

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Галунин Сергей Александрович
Должность: проректор по учебной работе
Дата подписания: 23.12.2025 13:42:26
Уникальный программный ключ:
08ef34338325bdb0ac5a47baa5472ce36cc3fc3b

Приложение к ОПОП
«Разработка программно-
информационных систем»



СПбГЭТУ «ЛЭТИ»
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

МИНОБРНАУКИ РОССИИ

федеральное государственное автономное образовательное учреждение высшего образования
«Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И.Ульянова (Ленина)»
(СПбГЭТУ «ЛЭТИ»)

РАБОЧАЯ ПРОГРАММА

ДИСЦИПЛИНЫ

«НИЗКОУРОВНЕВОЕ ПРОГРАММИРОВАНИЕ»

для подготовки бакалавров

по направлению

09.03.04 «Программная инженерия»

по профилю

«Разработка программно-информационных систем»

Санкт-Петербург

2025

ЛИСТ СОГЛАСОВАНИЯ

Разработчики:

старший преподаватель Халиуллин Р.А.

доцент, к.т.н., доцент Племянников А.К.

Рабочая программа рассмотрена и одобрена на заседании кафедры ИБ

24.01.2025, протокол № 1

Рабочая программа рассмотрена и одобрена учебно-методической комиссией

ФКТИ, 28.01.2025, протокол № 1

Согласовано в ИС ИОТ

Начальник ОМОЛА Загороднюк О.В.

1 СТРУКТУРА ДИСЦИПЛИНЫ

Обеспечивающий факультет	ФКТИ
--------------------------	------

Обеспечивающая кафедра	ИБ
------------------------	----

Общая трудоемкость (ЗЕТ)	3
--------------------------	---

Курс	2
------	---

Семестр	4
---------	---

Виды занятий

Электронные лекции (акад. часов)	34
----------------------------------	----

Электронные практические (академ. часов) (академ. часов)	34
--	----

Иная контактная работа (академ. часов)	1
--	---

Все контактные часы (академ. часов)	1
-------------------------------------	---

Самостоятельная работа, включая часы на контроль (академ. часов)	39
---	----

Всего (академ. часов)	108
-----------------------	-----

Вид промежуточной аттестации

Дифф. зачет (курс)	2
--------------------	---

2 АННОТАЦИЯ ДИСЦИПЛИНЫ

«НИЗКОУРОВНЕВОЕ ПРОГРАММИРОВАНИЕ»

Учебная дисциплина «Низкоуровневое программирование» формирует знания, умения и навыки, необходимые для разработки программного обеспечения с использованием методов и технологий низкоуровневого программирования и исследования функциональности программного обеспечения с использованием методов и технологий реверс-инжиниринга. В рамках учебной дисциплины «Низкоуровневое программирование» изучаются низкоуровневые интерфейсы взаимодействия, системные вызовы в операционных системах, форматы объектных и исполняемых файлов, инструментальные средства реверс-инжиниринга программного обеспечения. Практическая часть курса в виде практических работ нацелена на приобретение и закрепление умений и навыков разрабатывать, отлаживать, проверять работоспособность программного обеспечения, разработанного с помощью методов и технологий низкоуровневого программирования, исследовать функциональность программного обеспечения с использованием методов и технологий реверс-инжиниринга.

SUBJECT SUMMARY

«LOW-LEVEL PROGRAMMING»

The educational discipline "Low-level programming" develops the knowledge, skills and abilities necessary for software development using low-level programming methods and technologies and researching software functionality using reverse engineering methods and technologies. As part of the academic discipline "Low-level programming", low-level interaction interfaces, system calls in operating systems, object and executable file formats, and software reverse engineering tools are studied. The practical part of the course in the form of practical work is aimed at acquiring and consolidating the skills and abilities to develop, debug, test the functionality

of software developed using low-level programming methods and technologies, and explore the functionality of software using reverse engineering methods and technologies.

