Минобрнауки России

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Оренбургский государственный университет имени В.А. Бондаренко»

Кафедра математики и цифровых технологий

РАБОЧАЯ ПРОГРАММА

ДИСЦИПЛИНЫ

«Б1.Д.В.Э.4.2 Оптимизация программного кода»

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки
09.03.04 Программная инженерия
(код и наименование направления подготовки)

Разработка корпоративных систем и цифровых решений
(наименование направленности (профиля) образовательной программы)

Квалификация
Бакалавр

Форма обучения
Очная

Год набора 2026

Рабочая программа дисциплины *«Б1.Д.В.Э.4.2 Оптимизация программного кода»* рассмотрена и утверждена на заседании кафедры

Кафедра математики и цифровых технологий
наименование кафедры

протокол № _____от "___" _____ 20__г.

Заведующий кафедрой
Кафедра математики и цифровых технологий _____ А.Е. Шухман
наименование кафедры *подпись* *расшифровка подписи*

Исполнители:
Доцент _____ А.И. Сарайкин
должность *подпись* *расшифровка подписи*

_____
должность *подпись* *расшифровка подписи*

СОГЛАСОВАНО:
Председатель методической комиссии по направлению подготовки
09.03.04 Программная инженерия _____ И.П. Болодурина
код наименование *личная подпись* *расшифровка подписи*

Заведующий отделом формирования фонда и научной обработки документов
_____ С.А. Бикташиров
личная подпись *расшифровка подписи*

Уполномоченный по качеству института
_____ С.Н. Морозова
личная подпись *расшифровка подписи*

№ регистрации _____

© Сарайкин А.И., 2026
© ОГУ, 2026

1 Цели и задачи освоения дисциплины

Цель (цели) освоения дисциплины: формирование у студентов системного понимания факторов, влияющих на производительность программного обеспечения, и освоение практических навыков профилирования, анализа и оптимизации программного кода.

Задачи:

1. Изучение методов профилирования программного обеспечения с использованием современных инструментов для выявления критических по производительности участков кода.
2. Освоение анализа ассемблерного кода, генерируемого компилятором, и понимание влияния кода на C/C++ на итоговые машинные инструкции.
3. Применение методов оптимизации на уровне алгоритмов и структур данных, включая выбор оптимальных алгоритмов, кэширование, ленивые вычисления и снижение вычислительной сложности.
4. Освоение техник низкоуровневой оптимизации: оптимизация работы с памятью (кэш, предвыборка, локальность), векторизация (SIMD), оптимизация ветвлений и циклов, минимизация накладных расходов на вызовы функций.
5. Разработка многопоточного и асинхронного кода с учётом оптимизации использования ресурсов процессора и минимизации блокировок и накладных расходов на синхронизацию.
6. Проведение сравнительного анализа эффективности различных реализаций одного и того же функционала, обоснование выбора конкретного решения на основе измеренных метрик.
7. Понимание и учёт при разработке влияния настроек компилятора (флаги оптимизации, межпроцедурная оптимизация, профилирующая оптимизация) и операционной системы на итоговую производительность приложения.

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к дисциплинам (модулям) по выбору вариативной части блока Д «Дисциплины (модули)»

Пререквизиты дисциплины: *Б1.Д.В.8 Тестирование цифровых решений, Б1.Д.В.11 Интеграция информационных систем*

Постреквизиты дисциплины: *Отсутствуют*

3 Требования к результатам обучения по дисциплине

Процесс изучения дисциплины направлен на формирование следующих результатов обучения

Код и наименование формируемых компетенций	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций
ПК*-13 Способен разрабатывать, оптимизировать и отлаживать системное программное обеспечение	ПК*-13-В-1 Разрабатывает низкоуровневый код для встраиваемых систем и драйверов ПК*-13-В-2 Оптимизирует код под ограниченные ресурсы	**Знать:** методы оптимизации (кэш, память, тактовая частота процессора); инструменты профилирования; архитектуру компиляторов, оптимизации для выбранных архитектур

Код и наименование формируемых компетенций	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций
		процессоров. **Уметь:** анализировать бенчмарки, выявлять узкие места; разрабатывать код, оптимизированный для выбранной аппаратной архитектуры. **Владеть:** навыками чтения и оптимизации ассемблерного кода; методами оптимизации памяти и вычислений; навыками оптимизации под ограниченные ресурсы; методами оценки эффективности оптимизаций на основе метрик.

4 Структура и содержание дисциплины

4.1 Структура дисциплины

Общая трудоемкость дисциплины составляет 4 зачетные единицы (144 академических часа).

