

После прослушивания лекций у вас будет возможность пройти тестирование для участия в проектной работе. Прошедшие отбор поедут в выбранный вуз для создания проекта. Ниже перечислены темы проектов и университеты, в которые можно поехать.

В результате вы получите опыт разработки и оптимизации ПО под открытую архитектуру RISC-V. А также полноценный реализованный проект, который можно положить в портфолио. В процессе вам будут помогать опытные преподаватели ведущих технических вузов.

Можно выбрать город для участия

Нижний Новгород, ННГУ
Санкт-Петербург, СПбГУ
Новосибирск, НГУ

Дизассемблирование стартового кода платы Hifive Unmatched

Участникам предлагается попробовать себя в роли реверсера, научиться видеть в машинных кодах знакомые конструкции языка C, узнать как загружается многоядерная система, проинициализировать периферийные устройства в режиме bare metal, а также познакомиться с экзотическими инструментами. Есть здесь место и для программирования, с возможностью отправить решение в Open Source.

<p>Требующиеся знания</p> <p>Навык работать со справочником SnK, спецификацией RISC-V</p>	<p>Требующиеся навыки</p> <p>Умение читать код Assembler</p>
--	---

1

Нижний Новгород, ННГУ
Санкт-Петербург, СПбГУ

Использование внешней базы значений для формирования начальных условий в llvm-snipru

Есть классы инструкций, чувствительные к начальным значениям операндов. Например, для RISC-V это инструкции F/D расширений. Случайный выбор входных значений может быть недостаточен. Довольно эффективные дата-флоу подходы могут генерировать начальные значения с помощью символического исполнения и формальных методов. Нужно расширить RISC-V-бэкэнд для llvm-snipru с возможностью брать значения операций и ассоциированный контекст из базы данных, подаваемой на вход.

<p>Требующиеся знания</p> <p>Навык C++, CMake, Git, sqLite Инфраструктуры LLVM, Python, html</p>	<p>Требующиеся навыки</p> <p>Git Техника юнит-тестирования Базовые навыки CI/CD</p>
---	--

2

Санкт-Петербург, СПбГУ
Новосибирск, НГУ

Benchmark для процессора на архитектуре RISC-V

Проект направлен на разработку бенчмарка для тестирования производительности процессоров на архитектуре RISC-V. Бенчмарк будет включать в себя набор тестов, которые охватывают различные аспекты производительности, такие как целочисленные операции, операции с плавающей точкой, векторные вычисления и другие. Оценка фактических задержек при обращении к оперативной памяти компьютера и зашированным данным в процессоре.

<p>Требующиеся знания</p> <p>gcc/clang stake, performance profiler Архитектура ЭВМ</p>	<p>Требующиеся навыки</p> <p>Git Техника юнит-тестирования Базовые навыки CI/CD</p>
---	--

3

Санкт-Петербург, СПбГУ
Новосибирск, НГУ

Bootstrap RISC-V

Часть open-source сообщества озадачена вопросами создания полностью раскручиваемых систем (bootstrappable builds). Значимым результатом этих усилий является проект live-bootstrap — pipeline-сборки Linux-системы из минимальных сидов в изолирован окружении. Сейчас live-bootstrap работает только на архитектуре x86. Нужно исследовать возможность портирования используемого им загрузчика builder-hex0 на открытую архитектуру RISC-V и протестировать своё решение на реальном железе.

<p>Требующиеся знания</p> <p>gcc/clang stake, performance profiler Архитектура ЭВМ</p>	<p>Требующиеся навыки</p> <p>Git Техника юнит-тестирования Базовые навыки CI/CD</p>
---	--

4

Санкт-Петербург, СПбГУ
Новосибирск, НГУ

Watermark RISC-V

Цель проекта состоит в создании метода стеганографии для обнаружения источников утечки приватного программного обеспечения, написанного на RISC-V. Для достижения поставленной цели поставлена задача создать инструмент для сокрытия информации в исполняемом коде, используя функционально эквивалентные инструкции языка Assembler RISC-V.

<p>Требующиеся знания</p> <p>gcc/clang stake, performance profiler Архитектура ЭВМ</p>	<p>Требующиеся навыки</p> <p>Git Техника юнит-тестирования Базовые навыки CI/CD</p>
---	--

5

Санкт-Петербург, СПбГУ
Минск, БГУИР

Оптимизация программного декодера JPEG-формата для архитектуры RISC-V

В рамках проекта предлагается выполнить оптимизацию программной реализации декодера стандарта JPEG. Описать и оптимизировать для архитектуры RISC-V алгоритм быстрого дискретного косинусного преобразования. Встроить оптимизированный алгоритм в программную реализацию JPEG-декодера, протестировать и выполнить оценку производительности декодера на отладочной плате с RISC-V.

