

**Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)**

Кафедра информационных технологий
Факультет информационных технологий и анализа больших данных

Документ подписан усиленной неквалифицированной электронной подписью

Организация: Финансовый университет при Правительстве РФ

Сертификат: Zto7E4D3aezIHLPoZHqFvb5ocTrM7Tv8

Утверждено: Проректор по учебной и методической работе Е.А. Каменева

Дата: 11.02.2026 г.

А.В. Антипов, А.М. Ерофеев

Низкоуровневое программирование

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки:

09.03.04 - Программная инженерия,

Образовательная программа «Инженер-разработчик программного обеспечения»

Профиль:

«Инженер-разработчик программного обеспечения»

Рекомендовано

*Факультет информационных технологий и анализа больших данных
(протокол № 06 от 17.03.2026 г.)*

Одобрено

*Кафедра информационных технологий
(протокол № 1 от 27.02.2026 г.)*

Содержание

1. Наименование дисциплины	4
2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	4
3. Место дисциплины в структуре образовательной программы	7
4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	7
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий . . .	8
5.1. Содержание дисциплины	8
5.2. Учебно-тематический план	11
5.3. Содержание семинаров	12
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	14
6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	14
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю . . .	15
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	16
8. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины	27
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	29
10. Методические указания для обучающихся по освоению дисциплины	30
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	32

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	33
--	----

1. Наименование дисциплины

«Низкоуровневое программирование».

2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	
ОПК-5	Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	
Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	
Устанавливает простое программное обеспечение для информационных и автоматизированных систем	знать: простое программное обеспечение для информационных и автоматизированных систем уметь: устанавливать простое программное обеспечение для информационных и автоматизированных систем	
Настраивает программное и аппаратное обеспечение для решения конкретных задач предметной области	знать: программное и аппаратное обеспечение для решения конкретных задач предметной области уметь: настраивать программное и аппаратное обеспечение для решения конкретных задач предметной области	
Проводит анализ информационной инфраструктуры, выявляет слабые места, вырабатывает рекомендации для ее улучшения	знать: принципы выявления слабых мест в программных системах уметь: проводить анализ информационной инфраструктуры, выявлять слабые места, вырабатывать рекомендации для ее улучшения	
ПКП-4	Способность проектировать архитектуру и дизайн программной системы на основе требований к ней	
Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	

Декомпозирует задачу, решаемую программной системой на более простые, элементарные подзадачи, реализует простые задачи с помощью алгоритмизации и в виде программного модуля на языке программирования	знать: основные алгоритмы и структуры данных уметь: использовать на практике простые структуры данных, оценивать сложность алгоритмов
Демонстрирует знания основ теории архитектуры и дизайна программного обеспечения, разделения задачи на уровни абстракции, подбирает те или иные архитектурные решения на основе анализа требований к программной системе в целом	знать: основы теории архитектуры и дизайна программного обеспечения, разделение задачи на уровни абстракции уметь: подбирать те или иные архитектурные решения на основе анализа требований к программной системе в целом
Проводит интеграцию программных модулей, описывает, анализирует и модифицирует интерфейс программных модулей, понимает принципы разделения ответственности, модульности и инверсии зависимостей применительно к созданию программного обеспечения	знать: принципы разделения ответственности, модульности и инверсии зависимостей применительно к созданию программного обеспечения уметь: проводить интеграцию программных модулей, описывать, анализировать и модифицировать интерфейс программных модулей

Проектирует пользовательское взаимодействие с программной системой, включая пользовательский интерфейс, сценарии использования, механизмы аутентификации и разделения ролей, в том числе, с учетом требований информационной безопасности	знать: сценарии использования, механизмы аутентификации и разделения ролей, в том числе, с учетом требований информационной безопасности уметь: проектировать пользовательское взаимодействие с программной системой, включая пользовательский интерфейс
Адаптирует архитектуру и интерфейс программной системы под известную внешнюю среду, включая интерфейсы внешних программных систем	знать: интерфейсы внешних программных систем уметь: адаптировать архитектуру и интерфейс программной системы под известную внешнюю среду, включая интерфейсы внешних программных систем
Учитывает нефункциональные требования к программной системе в ее архитектуре, в том числе требования к надежности, согласованности, скорости работы программной системы	знать: архитектурные решения на основе нефункциональных требований к программной системе уметь: собирать, формулировать, систематизировать и анализировать нефункциональные требования к программной системе

3. Место дисциплины в структуре образовательной программы

Дисциплина «Низкоуровневое программирование» относится к «Модулю ”Системное программирование”».