Вид работы	Трудоемкость, академических часов	
	7 семестр	всего
Общая трудоёмкость	**144**	**144**
Контактная работа:	**55**	**55**
Лекции (Л)	18	18
Лабораторные работы (ЛР)	34	34
Консультации	1	1
Индивидуальная работа и инновационные формы учебных занятий	1,5	1,5
Промежуточная аттестация (зачет, экзамен)	0,5	0,5
Самостоятельная работа: *- выполнение курсового проекта (КП);* *- самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий;* *- изучение разделов курса в системе электронного обучения;* *- подготовка к лабораторным занятиям;* *- подготовка к рубежному контролю).*	**89** +	**89**
Вид итогового контроля (зачет, экзамен, дифференцированный зачет)	**экзамен**	

Разделы дисциплины, изучаемые в 7 семестре

№ раздела	Наименование разделов	Количество часов				
		всего	аудиторная работа			внеауд. работа
			Л	ПЗ	ЛР	
1	Введение в оптимизацию программного кода	18	2		4	12
2	Анализ ассемблерного кода и компиляция	18	2		4	12
3	Алгоритмическая оптимизация и структуры данных	18	2		4	12
4	Низкоуровневая оптимизация	24	4		6	14
5	Многопоточность и асинхронность	18	2		4	12
6	Сравнительный анализ и оценка эффективности	22	2		6	14
7	Практика оптимизации в индустрии	26	4		6	16
	Итого:	144	18		34	92
	Всего:	144	18		34	92

4.2 Содержание разделов дисциплины

№ 1. Введение в оптимизацию программного код

Цели и задачи оптимизации. Метрики производительности. Профилирование и бенчмаркинг. Инструменты профилирования: perf, VTune, Valgrind. Основные принципы оптимизации.

№ 2. Анализ ассемблерного кода и компиляция

Генерация ассемблерного кода компилятором. Чтение дизассемблированного кода. Влияние кода на C/C++ на машинные инструкции. Флаги оптимизации компилятора. Межпроцедурная оптимизация. Профилирующая оптимизация.

№ 3. Алгоритмическая оптимизация и структуры данных

Выбор оптимальных алгоритмов. Оценка вычислительной сложности. Кэширование. Ленивые вычисления. Снижение вычислительной сложности. Оптимизация структур данных.

№ 4. Низкоуровневая оптимизация

Оптимизация работы с памятью: кэш, предвыборка, локальность. Векторизация (SIMD). Оптимизация ветвлений и циклов. Минимизация накладных расходов на вызовы функций.

№ 5. Многопоточность и асинхронность

Разработка многопоточного и асинхронного кода. Оптимизация использования ресурсов процессора. Минимизация блокировок и накладных расходов на синхронизацию.

№ 6. Сравнительный анализ и оценка эффективности

Сравнительный анализ эффективности различных реализаций. Обоснование выбора решения на основе метрик. Влияние операционной системы на производительность. Особенности оптимизации для Astra Linux.

№ 7. Практика оптимизации в индустрии

Оптимизация высоконагруженных систем. Обработка логов, работа с базами данных, криптографические операции, обработка медиаконтента, задачи машинного обучения на граничных устройствах. Открытые бенчмарки (SPEC, TPC).

4.3 Лабораторные работы

№ ЛР	№ раздела	Наименование лабораторных работ	Кол-во часов
1	1	Профилирование программного обеспечения с использованием perf и Valgrind	4
2	2	Анализ ассемблерного кода, генерируемого компилятором	4
3	3	Алгоритмическая оптимизация: выбор структур данных и алгоритмов	4
4	4	Оптимизация работы с памятью и векторизация (SIMD)	4

№ ЛР	№ раздела	Наименование лабораторных работ	Кол-во часов
5	4	Оптимизация ветвлений и циклов	4
6	5	Разработка многопоточного кода с минимизацией блокировок	4
7	6	Сравнительный анализ эффективности реализаций	4
8	7	Оптимизация реального программного компонента	6
		Итого:	34

4.4 Курсовой проект (7 семестр)

Примерные темы курсового проекта:
1. Оптимизация алгоритма обработки логов для высоконагруженной системы.
2. Ускорение обработки веб-запросов в корпоративном приложении.
3. Снижение потребления памяти при работе с большими данными.
4. Оптимизация алгоритма сжатия данных для работы в реальном времени.
5. Векторизация криптографических операций с использованием SIMD.
6. Оптимизация многопоточной обработки медиаконтента.
7. Оптимизация задач машинного обучения на граничных устройствах.
8. Оптимизация системного ПО для работы в защищённой среде Astra Linux.
9. Профилирование и оптимизация библиотеки для работы с базами данных.
10. Оптимизация кода с использованием профилирующей оптимизации компилятора.