3 ОБЩИЕ ПОЛОЖЕНИЯ

3.1 Цели и задачи дисциплины

1. Целями изучения учебной дисциплины является приобретение теоретических знаний низкоуровневого программирования и формирование практических умений и навыков использовать изученные методы и технологии низкоуровневого программирования для реализации компьютерных программ (приложений). Теоретический базис учебной дисциплины основывается на знаниях, полученных при изучении программирования, информатики и математики.
2. Задачи изучения учебной дисциплины направлены на формирование профессиональной компетенции -способности разрабатывать информационные модели и применять их для решения задач профессиональной деятельности, в частности, с применением методов, технологий и инструментальных средств низкоуровневого программирования.
3. Учебная дисциплина обеспечивает получение знаний о низкоуровневых интерфейсах взаимодействия, интеграции низкоуровневых конструкций, системных вызовах в операционных системах, форматах объектных и исполняемых файлов для процессорных архитектур x86 (IA-32) и x86-64 (AMD64).
4. Учебная дисциплина вырабатывает умения интегрировать низкоуровневые конструкции в программы (приложения), организовывать низкоуровневое взаимодействие с операционной системой и другими программами (приложениями), изучать код программ (приложений), полученный с помощью инструментальных средств реверс-инжиниринга.
5. Результатом освоения учебной дисциплины является приобретение практических навыков использования инструментальных средств разработки программного обеспечения с использованием методов и технологий низкоуровневого программирования, отладки программного обеспечения, исследования функцио-

нальности программного обеспечения с использованием методов и технологий реверс-инжиниринга.

3.2 Место дисциплины в структуре ОПОП

Дисциплина изучается на основе ранее освоенных дисциплин учебного плана:

1. «Информатика»
2. «Программирование»
3. «Алгоритмы и структуры данных»
4. «Организация ЭВМ и систем»
5. «Основы программирования на языке Ассемблера»

3.3 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения образовательной программы обучающийся должен достичь следующие результаты обучения по дисциплине:

Код компетенции/ индикатора компетенции	Наименование компетенции/индикатора компетенции
ПК-0	Способен разрабатывать информационные модели и применять их для решения задач профессиональной деятельности
<i>ПК-0.1</i>	<i>Знает современные виды информационных моделей, применяемых при решении задач профессиональной деятельности</i>
<i>ПК-0.2</i>	<i>Создает и модифицирует информационные модели для решения задач профессиональной деятельности</i>
<i>ПК-0.3</i>	<i>Применяет информационные модели для решения задач профессиональной деятельности</i>

4 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1 Содержание разделов дисциплины

4.1.1 Наименование тем и часы на все виды нагрузки

№ п/п	Наименование темы дисциплины	ЭЛек, ач	ЭПр, ач	ИКР, ач	СР, ач
1	Введение	2			2
2	Инструментальные средства низкоуровневого программирования и реверс-инжиниринга	4	2		2
3	Низкоуровневые интерфейсы взаимодействия	4	4		4
4	Интеграция низкоуровневых конструкций в приложения, реализованные на языках программирования высокого уровня	4	4		4
5	Низкоуровневое программирование приложений для операционных систем Microsoft Windows	4	6	0.25	6
6	Низкоуровневое программирование приложений для UNIX-подобных операционных систем	4	6	0.25	7
7	Форматы объектных и исполняемых файлов для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	4	6	0.25	6
8	Технологии реверс-инжиниринга	6	6	0.25	6
9	Заключение	2			2
	Итого, ач	34	34	1	39
	Из них ач на контроль	0	0	0	0
	Общая трудоемкость освоения, ач/зе	108/3			

4.1.2 Содержание

№ п/п	Наименование темы дисциплины	Содержание
1	Введение	Структура и содержание курса. Понятия низкоуровневого программирования, языка программирования низкого уровня, машинного кода, реверс-инжиниринга. История появления и развития методов и технологий низкоуровневого программирования и реверс-инжиниринга. Сферы применения низкоуровневого программирования и реверс-инжиниринга.
2	Инструментальные средства низкоуровневого программирования и реверс-инжиниринга	Компилятор (транслятор) ассемблера: назначение, обзор распространенных компиляторов (трансляторов) ассемблера. Отладчик: назначение, обзор распространенных отладчиков. Нех-редактор: назначение, обзор распространенных hex-редакторов. Дизассемблер: назначение, обзор распространенных дизассемблеров.