4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

Очно-заочная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 7 (в часах)
Общая трудоёмкость дисциплины	3/108	108
Контактная работа – Аудиторные занятия	24	24
Лекции	8	8
Семинары, практические занятия	16	16
Самостоятельная работа	84	84
Вид текущего контроля	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	Зачет	Зачет

5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

5.1. Содержание дисциплины

Тема 1. Основные понятия и определения

Введение в низкоуровневое программирование. Цели и задачи курса. Роль курса в подготовке специалиста. Сведения об истории развития проблематики дисциплины. Обзор и общая характеристика языков программирования. Стандарты языков программирования. Понятие низкоуровневого программирования. Файловые системы с точки зрения системного программиста. Конфигурационная информация операционных систем. Сборка и установка ОС Линукс (ядра операционной системы). Стек протоколов TCP/IP с точки зрения системного программиста. Текстовые редактор vim.

Тема 2. Архитектура и системы команд современных микропроцессоров

Классическая архитектура ЭВМ, предложенная группой Дж. фон Неймана. Арифметико-логическое устройство. Основное (оперативное) запоминающее устройство. Иерархия памяти. Современные структуры микропроцессоров. Регистры общего назначения. Управление потоком команд.

Тема 3. Введение в язык C

Структура программы. Переменные, указатели, объявления в C. Преобразование типов данных в C. Логические, битовые, операции сдвига, сравнения в C. Квалифицирующие типы и спецификаторы функций в C. Директивы предпроцессора в C. Простые типы. Базовые операторы. Выражения и операции. Стандартные математические функции. Операторы выбора. Операторы цикла. Функции. Область видимости. Операции прерывания и безусловного перехода. Стил программирования.

Тема 4. Составные типы данных

Составные типы данных: массивы. Объявление и инициализация массивов. Обращение к элементам массива. Упорядочивание массивов. Составные типы данных: структуры и объединения. Расположение элементов структур в ОЗУ ЭВМ. Составные типы данных: строки. Объявление и инициализация строк. Функции для работы со строками. Составные типы данных: многомерные массивы. Инициализация многомерных массивов. Операции с многомерными массивами. Составные типы данных: битовые поля. Составные типы данных: перечисления.

Тема 5. Вспомогательные средства и приемы работы

Указатели и адреса памяти. Представление памяти процесса. Управление памятью. Области данных, текста программы, стека, кучи. Модели памяти. Динамические массивы. Преобразования типов данных. Логические, битовые операции, операции сдвига, операции сравнения. Квалифицирующие типы и спецификаторы функций. Директивы предпроцессора. Отличия между стандартами языка Си разных лет. Переменные окружения. Потоки ввода вывода. Передача параметров. Этапы создания исполняемого файла. Профилирование программы. Автоматизация проверки исходных текстов и создания исполняемых файлов. Автоматическая генерация документации. Отладка без специальных программ. Отладка с использованием gdb. Поиск утечек памяти valgrind. Си и ассемблер. Технологии дезассемблирования. Ассемблерные вставки в компиляторе gcc.

Тема 6. Использование библиотечных функций в С программах

Статические библиотеки в С. Динамические библиотеки в С. Реализация односвязных и многосвязных списков. Приёмы оценки вычислительной сложности алгоритмов в системном программировании. Библиотека libusb. Получение списка доступных устройств libusb_get_device_list(). Получение характеристик заданного устройства. Получение данных от USB манипулятора типа мышь. Определение libusb_transfer_type() устройства. Определение libusb_bos_type() устройства. Определение скорости передачи данных устройства libusb_speed(). Получение номера устройства на шине libusb_get_bus_number(). Получение порта устройства на шине libusb_get_port_number(). Получение родительского устройства libusb_get_parent(). Получение максимального

размера пакета устройства `libusb_get_max_packet_size()`.

