

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Институт информационных технологий, математики и механики

УТВЕРЖДЕНО
решением Ученого совета ННГУ
протокол № 10 от 27.08.2025

Рабочая программа дисциплины

Оптимизация производительности программ

Уровень высшего образования
Бакалавриат

Направление подготовки / специальность
02.03.02 - Фундаментальная информатика и информационные технологии

Направленность образовательной программы
Системное программирование

Форма обучения
очная

г. Нижний Новгород

2025 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.20 Оптимизация производительности программ относится к части, формируемой участниками образовательных отношений образовательной программы.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства	
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	Для текущего контроля успеваемости	Для промежуточной аттестации
ПК-ТОП_11: Способен разрабатывать, оптимизировать и отлаживать системное программное обеспечение	<p>ПК-ТОП_11.1: Разрабатывает низкоуровневый код для встраиваемых систем и драйверов</p> <p>ПК-ТОП_11.2: Оптимизирует код под ограниченные ресурсы</p> <p>ПК-ТОП_11.3: Работает с ОС, загрузчиками и аппаратурой</p> <p>ПК-ТОП_11.4: Отлаживает системное ПО без ОС</p>	<p>ПК-ТОП_11.1: ПК-ТОП_11.1. 3-1. Знает архитектуру современных процессоров, особенности работы с оборудованием процессора.</p> <p>ПК-ТОП_11.1. 3-2. Знает принципы взаимодействия ПО с аппаратурой.</p> <p>ПК-ТОП_11.1. У-1. Умеет разрабатывать низкоуровневый код для встроенного программного обеспечения и драйверов</p> <p>ПК-ТОП_11.1. У-2. Умеет разрабатывать драйверы для аппаратных устройств.</p> <p>ПК-ТОП_11.2: ПК-ТОП_11.2. 3-1. Знает методы оптимизации (кэш, память, тактовая частота процессора)</p> <p>ПК-ТОП_11.2. 3-2. Знает инструменты профилирования</p> <p>ПК-ТОП_11.5. 3-2. Знает архитектуру компиляторов, оптимизации для выбранных архитектур процессоров</p> <p>ПК-ТОП_11.2. У-1. Умеет анализировать бенчмарки, выявлять узкие места.</p> <p>ПК-ТОП_11.3: ПК-ТОП_11.3. 3-1. Знает</p>	Собеседование	Зачёт: Задачи

		<p>архитектуру ядра Linux ПК-ТОП_11.3. 3-2. Знает принципы работы встраиваемых операционных систем</p> <p>ПК-ТОП_11.3. У-1. Умеет портировать код между различными вариантами загрузчиков и ОС.</p> <p>ПК-ТОП_11.4: ПК-ТОП_11.4. 3-1. Знает методы анализа bare-metal сбоев</p> <p>ПК-ТОП_11.4. 3-2. Знает форматы бинарных файлов, работу с памятью.</p> <p>ПК-ТОП_11.4. У-1. Умеет отлаживать код .</p>		
<p>ПК-ТОП_6: Способен оптимизировать производительность программного обеспечения</p>	<p>ПК-ТОП_6.1: Выполняет мониторинг производительности программного обеспечения.</p> <p>ПК-ТОП_6.2: Выполняет оптимизацию программного кода.</p>	<p>ПК-ТОП_6.1: ПК-ТОП_6.1. 3-1. Знает методы и средства мониторинга производительности компьютерного программного обеспечения</p> <p>ПК-ТОП_6.1. 3-2. Знает метрики производительности программного обеспечения</p> <p>ПК-ТОП_6.1. 3-3. Знает современные инструменты мониторинга производительности программного обеспечения</p> <p>ПК-ТОП_6.1. У-1. Умеет применять методы и средства мониторинга производительности компьютерного программного обеспечения</p> <p>ПК-ТОП_6.1. У-2. Умеет интерпретировать диагностические данные мониторинга производительности компьютерного программного обеспечения</p> <p>ПК-ТОП_6.1. У-3. Умеет определять "узкие места" программного обеспечения.</p> <p>ПК-ТОП_6.2:</p>	Собеседование	<p>Зачёт: Задачи</p>

		<p>ПК-ТОП_6.2. 3-1. Знает методы и средства оптимизации производительности компьютерного программного обеспечения</p> <p>ПК-ТОП_6.2. 3-2. Знает современные инструменты оптимизации производительности программного обеспечения</p> <p>ПК-ТОП_6.2. У-1. Умеет оптимизировать программный код с использованием специализированных программных средств</p> <p>ПК-ТОП_6.2. У-2. Умеет выработать варианты оптимизации производительности компьютерного программного обеспечения.</p>		
--	--	---	--	--