№ п/п	Наименование темы дисциплины	Содержание
3	Низкоуровневые интерфейсы взаимодействия	Бинарный интерфейс приложения (Application Binary Interface, ABI): понятие, назначение. Соглашение о вызове функции (Function calling convention): понятие, назначение, обзор распространенных соглашений о вызове функции. Поддержка бинарного интерфейса приложения и соглашений о вызове функции в компиляторах.
4	Интеграция низкоуровневых конструкций в приложения, реализованные на языках программирования высокого уровня	Интеграция низкоуровневых конструкций: понятие, назначение. Компоновка объектных модулей, реализованных на ассемблере, с приложениями, реализованными на языках программирования высокого уровня. Ассемблерная вставка: понятие, назначение.
5	Низкоуровневое программирование приложений для операционных систем Microsoft Windows	Специфика низкоуровневого программирования приложений для операционных систем Microsoft Windows. Обзор внутреннего устройства операционных систем Microsoft Windows. Системные вызовы в операционных системах Microsoft Windows. Инструментальные средства низкоуровневого программирования приложений для операционных систем Microsoft Windows.
6	Низкоуровневое программирование приложений для UNIX-подобных операционных систем	Специфика низкоуровневого программирования приложений для UNIX-подобных операционных систем. Обзор внутреннего устройства UNIX-подобных операционных систем. Системные вызовы в UNIX-подобных операционных системах. Инструментальные средства низкоуровневого программирования приложений для UNIX-подобных операционных систем.
7	Форматы объектных и исполняемых файлов для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	Объектный файл: понятие, назначение, обзор распространенных форматов объектных файлов. Исполняемый файл: понятие, назначение. Форматы исполняемых файлов DOS. Формат исполняемых файлов Portable Executable (PE). Формат исполняемых файлов Executable and Linkable Format (ELF).
8	Технологии реверс-инжиниринга	Дизассемблирование: понятие, назначение. Анализ кода в дизассемблере. Декомпиляция. Отладка: понятие, назначение. Анализ кода в отладчике, противодействие обнаружению отладчиков. Деобфускация исполняемого кода.
9	Заключение	Обзор нововведений в методах и технологиях низкоуровневого программирования. Обзор нововведений в инструментальных средствах низкоуровневого программирования и реверс-инжиниринга. Перспективы развития технологий низкоуровневого программирования и реверс-инжиниринга.

4.2 Перечень лабораторных работ

Лабораторные работы не предусмотрены.

4.3 Перечень практических занятий

Наименование практических занятий	Количество ауд. часов
1. Использование инструментальных средств низкоуровневого программирования	2
2. Реализация вызова функции по разным соглашениям о вызове и анализ различий в разных соглашениях о вызове	4
3. Компоновка объектного модуля, реализованного на ассемблере, с приложением, реализованном на C/C++	4
4. Организация взаимодействия с операционной системой Microsoft Windows с помощью системных вызовов	6
5. Организация взаимодействия с UNIX-подобной операционной системой с помощью системных вызовов	6
6. Изучение формата исполняемых файлов Portable Executable (PE)	3
7. Изучение формата исполняемых файлов Executable and Linkable Format (ELF)	3
8. Дизассемблирование и исследование функциональности приложения	6
Итого	34

4.4 Курсовое проектирование

Курсовая работа (проект) не предусмотрены.

4.5 Реферат

Реферат не предусмотрен.

4.6 Индивидуальное домашнее задание

Индивидуальное домашнее задание не предусмотрено.

4.7 Доклад

Доклад не предусмотрен.

4.8 Кейс

Кейс не предусмотрен.

