

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

УТВЕРЖДЕНО

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

А.В. Дворкович

	Рабочая программа дисциплины (модуля)
по дисциплине:	Технологии оптимизирующей компиляции
по направлению:	Информатика и вычислительная техника
профиль подготовки:	Компьютерные технологии и вычислительная техника Физтех-школа Радиотехники и Компьютерных Технологий кафедра информатики и вычислительной техники
курс:	4
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 8 (весенний) - Дифференцированный зачет

Аудиторных часов: 30 всего, в том числе:

лекции: 30 час.

семинары: 0 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 15 час.

Всего часов: 45, всего зач. ед.: 1

Программу составили:

М.И. Нейман-Заде, канд. физ.-мат. наук

А.Л. Маркин

Программа обсуждена на заседании кафедры информатики и вычислительной техники 18.05.2020

Аннотация

Курс разработки оптимизирующих компиляторов посвящён основам построения компиляторов и описанию оптимизаций, применяемых в них. Курс охватывает темы, посвящённые каждому этапу работы компилятора - лицевой части, оптимизатору, хвостовой части. Слушатели изучат стадии лексического, синтаксического и семантического анализа, познакомятся с основными принципами кодогенерации, а также узнают о широком спектре оптимизаций начиная с локальных, и заканчивая межмодульными. Эти знания помогут как понимать логику работы самих компиляторов, так и видеть потенциальные узкие места производительности в пользовательском программном коде.

Курс проходит в формате лекционных занятий. Для успешного прохождения курса необходимо посещать лекции, слушать их и конспектировать, а также изучать дополнительную литературу.

1. Цели и задачи

Цель дисциплины

освоение студентами базовых знаний в области устройства и разработки оптимизирующих компиляторов, изучение различных способов представления программы, изучение классов оптимизирующих преобразований и построителей аналитической информации, ознакомление с организацией процесса разработки компилятора.

Задачи дисциплины

- лексического, синтаксического и семантического разбора программ на языках высокого уровня;
- алгоритмов поиска и преобразования графов, широко применяемых в оптимизирующей компиляции;
- классов оптимизирующих преобразований, их внутреннего устройства, а также сбора, построения и хранения аналитической информации, используемой в процессе работы оптимизаций;
- оптимизирующего планирования и генерации кода под целевую архитектуру;
- оптимизации работы с памятью;
- организации процесса разработки оптимизирующего компилятора.

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	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

новые научные результаты	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:

- ☐ общее устройство современного оптимизирующего компилятора;
- ☐ принципы лексического, синтаксического и семантического разбора программ;
- ☐ основные аналитические структуры данных, используемые в оптимизирующих компиляторах;
- ☐ основные алгоритмы на графах и деревьях;
- ☐ общее представление об оптимизирующих преобразованиях;
- ☐ общее представление о реализации крупного программного проекта в области компиляторов.

уметь:

- ☐ преобразовывать исходную программу на языке высокого уровня в промежуточное представление;
- ☐ анализировать промежуточное представление программы;
- ☐ строить аналитические структуры данных по промежуточному представлению;
- ☐ строить алгоритмы на аналитических структурах данных, используемых в оптимизирующем компиляторе;
- ☐ моделировать работу простых потоковых, цикловых и межпроцедурных оптимизаций на промежуточном представлении.

владеть:

- ☐ современными программными методиками, используемыми в оптимизирующих компиляторах;
- ☐ приемами организованной разработки и сопровождения крупного программного проекта.

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 в компиляторы	2			1
2	Лексический анализ	2			1
3	Синтаксический анализ	2			1
4	Семантический анализ	2			1
5	Промежуточное представление программы	2			1
6	Планирование и генерация кода	2			1
7	Введение в оптимизации	2			1

8	Локальные оптимизации	2			1
9	Оптимизации потока управления	2			1
10	Оптимизации памяти	2			1
11	Оптимизации циклов	4			2
12	Межпроцедурные и межмодульные оптимизации	2			1
13	Анализы указателей	4			2
Итого часов		30			15
Подготовка к экзамену		0 час.			
Общая трудоёмкость		45 час., 1 зач.ед.			