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очная
Общая трудоемкость, з.е.	2
Часов по учебному плану	72
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	16
- занятия семинарского типа (практические занятия / лабораторные работы)	16
- КСР	1
самостоятельная работа	39
Промежуточная аттестация	0
	Зачёт

3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

Наименование разделов и тем дисциплины	Всего (часы)	в том числе		
		Контактная работа (работа во взаимодействии с преподавателем), часы из них		Самостоятельная работа обучающегося, часы
		Занятия	Занятия	

	0 Ф 0	лекционного типа	семинарского типа (практические занятия/ лабора- торные работы), часы	0 Ф 0	0 Ф 0
Раздел I. Введение в предмет.	11	4	3	7	4
Раздел II. Оптимизация программ: алгоритмы и структуры данных .	17	5	2	7	10
Раздел III. Векторизация циклов .	15	2	3	5	10
Раздел IV Алгоритмическая и программная оптимизация для систем на базе процессоров архитектуры x86	18	2	6	8	10
Раздел V Оптимизация программ в условиях ограниченных ресурсов: оптимизация для систем на базе процессоров архитектуры RISC-V	10	3	2	5	5
Аттестация	0				
КСР	1			1	
Итого	72	16	16	33	39

Содержание разделов и тем дисциплины

Цели и задачи изучения дисциплины.

Целью освоения дисциплины «Оптимизация производительности программ» является освоение методов повышения производительности программ.

Задачи:

1. Изучение целей, критериев, методов и принципов оптимизации программ.
2. Изучение основных метрик, связанных с производительностью аппаратного и программного обеспечения.
3. Изучение особенностей вычислительных архитектур, влияющих на производительность.
4. Освоение методов замера времени, профилировки и анализа производительности программ.
5. Изучение основных подходов к повышению производительности программ.
6. Практика повышения производительности кода для процессоров архитектур x86 и RISC-V.

Раздел I. Введение в предмет.

- 1.1. Анализ производительности и оптимизация программ: введение в предмет (суть, цели, задачи, основные принципы, критерии, примеры, структура курса)
- 1.2. Архитектурные механизмы, влияющие на производительность. Уровни параллелизма
- 1.3. Основные метрики, методика их сбора и анализа.
- 1.4. Практическое использование оптимизирующего компилятора. Компиляция и сборка из командной строки. Использование профилировщика

Раздел II. Оптимизация программ: алгоритмы и структуры данных .

- 2.1. Алгоритмическая оптимизация. Вычислительная сложность, практические аспекты
- 2.2. Алгоритмическая оптимизация при реализации алгоритмов сортировки
- 2.3. Оптимизация структур данных при работе с разреженными матрицами

Раздел III. Векторизация циклов

- 3.1. Векторизация циклов. Общие принципы и использование компилятора
- 3.2. Векторизация циклов. Анализ с использованием специализированных инструментов

Раздел IV Алгоритмическая и программная оптимизация для систем на базе процессоров архитектуры

- 4.1. Рациональное использование иерархии памяти на примере задачи о транспонировании матриц. Использование профилировщика
- 4.2. Оптимизация вычислений, аспекты параллелизма, балансировка нагрузки. Задача о вычислении простых чисел
- 4.3. Пошаговая оптимизация программ в практических приложениях: вычисление формулы Блэка-Шоулса (векторизация, параллелизм, понижение точности, эквивалентные вычисления, NUMA).
- 4.4. Пошаговая оптимизация программ в практических приложениях: Фильтрация изображений.
- 4.5. Оптимизация вычислений в алгоритмах на графах. Матрично-векторная реализация алгоритма Брандеса.

Раздел V Оптимизация программ в условиях ограниченных ресурсов: оптимизация для систем на базе процессоров архитектуры RISC-V

- 5.1. Оптимизация программ для процессоров архитектуры RISC-V. Общие принципы и отличия от x86. Векторизация с использованием векторных расширений RVV
- 5.2. Различные приемы оптимизации программ для процессоров архитектуры RISC-V: оптимизация на примерах (фильтрация изображений, использование деревьев решений, алгоритмы BLAS, решение СЛАУ)
- 5.3. Оптимизация программ для процессоров RISC-V в задачах линейной алгебры

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

Для обеспечения самостоятельной работы обучающихся используются:
Открытые онлайн-курсы MOOC:

Материалы курса., <https://hpc-education.unn.ru/ru/обучение/курсы/оптимизация>.