<p>Требующиеся знания</p> <p>Базовое знание формата JPEG</p>	<p>Требующиеся навыки</p> <p>C/C++, CMake, ASM Отладка программ</p>
---	--

6

Санкт-Петербург, СПбГУ
Минск, БГУИР

Реализация программного генератора случайных числовых последовательностей с использованием SRAM/FLASH PUF на RISC-V

В рамках проекта необходимо реализовать программный генератор случайной числовой последовательности с использованием SRAM/FLASH PUF на отладочной плате с RISC-V. Обеспечить инфраструктуру для взаимодействия с ПК. Выполнить программную постобработку исходных данных SRAM/FLASH PUF, оценить быстродействие. Провести оптимизацию кода для RISC-V и оценить характеристики генерируемых числовых последовательностей.

<p>Требующиеся знания</p> <p>Физическая криптография ABC</p>	<p>Требующиеся навыки</p> <p>Владение C/C++, ASM Отладка программ</p>
---	--

7



Нижний Новгород

Нижний Новгород, ННГУ

Оптимизация функций компьютерного зрения на RISC-V

OpenCV — это библиотека с большим количеством готовых алгоритмов компьютерного зрения и функций, необходимых для их реализации. В рамках задачи предлагается оценить ускорение функций OpenCV без и с векторизацией на RISC-V и предложить оптимизацию. Решение участников могут быть приняты в основной код библиотеки и быть использованы миллионами пользователей.

<p>Требующиеся знания</p> <p>Знание C++</p>	<p>Требующиеся навыки</p> <p>Компиляция программ под Linux Опыт работы с многомерными массивами/тензорами</p>
--	--

8

Нижний Новгород, ННГУ

Запуск и оптимизация C# приложений на RISC-V

Поддержка новых архитектур — это вызов для разработчиков продуктов. Язык программирования C# и его инструменты разработки прошли несколько больших итераций, чтобы позволить пользователям компилировать и оптимально исполнять код на архитектуре RISC-V. В рамках задачи предлагается попробовать подготовить кросс-компилятор, запустить "Hello World" и выполнить написание более сложного приложения с возможностью оптимизации на языке C#.

<p>Требующиеся знания</p> <p>Знание ООП</p>	<p>Требующиеся навыки</p> <p>Опыт программирования на C#/Java приветствуется</p>
--	---

9

Нижний Новгород, ННГУ

Оптимизация кода с помощью векторного расширения RISC-V и моделирование производительности на потактовых симуляторах

Оптимизация приложений под новую архитектуру это непростая задача. Мы предлагаем реализовать ряд алгоритмов линейной алгебры и обработок изображений с использованием векторного расширения. Используя потактовый симулятор с генерацией трасс исполнения программ, отображающие работу конвейера процессора, оценить производительность оптимизаций и сравнить с реальными RISC-V платами.

<p>Требующиеся знания</p> <p>Язык C Команды linux cli Знание конвейера процессора</p>	<p>Требующиеся навыки</p> <p>Оптимизирующие компиляторы Параллельный кодинг</p>
--	--

10

Нижний Новгород, ННГУ

Методы оптимизации программ для процессоров RISC-V

В рамках проекта будут рассмотрены различные приемы оптимизации программ для процессоров архитектуры RISC-V, начиная от алгоритмической оптимизации и заканчивая низкоуровневыми преобразованиями кода. Участникам школы будет предложено реализовать алгоритмы паразитной сортировки, базовые алгоритмы линейной алгебры и финансовой математики. Студенты будут работать с алгоритмами, реализуют их на языке C/C++ с последующей оптимизацией кода для устройств с процессорами RISC-V.

<p>Требующиеся знания</p> <p>C и C++ Базовый курс высшей математики</p>	<p>Требующиеся навыки</p> <p>Оптимизирующие компиляторы Параллельный кодинг</p>
--	--

11

Санкт-Петербург

Санкт-Петербург, СПбГУ, ИТМО

Cloud-based визуализатор и инструмент для работы с Kanata трассами

Kanata-формат нужен для демонстрации работы конвейера процессора. Существует стандартный просмотрщик kanata-трасс в виде standalone-приложения, но в таком виде он не очень удобен для обмена трассой с коллегами или когда размер трассы составляет гигабайты. Кроме того, многие стандартные операции с каната трассами никак не интегрированы в текущие инструменты просмотра. Нужно закрыть эти недостатки с помощью web-инструмента, который визуализирует трассы, хранящиеся на сервере.

<p>Требующиеся знания</p> <p>Базовое представление о web-программировании</p>	<p>Требующиеся навыки</p> <p>Опыт работы с бэкенд-фреймворком и WebGL/WebAsm</p>
--	---

12

Санкт-Петербург, СПбГУ

llvm-exegesis — поддержка RVV, работа в open-source

llvm-exegesis — это open-source инструмент на базе LLVM фреймворка, позволяющий измерять точечные характеристики отдельных инструкций (latency/throughput). На текущий момент команда Symbioses добавляет в open-source поддержку RISC-V архитектуры, в 2025 году ожидается базовая поддержка для инструкций векторного расширения RVV. Проект предполагает поддержку измерений для отдельных специфических инструкций (редукций, перестановок).