4.9 Организация и учебно-методическое обеспечение самостоятельной работы

Изучение дисциплины сопровождается самостоятельной работой студентов с рекомендованными преподавателем литературными источниками и информационными ресурсами сети Интернет.

Планирование времени для изучения дисциплины осуществляется на весь период обучения, предусматривая при этом регулярное повторение пройденного материала. Обучающимся, в рамках внеаудиторной самостоятельной работы, необходимо регулярно дополнять сведениями из литературных источников материал, законспектированный на лекциях. При этом на основе изучения рекомендованной литературы целесообразно составить конспект основных положений, терминов и определений, необходимых для освоения разделов учебной дисциплины.

Особое место уделяется консультированию, как одной из форм обучения и контроля самостоятельной работы. Консультирование предполагает особым образом организованное взаимодействие между преподавателем и студентами, при этом предполагается, что консультант либо знает готовое решение, которое он может предписать консультируемому, либо он владеет способами деятельности, которые указывают путь решения проблемы.

Самостоятельное изучение студентами теоретических основ дисциплины обеспечено необходимыми учебно-методическими материалами (учебники, учебные пособия, конспект лекций и т.п.), выполненными в печатном или электронном виде.

По каждой теме содержания рабочей программы могут быть предусмотр-

рены индивидуальные домашние задания (расчетно-графические работы, рефераты, конспекты изученного материала, доклады и т.п.).

Изучение студентами дисциплины сопровождается проведением регулярных консультаций преподавателей, обеспечивающих практические занятия по дисциплине, за счет бюджета времени, отводимого на консультации (внеаудиторные занятия, относящиеся к разделу «Самостоятельные часы для изучения дисциплины»).

В случае применения ДОТ с заменой аудиторных занятий:

Самостоятельной записи на курс нет. Студент заходит на курс, используя логин/пароль от единой учетной записи университета (единый логин и пароль). Каждую неделю будет доступна новая тема курса: видеолекции, кратко раскрывающие содержание каждой темы, презентации и конспекты, с которыми обучающиеся смогут ознакомиться в любое удобное время. Все темы включают практические занятия, которые предусматривают самостоятельное выполнение заданий, а также задания с автоматической проверкой, результаты которых учитываются при общей аттестации полученных знаний. В конце каждой лекции необходимо пройти небольшой контрольный тест, который покажет насколько усвоен предложенный материал. Рекомендуем изучать материал последовательно, что существенно облегчит работу. У каждого контрольного задания имеется своя форма (тест или практическое задание) есть срок выполнения (окончательный срок), по истечении которого даже правильные ответы система принимать не будет! В расписании курса указан окончательный срок каждого задания, который варьируется от двух до четырех недель в зависимости от его сложности. Весь учебный курс рассчитан на 16 недель. Его итоги будут подведены в течение нескольких недель после его окончания.

Текущая СРС	Примерная трудоемкость, ач
Работа с лекционным материалом, с учебной литературой	6

Текущая СРС	Примерная трудоемкость, ач
Опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	6
Самостоятельное изучение разделов дисциплины	6
Выполнение домашних заданий, домашних контрольных работ	0
Подготовка к лабораторным работам, к практическим и семинарским занятиям	6
Подготовка к контрольным работам, коллоквиумам	0
Выполнение расчетно-графических работ	0
Выполнение курсового проекта или курсовой работы	0
Поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	0
Работа над междисциплинарным проектом	0
Анализ данных по заданной теме, выполнение расчетов, составление схем и моделей, на основе собранных данных	0
Подготовка к зачету, дифференцированному зачету, экзамену	15
ИТОГО СРС	39

5 Учебно-методическое обеспечение дисциплины

5.1 Перечень основной и дополнительной литературы, необходимой для освоения дисциплины