Иные учебно-методические материалы:

- Гергель В.П. Курс «Основы параллельных вычислений».
<http://www.intuit.ru/studies/courses/1091/293/info>
- Гергель В.П. Курс «Теория и практика параллельных вычислений».
<https://intuit.ru/studies/courses/1156/190/info>
- Немнюгин С. Курс «Основы параллельного программирования с использованием MPI».
<http://www.intuit.ru/studies/courses/1090/294/info>
- Бахтин В. Курс «Параллельное программирование с OpenMP».
<http://www.intuit.ru/studies/courses/1111/295/info>
- Баркалов К.А. и др. Курс «Intel Parallel Programming Professional»
<https://intuit.ru/studies/courses/4447/983/info>
- Ануфриенко А., Идрисов Р. Курс «Введение в оптимизацию производительности приложений с использованием компиляторов Интел» <https://intuit.ru/studies/courses/707/563/info>

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Собеседование) для оценки сформированности компетенции ПК-ТОП_11:

1. Назовите особенности векторизации кода для процессоров архитектуры RISC-V.
2. Назовите основные приемы оптимизации кода для процессоров архитектуры RISC-V.
3. Как векторизовать цикл с использованием инструкций векторного расширения RVV 1.0?
4. Можно ли векторизовать цикл, если в нем вызываются стандартные математические функции (sqrt, sin, cos и др.)?

5.1.2 Типовые задания (оценочное средство - Собеседование) для оценки сформированности компетенции ПК-ТОП_6:

1. Перечислите архитектурные механизмы, влияющие на производительность программ.
2. Назовите цели и задачи оптимизации программ.
3. Сформулируйте критерии оптимизации программ.
4. Назовите основные метрики, которые применяются для оценки производительности программ. Поясните их смысл и назначение.
5. Как вычислить пиковую производительность процессора?
6. Как определить пропускную способность подсистемы памяти?
7. Поясните, в чем суть алгоритмической оптимизации. Приведите примеры использования алгоритмической оптимизации для ускорения вычислений.
8. Векторизации и какова ее роль. Приведите примеры циклов, которые могут быть векторизованы. В каком случае цикл не может быть векторизован либо его векторизация не эффективна? Приведите примеры.
9. В каком случае требуется балансировка вычислительной нагрузки? Приведите соответствующий пример задачи.
10. Какие особенности подсистемы памяти влияют на производительность программ?
11. Приведите пример алгоритма, в котором разные способы работы с памятью существенно влияют на время вычислений. Поясните основные идеи оптимизации.

Критерии оценивания (оценочное средство - Собеседование)

Оценка	Критерии оценивания
зачтено	Студент дал развернутый ответ на все вопросы без существенных ошибок.
не зачтено	При ответе студент допускает грубые ошибки в основном материале.

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатора достижения компетенций)	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	не зачтено			зачтено			
<u>Знания</u>	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	Уровень знаний ниже минимальных требований. Имели место грубые ошибки	Минимально допустимый уровень знаний. Допущено много негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет.	Уровень знаний в объеме, превышающем программу подготовки.
<u>Умения</u>	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые с недочетами	Продемонстрированы все основные умения. Решены все основные задачи с отдельными несущественными недочетами, выполнены все задания в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
<u>Навыки</u>	Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки	Имеется минимальный набор навыков для решения стандартных задач с некоторыми недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторыми недочетами	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов	Продемонстрирован творческий подход к решению нестандартных задач

Шкала оценивания при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой

	отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично».
	очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо».
	удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
не зачтено	неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно».
	плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Задачи) для оценки сформированности компетенции ПК-ТОП_11

1. Реализовать параллельный векторизованный алгоритм для умножения плотных матриц (по определению). Провести вычислительные эксперименты, проверить асимптотику, сравнить производительность при разных порядках циклов. Объяснить наблюдаемые эффекты. Использовать вычислительную систему с многоядерным процессором архитектуры RISC-V.
2. Реализовать параллельный векторизованный алгоритм для умножения плотных матриц (блочный алгоритм). Провести вычислительные эксперименты, определить наилучшие размеры блока, сравнить производительность при разных порядках циклов и размерах блока. Объяснить наблюдаемые эффекты. Использовать вычислительную систему с многоядерным процессором архитектуры RISC-V.
3. Реализовать параллельный алгоритм фильтрации изображений. Провести вычислительные эксперименты, оценить эффективность распараллеливания, изучить аспекты производительности. Использовать вычислительную систему с многоядерным процессором архитектуры RISC-V.
4. Реализовать параллельный векторизованный алгоритм умножения разреженной матрицы, представленной в формате CRS, на плотный вектор. Провести вычислительные эксперименты, оценить эффективность распараллеливания и векторизации. Использовать вычислительную систему с многоядерным процессором архитектуры RISC-V.
5. Выполнить оптимизацию кода учебной программы, подготовленной преподавателем, для процессора архитектуры RISC-V. Объяснить использованные приемы оптимизации, исследовать их вклад в сокращение времени работы программы.

