LAVRUSHCHEV IVAN

Student ~ Developer

github.com/livlavr
gitlab.com/livlavr
Moscow, Russia
lavrushchev.iv@phystech.edu
+7 (915) 693 01 18
livlavr.github.io/LavrushchevMIPT.io

DEVELOPER

Results-driven MIPT student and C developer with knowledge of x86-64 assembly and basics of Python. I'm focused on **improving my C++** programming skills and expanding my computer science knowledge. I am particularly interested in projects involving **optimizations**, **OS internals** and **artificial intelligence**.

SKILLS

Tools: Git, Make/CMake, Linux, Radare2, Latex, Graphviz, Doxygen

Libraries: SFML, GTK

Languages: C, Bash, x86-64 assembly (NASM), ARM64, Python

EDUCATION

Present — **Moscow Institute of Physics and Technology** — **University | Moscow, Russia**
System programming and applied mathematics (undergraduate)

PROJECTS

2025 in process — **Mandelbrot Set** — **github.com/livlavr/MandelbrotSet**
Lab work involving comparison of different types of CPU rendering, testing with the *Mandelbrot set*. Three versions were compared: basic, with **compiler optimizations**, and with direct **Arm Neon Intrinsics**.
`C/C++ / CMake / SFML / SIMD`

2025 finished — **Printf** — **github.com/livlavr/Printf**
An educational project implementing the printf function from the standard C library, supporting specifications: *%(b|c|d|o|s|u|x)*. **The aim** of this project was to work with **calling conventions**, use the **stack frame** and **function arguments** in x86-64 assembly, and **identify security vulnerabilities** of the standard printf function.
`x86-64 assembly / NASM`

2024 – 2025 in process — **Programming Language** — **github.com/livlavr/Language**
Implementation of my own compiled programming language. The code is first processed by **lexical analysis**, then by syntax analysis using **recursive descent**, translated into a syntax tree with a name table, and finally into an assembly file (pseudo-assembly for the **CPU Emulator**). This assembly file is processed by my own SPU. I plan to compile the program into full-fledged x86-64 assembly. The project supports cross-compilation with the SPU and languages of other group students, thanks to a common AST standard.
`C/C++ / Make / Graphviz`

2024 finished — **CPU Emulator (SPU)** — **github.com/livlavr/SPU**
A virtual machine emulating the interaction between the CPU and assembly instructions (**Harvard architecture**). The project consists of two parts: an **assembler for pseudo-assembly**, which translates assembly instructions into byte code, and a **processor emulator** that processes this byte code. It also supports basic video memory operations.
`C / Make`

ACCOLADES

2025 — **Rosatom Industry Mathematics Olympiad**
1st degree diploma

2024 — **Computer Security I.Y. Verchenko Interregional Olympiad**
2st degree diploma

LANGUAGES

English - B2, **Russian** - native