Mahesh Vegiraju

email:mvegiraj@ucsc.edu ○ in/MaheshVegiraju ○ github:Mahesh-Vegiraju

Education

University of California, Santa Cruz **Expected — June 2023**
Bachelor of Science Computer Science *GPA: 3.5*
Minor in Biology

Relevant CS Coursework: Programming Abstractions in Python, Assembly Language & Computer Systems, Embedded Systems & C Programming, Data Structures and Algorithms, Computer Architecture, Introduction to Computer Networks, Computer Systems Design, Foundations of Programming Languages, Introduction to Software Engineering
Relevant Biology Coursework: General Chemistry I, General Chemistry II, General Chemistry III, Cell & Molecular Biology, Biology: Development & Physiology, Genetics
Involvements: Santa Cruz NeuroTech, Cycling Club, Brooks Lab

Experience

Brooks Lab **Feburary 2022 - Present**
Undergraduate Researcher *Santa Cruz, CA*

- Working with long read & short read simulated data to compare statistical data between the two
- Working with technologies like FLAIR, JAFFAR, AERON, ARIBA and STAR

CalFee Design **November 2021 - Feburary 2022**
Independent Contractor *Santa Cruz, CA*

- Worked with motor controllers, hall sensors, buttons and arduinos to create the brains of an ebike system
- Implemented quality of life features like dynamic motor activation and smooth motor ramp up/down

Acorn Basket Studios **June 2019 - August 2019**
Software Engineering Intern - Game Mechanics *San Jose, CA*

- Helped implement object collision & procedural generation of the game map
- Worked in the ActionScript framework to make the game accessible to a wider audience

Projects

Huffman Compression Algorithm

- Implemented Huffman Compression and Decompression in C
- Created fundamental data structures including nodes, priority queues and stacks
- Used these fundamental data structures in tandem

Pintos Project

- Changed the way Pintos puts threads to sleep as to not use busy wait
- Implemented a priority ready queue that takes into account processes' priority when scheduling onto the CPU
- Implemented system calls such as exec, wait, etc

Hamming Encoding

- Implemented the error correction Hamming encoding algorithm in C
- Used data structures like bit vectors and bit matrices

Simple Motion Detection

- Implemented a simple motion detection algorithm
- Created sliding window Averagerators

Faster Merge Sort

- Made Merge Sort run faster using UNIX and POSIX child processes
- Also made a multi-threaded implementation using POSIX threads

Skills

Programming Languages

- Python, Java, C, C++, MIPS Assembly, RISC-V Assembly

Technologies

- Git, Bash, LaTeX, ActionScript, BeautifulSoup