Stemporzhetskaia Ludmila

Email:

GitHub

GPA

📞 +7 (909) 432-90-83

✉ stemporzhetskaia.lv@phystech.edu

✉ StemporzheckayaLudmila@yandex.ru

 ludmilastemp

8.4

EDUCATION

Advanced Educational Scientific Center (faculty) – Kolmogorov's school of Moscow State University

Moscow, 2021 - 2023

Moscow Institute of Physics and Technology

Dolgoprudny, 2023 - now (2nd year)

WORK EXPERIENCE

"Huawei Technologies Co.,Ltd". MPOL (Moscow Processor Optimization Lab)

Moscow, July 2024 - December 2024

Maintenance LLVM - BOLT.

COURSES

Concurrency

2025, February 2025 - June 2025

Synchronization primitives for threads (mutex, spinlock, wait group, semaphore, condition variable). Thread pool, coroutine, future, fiber. Synchronization primitives for fibers (event, mutex, spinlock, wait group). The memory model.

Architecture of computers and operating systems

2024, September 2024 - January 2025

Representation of data in a computer. Files, file system. x86-64, ARM assembler. Memory in the OS. Processes, threads. Synchronization and interprocess communication.

Algorithm development and analysis

2024, February 2024 - June 2024

Data structures: stack, queue, binary/binomial/AVL/splay/B/red-black trees, segment tree/fanfiction, hash table, binary/binomial/Fibonacci heaps.

Graph algorithms and dynamic programming

2024, September 2024 - January 2025

Dynamic programming, DP by profile. The Strassen algorithm. Convex hull trick. The count. Kruskal's algorithm. The gamma algorithm. The Prima algorithm. The Borovki algorithm. The Faraday-Colton and Bender algorithm. Dijkstra's Algorithm. The Ford-Bellman algorithm. The Floyd-Warshall algorithm. Algorithms A* and IDA*.

PROJECT

Photoshop   *https://github.com/ludmilastemp/Photoshop*

2024, October

Graphic editor.

– SFML Graphics Library
– The MVC (Model-View-Controller) concept
– Plugin system

Draw Sphere   *https://github.com/ludmilastemp/DrawSphere*

2024, September

Rendering of a 3D sphere. Multiple light sources. Ray casting rendering method.

– SFML Graphics Library
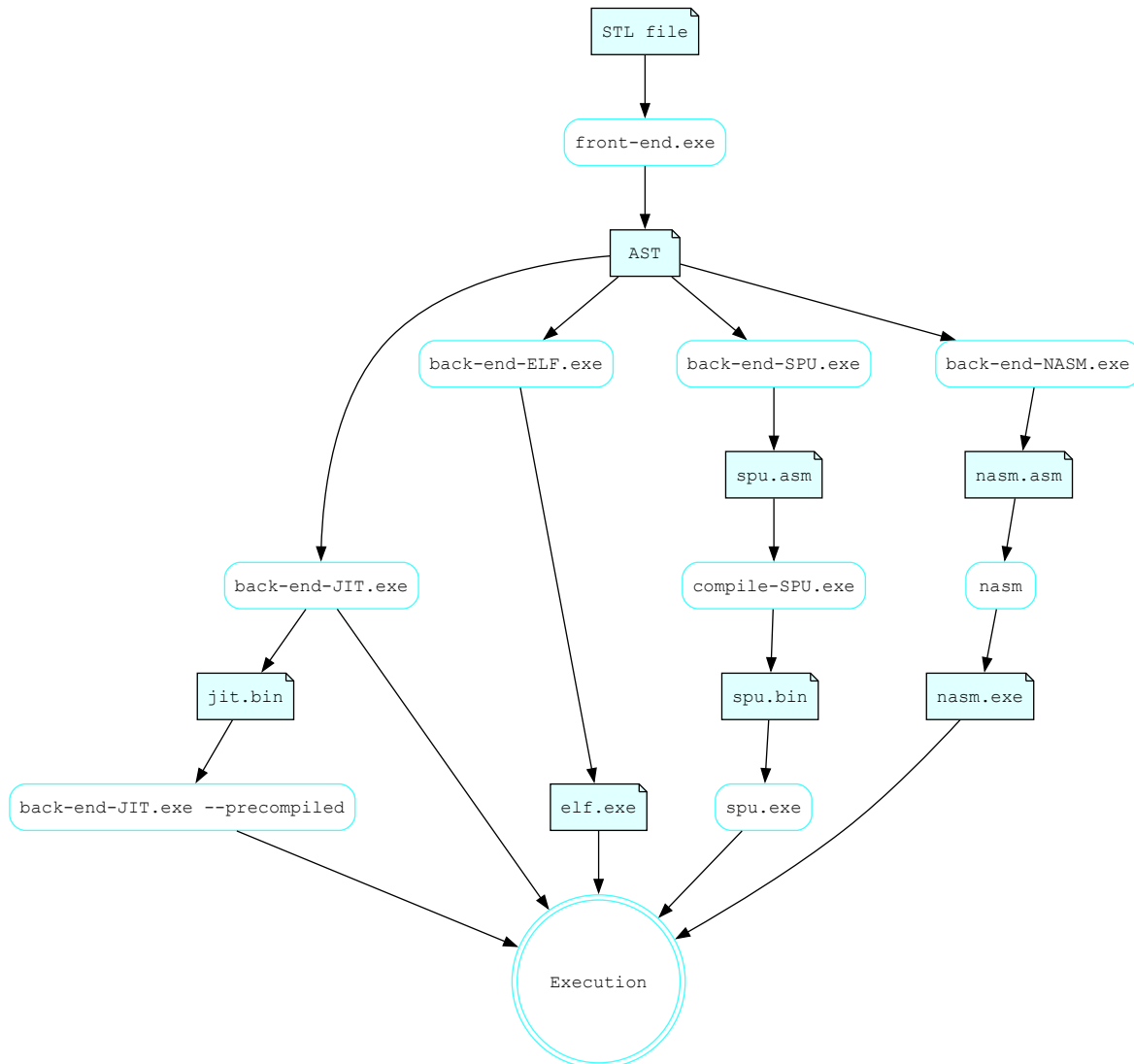– Rendering methods
– Color mixing