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

Семестр: 8 (Весенний)

1. Введение в компиляторы

Основные понятия и виды трансляторов. Устройство современных компиляторов. История развития компиляторов.

2. Лексический анализ

Определение и задача лексического анализа. Регулярные выражения. Практические вопросы создания лексических анализаторов.

3. Синтаксический анализ

Определение и задача синтаксического анализа. Формальные грамматики. Деревья разбора.

4. Семантический анализ

Определение и задача семантического анализа. Таблицы символов. Области видимости. Типы данных.

5. Промежуточное представление программы

Определение и виды промежуточных представлений. Основные языковые конструкции в представлении.

6. Планирование и генерация кода

Основные стадии работы хвостовой части компилятора. Алгоритмы планирования инструкций. Алгоритмы распределения регистров.

7. Введение в оптимизации

Определение и задача оптимизаций. Классификация оптимизаций.

8. Локальные оптимизации

Локальные оптимизации потока данных. Оптимизации алгебраических выражений. Удаление мёртвого кода.

9. Оптимизации потока управления

Задача оптимизации потока управления. Упрощение условных конструкций. Сортировка графа потока управления. Оптимизация конструкций switch.

10. Оптимизации памяти

Виды памяти в программе. Оптимизация работы со стеком. Оптимизация работы с массивами.

11. Оптимизации циклов

Основные понятия циклов. Оптимизации циклов для устранения избыточности. Оптимизации циклов для работы с кешем. Оптимизации циклов для уплотнения широкой команды.

12. Межпроцедурные и межмодульные оптимизации

Оптимизации вызовов функций. Проблемы неявных вызовов функций. Проблемы и способы решения межпроцедурных оптимизаций в разных.

13. Анализы указателей

Понятие указателей и их роль во время оптимизации. Определение и задача анализа указателей. Классификация и виды анализов указателей.

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

Компьютер и мультимедийное оборудование (проектор, звуковая система).

6. Перечень рекомендуемой литературы

Основная литература

1. Компиляторы: принципы, технологии и инструментарий [Текст] : [учеб. пособие для вузов] / Ахо, А. В. [и др.] ; [пер. с англ. и ред. И. В. Красикова] .— 2-е изд. — М. : Вильямс, 2011 .— 1184 с.

Дополнительная литература

1. Искусство программирования для ЭВМ [Текст] : в 7 т. Т. 2 : Получисленные алгоритмы : [учеб. пособие для вузов] / Д. Кнут ; пер. с англ. Г. П. Бабенко [и др.] ; под ред. К. И. Бабенко, В. С. Штаркмана .— М. : Мир, 1977 .— 724 с.

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

Не используются

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. MS PowerPoint , демонстрация презентаций.

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 дифференцированному зачету.

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)

по направлению:	Информатика и вычислительная техника
профиль подготовки:	Компьютерные технологии и вычислительная техника Физтех-школа Радиотехники и Компьютерных Технологий кафедра информатики и вычислительной техники
курс:	4
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 8 (весенний) - Дифференцированный зачет

Разработчики:

М.И. Нейман-Заде, канд. физ.-мат. наук

А.Л. Маркин

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 «Технологии оптимизирующей компиляции» обучающийся должен:

знать:

- ☐ общее устройство современного оптимизирующего компилятора;
- ☐ принципы лексического, синтаксического и семантического разбора программ;
- ☐ основные аналитические структуры данных, используемые в оптимизирующих компиляторах;
- ☐ основные алгоритмы на графах и деревьях;
- ☐ общее представление об оптимизирующих преобразованиях;
- ☐ общее представление о реализации крупного программного проекта в области компиляторов.

уметь:

- ☐ преобразовывать исходную программу на языке высокого уровня в промежуточное представление;
- ☐ анализировать промежуточное представление программы;
- ☐ строить аналитические структуры данных по промежуточному представлению;
- ☐ строить алгоритмы на аналитических структурах данных, используемых в оптимизирующем компиляторе;
- ☐ моделировать работу простых потоковых, цикловых и межпроцедурных оптимизаций на промежуточном представлении.

