

**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

	Рабочая программа дисциплины (модуля)
по дисциплине:	Введение в архитектуру вычислительных систем и язык ассемблера
по направлению:	Информатика и вычислительная техника
профиль подготовки:	Компьютерные технологии и вычислительная техника Физтех-школа Радиотехники и Компьютерных Технологий кафедра радиоэлектроники и прикладной информатики
курс:	1
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 2 (весенний) - Дифференцированный зачет

Аудиторных часов: 120 всего, в том числе:

лекции: 0 час.

семинары: 120 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 60 час.

Всего часов: 180, всего зач. ед.: 4

Количество контрольных работ, заданий: 4

Программу составил: И.Р. Дединский, старший преподаватель

Программа обсуждена на заседании кафедры радиоэлектроники и прикладной информатики 30.01.2024

Аннотация

Курс представляет собой введение в архитектуру вычислительных систем. Практическая часть содержит задачи достаточно большого объема. Задачи подобраны по большей части таким образом, что в конце курса каждый студент самостоятельно реализует бинарный транслятор в нативный код Intel x86_64 с байт-кода софт-процессора, реализованного в прошлом семестре, либо кодогенератор в нативный код Intel x86_64 для собственного языка программирования высокого уровня, реализованного в прошлом семестре. Также в курсе уделяется большое внимание защите данных, надежности программ, механизмам взлома программ и противодействия ему.

Сложность задач курса легко регулируется их функциональным наполнением.

Для обучения используются следующие принципы:

1. Во главу угла ставится задача, ее решение и, главное, путь от задачи к решению. Во всякой задаче подчеркивается разделение на идею решения и технологию реализации.
2. Самостоятельность решения является ключевым условием.
3. Понимание студентами тех средств, с помощью которых он решил задачу, ставится выше уровня самих средств решения.
4. Аккуратность и надежность решения ставятся выше «программистских трюков», иногда позволяющих в отдельных случаях добиться несколько лучших результатов.

Преподавание курса ведется в предположении, что студенты уже знают язык Си.

1. Цели и задачи

Цель дисциплины

– познакомить студентов с базовыми принципами организации внутренней организации компьютерных систем, с базовыми принципами организации операционных систем, а также абстракций и интерфейсов, которые предоставляются программисту для взаимодействия с операционной системой.

Задачи дисциплины

Задача дисциплины заключается в демонстрации базовых принципов низкоуровневого программирования.

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	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности

решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	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:

- основы работы в UNIX- подобных системах;
- основы низкоуровневого программирования;
- основы машинного кода, языков ассемблера;
- различные пути повышения производительности программы.

уметь:

- создавать программы на языках Си и Ассемблер;
- работать в UNIX- подобных средах;
- создавать программы на языках Си и Ассемблер без использования высокоуровневых библиотек.

владеть:

- навыками ведения простейших программных проектов в системах контроля версий.

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	Представление информации в ВС		15		8
2	Структура ВС		15		8
3	Хранение данных и иерархия памяти		15		8
4	Архитектура процессора		15		8
5	Машинное представление программ		15		8
6	Оптимизация программ		15		8
7	Компиляторные технологии		30		12
Итого часов			120		60
Подготовка к экзамену		0 час.			

Общая трудоёмкость	180 час., 4 зач.ед.
--------------------	---------------------

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

Семестр: 2 (Весенний)

1. Представление информации в ВС

Двоичная система счисления. Шестнадцатеричная нотация. Слова и размеры данных. Представления целых чисел в форме с фиксированной точкой (представление беззнаковых чисел, представление знаковых чисел в прямом и дополнительном кодах). Особенности сложения и вычитания целых чисел. Флаги. Представление вещественных чисел в форме с плавающей точкой. Размещение числовых данных в памяти. Двоично-десятичные числа. Представление нечисловой информации.

2. Структура ВС

Уровни абстрактного представления ЭВМ, язык Ассемблера и машинные команды среди них. Элементы и контекст машинного представления информации. Трансляция и интерпретация программ и команд. Краткое описание устройств ЭВМ и схема их взаимодействия. Структура центрального процессора (ЦП). Регистры, арифметико-логическое устройство, устройство управления. Схема работы ЭВМ. Одновременность и параллельность. Использование отладчика.

3. Хранение данных и иерархия памяти

Технологии хранения данных. Оперативная память ЭВМ. Ячейки, адреса, машинные слова, разряды, биты. Локальность. Иерархия видов памяти и принцип кэширования. Кэширование и иерархия устройств хранения. Создание кэш-ориентированных программ. Влияние кэш-памяти на производительность.