STL language

2023, December - 2024, May

The software infrastructure of the STL programming language. Front-end: lexer, recursive descent-based parser.
4 different back-ends:

· *back-end-ELF* (from the AST tree to the ELF executable file)
· *back-end-JIT* (JIT (Just-In-Time) AST tree to x86 binary file)
· *back-end-NASM* (from the AST tree to a NASM language file)
· *back-end-SPU* (from the AST tree to a file in the language for STL-SPA)

```
                              ┌──────────┐
                              │ STL file │
                              └──────────┘
                                   │
                                   ▼
                           ┌───────────────┐
                           │ front-end.exe │
                           └───────────────┘
                                   │
                                   ▼
                                ┌─────┐
                                │ AST │
                                └─────┘
```

- Front-end/Back-end of the compiler
- Encoding of x86 instructions
- The structure of the ELF executable file
- JIT compilation

HashTable

2024, April

Hash table (chain method). The study of various hash functions and their load factors. Implementation of 3 optimization methods: assembler function, assembler insertion, SIMD instructions. The overall acceleration was +300%.

- The data structure is a hash table
- SIMD vector instructions
- Inline Assembly
- Linux perf utility

Mandelbrot

https://github.com/ludmilastemp/Mandelbrot

2024, March

Drawing the Mandelbrot set. The study of SIMD vector instructions (AVX, SSE extensions) and program acceleration using them. 4 iterations were performed, after which an acceleration of 3.5 times was obtained.

– SIMD vector instructions
– Graphics Library txlib
– Linux perf utility

Wolfram

https://github.com/ludmilastemp/Wolfram

2023, November

A program for working with mathematical expressions. Processing an expression, simplifying it, and calculating the first and second derivatives. All intermediate results with a detailed description are written to a file in LaTeX format.

– Expression tree
– Dot, Graphviz
– LaTeX

Akinator

https://github.com/ludmilastemp/Akinator

2023, November

The Akinator game. The essence of the game is to guess the hidden object by the program using simple questions (yes/ no answers). If the game does not know such an object yet, it is suggested to add it to the database. You can also request a description of the object. All data is stored in a binary tree.

– The data structure is a binary tree
– Dot, Graphviz

SPU

https://github.com/ludmilastemp/CPU

2023, October

Stack VM and compiler. An assembly language text file is processed, lexical errors are searched, and labels are placed. All commands are encoded and a file with executable bytecode is obtained.

– Assembler/disassembler
– Encoding instructions, ISA (instruction set architecture)
– Working with labels (fixups)
– Working with text and binary files