владеть:

- ☐ современными программными методиками, используемыми в оптимизирующих компиляторах;
- ☐ приемами организованной разработки и сопровождения крупного программного проекта.

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

Задания для самостоятельного выполнения:

1. Для выбранного языка программирования изучить историю появления и развития, дать описание существующих реализаций, оценить влияние на последующие идеи и разработки.
2. Изучить существующие алгоритмы реализации регулярных выражений, выяснить какой именно алгоритм применяется а заданном компиляторе.
3. Изучить существующие генераторы синтаксических анализаторов, выяснить какой именно алгоритм применяется а заданном компиляторе.
4. Для заданного языка дать описание системы типов, назвать её сильные и слабые стороны.
5. Изучить промежуточные представления в заданном компиляторе, описать в его терминах основные программные конструкции.
6. Изучить алгоритмы планирования ресурсов и кода, подготовить пример для заданной аппаратной платформы.
7. Подготовить список основных оптимизаций для заданного компилятора
8. Изучить дополнительные локальные оптимизации, дать их описание, предназначение, сильные и слабые стороны
9. Изучить дополнительные оптимизации потока управления, дать их описание, предназначение, сильные и слабые стороны
10. Изучить дополнительные оптимизации памяти, дать их описание, предназначение, сильные и слабые стороны
11. Изучить дополнительные оптимизации циклов, дать их описание, предназначение, сильные и слабые стороны
12. Изучить дополнительные глобальные оптимизации, дать их описание, предназначение, сильные и слабые стороны
13. Изучить дополнительные анализы указателей, дать их описание, предназначение, сильные и слабые стороны

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

Перечень контрольных вопросов для сдачи дифференцированного зачёта в 8-ом семестре;

1. Понятие компилятора и прочие виды трансляторов.
2. Логическое устройство компилятора и сопутствующих инструментов.
3. История появления и развития компиляторов.
4. Задачи лексического анализа и способы описания лексем.
5. Реализация регулярных выражений в коде.
6. Понятие грамматики, их виды.
7. Задачи синтаксического анализа, способы описания синтаксиса.
8. Дерево разбора программы: предназначение, основные проблемы, способы решения.
9. Семантический анализ. Системы типов языка.
10. Реализация типов в языке. Представление, взаимодействие.

11. Объекты программы. Время жизни, видимость.
12. Промежуточное представление. Виды, примеры, различия.
13. Промежуточное представление. Основные конструкции: условия, циклы, вызовы.
14. Граф потока управления. Предназначение, основные конструкции: условия, циклы, вызовы.
15. Нумерация графа.
16. Циклы в графе потока управления. Определения, связанные понятия.
17. Граф вызовов.
18. Поток данных.
19. Планирование операций.
20. Распределение регистров.
21. Задачи и классификация оптимизаций.
22. Peephole: набор локальных оптимизаций.
23. Оптимизация алгебраических выражений. Сбор общих подвыражений.
24. Удаление мёртвого кода.
25. Устранение избыточности графа потока управления: лишние и недостижимые узлы.
26. Оптимизация условных конструкций.
27. Устранение переходов и улучшение работы с кэшем инструкций.
28. Слияние условных конструкций (if-conversion).
29. Понятие и структура цикла.
30. Оптимизации циклов для улучшения работы с кэшем данных.
31. Инварианты в цикле и способы их устранения.
32. Раскрутка цикла.
33. Программная конвейеризация цикла.
34. Модели сборки приложений.
35. Межпроцедурные оптимизации над вызовами функций.
36. Девиртуализация и подстановка неявных вызовов.
37. Предназначение анализов указателей и их взаимодействие с оптимизациями.
38. Виды анализов указателей, хранение результатов анализов.
39. Алгоритмы выполнения анализов указателей.

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

Дифференцированный зачет проводится в устной форме.

При проведении зачета обучающемуся предоставляется 20 минут на подготовку. Опрос обучающегося проводится в течение 30 минут.

Во время проведения экзамена и зачета обучающиеся могут пользоваться программой дисциплины.