<p>Требующиеся знания</p> <p>C/C++ Знание инфраструктуры LLVM</p>	<p>Требующиеся навыки</p> <p>Оптимизирующие компиляторы Параллельный кодинг</p>
--	--

13

Санкт-Петербург, ИТМО

Запуск и измерение RVV бенчмарков на open-source RISC-V VPU (Ara2, Saturn)

На сегодняшний день растёт число open-source-impleментаций RISC-V процессоров различной сложности, вплоть до высокопроизводительных ядер, отверифицированных и используемых в индустрии. Некоторые из них реализуют стандартный набор расширений rv32i64dc, к которым можно добавить внешний векторный модуль для RVV расширения (Ara, Ara2, Saturn Vectors, Vicuna, etc).

<p>Требующиеся знания</p> <p>RISC-V ISA Параллельные алгоритмы Микроархитектура ядра</p>	<p>Требующиеся навыки</p> <p>Опыт работы с бэкенд-фреймворком и WebGL/WebAsm</p>
---	---

14

Санкт-Петербург, СПбГУ

Оптимизации для плотных матриц на RISC-V с GPU от Imagination Technologies

Сейчас графические ускорители используются не только для вычислений общего назначения. Достижение максимальной производительности на реальных устройствах требует глубокого знания их архитектуры и написания достаточно специфичного кода. При этом, хотя индустрия OpenCL и подражает переносимости, специфика того или иного вендора также накладывает ограничения на возможные оптимизации и процесс работы с устройством. Нужно изучить эту связь и понять ее сильные и слабые стороны.

<p>Требующиеся знания</p> <p>C Сборка/отладка Профиллирование Архитектура процессора</p>	<p>Требующиеся навыки</p> <p>Читать код (Rust, Python) Поиск и анализ информации</p>
---	---

15

Санкт-Петербург, СПбГУ

Базовые оптимизации для плотных матриц на RISC-V GPGPU Vortex

В современном мире графические ускорители общего назначения используются для вычислений общего назначения. Достижение максимальной производительности на реальных устройствах требует глубокого знания их архитектуры и написания специфичного кода. Архитектура RISC-V набирает популярность. В частности, разрабатывается открытый графический ускоритель Vortex. Нужно изучить особенности архитектуры Vortex, научиться с ним работать, изучить, как текущая версия отзывается на типичные оптимизации.

<p>Требующиеся знания</p> <p>C Сборка/отладка Профиллирование Архитектура процессора</p>	<p>Требующиеся навыки</p> <p>Чтение кода Отладка Базовый shell scripting</p>
---	---

16

Санкт-Петербург, СПбГУ

Бенчмаркинг методов дедупликации на RISC-V

Для существующих методов дедупликации на наборе данных Emgrip предлагается провести запуск на платформах RISC-V, а также выполнить замер производительности с использованием уже имеющихся инструментов, и, при необходимости, выполнить разработку необходимых для бенчмаркинга компонент.

<p>Требующиеся знания</p> <p>Архитектура СХД Задача дедупликации</p>	<p>Требующиеся навыки</p> <p>Читать код (Rust, Python) Поиск и анализ информации</p>
---	---

17

Санкт-Петербург, СПбГУ

Поддержка популярных в кругу любителей плат RISC-V в Yocto/Poky

На сегодняшний день большинство плат с процессорами RISC-V поставляется со своими сборками ОС на основе ядра Linux. Разнообразие версий библиотек и компиляторов затрудняет работу с платами, а зачастую делает просто невозможным или крайне сложным запуск тестов и бенчмарков. В рамках проекта предлагается провести анализ поддержки RISC-V в Yocto. Изучить, как именно сделан конкретный дистрибутив, поставляемый производителем. Понять, за счёт чего плата грузится и с какими патчами ядра она работает.

<p>Требующиеся знания</p> <p>Общая структура Yocto Порядок загрузки процессора</p>	<p>Требующиеся навыки</p> <p>Чтение кода Отладка Базовый shell scripting</p>
---	---

18

Санкт-Петербург, СПбГУ

Системы непрерывной интеграции для проектов под RISC-V

Современные проекты с открытым исходным кодом не умеют собираться в режиме нативной сборки. Студентам нужно выяснить, какие есть основные виды проектов и помочь им, создав необходимый удобный инструментарий. Предлагается реализовать часть требуемой функциональности, которая позволит разработчикам легче добавлять RISC-V в список платформ, на которых их библиотеки и утилиты работают.

<p>Требующиеся знания</p> <p>Современный процесс разработки ПО CI Кросс-компиляция</p>	<p>Требующиеся навыки</p> <p>Отладка Работа в консоли, базовый shell scripting</p>
---	---

19