

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Московский физико-технический институт  
(национальный исследовательский университет)»**

**УТВЕРЖДЕНО**

**Директор физтех-школы  
радиотехники и компьютерных  
технологий**

**Д.А. Гаврилов**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Оптимизирующие компиляторы
<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Компьютерные технологии и вычислительная техника Физтех-школа Радиотехники и Компьютерных Технологий кафедра перспективных вычислительных технологий
<b>курс:</b>	4
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 8 (весенний) - Экзамен

Аудиторных часов: 30 всего, в том числе:

лекции: 30 час.

семинары: 0 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 30 час.

Подготовка к экзамену: 30 час.

Всего часов: 90, всего зач. ед.: 2

Программу составил: К.В. Юхин, старший преподаватель

Программа обсуждена на заседании кафедры перспективных вычислительных технологий 16.05.2023

## Аннотация

Курс «Оптимизирующие компиляторы» разработан с целью знакомства студентов с основами работы современных оптимизирующих компиляторов, особенностях их разработки и применения. В результате изучения курса у студентов будут сформированы знания: об архитектурах основных современных микропроцессоров; об основных структурах данных и алгоритмах, применяемых в компиляторах; об основных компонентах современного компилятора. В качестве практических навыков студенты научатся писать компиляторно-ориентированный код, позволяющий исполнять скомпилированную программу максимально эффективно.

### 1. Цели и задачи

#### Цель дисциплины

- знакомство студентов с основами работы современных оптимизирующих компиляторов.

#### Задачи дисциплины

- формирование общего представления об архитектурах основных современных микропроцессоров;
- знакомство с основными структурами данных и алгоритмами применяемыми в компиляторах.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
ОПК-3 Способен составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты)	ОПК-3.3 Владеет методами визуального и графического представления результатов научной (научно-технической, инновационной технологической) деятельности в виде отчетов, научных публикаций
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
	ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации
	ПК-2.2 Способен планировать и проводить научные исследования самостоятельно или в качестве члена (руководителя) малого научного коллектива
	ПК-2.3 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- различия архитектур современных микропроцессоров с точки зрения оптимизирующего компилятора;
- основные компоненты современного компилятора;
- понятие алгоритмической сложности;
- основы теории графов и базовые алгоритмы на графах в компиляторах;
- основные оптимизации, применяемые в современных компиляторах.

уметь:

- писать компиляторно-ориентированный код, позволяющий исполнять скомпилированную программу максимально эффективно.

владеть:

- математическими основами теории графов, применяемыми при создании компилятора.

### 4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

#### 4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Введение. Понятие компиляции, виды оптимизирующей компиляции. Архитектуры современных микропроцессоров	2			2
2	Внутреннее представление программы и его построение	2			2
3	Управляющий граф, построение и использование	2			2
4	Анализ потока данных на основе решения системы уравнений	2			2
5	Динамический и статический профилировщики программ	2			2
6	Мелко-зернистые оптимизации	2			2
7	Сбор общих подвыражений, удаление избыточных операций чтения из памяти	1			1
8	Подстановка констант, удаление мертвого кода	2			2
9	Межпроцедурный анализ программ	2			2
10	Межпроцедурные оптимизации	2			2
11	Анализ зависимостей в цикловых регионах программы	2			2
12	Цикловые оптимизации и условия их корректности	2			2
13	Оптимизации доступа в память	2			2
14	Векторные инструкции	2			2
15	Машинная модель	2			2

16	Анализ и оптимизации на предикатном коде	1			1
Итого часов		30			30
Подготовка к экзамену		30 час.			
Общая трудоёмкость		90 час., 2 зач.ед.			

#### 4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

##### Семестр: 8 (Весенний)

#### 1. Введение. Понятие компиляции, виды оптимизирующей компиляции. Архитектуры современных микропроцессоров

Виды оптимизирующей компиляции. Основные принципы сравнения архитектур на пакетах SPEC. Общая схема компиляции: входные языки, оптимизатор, целевые платформы. Контексты в которых работает оптимизатор: языки высокого уровня – C, C++, F90, двоичный код – x86, динамическая трансляция – JAVA byte code optimizer. Архитектуры для которых делается оптимизированный код: RISC, CISC, Superscalar, VLIW, EPIC, multithreading. Фазы компиляции и сборки программы: лексический анализ, синтаксический анализ, семантический анализ, оптимизация, кодогенерация, линковка, сброс отладочной информации. Основные проблемы, возникающие в процессе решения задач оптимизирующей компиляции: определение независимых вычислений на уровне отдельных операций (fine grain parallelism) и на уровне блоков программы (coarse grain parallelism), скорость компиляции, объем занимаемой памяти, теоретические ограничения на сложность анализа, оптимальное использование ресурсов архитектуры, увеличение размера программы, коррекция аналитической информации в процессе глобального планирования, отсутствие достоверной информации о реальном поведении программы, доказательство корректности преобразований, сброс отладочной информации в случае оптимизированного кода.

