Last Updated on June 6th, 2024

Aiden Munro

github.com/aidenfmunro manro.e@phystech.edu +7 915 298 11 30 Moscow, Russia

Education

MIPT 2023 - 2027

Department of Radio Engineering and Cybernetics *Overall GPA: 7.5/10, Programming GPA: 6.5/10*

Coursework

Courses: C Programming, Assembly, Discrete Math, Linear Algebra, Calculus, Physics, Probability & Statistics

Awards: Russian National Olympiad in Mathematics Region Prize Winner (4x), MIPT Olympiad in Mathematics Prize Winner, MSU Olympiad in Physics Prize Winner, BMSTU Olympiad in Physics Prize Winner, Physics EGE 100 points.

Skills

Languages: C/C++, NASM, Python, LaTeX, DOT
Tools: Git/GitHub, Linux, VS Code, Make, GDB, Kcachegrind, perf, hotspot, IDA, SDL, SFML

Projects

CPU [GitHub] | *C/C++, SDL* Oct. 2023

- An implementation of my own simplified virtual machine that simulates the work of a single processor unit. The project is split into three seperate programs: an assembler, a disassembler, and the machine itself. The assembly language is made by me. As an additional task the ASCII represantation of the famous Bad Apple video was ran on my CPU using it's RAM as video memory.

Cache-friendly doubly linked list [GitHub] | *C/C++, DOT* Nov. 2023

- A cache friendly doubly linked list is a data structure based on nodes. In a cache friendly doubly linked list, each node contains pointers to the previous and next nodes in the list, allowing for efficient traversal in both directions. Additionally, the nodes are arranged contiguously in memory, minimizing the number of cache misses when accessing consecutive nodes. Also the canonical version of the linked list was made. For better debugging experience DOT language was used to dump images of the current state of the linked list.

Programming language [GitHub] | *C/C++, DOT* Feb. 2024

- My Scottish programming language. It consists of a Tokenizer, Parser & Compiler. The Tokenizer reads the source file and creates token. The Parser uses recursive descent algorithm to build an Abstract Syntax Tree. The Compiler then analyzes the AST and translates it into instructions of my assembly language.

Mandelbrot Set [GitHub] | *C/C++, SFML* Mar. 2024

- Visualization of the Mandelbrot fractal. SIMD instructions were used to optimize image processing.

Hash Table optimization [GitHub] | *C/C++, Python, NASM, GCC inline assembly, perf, hotspot* Apr. 2024

- Hash Table data structure implementation using my linked list. This project is split into 2 parts: Hash functions analysis & Hash Table optimization. Histograms of different hash distributions were made using matplotlib in python. Further optimizations were made using GCC inline assembly, assembly written function & intrinsics that were analyzed using hotspot & perf.

Soft skills

- Adaptability, self motivation, attention to detail, verbal communication, emotional intelligence, work ethic, resilience, humility.