№ п/п	Название, библиографическое описание	К-во экз. в библ.
Основная литература		
1	Зубков С. В. Assembler. Для DOS, Windows и Unix : учебное пособие / С. В. Зубков, 2008. -640 с. -Текст : электронный.	неогр.
2	Калашников О. Ассемблер — это просто. Учимся программировать. 2 изд. / О. Калашников, 2014. -336 с. -Текст : электронный.	неогр.
3	Касперски К. Искусство дизассемблирования / К. Касперски, Е. Рокко, 2009. -896 с. -Текст : электронный.	неогр.
4	Касперски К. Техника отладки программ без исходных текстов / К. Касперски, 2005. -832 с. -Текст : электронный.	неогр.
5	Панов А. Реверсинг и защита программ от взлома / А. Панов, 2006. -256 с. -Текст : электронный.	неогр.
6	Пирогов В. Ассемблер для Windows. 4 изд. / В. Пирогов, 2007. -896 с. -Текст : электронный.	неогр.
7	Внутреннее устройство Windows. 7-е изд. / М. Руссинович, Д. Соломон, А. Ионеску и др., 2021. -944 с. -Текст : электронный.	неогр.
8	Б. Уорд Внутреннее устройство Linux / Уорд Б., 2021. -384 с. -Текст : электронный.	неогр.
9	Юров, Виктор Иванович. Assembler : учеб. пособие для вузов по направлению подгот. дипломир. специалистов "Информатика и вычисл. техника" / В.И.Юров, 2003. -636 с.	123
10	Юров, Виктор Иванович. Assembler : практикум : учеб. пособие для вузов по направлению подгот. дипломир. специалистов "Информатика и вычисл. техника" / В.И.Юров, 2004. -398 с.	117
Дополнительная литература		
1	Аблязов Р. З. Программирование на ассемблере на платформе x86-64 / Р. З. Аблязов, 2011. -304 с. -Текст : электронный.	неогр.
2	Бунаков П. Ю. Машинно-ориентированные языки программирования. Введение в ассемблер : учебное пособие для вузов / П. Ю. Бунаков, 2023. -144 с. -Текст : электронный.	неогр.
3	Ермакова Н. А. Основные алгоритмы обработки массивов на примере языка программирования ассемблер [Электронный ресурс] : учебно-методическое пособие для студентов направления 23.05.05 «системы обеспечения движения поездов» специализация «телекоммуникационные системы и сети железнодорожного транспорта», «автоматика и телемеханика на железнодорожном транспорте» (специалитет), 2019. -52 с.	неогр.

№ п/п	Название, библиографическое описание	К-во экз. в библ.
4	Йо В. Г. Программирование на ассемблере x64. От начального уровня до профессионального использования AVX64 : руководство / В. Г. Йо, 2021. -332 с. -Текст : электронный.	неогр.
5	Куссвюрм Д. Профессиональное программирование на ассемблере x64 с расширениями AVX, AVX2 и AVX-512 / Д. Куссвюрм, 2021. -628 с. -Текст : непосредственный.	неогр.
6	Максимов А. В. Оптимальное проектирование ассемблерных программ математических алгоритмов: теория, инженерные методы : учебное пособие / А. В. Максимов, 2021. -192 с. -Текст : электронный.	неогр.
7	Максимов А. В. Оптимальное проектирование ассемблерных программ математических алгоритмов: лабораторный практикум : учебное пособие / А. В. Максимов, Е. А. Максимова, 2022. -128 с. -Текст : электронный.	неогр.
8	Фуско Дж. Linux. Руководство программиста. / Дж. Фуско, 2021. -448 с. -Текст : электронный.	неогр.
9	Штеренберг С. И. Ассемблер в задачах защиты информации [Электронный ресурс] : учебное пособие, 2019. -82 с.	неогр.
10	Щупак Ю. А. Win32 API. Разработка приложений для Windows / Ю. А. Щупак, 2010. -592 с. -Текст : электронный.	неогр.

