

## 1. Ты уже выбрал тему для диплома?

Сейчас я работаю над темой: «Программная реализация параллельных методов решения задач глобальной оптимизации».

Я выбирал между компьютерной графикой, суперкомпьютерным моделированием, разработкой под RISC-5 и выбранной темой. Я сделал выбор в пользу более общей темы, которая имеет место быть в разных областях, ведь сейчас в НРС никак без параллельных вычислений. Причём используются параллелизм разных уровней: параллелизм уровня инструкций (конвейер), параллелизм на уровне потоков (multithreading), параллелизм на уровне данных (SIMD), параллелизм на уровне узлов (MPI). Поэтому тема на мой взгляд является актуальной.

Сейчас я официально реализовал оптимизированную, но не параллельную версию эвристического алгоритма поиска глобального экстремума. Однако у меня есть версия, использующая OpenMP.

## 2. Почему ты выбрал стажировку в разработке, а не, например, в тестировании?

Я выбрал C++, потому что знаю его лучше всего. По сути, три года в университете занимался его изучением, даже был курс под названием «Профессиональный C++». Его концепция мне ближе всего. А в целом разработка мне кажется интереснее, чем тестирование.

## 3. Проходил ли ты какие-нибудь курсы или читал книги по программированию вне рамок университета? Если да, то какие.

Я занимался олимпиадным программированием с Алексеем Шмелёвым, это в университете, но необязательная дисциплина. На занятиях читались лекции, а дома нужно было решать тренировки на codeforces.

Смотрел курс лекций по Архитектурам ЭВМ на youtube, поскольку не хватало понимания по лекциям в университете:

<https://www.youtube.com/playlist?list=PLnseyzyGdZdfv8H7LkvyVVE33fbBZaSdH>

Совсем немного читал «Совершенный код» от Стива Макконела (книга разделена на части, я читал первую)

Также читал книгу «Практическая статистика для специалистов Data Science» от Питера Брюса, Эндрю Брюса и Питера Гедека, но меня больше интересовала мат. часть.

## 4. Как ты оцениваешь свои знания по C++ в следующих областях?

Области:

- Объектно-ориентированное программирование (ООП) – уверенно использую
- Многопоточность (multithreading) – знаком, но мало использовал
- Шаблоны (templates) – уверенно использую
- Стандартная библиотека (STL) – уверенно использую
- Управление памятью (memory management) – уверенно использую
- Побитовые операции (bitwise operations) – знаком, но мало использовал

## 5. Чем Python отличается от C++?

В первую очередь Python – интерпретируемый язык, а C++ – компилируемый. Это означает, что выполнение программы Python происходит непосредственно во время работы, виртуальная машины переводит инструкции программы в низкоуровневые команды. Это не очень эффективно поскольку появляется ещё один уровень абстракции между программой и аппаратным обеспечением. Но плюсы у этого тоже есть, программы python являются кроссплатформенными. Компиляция в свою очередь переводит исходный код в машинный, создавая исполняемый файл, который может быть выполнен непосредственно на процессоре без дополнительной интерпретации. Если вдаваться в подробности, то преобразование src файла выглядит следующим образом:

Препроцессор: source.cpp -> source.i

Компилятор: source.i -> source.s

Ассемблер: source.s -> source.o

Компоновщик: source.o -> source

Если файлом будет несколько, то компоновщик соберёт их в один исполняемый. Однако до этапа компоновки каждый исходный файл транслируется отдельно. То есть файл `spp` является единицей трансляции.

Это их самое главное отличие, из которого следуют области использования каждого языка, `c++` там, где нужно быстро, `python` – где нужно удобно.

Есть ещё более тонкие отличия, связанные с управлением памятью, ООП и т. д., но это слишком большой объём текста получится.

#### **6. Какой самый сложный и/или интересный проект ты писал на C++?**

Всё-таки данная лабораторная является моим самым сложным проектом, поскольку он самый большой, здесь используется и ООП, и шаблоны, и алгоритмы, и структуры данных.

[https://github.com/Arseny-Korobeinikov/mp2-practice/tree/05\\_lab](https://github.com/Arseny-Korobeinikov/mp2-practice/tree/05_lab)

#### **7. Как ты оцениваешь свои знания английского языка?**

Я не занимался английским с репетитором, но в связи с тем, что часто приходится читать текст на английском, то навык вырабатывается. Читать то, что на сайте могу, при случае незнакомого слова просто вбиваю его в переводчик.

#### **8. Работал ли ты с Linux?**

С Linux по большей части работал в рамках IT-lab, работал на Ubuntu. Писал два похожих приложения, пытаясь в одном из них добиться векторизации и использования FMA.

Кроме того, так же, как и все, работал на виртуальной машине Linux 2.4 (32 bit), в рамках курсов "Архитектура ЭВМ", "Операционные системы" и "Компьютерные сети". Работал в терминале, без использования графического интерфейса.

#### **9. Как оцениваешь свои знания по алгоритмам?**

Вычислять Big O умею, прошел специальный курс в университете, на котором занимались этим ("Алгоритмы и Анализ сложности"). Ну и так это повсеместно обсуждается при работе с алгоритмами. На том же олимпиадном программировании стремятся к  $O(n)$  или  $O(n \log n)$ .