Steven Xia

xia.steven.xia@gmail.com | (302) 482 – 5827 | 230 W 56th St, Apt 68B, New York, NY 10019 | xia-steven.github.io

EDUCATION

University of Michigan College of Engineering, Honors Program **Ann Arbor, MI**
Bachelor of Science and Engineering in Computer Science *Expected Graduation: May 2024*
GPA: 3.94 / 4.0
Past Coursework: Advanced Operating Systems, Networking, Databases, Web Systems, Parallel GPU Programming, Game Development, Data Structures and Algorithms, Computer Organization, Computational Theory, Circuits, Linear Algebra, Differential Equations, Discrete Math
Upcoming Coursework: Distributed Systems, Compiler Construction, Computer Security, Computer Architecture, Embedded Systems

University of Michigan Ross School of Business, Business Minor **Ann Arbor, MI**
Coursework: Business Strategy, Marketing, Finance, Accounting, Economics *Expected Graduation: May 2024*

EXPERIENCE

University of Michigan College of Engineering **Ann Arbor, MI**
Teaching Assistant for EECS 482: Operating Systems *August 2022 – Present*

- Assisted Professors Manos Kapritsos and Baris Kasikci in operating an advanced-level Operating Systems course, teaching concurrent programming, kernel-level thread implementation, scheduling, virtual memory, filesystems, networking, distributed systems, etc.
- Authored a comprehensive C++20 guide about safety, efficiency, style, advanced / modern features, design patterns, and debugging
- Led weekly lab sessions and hosted 1-on-1 office hours to reinforce understanding and guide project completion
- Previously taught EECS 484: Database Management Systems, an advanced-level SQL and C++ database engineering course

Amazon Web Services (AWS Serverless Compute Fabric) **New York, NY**
Software Development Engineering Intern *May 2023 – July 2023*

- Developed a novel Rust metric emission mechanism to emit metrics out of virtual machines used by multiple internal customers, which supports unlimited concurrent metrics emitted from any number of virtual machines, while enforcing robust per-VM rate limits
- Designed a distributed pipeline involving partitioning, data filtering, demand-based scaling with security and fault tolerance guarantees
- Implemented fine-grained locking and log batching, which decreased CPU idle time and led to 50% decrease in Lambda invocations
- Performed extensive design research and cost analysis; wrote internally reviewed documentation on the pipeline's infrastructure design

CyberCube Analytics **New York, NY**
Data Analyst Intern *May 2022 – August 2022*

- Developed Python filtering algorithms that process risk data, update company databases, and match vulnerabilities to client companies
- Migrated resources to a new AWS account and set up infrastructure using Terraform and AWS CloudFormation
- Set up CI/CD pipeline for the Data Collection Team's AWS ECS workflow using GitHub Actions

PROJECTS

Michigan Mars Rover Project Team (C++20, CUDA) (https://github.com/umrover/mrover-ros)

- Competed in the University Rover Challenge in a team of over 100 students and received 1st place in autonomy portion
- Developed an embedded and templated C++20 units library that supports all SI units and improved compile time by over 50x
- Implemented and tested visual odometry that improved local positioning in addition to the existing global GPS positioning

Multiprocessor Thread Library (C++20, C)

- Implemented a kernel-level thread library that supports the creation of threads, synchronization mechanisms including mutex, condition variable, semaphore, latch, barrier, upgradable reader/writer mutex, recursive mutex, and RAII lock wrappers, and stop token
- Designed a priority-based scheduling mechanism that favors interactive workloads by tracking a thread's CPU time

Network File Server (C++20, Boost)

- Implemented a multithreaded file server that supports concurrent operations to read, write, create, and delete files / directories
- Used Boost library's upgradable reader/writer locks to maximize concurrency as threads are bound by slow network and disk I/O
- Minimized expensive disk I/O by eagerly detecting and rejecting invalid requests, as well as caching directory entry indices

Virtual Memory Pager (C++20)

- Implemented system calls supporting process creation / deletion / forking, memory allocation / deallocation, and fault handling
- Simulated a least-recently-used eviction policy using the Clock Algorithm that supports eviction to the swap-space and files
- Minimized expensive page faults and disk I/O by implementing a copy-on-write mechanism when handling process creation / forks

Haunt (Unity, C#) (https://github.com/xia-steven/Haunt)

- Developed a single-player 2.5D Bullet-Hell video game with a team of 4 other students using the Unity Engine and C#
- Designed an adaptive A-Star pathfinding algorithm that changes enemy behavior based on player location, obstacles, and game state
- Used Jira to efficiently track project progress including sprint planning, task assignment, time estimation, category separation

SKILLS

Languages / Tools: C++ / C, Rust, Python, C#, SQL, GDB, Valgrind, Gcov, CMake, Git, *Go*, Verilog*, STM32*, ARM*, x86**
Libraries / Technologies: STL (C++20), CUDA, C++ / Rust concurrency, Unix, MacOS, Unity, AWS (Certified Cloud Practitioner), Jira

* *Italicized skills will be obtained in the upcoming school year.*