5.2. Учебно-тематический план

№ п/п	Наименование тем(разделов) дисциплины	Трудоемкость в часах				
		Всего	Контактная работа - Аудиторная работа			Самостоя- тельная работа
			Общая, в т.ч.:	Лекции	Семинары, практическ- ие занятия	
1	Основные понятия и определе- ния	17	3	1	2	14
2	Архитектура и системы команд современных микропроцессоров	18	4	2	2	14
3	Введение в язык С	17	3	1	2	14
4	Составные типы данных	17	3	1	2	14
5	Вспомогательные средства и приемы работы	19	5	1	4	14
6	Использование библиотечных функций в С программах	20	6	2	4	14
	Итого	108	24	8	16	84

Формы текущего контроля успеваемости согласно учебному плану: **Контрольная работа.**

5.3. Содержание семинаров

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Основные понятия и определения	Понятие низкоуровневого программирования.	Занятия в интерактивной форме в виде дискуссий.
Архитектура и системы команд современных микропроцессоров	Классическая архитектура ЭВМ, предложенная группой Дж. фон Неймана.	Занятия в интерактивной форме в виде дискуссий.
Введение в язык С	Структура программы. Переменные, указатели, объявления в С. Преобразование типов данных в С. Логические, битовые, операции сдвига, сравнения в С. Директивы предпроцессора в С. Простые типы. Базовые операторы. Выражения и операции. Стандартные математические функции. Операторы выбора. Операторы цикла. Функции. Область видимости.	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений.
Составные типы данных	Составные типы данных: массивы. Объявление и инициализация массивов. Обращение к элементам массива. Упорядочивание массивов. Составные типы данных: структуры и объединения. Расположение элементов структур в ОЗУ ЭВМ. Составные типы данных: строки. Объявление и инициализация строк. Функции для работы со строками. Составные типы данных: многомерные массивы. Инициализация многомерных массивов. Операции с многомерными массивами.	Занятия в интерактивной форме в виде дискуссий.
Вспомогательные средства и приемы работы	Указатели и адреса памяти. Представление памяти процесса. Управление памятью. Области данных, текста программы, стека, кучи. Модели памяти. Динамические массивы. Автоматизация проверки исходных текстов и создания исполняемых файлов. Автоматическая генерация документации. Отладка без специальных программ. Отладка с использованием gdb. Поиск утечек памяти valgrind. Си и ассемблер. Технологии дезассемблирования. Ассемблерные вставки в компиляторе gcc.	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений.

Использование библиотечных функций в С программах	Библиотека libusb. Получение списка доступных устройств libusb_get_device_list(). Получение характеристик заданного устройства. Получение данных от USB манипулятора типа мышь. Определение libusb_transfer_type() устройства. Определение libusb_bos_type() устройства. Определение скорости передачи данных устройства libusb_speed(). Получение номера устройства на шине libusb_get_bus_number(). Получение порта устройства на шине libusb_get_port_number().	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений.
--	---	---

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Основные понятия и определения	Стандарты языков программирования.	Работа с учебной литературой. Разбор вопросов по теме занятия.
Архитектура и системы команд современных микропроцессоров	Основное (оперативное) запоминающее устройство.	Работа с учебной литературой. Разбор вопросов по теме занятия. Выполнение и защита домашней контрольной работы.
Введение в язык С	Операции прерывания и безусловного перехода. Стил программирования.	Работа с учебной литературой. Разбор вопросов по теме занятия.
Составные типы данных	Составные типы данных: перечисления.	Работа с учебной литературой. Разбор вопросов по теме занятия.
Вспомогательные средства и приемы работы	Профилрование программы. Автоматизация проверки исходных текстов и создания исполняемых файлов.	Работа с учебной литературой. Разбор вопросов по теме занятия.
Использование библиотечных функций в С программах	Получение родительского устройства libusb_get_parent(). Получение максимального размера пакета устройства libusb_get_max_packet_size().	Работа с учебной литературой. Разбор вопросов по теме занятия.

