Zhitnikov
Andrey

CONTACT

8 (926) 191-37-08

anzhiday@gmail.com

andrushechka37

SKILLS

Languages:

- C
- x86 Assembler

Instruments:

- Git
- Perf
- IDA
- Make
- Graphviz

SOFT SKILLS

- Hard-working
- Active
- Sociable
- Creative
- Responsible
- Punctual
- Reliable

EDUCATION

- Moscow Institute of Physics and Technology (MIPT) *2023 - now*

Bachelor of Applied Math & Physics, first-year
Department of Radio Engineering and Computer Technology
GPA(general) 6,85/10
GPA(informatics) 6,50/10

- Systems Programming Course by Ilya Dedinsky *2023-2024*

First semester: 3/10
Second semester: 10/10

MAIN PROJECTS

HashTable *04.2024*

The hash table based on my cache-friendly doubly-linked list with fast O(n) linearization. The hash table's performance is 7 times faster than base version through profiling and optimization. This optimization involved utilizing assembly language and SIMD vectorization, guided by researching various hash functions, identifying performance bottlenecks, and determining the optimal load factor for the hash table.

CringeLanguage *12.2023*

This is a compiler for an original, programming language designed for my custom stack-based virtual machine. The compiler follows a traditional architecture, incorporating a lexer, parser, code generator and backend.

X86 Backend *05.2024*

Backend translates the abstract syntax tree (AST) of my language into x86 assembly code. Utilizing an intermediate representation (IR), the backend facilitates code optimization. The performance is 3 times faster compared to a virtual CPU.

Mandelbrot *03.2024*

Visualizes the Mandelbrot set using the SFML library. The comparison of performance under different optimizations, specifically loop-unrolling and the use of SIMD instructions. The performance is 7.5 times faster than a default version.

WolframOmega *11.2023*

The program takes a mathematical expression as input and computes its derivative. It uses a recursive descent parser to build an abstract syntax tree (AST) of the expression, and then builds a new AST for the derivative. Performs some simplifications, such as removing unnecessary elements. It can render the original expression and its derivative in LaTeX format. Uses Graphviz to generate easily inspect-able trees from mathematical expressions.

DosProjects *02.2024*

- First project is a debugging tool that displays hardware registers in real time as they change upon usage by active program. It fakes preemptive multitasking on DOS by taking control of asynchronous system timer interrupt. It also responds to key bindings by hooking to keyboard interrupt.

- The goal of the second project was to identify vulnerabilities in the program, exploit them, and access data without knowing the password. This task involved reverse engineering of an executable file using IDA disassembler.