5.3.2 Типовые задания (оценочное средство - Задачи) для оценки сформированности компетенции ПК-ТОП_6

1. Реализовать параллельный векторизованный блочный алгоритм для умножения плотных матриц. Провести вычислительные эксперименты, проверить асимптотику, сравнить производительность с базовой версией программы (умножение по определению). Использовать вычислительную систему с многоядерным процессором архитектуры x86.
2. Реализовать последовательный алгоритм сортировки Хоара (быстрая сортировка). Провести вычислительные эксперименты, проверить асимптотику, изучить аспекты производительности. Использовать вычислительную систему с процессором архитектуры x86.
3. Реализовать параллельный алгоритм фильтрации изображений. Провести вычислительные эксперименты, оценить эффективность распараллеливания, изучить аспекты производительности. Использовать вычислительную систему с многоядерным процессором архитектуры x86.
4. Реализовать параллельный алгоритм нахождения простых чисел (решето Эратосфена). Провести вычислительные эксперименты, оценить эффективность распараллеливания при использовании разных методов распределения вычислительной нагрузки. Использовать вычислительную систему с многоядерным процессором архитектуры x86.
5. Реализовать параллельный векторизованный алгоритм умножения разреженной матрицы, представленной в формате CRS, на плотный вектор. Провести вычислительные эксперименты, оценить эффективность распараллеливания и векторизации. Использовать вычислительную систему с многоядерным процессором архитектуры x86.
6. Выполнить оптимизацию кода учебной программы, подготовленной преподавателем, для процессора архитектуры x86. Объяснить использованные приемы оптимизации, исследовать их вклад в сокращение времени работы программы.

Критерии оценивания (оценочное средство - Задачи)

Оценка	Критерии оценивания
зачтено	Выполнена основная часть задания, возможно с незначительными недочетами.
не зачтено	Выполнено менее половины задания, есть существенные недочеты.

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Теория и практика параллельных вычислений / Гергель В.П. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=663423&idb=0>.
2. Введение в принципы функционирования и применения современных мультиядерных архитектур (на примере Intel Xeon Phi) / Гергель В.П., Мееров И.Б., Бастраков С.И., Горшков А.В., Козинев Е.А., Линева А.В., Сиднев А.А., Сысоев А.В. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=662900&idb=0>.
3. Пирова Анна Юрьевна. Параллельные алгоритмы разделения графов : учебно-методическое пособие / А. Ю. Пирова ; ННГУ им. Н. И. Лобачевского. - Нижний Новгород : Изд-во ННГУ, 2019. - 20 с. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=794851&idb=0>.

4. Малявко Александр Антонович. Параллельное программирование на основе технологий openmp, cuda, opencl, mpi : учебник для вузов / А. А. Малявко. - 3-е изд. - Москва : Юрайт, 2025. - 135 с. - (Высшее образование). - ISBN 978-5-534-14116-0 : 549.00. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=964557&idb=0>.
5. Уильямс Э. Параллельное программирование на C++ в действии. Практика разработки многопоточных программ : монография / Уильямс Э. - Москва : ДМК-пресс, 2023. - 674 с. - ISBN 978-5-89818-319-6., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=878880&idb=0>.
6. Восс М. Параллельное программирование на C++ с помощью библиотеки TBB : монография / Восс М.; Асенхо Р.; Рейндерс Дж. - Москва : ДМК-пресс, 2020. - 674 с. - ISBN 978-5-97060-864-7., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=735942&idb=0>.

Дополнительная литература:

1. Параллельное программирование с использованием OpenMP / Левин М.П. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=663520&idb=0>.
2. Методические указания по самостоятельному изучению литературы по информационным технологиям : учебно-методическое пособие / В. П. Гергель, С. Н. Карпенко, Г. В. Кузенкова [и др.] ; ННГУ им. Н. И. Лобачевского, Институт информационных технологий, математики и механики. - Нижний Новгород : Изд-во ННГУ, 2018. - 21 с. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=822383&idb=0>.
3. Программирование на современных мультиядерных архитектурах (на примере Intel Xeon Phi) / Гергель В.П., Мееров И.Б., Бастраков С.И. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=663288&idb=0>.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

Программное обеспечение:

1. Операционные системы семейства Microsoft Windows, лицензия по подписке Microsoft Imagine.
2. Браузер Google Chrome, предоставляется бесплатно на условиях лицензионных соглашений на программное обеспечение с открытым исходным кодом.
3. Среда разработки семейства Microsoft Visual Studio <https://visualstudio.microsoft.com/ru/>

Интернет-ресурсы:

- Материалы курса. <https://hpc-education.unn.ru/ru/обучение/курсы/оптимизация>

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 02.03.02 - Фундаментальная информатика и информационные технологии.

Автор(ы): Мееров Иосиф Борисович, кандидат технических наук, доцент.

Заведующий кафедрой: Мееров Иосиф Борисович, кандидат технических наук.

Программа одобрена на заседании методической комиссии от 25.06.2025, протокол № Протокол №11.