Yifan (York) Liu

yil173@ucsd.edu 510-612-0541

Education

University of California - San Diego
M.S. in Computer Science
Sep 2021 - Jun 2023

Washington University in St Louis
B.S. in Computer Science
Aug 2019 - May 2021
GPA: 3.99/4.00
Honors: Outstanding Senior Award
Summa Cum Laude

University of California - Santa Cruz
Computer Science
Sep 2017 - Jun 2019
GPA: 3.99/4.00
Transferred

Courses & Skills

Systems Software
Computer Networks
Systems Security
Operating Systems
Multicore Computing
Theory of Parallel Systems
Programming Systems & Languages

Machine Learning
Artificial Intelligence

Data Structures
Analysis of Algorithms
Object Oriented Programming
Front End Development

Number Theory & Cryptography
Probability and Statistics
Graph Theory & Combinatorics
Discrete Mathematics & Logic
Linear Algebra
Real Analysis

Programming Languages & Tools

C/C++
Java
Python
Lua
Shell
SQL
Javascript

HTML/CSS
Ruby
LISP

Docker
Git
Linux
Latex

Contests

CTF - STLCyberCon
First Place (Team), Nov 2020

Asia-Pacific Informatics Olympiad
Bronze Medal, 2016

Experiences

Undergraduate Researcher @ Washington University in St Louis Feb 2020 - Sep 2021

- Worked under Dr. I-Ting Lee's Lab on a work-stealing scheduler called Interactive-Cilk.
- Rewrote the well-known web server NGINX using Interactive-Cilk and C++, built containerized benchmarking software, and performanced-engineered the server.
- As a second author, submitted papers to PLDI 2021 and ASPLOS 2022.

Full-stack Game Developer @ Maximillian Studios™ Oct 2017 - Jul 2019

- Designed and Published a first-person shooter game Operation Scorpion on Roblox, which has accumulated 10 million visits and thousands of Youtube Videos.
- As the only programmer in the start-up game studio, managed a codebase of 30000+ lines.
- Deployed a backend server for player data storage using CentOS, postgreSQL and ExpressJS.
- Implemented the in-game UI, procedural animation, inverse kinematics, matchmaking system, player progression system, gunplay, and team-based game logic in Lua.

Teaching Assistant @ Washington University in St Louis Feb 2020 - May 2021

- Served as a teaching assistant for two courses (Intro to Systems Software and Analysis of Algorithms) for multiple semesters.
- Graded homeworks and exams. Held office hours answering students' questions.

Projects

- Designed and implemented a Vocabulary Memorizer using HTML, CSS, javascript, Google Sheets API, and Google OAuth2 API. The user can import/edit the word list in a Google Sheet, and start memorizing words in a flashcard mode, where the front of the flashcard is the word and the back of the card is its meaning. The front and back can hold texts in any language, which is useful when learning a new language and memorizing unfamiliar words.

- Built a web-based todo list using React, where the user can add a todo item, mark/unmark a todo item as done, and remove a todo item.

- Using shell scripts, built a **zoom classroom launcher** during quarantine where the user enters a partial name of the class and the application will open the zoom meeting based on a configurable list of zoom links. On Linux, the application can also be configured to open a zoom class automatically at class time with the help of a built-in scheduling utility called cron.

- Performed **cache analysis** on several cache oblivious/aware algorithms in C++: Implemented mergesort, K-way mergesort, funnelsort, blocked matrix multiplication, and matrix multiplication in Z-layout; Wrote a cache simulator based with several replacement policy such as LRU, RR, FIFO, MRU, NMRU; Compared each algorithm's cache miss rate to its theoretical I/O complexity.

- Designed and Implemented an **interactive command-line dictionary** with shell scripts and various open-source utilities like fzf and w3m. Also wrote a small program in C to parse StarDict dictionaries.

- Wrote an **interactive shell** using C and low-level system calls such as fork-exec-wait.

- Built a **memory allocator** that supports malloc, free, realloc, and calloc. This allocator is an implementation of segregated list in C, and it beats the performance of glibc's malloc in single core performance.