



DENIS MELNIKOV

+7 931 511 58 76

melnikov.denis@phystech.edu

[@daikofnetu](#)

[Dinichthys](#)

[Moscow, Russia](#)

EDUCATION

Bachelor | *First course, GPA: 8.43*

Sep. 2024 – Present

Phystech School of Radio Engineering and Computer Technology

Moscow Institute of Physics and Technology

Dolgoprudny, Russia

PROJECTS

Programming Language | *C, make*

Language on GitHub

Project Goal: Implement a compiler for a custom language with its own syntax, targeting a custom architecture. The front-end module uses a recursive descent parser to analyze the program text and construct an AST (Abstract Syntax Tree). Subsequently, the back-end module translates the AST into assembly code. Finally, the assembly code is converted into binary representation, which is executed by the processor from the SPU project.

SPU | *C, make*

SPU on GitHub

Project Goal: Implement a functional processor simulator with three components:

- **Assembler:** Translates assembly code to binary representation.
- **Disassembler:** Reverts binary representation to assembly code.
- **Processor:** Executes the binary instructions.

Printf | *Assembly (NASM), C, make*

Printf on GitHub

Project Goal: Implement a simplified version of the C function **printf** in assembly, supporting the following format specifiers: %c, %s, %d, %b, %o, %x, and %f. The parser uses a **JUMP-table** to efficiently handle specifier letters.

Additional Features:

- **File output:** Writing formatted text to files
- **Color support:**
 - * For console output: Insertion ANSI escape sequences for colored text
 - * For file output: Usage of HTML-style color tags (e.g.,)
- **Floating-point support:** Handles %f specifier for floating-point numbers

Mandelbrot Set | *C, CMake*

Mandelbrot Set on GitHub

Project Goal: Try to optimize the work of the program, which draws Mandelbrot Set and makes calculations for every point on the screen.

There are three methods for optimization:

- **Arrays:** Usage of arrays in a specific way, so the compiler translates the operations with it to SIMD instructions.
- **SIMD Instructions:** Usage of intrinsics will optimize the process because one operation will be used for several data at one moment.
- **Loop unrolling:** Loop unrolling helps to execute data-independent instructions in a parallel way, because the pipeline of the processor will be full filled of instructions, which won't follow to **bubble** and **stall**.

Stack | *C, make*

Stack on GitHub

Project Goal: Write a data structure **stack** with several levels of protections:

- **Hash Protection:** This level of protection will count a hash of stack's structure and it's elements and compare it with the correct value.
- **Canary Protection:** This level of protection will add the flag-elements before and after stack's structure and array of it's elements to protect it from vulnerabilities such as buffer overflow.

- **Encapsulation:** This level of protection will oblige the user to work with stack only with functions to exclude the possibility of changing the value of stack elements outside of functions.

CrackMe | *Assembly (TASM), C, make*

CrackMe on GitHub

Project Goal: Write a program with two vulnerabilities, exchange it with another student and try to crack his program. Also there was a research and comparison of three disassemblers based on the program 'The Bomb Lab':

- IDA
- Radare2
- Ghidra

ACHIEVEMENTS FOR THE PREVIOUS YEAR

Open School Olympiad The winner of the Math Olympiad from ITMO university.	2024
Interregional Olympiad of schoolchildren named after I.Ya. Verchenko Medalist of the Math Olympiad	2024
The Formula of Unity Medalist of the Physics Olympiad	2024
Phystech School Olympiad Medalist of the Math Olympiad	2024
Rosatom Branch Physics and Mathematics Olympiad for Schoolchildren Medalist of the Math Olympiad	2024
Rosatom Branch Physics and Mathematics Olympiad for Schoolchildren Medalist of the Physics Olympiad	2024
Regional stage of the All-Russian School Olympiad Medalist of the Informatics Olympiad	2024
Regional stage of the All-Russian School Olympiad Medalist of the Math Olympiad	2024
Regional stage of the All-Russian School Olympiad Medalist of the Physics Olympiad	2024

INTERESTS

Profilers
Architectures CPU/GPU
Compilers

HARD SKILLS

Programming Languages: C, Assembly (NASM, TASM, x86-64), Python
Tools: CMake, Make, IDA, Ghidra, Radare2, edb, gdb
Document Creation: Microsoft Office, LaTeX, Markdown

LANGUAGES

Russian: Native
English: B1

SOFT SKILLS

Communication skills
Self-confidence
Responsibility