HAO TUONG LAM

PROFILE

I am a Senior at the University of Washington, majoring in Electrical Engineering. However, for the last two years, I have developed a new interest in Computer Science. Not only that I see this position as an ideal opportunity to explore my new interest, but also to contribute the skills I have learned. Since last year, I have completed most of the online curriculum in App Academy. When I am free, I work on Leetcode problems, learning about Distributed System Designs and Ruby on Rails. Now, I am learning about computer architecture because I believe that understanding how things work at the lowest level helps me become a more well-rounded engineer. Computer Science topics are so compelling that I am willing to self-taught myself all the skills that are needed to become a software engineer. I fell in love with Computer Science because it satisfies my curiosity. The deeper I get into this field, the more that I want to learn. I believe that learning is a lifelong process. I work really hard. I am persistent and I am responsible for my work.

EDUCATION

Bachelor of Science in Electrical Engineering

Concentration in Embedded Computing Systems

University of Washington - Seattle

2019 - Present

Major GPA: 3.80

Honors:

- Annual Dean's List Award
- Top 6 of DECA Area 5 Competition (2018)

EXPERIENCES

RISC-V Core on FPGA *Jan- March 2023*

- 37 RV32 Base Instruction Set was implemented in Verilog and run on FPGA board
- Can run arbitrary code that had compiled to RV32 executable
- Pipelining stages are added to increase throughput
- Demonstrate C printf() working by displaying string on FPGA 7 segment displays

Password Based Door Locked *Jan- March 2023*

- Implement a security system that can capture photos of suspicious intruder when the enter password is wrong
- Photos will be send to user email along with newly random generated password
- Face recognition is added using universal encoding

File System Crawler, Indexer, and Search Engine on web server *Jan- March 2023*

- Implement LinkedList and HashTable modules and use these modules to build System Crawler, Indexer, and Search Engine
- System Crawler reads content of a file into memory, parses it into a series of words, and builds a linked list of (word, position) information. Indexer converts a series of these linked lists into an in-memory inverted index. Search Engine uses the inverted index to build a query processor that has a console-based interface
- Improve previous system by moving the in-memory inverted index to an on-disk index
- Implement a multithread web server front-end to the previously implemented query processor

MY SKILLS

Programming(OOP, C, Ruby)+Debugging Skills	★★★★★★★★★★
Digital Design(HDL) & Assembly(RISCV, x86-64)	★★★★★★★★★☆
Embedded Programming - IoT	★★★★★★★★★☆
Data Structure & Algorithms	★★★★★★★★★☆
System Programming	★★★★★★★★☆☆
Web Development(Rails)	★★★★★★★★☆☆

HAO TUONG LAM

EXTRAS

Language Skills

Fluent in English and Vietnamese
Intermediate experience in Mandarin

Office Skills

Certified with Microsoft Office Applications: Word, Excel, and Access (2018)

CONTACT

Address

627 S 186th ST, Burien, WA 98148, USA

Phone

+1 (206) 960 - 3660

Email

tuonghao2001@gmail.com

GitHub

https://github.com/haolam05

HOBBIES

Working out
Hiking
Playing Cards
Reading

EXPERIENCES CONTINUED

8-bit breadboard CPU, 6502 Computer, Worst Video Card *Jul - Sept 2022*

- Even though I am interested in software, I believe that understanding what happens at the lowest level helps me solve more problems
- Inspired by Ben Eater Youtube Channel, I learned how a CPU works by building one on a breadboard from scratch
- One of the most interesting things that I learned from building a CPU was that a Turing complete machine only needs 1 conditional jump
- Before microprocessor was vague and like a magical black box for me, after building the CPU on breadboard, I understand that it is a fancier version of it
- EEPROM was used to write instructions(given by the 6502 microprocessor) for programs
- Learned how to interface between the computer and other components such as LCD
- Learned how CPU handles interrupts from a variety of sources such as the PS/2 keyboard
- Builded a basic video card from scratch using VGA monitor
- Generated Vsync and Hsync signals for the VGA using a series of counters.
- Generated 4 different voltage levels(0 - 0.7V) for each RGB signal using 2 digital signals

Memory Management *April 2022*

- Created a memory management package similar “malloc” and “free” but simpler to understand of how C manage memory under the hood
- Wrote 2 main methods: freemem() and getmem(), which allocates appropriate memory block size when being asked and merge it back to the list when being freed
- Learned how to write a makefile

100 Best Websites *March 2022*

- Used linux commands such as wget, grep, and sed to pull and extract the necessary content down from a website and check to see whether these websites are accessible
- Wrote shell scripts to print the the result to standard output and plot using gnuplot

Husky Map *(Autocomplete, Priority Queues, and Shortest Path)* *Jan 2022*

- Worked in a groups to gained a deep understanding regarding data structures and algorithms during this project
- Experienced with running tests, analyzed runtimes among different approaches as well as space occupied

Turbo Chat *Nov 2021*

- Built a real-time chat app, including video call, infinite scroll bar, group/private chat, user's online statuses, preview messages... to gain a deeper understanding of Rails
- Took advantage of several gems such as Devise to deal with sign in and sign up process and Pagy to track pages and create infinite scroll bars
- Used hotwire and turbo stream to create a realtime chatting experience without refreshing the page for every new message
- Utilized Stimulus to track adb change user's statues in the frontend and SQL queries to ensure correct messages are returned in private/public rooms in the backend
- Experienced with deployment process using Heroku
- Challenged myself to add facetime feature by using OpenTok gem

Talker *Aug - Sept 2021*

- This was the first website that I built, and the key lesson that I learned is "A website can only recovered one's password if the password was not hashed in the first place"
- Built a social networking website, sign in and sign up process, modules, and user's identity protection...
- This project was done without using any existing frameworks with the goal is to understand the materials at a deeper level
- Learned basic concepts of web security, MVC model, and used Bootstrap and JavaScript for frontend designs

Tic Tac Toe, Eight Queens, Chess, Poker, Maze Solver… *April - Dec 2020*

- Inspired by the power of software in my freshman year, I decided to went through AppAcademy curriculum (App Academy Open) by myself
- Finished every single exercise and projects from *Welcome to Coding* to *SQL* and *Data Structure and Algorithm* sections
- Gained a strong foundation of classes and OOP through building many board games with the most challenging project is Chess
- Acquired a strong understanding of different data structures advantages and disadvantages along with space and time efficiency