5 Учебно-методическое обеспечение дисциплины

5.1 Основная литература

1. Ахо, А. Компиляторы [Текст] : принципы, технологии, инструменты: пер. с англ. / А. Ахо, Р. Сети, Дж. Ульман. - М. : Вильямс, 2003. - 768 с. : ил. - Библиогр.: с. 742-767. - ISBN 5-8459-0189-8.

2. Практикум по методам параллельных вычислений [Текст] : учеб. для вузов / А. В. Старченко [и др.]; под ред. А. В. Старченко ; Томс. гос. ун-т. - М. : Изд-во Моск. ун-та, 2010. - 200 с. : ил. - (Суперкомпьютерное образование / Суперкомпьютер. консорциум ун-тов России). - Библиогр.: с. 194-195. - ISBN 978-5-211-05976-4.

5.2 Дополнительная литература

1. Финогенов, К. Г. Использование языка Ассемблера [Текст] : учеб. пособие / К. Г. Финогенов. - М. : Горячая линия-Телеком, 2004. - 438 с. : ил. - (Учебное пособие для вузов) - ISBN 5-93517-118-X.

2. Конова, Е. А. Алгоритмы и программы. Язык С++ [Текст] : учебное пособие для студентов, обучающихся по направлению "Прикладная информатика" / Е. А. Конова, Г. А. Поллак.- 3-е изд., стер. - Санкт-Петербург : Лань, 2018. - 384 с. : ил. - (Учебники для вузов. Специальная литература). - Библиогр.: с. 382. - ISBN 978-5-8114-2020-9.

3. Вирт, Н. Разработка операционной системы и компилятора. Проект Оберон = Tne Design of an Operating System and Compiler. Project Oberon [Текст] / Н. Вирт, Ю. Гуткнехт; пер. с англ. Е. В. Борисова, Л. Н. Чернышева. - М. : ДМК Пресс, 2012. - 560 с. - Парал. тит. л. англ. - Библиогр. в конце гл. - ISBN 978-5-94074-672-0.

5.3 Периодические издания

1. Программирование (ИПМ им. М.В. Келдыша РАН): журнал. - Москва: издательство «Наука».
2. Вестник компьютерных и информационных технологий: журнал. - Москва: ИД «Спектр».
3. Информационные технологии: журнал. - Москва: Агентство «Роспечать».

5.4 Интернет-ресурсы

− www.elibrary.ru - научная электронная библиотека. Журналы, книги, патенты;

− SCOPUS [Электронный ресурс]: реферативная база данных / компания Elsevier. – Режим доступа: https://www.scopus.com/, в локальной сети ОГУ.

5.5 Программное обеспечение, профессиональные базы данных и информационные справочные системы

1. Операционная система РЕД ОС.
2. Пакет офисных приложений «МойОфис Образование».
3. Университетская платформа электронного обучения «Электронные курсы ОГУ в системе обучения Moodle» (http://moodle.osu.ru).
4. Для работы с ресурсами Интернет - веб-браузер Яндекс https://yandex.ru/.
5. ГАРАНТ Платформа F1 [Электронный ресурс]: справочно-правовая система. / Разработчик ООО НПП «ГАРАНТ-Сервис», 119992, Москва, Воробьевы горы, МГУ, [1990–2026]. – Режим доступа в сети ОГУ http://garant.net.osu.ru
6. Автоматизированная интерактивная система сетевого тестирования - АИССТ (зарегистрирована в РОСПАТЕНТ, Свидетельство о государственной регистрации программы для ЭВМ №2011610456, правообладатель – Оренбургский государственный университет), режим доступа - http://aist.osu.ru.
7. Набор свободных компиляторов для различных языков программирования GCC (GNU Compiler Collection).
8. Инструменты профилирования: perf, Intel VTune Amplifier, AMD uProf, Valgrind (Cachegrind, Callgrind).
9. Бесплатные инструменты для написания и редактирования компьютерного кода – VS Code, Eclipse.

6 Материально-техническое обеспечение дисциплины

Учебные аудитории для проведения занятий лекционного типа, семинарского типа, курсового проектирования, для проведения групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Аудитории оснащены комплектами ученической мебели, техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения лабораторных занятий используется компьютерный класс, оснащенный:

- персональными компьютерами с процессорами архитектуры x86-64;
- операционной системой РЕД ОС, Linux (Astra Linux, Ubuntu);
- компиляторами GCC;
- инструментами профилирования: perf, Intel VTune Amplifier, Valgrind;
- доступом к сети «Интернет»;
- проекционным оборудованием для демонстрации учебных материалов.

Помещение для самостоятельной работы обучающихся оснащено компьютерной техникой, подключенной к сети «Интернет», и обеспечением доступа в электронную информационно-образовательную среду ОГУ.