Yurii Shmal

yshmal@andrew.cmu.edu • 412-482-0372 • yuriishmal.site • linkedin.com/in/yurii-shmal-1a5800245/

Education

Carnegie Mellon University USA.
Bachelor of Science in Computer Science — Concentration in Computer Systems Expected December 2026

- Coursework: Introduction to Probability for Computer Scientists, Introduction to Machine Learning, Intro to Computer Systems, Parallel and Sequential Data Structures and Algorithms, Algorithm Design and Analysis, Networking and the Internet, Database Systems, Management of Computer and Information Systems, Introduction to Computer Security
- Cumulative GPA: 3.71 / 4.0

Work Experience

Charlie - a CMU Startup Remote from Saudi Arabia
Full-Stack Software Engineer Intern May 2023 - August 2023

- Developed, published, and updated a full stack web app (https://charlie-nine.vercel.app/) coded primarily with TypeScript, HTML, CSS using the following technologies: NodeJS, NextAuth, Tailwind CSS, and MongoDB
- Leveraged human-computer interaction principles to ensure a seamless website experience in an agile environment

Projects

x86 Operating System Kernel September 2026

- Built a 32-bit x86 operating system kernel supporting processes and threads, 25 system calls, user/kernel mode transitions, and interrupt and exception handling.
- Designed an O(1) round-robin scheduler with timer-driven preemption and low-level context switching; implemented blocking mutexes, condition variables, spinlocks, and lost-wakeup-safe synchronization.
- Implemented per-process virtual memory with page tables, zero-fill-on-demand, page allocation/removal, safe user-memory access, and address-space cloning for fork.
- Implemented fork, exec, wait, and process/thread termination, including ELF program loading, orphan reparenting, exit-status management, and deferred thread cleanup via a dedicated reaper thread.
- Implemented 25 system calls supporting process/thread management, virtual memory, scheduling, console I/O, and user-level exception handling.

BusTub Database May 2025

- Designed and implemented a thread-safe Buffer Pool Manager with an LRU-K page replacement policy to efficiently cache disk pages and support multi-threaded I/O operations.
- Developed a B+Tree index with a latch-crabbing concurrency control scheme
- Built and optimized core query execution operators (e.g., Sequential Scan, Hash Join, Aggregation)
- Engineered a MVCC system to handle concurrent transactions and ensure data isolation and atomicity
- Implemented a series of query optimizer rules to rewrite logical query plans into more efficient physical plans

HTTP Server and Client December 2024

- Implemented HTTP Server in C capable of supporting multiple connections using IO multiplexing with poll, parsing and handling HTTP GET, HEAD, and POST requests
- Implemented Client in C capable of setting up multiple concurrent connections to the same server for faster page load times

Skills

Programming languages: Python, C, C++, Java, JavaScript, TypeScript, Standard ML, Kotlin
Data Science/ML: Numpy, Pandas, Matplotlib, Seaborn, PyTorch, scikit-learn, TensorFlow
Full Stack Development/Cloud: HTML/CSS, NodeJS, React, MongoDB, sqlite3, AWS development, Oracle cloud
Work Applications: Microsoft Office Suite, Google Workspace, ChatGPT, Codex, Cursor, Visual Studio Code, Vim
Debugging Applications: gdb, Valgrind