#### 2. Внутреннее представление программы и его построение

Типы внутреннего представления. Виртуальные регистры как удобное средство для проведения оптимизаций. Abstract Syntax Tree (AST), High Intermediate Representation (HIR) == AST, Medium IR (MIR) Low IR (LIR). Обсуждение примера, иллюстрирующего различия между HIR, MIR and LIR. Проблема выбора последовательности оптимизаций

#### 3. Управляющий граф, построение и использование

Линейный участок, расширенный линейный участок, алгоритм построения управляющего графа, обходы управляющего графа: сначала в глубину обход. Обходы управляющего графа: обход в предварительном порядке, обход в постпредварительном порядке, обход сначала в ширину, доминаторы и постдоминаторы, циклы и сильно связанные компоненты, дерево циклов. Определение зависимостей между обращениями в память в ациклическом участке программы. Различные техники: переменные видимые за пределами процедуры, разрешение на основе типовой информации, диапазоны значений, динамическое определение зависимостей

#### 4. Анализ потока данных на основе решения системы уравнений

Алгоритм поиска достигающих определений реализованный на базе битовых векторов. Анализ потока данных на основе разреженных представлений потока данных: фронт доминирования, итерационный фронт доминирования, форма статического единственного присваивания, граф определений использований

#### 5. Динамический и статический профилировщики программ

Вероятности дуг, вероятности выхода из цикла, счетчики линейных участков, инструментирование кода, распространение счетчиков, коррекция профиля, профилирование данных, профилирование зависимостей, эвристики статического профилирования

## 6. Мелко-зернистые оптимизации

Мелко-зернистые оптимизации. Оптимизации на основе результатов распространения битов значений. Удаления избыточных знаковых расширений. Балансировка и реассоциация выражений. Оптимизация удаления избыточных операций передачи управления с использованием метода дублирования вычислений

## 7. Сбор общих подвыражений, удаление избыточных операций чтения из памяти

Удаление избыточных записей в память, вынесения инвариантных операций из циклов, удаление частично избыточных вычислений

## 8. Подстановка констант, удаление мертвого кода

Глобальное распространение копий, определение индуктивных переменных циклов, удаление индуктивных переменных, понижение силы операций.

## 9. Межпроцедурный анализ программ

Граф вызовов, чувствительность анализа к потоку управления и месту вызовов процедур. Межпроцедурные распространители информации о программе: определение и использование переменных, взятие адреса переменной, межпроцедурная протяжка констант.

## 10. Межпроцедурные оптимизации

Подстановка процедур, подстановка копий процедур (клонирование), частичная постановка процедур, удаление хвостовой рекурсии, удаление мертвых процедур, реорганизация данных программы.

## 11. Анализ зависимостей в цикловых регионах программы

Гнездо циклов, пары операций для анализа, лексикографический порядок на отношении зависимости, вектор направления зависимости, вектор расстояния зависимости, решатель систем линейных Диофантовых уравнений и неравенств (метод Фурье, Симплекс метод, целочисленный Симплекс метод).

## 12. Цикловые оптимизации и условия их корректности

Раскрутка циклов, слияние циклов, открутка первых итераций цикла в скалярную часть программы, разрезание циклов, постановка цикла под условие использования его побочных эффектов, динамический анализ зависимостей в циклах.

## 13. Оптимизации доступа в память

Уменьшение промахов в КЭШ'и различных назначений и уровней (кэш данных, кэш инструкций, различные уровни кэш памяти): предподкачка данных в кэш, предподкачка инструкций в кэш, увеличение форматов чтения и записи в память, перестановка циклов, компактизация циклов.

## 14. Векторные инструкции

На примере SSE расширения для x86 архитектуры. Алгоритмы и методы использования векторных операций в целях оптимизации (SIMD). Средства и методы распараллеливания программ на несколько процессов.

## 15. Машинная модель

Выбор инструкций при трансляции MIR в LIR. Планирование кода: граф зависимостей, планирование на основе списков и приоритетов операций. Циклическое и ациклическое глобальное планирование.

## 16. Анализ и оптимизации на предикатном коде

Планирование цикла с конвейеризацией и использованием аппаратной поддержки для конвейеризации.

## 5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Необходимое оборудование для лекций: компьютер и мультимедийное оборудование (проектор, звуковая система).

Необходимое программное обеспечение.

Обеспечение самостоятельной работы: доступ к научным библиотекам.

## 6. Перечень рекомендуемой литературы

### Основная литература

1. Никлаус Вирт, Построение компиляторов, ДМК, Москва, 2010.
2. Keith Cooper, Engineering a Compiler, Second Edition, Morgan Kaufmann, Burlington, 2011.
3. Steven S. Muchnick, "Advanced Compiler Design and Implementation", Morgan Kauffman, San Francisco, 1997.

### Дополнительная литература

1. Loren Taylor Simpson, "Value-Driven Redundancy Elimination", Ph.D. Thesis, Rice University, 1996
2. David F. Bacon, Susan L. Graham, and Oliver J. Sharp, Compiler Transformations for High-Performance Computing, Computer Science Division, University of California, Berkeley, California 94720
3. John R. Ellis. Bulldog: A compiler for VLIW Architectures. MIT Press, 1985
4. Dick Grune, Henri E. Bal, Criel J.H. Jacobs and Koen G. Langendoen, Modern Compiler Design, by John Wiley & Sons, Ltd, 2000.
5. Randy Allen, Ken Kennedy, Optimizing Compilers for Modern Architectures. 2002 by Academic Press.