6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

Примерный вариант контрольной работы

1. Реализовать программу, получающую список всех подключенных к компьютеру USB устройств с использованием libusb. Для каждого найденного устройства напечатать его класс, идентификатор производителя и идентификатор изделия.
2. Изучить состав и характеристики, обнаруженных с помощью реализованной программ USB устройств.
3. Дополнить программу, реализованную в п. 2 функцией печати серийного номера USB устройства. Для написания функции рекомендуется использовать функции libusb_open, libusb_close, libusb_get_string_descriptor_ascii для печати поля iSerialNumber дескриптора устройства.

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Кафедры информационных технологий Факультета информационных технологий и анализа больших данных.

7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы содержится в разделе 2. *Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине*

Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний

ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем

1) Устанавливает простое программное обеспечение для информационных и автоматизированных систем

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: простое программное обеспечение для информационных и автоматизированных систем

Уметь: устанавливать простое программное обеспечение для информационных и автоматизированных систем

Типовые контрольные задания

С помощью логических операций реализуйте запись битов в данной ячейки памяти, размером в 2 байта, в обратном порядке.

Реализуйте с помощью логических операций, при которой старший байт слова логически складывается с младшим байтом, записанным в обратном порядке.

2) Настраивает программное и аппаратное обеспечение для решения конкретных задач предметной области

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: программное и аппаратное обеспечение для решения конкретных задач предмет-

ной области

Уметь: настраивать программное и аппаратное обеспечение для решения конкретных задач предметной области

Типовые контрольные задания

Используя Valgrind (cachegrind) сформировать аннотированный исходный текст программы и оценить число промахов по кеш-памяти данных и инструкций, которые произошли во внутреннем цикле функций.

Используя компилятор GCC, скомпилируйте подготовленную программу с включением отладочной информации (всех уровней). На сколько изменяется размер исполняемого файла в зависимости от уровня детализации отладочной информации?

3) Проводит анализ информационной инфраструктуры, выявляет слабые места, вырабатывает рекомендации для ее улучшения

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: принципы выявления слабых мест в программных системах

Уметь: проводить анализ информационной инфраструктуры, выявлять слабые места, вырабатывать рекомендации для ее улучшения

Типовые контрольные задания

Выделите отладочную информацию в отдельный файл с включением ссылки на него в исполняемый файл.

Проведите отладку программы в GDB.

ПКП-4 Способность проектировать архитектуру и дизайн программной системы на основе требований к ней

1) Декомпозирует задачу, решаемую программной системой на более простые, элементарные подзадачи, реализует простые задачи с помощью алгоритмизации и в виде программного модуля на языке программирования

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: основные алгоритмы и структуры данных

Уметь: использовать на практике простые структуры данных, оценивать сложность алгоритмов

Типовые контрольные задания

Реализуйте логическую арифметику, при которой в одном байте хранилось бы 8 логических переменных.

Реализуйте с помощью логических операций, при которой старший байт слова логически умножается на младший байт, записанный в обратном порядке.

2) Демонстрирует знания основ теории архитектуры и дизайна программного обеспечения, разделения задачи на уровни абстракции, подбирает те или иные архитектурные решения на основе анализа требований к программной системе в целом

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: основы теории архитектуры и дизайна программного обеспечения, разделение задачи на уровни абстракции

Уметь: подбирать те или иные архитектурные решения на основе анализа требований к программной системе в целом

Типовые контрольные задания

Используя Valgrind (cachegrind) сформировать аннотированный исходный текст программы и оценить число промахов по кеш-памяти данных и инструкций, которые произошли во внутреннем цикле функций `dgemm_def`, `dgemm_transpose`.

Выделите отладочную информацию в отдельный файл с включением ссылки на него в исполняемый файл. Проведите отладку программы в GDB.

3) Проводит интеграцию программных модулей, описывает, анализирует и модифицирует интерфейс программных модулей, понимает принципы разделения ответственности, модульности и инверсии зависимостей применительно к созданию программного обеспечения

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: принципы разделения ответственности, модульности и инверсии зависимостей применительно к созданию программного обеспечения

Уметь: проводить интеграцию программных модулей, описывать, анализировать и модифицировать интерфейс программных модулей

Типовые контрольные задания

При помощи профилировщика `perf` получить аннотированный отчет с оценкой числа промахов при обращении к кеш-памяти в функциях `dgemm_def`, `dgemm_transpose`.

