

# Мерзляков Арсений

 <https://github.com/ogkisque> |  [merzliakov.ad@phystech.edu](mailto:merzliakov.ad@phystech.edu) |  [https://t.me/og\\_kisque](https://t.me/og_kisque) |  +7 902 864 54 72



## ОБРАЗОВАНИЕ

---

- МБОУ ФМЛ №31 г. Челябинска (4-5)
- МФТИ ФРКТ, 1 курс (средний балл 9)
- Курс Дединского И. Р. "Введение в эмуляцию вычислительных систем, компиляторные технологии и промышленное программирование" (10)

## НАВЫКИ

---

- Языки программирования: C, Assembly, Python
- Прочее: make, git, latex, callgrind, edb, dot, SDL2
- Языки: русский, английский (B1-B2)

## ПРОЕКТЫ

---

- **Стэк** (<https://github.com/ogkisque/Stack>)

Реализация структуры стек с защитой канарейками и хэшированием. Также поддерживается создание целочисленных и вещественных стеков в одной программе (через кодогенерацию, дженерики)

- **Виртуальный процессор** (<https://github.com/ogkisque/Processor>)

Процессор, исполняющий программы на "своём" ассемблере. Состоит из ассемблера (генерирует байт-код из программы на ассемблере), дизассемблера (генерирует программу на ассемблере из байт-кода) и процессора (исполняет байт-код). Также реализована графика и отображение через неё картинок формата ppm.

- **Дифференциатор** (<https://github.com/ogkisque/Differenciator>)

Программа, считающая производную функции. Состоит из парсера выражения в дерево, его упрощения и расчёта. Считается производная, разложение по формуле Тейлора до заданного порядка. В случае функции нескольких переменных можно считать частные производные и вычислять полный дифференциал. Результат вместе с графиками функции, её разложения, проведенной касательной печатается в виде отчета в latex.

- **Множество Мандельброта** (<https://github.com/ogkisque/Mandelbrot>)

Отрисовка множества Мандельброта через SDL2 и её оптимизация через SIMD инструкции. Измерение производительности через rdtsc.

- **Хэш-таблица** (<https://github.com/ogkisque/Hash-Table>)

Реализация хэш-таблицы. Исследование эффективности различных хэш-функций. Оптимизация поиска по хэш-таблице через ассемблерную вставку, ассемблерную функцию и SIMD инструкции. Измерение производительности и поиск "горячих точек" через callgrind.

- **Язык программирования** (<https://github.com/ogkisque/My-language>)

Написание своего языка программирования. Состоит из 3 этапов - middle-end (генерация AST дерева), middle-end (упрощение дерева) и back-end (генерация исполняемого файла). Последний реализован в 3 вариантах: ассемблер для виртуального процессора, elf файл (вместе с nasm) и LLVM IR формат.

## ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ

---

- Социальные навыки: ответственность, трудолюбие, быстрая обучаемость
- Достижения: международная Жаутыковская олимпиада по физике (бронза), призер и победитель различных олимпиад школьников по физике и информатике (всероссийская, физтех, росатом)
- Увлечения: футбол, музыка

Программированием увлекаюсь с 5 класса. Попробовал разные языки и области: сначала Scratch, Pascal, затем Java (+ под Android), Javascript + html + css, Python, C++. На первом курсе C.