Tyler Neal

Software Engineer

Chesterfield, MI, 48047 | +1 810-278-2749 | tylerneal.dev@gmail.com | github.com/tp-neal | linkedin.com/in/tyler-neal-dev

SUMMARY

Recent graduate from Michigan Technological University with extensive experience in systems programming and low-level C/C++ development. Demonstrated proficiency in compiler design, cache analysis, and concurrent programming through multiple academic and team projects. Seeking opportunities to apply analytical skills to complex systems challenges in high-performance computing, distributed systems, or system architecture.

EDUCATION

Bachelor's of Science in Computer Science, Major GPA: 3.43 — 2020 – 2024
Michigan Technological University — *Houghton, MI*

PROJECTS

C-Like Language Compiler | C++, MIPS Assembly, Yacc, Flex — September 2024 – December 2024

- Developed a multi-stage compiler translating a C-like language into MIPS assembly using Flex (lexical analysis) and Yacc (syntax parsing).
- Implemented semantic analysis and code generation utilizing abstract syntax trees (AST's) and symbol tables.
- Supported core language features including control flow constructs, arithmetic expression evaluation, and variable scope management.

Multi-Layer Cache Simulator | Research Project | C, Spec95 — March 2024 – April 2024

- Designed and implemented a multi-layer (L1, L2, L3) direct-mapped write-back trace-based cache simulator in C
- Analyzed cache hit/miss rate, and memory access time using trace-based simulation with the Spec95 benchmark suite; validating trace files up to 6.15GB.
- Authored a technical paper detailing simulator design, performance metrics, and cache management techniques.

Concurrent Systems Programming | C, POSIX, Threading — September 2023 – December 2023

- Implemented client-server RPC communication using C sockets and process forking, managing data serialization and network byte order conversion.
- Engineered multithreaded programs using mutexes, semaphores, as well as Mesa and Hoare-style monitors to ensure data integrity and prevent race conditions.
- Developed a command logging system for a custom bash interpreter, leveraging process forking and inter-process communication (IPC) via pipes.
- Implemented lossless Hoffman Encoding algorithm, comparing compression with Canterbury benchmark.

TECHNICAL SKILLS

Systems Programming:

- Memory Management, Process Control, Cache Optimization, Binary Analysis, Threading, Synchronization

Operating Systems & Development:

- Linux, System Calls, Process Management, GDB, Make, Valgrind, Git/GitHub

Languages & Technologies:

- C, C++, JavaScript, Python, HTML, CSS, SQL, Java, C#, Bash, MIPS Assembly