Реализовать программу, получающую список всех подключенных к компьютеру USB устройств с использованием `libusb`.

4) Проектирует пользовательское взаимодействие с программной системой, включая пользовательский интерфейс, сценарии использования, механизмы аутентификации и разделения ролей, в том числе, с учетом требований информационной безопасности

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: сценарии использования, механизмы аутентификации и разделения ролей, в том числе, с учетом требований информационной безопасности

Уметь: проектировать пользовательское взаимодействие с программной системой, включая пользовательский интерфейс

Типовые контрольные задания

Реализуйте логическую арифметику, при которой в одном байте хранилось бы 8 логических переменных.

Реализуйте с помощью логических операций, при которой старший байт слова логически умножается на младший байт, записанный в обратном порядке.

5) Адаптирует архитектуру и интерфейс программной системы под известную внешнюю среду, включая интерфейсы внешних программных систем

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: интерфейсы внешних программных систем

Уметь: адаптировать архитектуру и интерфейс программной системы под известную

внешнюю среду, включая интерфейсы внешних программных систем

Типовые контрольные задания

Реализуйте с помощью логических операций, при которой старший байт слова логически складывается с младшим байтом, записанным в обратном порядке.

Реализуйте с помощью логических операций, при которой старший байт слова логически умножается на младший байт, записанный в обратном порядке.

б) Учитывает нефункциональные требования к программной системе в ее архитектуре, в том числе требования к надежности, согласованности, скорости работы программной системы

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: архитектурные решения на основе нефункциональных требований к программной системе

Уметь: собирать, формулировать, систематизировать и анализировать нефункциональные требования к программной системе

Типовые контрольные задания

Используя компилятор GCC, скомпилируйте подготовленную программу с включением отладочной информации (всех уровней). На сколько изменяется размер исполняемого файла в зависимости от уровня детализации отладочной информации?

Выделите отладочную информацию в отдельный файл с включением ссылки на него в исполняемый файл. Проведите отладку программы в GDB.

Примеры практико-ориентированных заданий

Задание 1. Разработайте программу на языке C, которая:

- Выводит список всех файлов в заданном каталоге с атрибутами (права доступа, владелец, размер, время модификации)
- Реализует рекурсивный обход подкаталогов
- Формирует отчет о занимаемом дисковом пространстве по типам файлов

Задание 2.Выполните сборку ядра Linux из исходных кодов:

- Загрузите исходники ядра с kernel.org
- Сконфигурируйте ядро с отключением ненужных модулей для виртуальной машины
- Соберите ядро и модули, установите загрузчик
- Проведите тестирование в виртуальной среде (VirtualBox/VMware)

Задание 3. Разработайте библиотеку функций для манипуляции битами (моделирование работы с регистрами процессора):

- Установка/сброс/инверсия отдельных битов в 32-битном слове
- Извлечение битового поля заданной длины и позиции
- Циклический сдвиг влево/вправо
- Подсчёт установленных битов (population count)

Задание 4.Разработайте программу умножения матриц с двумя вариантами реализации:

- Стандартный алгоритм (i-j-k)
- Оптимизированный с учётом кэш-строк (транспонирование, блочное умножение)

Используя Valgrind (cachegrind), проанализируйте:

- Число промахов кэша данных (D-cache misses)
- Число промахов кэша инструкций (I-cache misses)
- Зависимость производительности от размера матрицы

Задание 5. Реализуйте двусвязный список для хранения структур данных фиксированного размера без использования стандартной библиотеки:

- Выделение памяти через mmap/munmap вместо malloc/free

- Функции вставки, удаления, поиска с $O(1)$ для известного узла
- Итератор для обхода списка
- Проверка целостности структуры (отсутствие циклов, корректность указателей)

Задание 6. На основе предоставленного «багового» кода (программа с утечками памяти и segmentation fault):

- Проведите отладку с использованием GDB (точки останова, просмотр стека, backtrace)
- Выявите утечки памяти с помощью Valgrind (memcheck)
- Проведите профилирование с помощью perf/gprof для выявления «узких мест»
- Подготовьте исправленную версию с Makefile, включающим цели debug, release, profile