4. Архитектура процессора

Архитектура системы команд X86. Способы задания операндов. Система команд как важнейшая характеристика ЭВМ. Разнообразие систем команд в реальных ЭВМ (CISC, RISC и др.). Понятие цифрового конструирования и язык управления аппаратурой. Последовательная реализация X86. Основные принципы конвейеризации. Конвейерная реализация X86. Работа с внешними по отношению к CPU устройствами. Механизм прерываний.

5. Машинное представление программ

Кодирование программ. Форматы данных. Обращение к данным. Арифметические и битовые операции. Команды управления. Процедуры. Массивы. Неоднородные конструкции данных. Указатели. Некорректные ссылки и переполнение буфера. 64-битное расширение IA-32. Программы с плавающей точкой.

6. Оптимизация программ

Возможности и ограничения оптимизирующих компиляторов. Измерение производительности программ. Исключение неэффективности циклов. Уменьшение количества вызовов процедур. Исключение ненужных ссылок в память. Понятие о современном процессоре. Разворачивание циклов. Увеличение степени параллелизма. Результат оптимизации кода. Ограничители производительности. Производительность памяти. Обнаружение и исключение мест потери производительности.

7. Компиляторные технологии

Задача бинарной трансляции. JIT- и AOT-компиляция. Реализация бинарной трансляции. Решение проблем с доступом к памяти. Кодогенерация в компиляторах. Структура ELF-файла. Итоговая семестровая работа.

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

Учебная аудитория, оснащенная компьютером и мультимедийным оборудованием (проектор, звуковая система).

6. Перечень рекомендуемой литературы

Основная литература

Фонд литературы кафедры:

1. В. Н. Пильщиков. Введение в язык ассемблера.
2. Р. Э. Брайант, Д. Р. О'Халларон. Компьютерные системы: архитектура и программирование. СПб.: БХВ, 2005.
3. В. И. Юров. Ассемблер. Учебник для вузов. СПб, Питер, 2010

Дополнительная литература

нет

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

1. <https://www.laruence.com/sse>
2. <http://godbolt.org>

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

Операционная система Linux. Среда программирования.

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

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 проработку учебного материала (по учебной литературе) подготовку к практическим занятиям.

Промежуточный контроль знаний проводится в виде сдачи проектных заданий.

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 радиоэлектроники и прикладной информатики
курс:	1
квалификация:	бакалавр
Семестр, формы промежуточной аттестации: 2 (весенний) - Дифференцированный зачет	
Разработчик:	И.Р. Дединский, 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 «Введение в архитектуру вычислительных систем и язык ассемблера» обучающийся должен:

знать:

- основы работы в UNIX- подобных системах;
- основы низкоуровневого программирования;
- основы машинного кода, языков ассемблера;
- различные пути повышения производительности программы.

уметь:

- создавать программы на языках Си и Ассемблер;
- работать в UNIX- подобных средах;
- создавать программы на языках Си и Ассемблер без использования высокоуровневых библиотек.

владеть:

- навыками ведения простейших программных проектов в системах контроля версий.

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

Примеры заданий:

1. Реализовать простейшую программу на языке ассемблера.
2. Реализовать программу, работающую с видеопамятью.
3. Реализовать программу, осуществляющую перехват прерываний.
4. Реализовать взлом программы.
5. Реализовать простейший вариант функции printf.
6. Реализовать программу, рассчитывающую множество Мандельброта, и оптимизировать ее.
7. Реализовать хеш-таблицу и оптимизировать ее работу.
8. Реализовать бинарный транслятор в нативный код Intel x86_64.

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

1. Система Linux, виртуальная машина. Инструменты для написания, компиляции и отладки программ.
2. Представление целых чисел. Обратный дополнительный код, битовые операции. Знаковые и беззнаковые числа. Undefined Behaviour.
3. Язык ассемблера x86_64, базовые конструкции; работа с регистрами и с памятью.
4. Архитектура CISC v.s. RISC. gdb и objdump. Стек вызовов, соглашения о вызовах x86_64.
5. Реализация функций в языке ассемблера, соглашения о вызовах.
6. Работа с памятью, кеш-память.
7. Механизм прерываний.
8. Суперскалярная архитектура.
9. Конвейер и исполнение машинного кода.
10. Технологии взлома программ и противодействия ему.
11. Системные вызовы через int 0x80 и vds0 (sysenter/syscall).
12. Файловые дескрипторы, open, read и write.
13. Выравнивание данных и векторные инструкции.
14. Технологии низкоуровневой оптимизации программ.
15. Отображение ELF файла на память.
16. Технологии бинарной трансляции.

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 всесторонние, 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 по итогам текущей успеваемости и сдачи заданий, предусмотренных программой дисциплины.