## 7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Сборники тезисов докладов конференций (Оптимизирующие компиляторы), доступные через Internet научные и научно-технические журналы (IEEE, ACM и др.), электронные конспекты лекций, учебные пособия и сборники задач, разработанные для данного курса.

## 8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

Мультимедийные технологии, презентации PowerPoint.

## 9. Методические указания для обучающихся по освоению дисциплины (модуля)

Студент, изучающий дисциплину, должен с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике. В результате изучения дисциплины студент должен знать основные определения, понятия.

Успешное освоение курса требует напряжённой самостоятельной работы студента. В программе курса приведено минимально необходимое время для работы студента над темой. Самостоятельная работа включает в себя:

- чтение и конспектирование рекомендованной литературы;
- проработку учебного материала (по конспектам лекций, учебной и научной литературе);
- подготовку к экзамену.

Руководство и контроль за самостоятельной работой студента осуществляется в форме индивидуальных консультаций.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов, следует обращаться за консультациями к преподавателю.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Компьютерные технологии и вычислительная техника Физтех-школа Радиотехники и Компьютерных Технологий кафедра перспективных вычислительных технологий
<b>курс:</b>	4
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 8 (весенний) - Экзамен

**Разработчик:** К.В. Юхин, старший преподаватель



## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
ОПК-3 Способен составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты)	ОПК-3.3 Владеет методами визуального и графического представления результатов научной (научно-технической, инновационной технологической) деятельности в виде отчетов, научных публикаций
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
	ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации
	ПК-2.2 Способен планировать и проводить научные исследования самостоятельно или в качестве члена (руководителя) малого научного коллектива
	ПК-2.3 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Оптимизирующие компиляторы» обучающийся должен:

### знать:

- различия архитектур современных микропроцессоров с точки зрения оптимизирующего компилятора;
- основные компоненты современного компилятора;
- понятие алгоритмической сложности;
- основы теории графов и базовые алгоритмы на графах в компиляторах;
- основные оптимизации, применяемые в современных компиляторах.

### уметь:

- писать компиляторно-ориентированный код, позволяющий исполнять скомпилированную программу максимально эффективно.

### владеть:

- математическими основами теории графов, применяемыми при создании компилятора.

## 3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

С целью контроля освоения обучающимися учебного материала проводится устный опрос в начале занятия по теме прошлой лекции или в конце занятия по пройденной теме.

#### **4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся**

1. Понятие компиляции, виды компиляции, основные стадии компиляции
2. Внутреннее представления программы. Виды промежуточного кода
3. Виды потоковых уравнений. Метод эффективного поиска решения потокового уравнения
4. Понятие управляющего графа. Алгоритм построения управляющего графа
5. Определение отношения доминирования и обратного доминирования. Алгоритм построения отношения доминирования
6. Дерево доминаторов, фронт доминирования (+ алгоритм построения)
7. Виды профилировки программы. Преимущества и недостатки
8. Мелкозернистые оптимизации: область действия, основные преобразования на примере
9. Сбор общих подвыражений: описание алгоритма
10. Межпроцедурный анализ: основные оптимизации на примере
11. Анализ зависимостей в циклах: виды зависимостей, понятие пространства итераций
12. Основные виды цикловых оптимизаций на примере. Условие корректности преобразований
13. Оптимизация доступов в память: область применимости, абстракция модели памяти в компиляторе
14. Векторизация: векторные машины, векторные инструкции, основные этапы векторизации кода на примере
15. Генерация кода: основные этапы, представление программы в момент кодогенерации, простейший алгоритм связывания регистров. Планирование кода
16. Понятие предикатного кода, генерация предикатного кода компилятором

Примеры билетов для проведения экзамена:

Билет 1.

1. Понятие компиляции, виды компиляции, основные стадии компиляции
2. Понятие предикатного кода, генерация предикатного кода компилятором

Билет 2.

1. Внутреннее представления программы. Виды промежуточного кода
2. Генерация кода: основные этапы, представление программы в момент кодогенерации, простейший алгоритм связывания регистров. Планирование кода

#### **Критерии оценивания**

Оценка отлично (10) выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (9) выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (8) выставляется студенту, показавшему систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.

Оценка хорошо (7) выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты.

Оценка хорошо (6) выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Оценка хорошо (5) выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.

Оценка удовлетворительно (4) выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.

Оценка удовлетворительно (3) выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.

Оценка неудовлетворительно (2) выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных принципов и не умеет использовать полученные знания при решении типовых задач.

Оценка неудовлетворительно (1) выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубейшие ошибки в формулировках базовых понятий дисциплины и вообще не имеет навыков решения типовых практических задач.

## **5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Экзамен проводится в устной форме.

При проведении устного экзамена обучающемуся предоставляется не менее 60 минут на подготовку. Опрос обучающегося на устном экзамене не должен превышать двух астрономических часов.

Во время проведения экзамена обучающиеся могут пользоваться программой дисциплины.