Задание 7. Разработайте модуль на ассемблере x86-64 (синтаксис AT&T) с интерфейсом на C:

- Функция вычисления контрольной суммы CRC32 с использованием аппаратной инструкции crc32
- Функция копирования памяти с использованием SIMD-инструкций (SSE/AVX)
- Корректное соблюдение соглашения о вызовах System V AMD64 ABI (регистры, стек, выравнивание)

Обеспечьте сборку через GCC с ассемблерными вставками или отдельным .S файлом.

Задание 8. Используя библиотеку libusb, разработайте приложение для работы с USB HID-устройством (мышь или клавиатура):

- Перечисление всех HID-устройств с фильтрацией по классу
- Асинхронное чтение входных отчётов (input reports) с использованием libusb_submit_transfer
- Визуализация полученных данных (координаты, состояние кнопок) в текстовом режиме

- Корректная обработка отключения устройства и ошибок шины

Задание 9. Выполните анализ предоставленного бинарного файла (ELF):

- Определите архитектуру, тип файла, точку входа (readelf, objdump)
- Проанализируйте таблицы символов и релокации
- Проведите дизассемблирование критических функций (objdump, radare2 или Ghidra)
- Выявите используемые библиотеки (динамические зависимости)
- Определите механизмы защиты (NX, ASLR, stack canaries) через checksec или аналоги

Задание 10. Разработайте многоступенчатый загрузчик (bootloader) для x86 в реальном режиме:

- Stage 1 (512 байт): Загрузочная запись MBR, вывод сообщения на экран через BIOS int 10h, загрузка Stage 2 с диска (int 13h)
- Stage 2: Переключение в защищённый режим (32-bit), настройка GDT, базовый драйвер VGA-текстового режима
- Stage 3 (на C): Простой shell с командами help, reboot, meminfo (определение объёма памяти через E820)

Сборка через кросс-компилятор i686-elf-gcc и линковку с пользовательским linker script.

Примерные вопросы для подготовки к зачету

1. Введение в низкоуровневое программирование
2. Цели и задачи курса
3. Роль курса в подготовке специалиста
4. Сведения об истории развития проблематики дисциплины
5. Обзор и общая характеристика языков программирования
6. Стандарты языков программирования
7. Понятие низкоуровневого программирования
8. Файловые системы с точки зрения системного программиста

9. Конфигурационная информация операционных систем
10. Сборка и установка ОС Линукс (ядра операционной системы)
11. стек протоколов TCP/IP с точки зрения системного программиста
12. Текстовые редактор vi
13. Классическая архитектура ЭВМ, предложенная группой Дж. фон Неймана
14. Арифметико-логическое устройство
15. Основное (оперативное) запоминающее устройство
16. Иерархия памяти
17. Современные структуры микропроцессоров
18. Регистры общего назначения
19. Управление потоком команд
20. Структура программы на языке C
21. Переменные, указатели, объявления в C
22. Преобразование типов данных в C
23. Логические, битовые, операции сдвига, сравнения в C
24. Квалифицирующие типы и спецификаторы функций в C
25. Директивы предпроцессора в C
26. Простые типы
27. Базовые операторы
28. Выражения и операции
29. Стандартные математические функции
30. Операторы выбора
31. Операторы цикла
32. Функции
33. Область видимости
34. Операции прерывания и безусловного перехода
35. Стили программирования
36. Составные типы данных: массивы
37. Объявление и инициализация массивов
38. Обращение к элементам массива
39. Упорядочивание массивов
40. Составные типы данных: структуры и объединения
41. Расположение элементов структур в ОЗУ ЭВМ
42. Составные типы данных: строки
43. Объявление и инициализация строк

