PAVEL SHUMILO

Software Developer and Researcher

@ shumilo.pd@phystech.edu   📞 +79673513157   📍 Dolgoprudny, Russia   Pave2005

in pavel05sh   🌐 vk.com/pavel05sh

EXPERIENCE

Software Engineer and Researcher
Institute for System Programming RAS

📅 June 2024 – Sept. 2024   📍 Moscow, Russia

- Implemented detectors for the static analyzer Svace (SharpChecker)
- Developed a tool for error detection in code using LLM, based on summaries generated by LLM from summaries created during analysis with Svace

EDUCATION

Department of Radio Engineering and Cybernetics
Moscow Institute of Physics and Technology (MIPT)

📅 Sept. 2023 – Ongoing   📍 Dolgoprudny, Russia

- Major in Mathematics and Computer Science
- **GPA (General)** 8.28/10   **GPA (IT)** 8/10   **GPA (Math)** 8.13/10

System Programming and Compiler Technologies Course
MIPT, Huawei, ISP RAS

📅 Sept. 2023 – June 2024   📍 Dolgoprudny, Russia

Introduction to Industrial Programming in C++ Course
MIPT, YADRO

📅 Sept. 2024 – Ongoing   📍 Dolgoprudny, Russia

Computer Architecture Course
MIPT, Sber

📅 Jan. 2025 – Ongoing   📍 Dolgoprudny, Russia

TECH SKILLS

Algorithms and Data Structures	Math/Linear Algebra
Probability and Discrete analysis	Processor Architecture
Optimizations	Multithreading
Physics and Theoretical mechanics	

SOFTWARE SKILLS

C/C++   Assembly   Python   C#   Google Test
Google Benchmark   Linux   Git   OpenCL   SFML   SDL
CMake/Make   Perf   IDA   Bash   DOT   Doxygen
Graphviz

PROJECTS

Hash Table

- A hash table optimized using inline assembly, C intrinsic functions, and external assembly functions. To improve performance, loop unrolling was applied, which also helped reduce execution time.
- C, Assembly, Perf
- [github.com/hash_table]

AVL Tree

- Implementation of a balanced AVL search tree with a function to calculate the distance between any two nodes. This implementation outperforms the standard std::set data structure in terms of performance.
- C++, CMake, Google Test
- [github.com/AVL]

Binary Translator

- This program converts the binary representation for **the processor emulator** into instruction sets compatible with the Intel x86 architecture. The conversion result is stored in a designated location within the main memory. Additionally, the -S command can be used to view the corresponding x86-64 Assembly code. The program employs double-precision arithmetic and utilizes AVX-256 instructions.
- C, Assembly, Linux
- [github.com/binary_translator]

SOFT SKILLS

Sociable   Purposeful   Hard-working

LANGUAGES

Russian (native)   ●●●●●
English (B2)   ●●●●○