller to enable rapid testing of mic/sensor streaming interface to maximize bandwidth and minimize packet loss.

Siemens

May 2023 – Aug. 2023

Software Engineer Intern

Troy, MI

- Managed DB and environment instances in **AWS** using **EC2** and **S3** while reducing development server costs by up to 5% to deploy testing and demo environments.
- Resolved critical bugs in dropdown menus on **C++** server-side code by refactoring with smart pointers, fixing product inconsistency and resolving 8 memory leaks per dropdown interaction.

Shade

Apr. 2023 – Sept. 2023

Audio Consultant

Ann Arbor, MI

- Assisted founders with pivot and development of AI audio file tagging system and music production tool, leveraging 4 years of music production expertise, resulting in doubling of total addressable market and 10+ new features.

PROJECTS

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

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

Virtual Memory Pager | C++, Virtual Memory, Page Faults, Process Lifecycle Management

- Designed a virtual memory pager which managed multiple processes and supported swap-backed and file-backed memory pages (similar to Unix `mmap()`). Managed process creation, page faults & MMU bits, process forking, process destruction, and swap disk all while supporting copy-on-write.

Multithreaded Network Fileserver | C++, Boost Library, Threads, Sockets

- Built a heavily concurrent, crash consistent network fileserver supporting multiple users and nested files/folders.
- Utilized committing writes to enable crash consistency, Boost threads and upgradeable reader-writer locks to optimize for maximum concurrency, and POSIX sockets to enable network communication with clients.

Delay Audio Effect Plugin | C++, JUCE Library, Digital Signal Processing

- Built an audio delay effect for DAW plugins using JUCE. Features include DAW tempo sync, stereo ping pong delay, low and high cut on wet signal, dry/wet mix, output gain, and level meter.
- Implemented custom circular buffer delay line with Hermite-interpolation, one-pole filtering for analog emulation, linear parameter smoothing to reduce zipper noise, and delay time crossfading to reduce artifacts

TECHNICAL SKILLS

Languages : C/C++, Python, ARM, x86, C#, HTML/CSS, Javascript/Typescript, SQL, Verilog

Technologies : Next.js/React.js, Git, gdb, Linux, AWS, Flask, MapReduce, Sockets, Multithreading, Networks, JUCE

Misc. : Ableton Live, FL Studio, Adobe Lightroom, Pioneer Rekordbox

Interests : Music Production/Audio, Consumer Tech, Soccer/Running/Gym