Trunov Vladimir

📫 trunov.vv@phystech.edu ☎ +7(902) 366-0322 GitHub: git

Education

MIPT, Dolgoprudny **Dolgoprudny, Russia**

Radio engineering and computer technology, Applied Mathematics and Physics 2021–2025

GPA (general) 7.29/10

Coursework:

- System programming from ISP RAS and Huawei
- C++ course (Second year ISR RAS and Huawei course)
- LLVM-practise course from Huawei

Internships: Compilers Department ISP RAS 07.2022-08.2022

Job experience: 09.2022-present ISP RAS

Internships

Compilers Department ISP RAS **July 2022- August 2022**

Job experience

Compilers Department ISP RAS **September 2022 - present**

My work is related to writing C++ code checkers for various guidelines using Clang Static Analyzer and Clang Tidy.

C projects

Software CPU **October, 2021**

The software processor is implemented in a language based on its own stack (2d task). It supports working with the stack, RAM and registers. The project was written at the very beginning of the course (3rd task).

Cache-friendly List **November, 2021**

Cache-friendly List is built using an array and has a linearization function, all this ensures maximum accessibility to caching. In this project, Dot was used for the first time to dump using graphviz.

Programming language **December, 2021**

This project was the last one in the first semester. It uses: a binary tree, a stack, and a software processor. Recursive descent and prefix tree structure were implemented.

Binary Translator **May, 2022**

The binary translator translates the binary code of my software processor into the binary code of the x86-64 architecture processor. Support for all functions of the software processor (including input, output of numbers) has been preserved.

System programming repository **September-December, 2022**

This is a training repository for the system programming course in the 3rd semester of MIPT. The repository includes the execution of tasks related to multithreading and using the Linux API

C + Asm projects

Mandelbrot-Set & Alpha Blending **April, 2022**

- A program that draws the Mandelbrot set using the SFML graphics library. The performance of different versions was compared, including various AVX instructions.
- Implementation of alpha blending with and without SSE instructions, performance has been investigated. The SFML graphic library was used for visualization.

Hash-table **April, 2022**

The hash table is implemented based on my Cache-friendly List. Various hash functions were investigated and optimizations were carried out, including writing functions in assembly language and using AVX instructions.

Pure Assembly projects

Printf asm version **March, 2022**

An analog of the Printf() library function was written. All specifiers are supported, and there is also support for an additional specifier - %b, which prints a number in binary representation.

Resident regs viewer **March, 2022**

The program is written for the DOS system on tasm. With the help of interrupt interception, real-time display of processor registers in a frame is implemented.

C++ projects

Ray-Tracing **September, 2022**

Repository has 2 branches with ray-casting and ray-tracing (currently doesn't work properly). The calculation includes direct lighting and diffuse (examples can be found in the repository README), an interactive light source is implemented.

PaintBucket visualizer **March, 2023**

The repository is a minimal analogue of programs like Paint. The main feature is the pixel-by-pixel rendering of the fill tool. Thanks to this, you can visualize and analyze the execution speed of a specific fill algorithm.

My own STL structures **March, 2023**

The repository includes several structures that are an alternative to the standard containers presented in the STL. The repository is educational, so there is no semantic connection of some parts. Currently there is a version of std in the repository::vector and the unusual use of templates that do not belong to the STL.

LLVM Projects

LLVM Callgraph **March, 2023**

The repository is a fork from the official LLVM repository, which implements a self-written tool for collecting Callgraph, the construction is carried out using graphviz. The program can work both statically and dynamically (edges in the graph gain weights). Multi-module projects are supported for more details, see the README

Programming Language with LLVM **October, 2021**

Currently, the project is still in development, it will not be completed until May. The programming language is an interface for further compilation in LLVM IR. Closer integration with proprietary tools written using LLVM infrastructure will also be added.

Skills

Programming languages: C/C++, x86-64 Assembly, LaTex

Knowledge: Data structures and algorithms, Concurrency paradigms, LLVM, Clang Static Analyzer, Clang-Tidy, Linux

Tools: make, CMake, gdb, git/github/gitlab, vim/nvim, perf

Personal qualities: Quick involvement in work, the ability to study theory for a long time, sociability