44. Функции для работы со строками
45. Составные типы данных: многомерные массивы
46. Инициализация многомерных массивов
47. Операции с многомерными массивами
48. Составные типы данных: битовые поля
49. Составные типы данных: перечисления
50. Указатели и адреса памяти
51. Представление памяти процесса
52. Управление памятью
53. Области данных, текста программы, стека, кучи
54. Модели памяти
55. Динамические массивы
56. Преобразования типов данных
57. Логические, битовые операции, операции сдвига, операции сравнения
58. Квалифицирующие типы и спецификаторы функций
59. Директивы предпроцессора
60. Отличия между стандартами языка Си разных лет
61. Переменные окружения
62. Потоки ввода вывода
63. Передача параметров
64. Этапы создания исполняемого файла
65. Профилирование программы
66. Автоматизация проверки исходных текстов и создания исполняемых файлов
67. Автоматическая генерация документации
68. Отладка без специальных программ
69. Отладка с использованием gdb
70. Поиск утечек памяти valgrind
71. Си и ассемблер
72. Технологии дезассемблирования
73. Ассемблерные вставки в компиляторе gcc
74. Статические библиотеки в С
75. Динамические библиотеки в С
76. Реализация односвязных и многосвязных списков
77. Приёмы оценки вычислительной сложности алгоритмов в системном программировании

- 78. Библиотека libusb
- 79. Получение списка доступных устройств libusb_get_device_list()
- 80. Получение характеристик заданного устройства
- 81. Получение данных от USB манипулятора типа мышь
- 82. Определение libusb_transfer_type() устройства
- 83. Определение libusb_bos_type() устройства
- 84. Определение скорости передачи данных устройства libusb_speed()
- 85. Получение номера устройства на шине libusb_get_bus_number()
- 86. Получение порта устройства на шине libusb_get_port_number()
- 87. Получение родительского устройства libusb_get_parent()
- 88. Получение максимального размера пакета устройства
libusb_get_max_packet_size()

**Промежуточная аттестация обучающихся проводится в формате тестирования.*

8. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины

Основная литература:

1. Рацеев, С. М. Программирование на языке Си [Электронный ресурс] : учебное пособие для вузов / Рацеев С. М. 2-е изд., стер. Санкт-Петербург : Лань, 2023 332 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/351863> ISBN 978-5-507-47236-9. [БИК ID: RU-LAN-BOOK-351863]
2. Ларина, Татьяна Борисовна Низкоуровневые языки : Учебное пособие / Российский университет транспорта (МИИТ) Москва : Федеральное государственное автономное образовательное учреждение высшего образования «Российский университет транспорта», 2018 147 с. ВО - Бакалавриат <https://znanium.com/catalog/document?id=415617>. [БИК ID: RU\{\{infra-m\}\}znanium\{\}bibl\{\}1895286]

Дополнительная литература:

1. Пичугин, В. Н. Теория трансляций и низкоуровневое программирование [Электронный ресурс] : учеб. пособие / Пичугин В. Н., Солдатов А. А. Чебоксары : ЧГУ им. И. Н. Ульянова, 2024 92 с. Книга из коллекции ЧГУ им. И. Н. Ульянова - Информатика <https://e.lanbook.com/book/482207> ISBN 975-8-7677-3830-4. [БИК ID: RU-LAN-BOOK-482207]
2. Трегубов, В. М. Низкоуровневое программирование [Электронный ресурс] : учебно-методическое пособие / Трегубов В. М., Шакирзянов Р. М. Казань : КНИТУ-КАИ, 2017 68 с. Рекомендовано к изданию Учебно-методическим управлением КНИТУ-КАИ Книга из коллекции КНИТУ-КАИ - Информатика <https://e.lanbook.com/book/366461> ISBN 978-5-7579-2250-8. [БИК ID: RU-LAN-BOOK-366461]
3. Бунаков, П. Ю. Машинно-ориентированные языки программирования. Введение в ассемблер [Электронный ресурс] : учебное пособие для вузов / Бунаков П. Ю. Санкт-Петербург : Лань, 2023 144 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/302627> ISBN 978-5-507-45490-7. [БИК ID: RU-LAN-BOOK-302627]
4. Романов, С. Л. Работа в операционной среде Linux: практикум для вузов [Электронный ресурс] : практикум / Романов С. Л. Санкт-Петербург : БГТУ "Военмех" им. Д.

