William Greenwald

(619) 371-2170 | walexgreen@gmail.com | linkedin.com/in/wagreenwald | william-greenwald.com

Education

University of California - Santa Barbara Goleta, CA

Bachelor of Science, Computer Science *Sept. 2021 – June 2025*

Relevant CS: Object-Oriented Design; Computer Organization & Architecture; Data Structures & Algorithms; Digital Design; Operating Systems; Computer Networking; Artificial Intelligence; Compiler Design; Cryptography
Relevant Math: Calculus 1-3, Linear Algebra, Differential Equations, Discrete Math, Probability & Statistics

Technical Skills

Languages: C, C++, Python, Java, MIPS Assembly, HTML, CSS, Verilog, OCaml
Tools & Tech: (Neo)Vim, VS Code, GitHub, Perforce, NumPy, Sockets, Makefile, SCons, LaTeX, pyRTL, T32
Areas: Operating Systems, TCP/IP Networking, Computer Architecture/Design, Data Structures
Other: Agile (Scrum, Kanban), Native Bilingual (Spanish, English), Guitar, Piano, Club Ultimate Frisbee

Experience

5G Software Engineer Intern June 2024 – Sept. 2024

Qualcomm *Boulder, CO*

- Worked with the 5G Beam Management team to help improve memory issues found in off-target testing platform.
- Decreased a test structures memory by ∼88% and updated test validation, substantially decreasing memory for tests containing this structure. Coordinated cross-team efforts to update test generation scripts, preparing for release.
- Implemented dynamic/chunk allocation and freeing of off-target tests & XML in C++, extending the maximum simulated test time from ∼1.5s up to a theoretically unlimited amount of time.
- Integrated and deployed code to unified 4G/5G codebase, granting support to run more robust and extensible tests.

Embedded System Software Engineer Intern June 2023 – Sept. 2023

Qualcomm *San Diego, CA*

- Worked with the Multiprocessor Communications team to implement the usage of an open source IP stack to communicate between 2 processors within a Qualcomm proprietary system on chip (SOC).
- Created a custom physical transport layer in C for the IP stack by leveraging APIs developed by other teams.
- Wrote BSD socket & raw API tests, which were cross-compiled for a Qualcomm platform for testing.
- Traced through the executable/C code using T32 to debug runtime errors, memory issues, and interrupt system.

Projects

AutomataScript Programming Language (In Progress) | *C++, Makefile*

- Designed and implemented a custom programming language, using C++ to generate assembly code.
- Created support for automata-based pattern matching and regex transformations while keeping syntax simple.
- Built a tokenizer by leveraging a custom NFA structure, creating nodes & transitioning between states.
- Created an LL(1) grammar which defined the language syntax, referenced to create AST & handle errors.
- Translated the validated AST to C++ code after completing frontend, allowing AutomataScript to have high portability by using the users compiler, generating an executable with a single command.

System Call & Pipe Integrated Operating System | *C, Makefile*

- Developed an operating system in C, which included functionality to handle traps and interrupts.
- Implemented the availability of many basic system calls and pipes, enabling reliable inter-process communication.
- Gained an understanding of how operating systems aspects such as threading, time sharing, multiprogramming, and memory management when handling multiple processes.

Array Encoder/Decoder & Python Instruction Assembler | *MIPS Assembly, Python*

- Assembly program, takes an array of integers, calls an encode and decode routine, which takes the values and applies/undoes arithmetic (power, subtraction) to them and returns/prints values.
- Python program that takes in a MIPS assembly instruction and displays binary & hex of instruction on tkinter.
- Gained assembler knowledge with Python assembler; worked with jumps, the stack, & branches in assembly.