- Ф. Устинова, 2017 74 с. Книга из коллекции БГТУ "Военмех" им. Д. Ф. Устинова - Информатика <https://e.lanbook.com/book/121866>. [БИК ID: RU-LAN-BOOK-121866]
5. Золкин, А. Л. Машинно-ориентированные языки программирования в сфере искусственного интеллекта [Электронный ресурс] : учебное пособие для вузов / Золкин А. Л. Санкт-Петербург : Лань, 2025 168 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/494978> ISBN 978-5-507-52465-5. [БИК ID: RU-LAN-BOOK-494978]

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Информационно-образовательный портал Финансового университета при Правительстве Российской Федерации <http://portal.ufrf.ru/>
2. Электронная библиотека <http://grebennikon.ru>
3. Библиотечно-информационный комплекс Финансового университета при Правительстве РФ <http://www.library.fa.ru>
4. Электронно-библиотечная система издательства «ЮРАЙТ» <https://www.biblio-online.ru/>
5. Электронно-библиотечная система издательства "Лань" <https://e.lanbook.com/>
6. Электронно-библиотечная система BOOK. RU <http://www.book.ru>
7. Электронно-библиотечная система "Университетская библиотека ОНЛАЙН" <http://biblioclub.ru/>
8. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>
(<http://library.fa.ru/files/elibfa.pdf>)
9. • Электронно-библиотечная система Znanium <http://www.znanium.ru/>
10. • Электронно-библиотечная система издательства Проспект <http://ebs.prospekt.org/books>
11. • Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>

10. Методические указания для обучающихся по освоению дисциплины

Основные этапы работы студента по дисциплине «Низкоуровневое программирование»:

1. Предварительная ориентировка в подлежащем изучению учебном материале по программе.
2. Ознакомление с рекомендованной учебной литературой.
3. Слушание и конспектирование лекций, а также выполнение других видов учебной работы.
4. Планирование самостоятельной работы.
5. Обобщение и систематизация информации, почерпнутой из лекций и прочитанной литературы.
6. Выполнение контрольной работы.

Студентам при подготовке к контрольной работе следует использовать нормативные документы Финансового университета, Методические рекомендации по планированию и организации внеаудиторной самостоятельной работы студентов по образовательным программам бакалавриата и магистратуры в Финансовом университете, утвержденные приказом Финуниверситета от 11.05.2021 г. № 1040/о (см. сайт Финансового Университета: на главной странице раздел "Наш университет" → "Единая правовая база Финуниверситета" → "Организация учебного процесса" → "Нормативные документы по самостоятельной работе").

Рекомендации по работе с учебным материалом:

1. Осознавайте наличный уровень полученных вами знаний.
2. В ситуации непонимания нужно выявить тот первичный уровень и факторы непонимания, которые стали препятствием понимания последующего.
3. Задавайте сами себе вопросы и пытайтесь ответить на них.

Рекомендации по работе на лекции и с лекционным материалом:

1. Основная задача на лекции – осмысление излагаемого в ней материала. Для этого необходимо слушать лекцию с самого начала, не упуская общих, ориентирующих в материале рассуждений и установок лектора.
2. Ведение записей на лекции важно и полезно для лучшего осмысливания материала,

для сохранения информации, с целью ее дальнейшего использования.

3. Для облегчения записи рекомендуется применять сокращения повторяющихся терминов или хорошо известных понятий.

Рекомендации по работе с литературой:

1. Если возникли затруднения при разыскивании материала, по какому-либо конкретному вопросу, следует обратиться к предметному указателю, напечатанному, как правило, в конце каждого литературного источника.

2. Предметный указатель – это алфавитный список основных научных понятий (терминов), содержание которых раскрыто в книге, рядом с термином стоят числа, обозначающие номера страниц, на которых изложен материал, относящийся к данному понятию.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем

Комплект лицензионного программного обеспечения:

1. Astra Linux
2. Libre Office
3. Kaspersky

Современные профессиональные базы данных и информационные справочные системы:

1. Информационно-правовая система «Гарант»
2. Информационно-правовая система «Консультант Плюс»
3. Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>

Сертифицированные программные и аппаратные средства защиты информации:

1. не предусмотрены

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

1. **Учебная аудитория** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), набор демонстрационного оборудования (проектор, экран)