5.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», используемых при освоении дисциплины

№ п/п	Электронный адрес
1	AMD64 Architecture Programmer's Manual, Volumes 1-5 https://www.amd.com/content/dam/amd/en/documents/processor-tech-docs/programmer-references/40332.pdf
2	AMD Developer Central https://www.amd.com/en/developer.html
3	Developer Guides, Manuals & ISA Documents https://developer.amd.com/resources/developer-guides-manuals/
4	Flat Assembler https://flatassembler.net/
5	Ghidra Software Reverse Engineering Framework https://github.com/NationalSecurityAgency/ghidra
6	HT Editor https://hte.sourceforge.net/
7	Intel® 64 and IA-32 Architectures Software Developer's Manual Combined Volumes: 1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, and 4 https://cdrdv2.intel.com/v1/dl/getContent/671200
8	PE Format https://docs.microsoft.com/en-us/windows/win32/debug/pe-format
9	SCO Developer Specs https://www.sco.com/developers/devspecs/
10	X86 Opcode and Instruction Reference http://ref.x86asm.net/

5.3 Адрес сайта курса

Адрес сайта курса: <https://vec.etu.ru/moodle/course/view.php?id=21263>

6 Критерии оценивания и оценочные материалы

6.1 Критерии оценивания

Для дисциплины «Низкоуровневое программирование» предусмотрены следующие формы промежуточной аттестации: зачет с оценкой.

Зачет с оценкой

Оценка	Описание
Неудовлетворительно	Выставляется студенту, продемонстрировавшему существенные пробелы в знаниях основного учебного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий.
Удовлетворительно	Выставляется студенту, продемонстрировавшему знание основного учебного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющемуся с выполнением заданий, предусмотренных программой, обладающему необходимыми знаниями, но допустившему неточности в ответах на аттестационном испытании и при выполнении учебных заданий.
Хорошо	Выставляется студенту, продемонстрировавшему полное знание учебного материала, успешно выполнившему предусмотренные программой задачи, освоившему основную рекомендованную литературу, показавшему систематический характер знаний по дисциплине и способному к их самостоятельному пополнению и обновлению в ходе дальнейшей учебы и профессиональной деятельности.
Отлично	Выставляется студенту, продемонстрировавшему всестороннее систематическое знание учебного материала, умение свободно выполнять практические задания, освоившему основную литературу и ознакомившемуся с дополнительной литературой, рекомендованной рабочей программой дисциплины, усвоившему взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии, проявившему творческие способности в понимании, изложении и использовании учебного материала.

Особенности допуска

Для получения допуска к зачету с оценкой каждому студенту необходимо посетить не менее 80% лекций (в дистанционном формате) и выполнить все контрольные и практические работы. Зачет с оценкой проводится в письменной или смешанной форме (устные ответы на вопросы и письменное решение задачи), по усмотрению преподавателя, проводящего зачет с оценкой. При проведении зачета с оценкой в письменной форме каждому студенту выделяется не менее 2 (двух) академических часов для ответов на вопросы и решения задачи. При проведении зачета с оценкой в смешанной форме каждому студенту выделяется не менее 1 (одного) астрономического часа для подготовки устных ответов на вопросы и письменного решения задачи. Во время зачета с оценкой всем студентам запрещено использовать любые источники информации (в том числе электронные устройства). По усмотрению преподавателя, проводящего зачет с оценкой, все студенты вправе использовать твердые рукописные конспекты лекций.

6.2 Оценочные материалы для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине

Вопросы к дифф.зачету

№ п/п	Описание
1	Низкоуровневое программирование: понятие. Сходства и различия между низкоуровневым и высокоуровневым программированием.
2	Реверс-инжиниринг: понятие. Сферы, в которых применяется реверс-инжиниринг.
3	Нех-редактор: понятие, назначение. Дизассемблер: понятие, назначение.
4	Сферы, в которых применяются низкоуровневое программирование и реверс-инжиниринг.
5	Понятия бинарного интерфейса приложения и соглашения о вызове функции. Наиболее распространенные соглашения о вызове функции.
6	Особенности внутреннего устройства операционной системы Microsoft Windows. Основные подсистемы операционной системы Microsoft Windows.
7	Системные вызовы в операционной системе Microsoft Windows. Способы обращения к системным вызовам в операционной системе Microsoft Windows.

8	Особенности внутреннего устройства UNIX-подобной операционной системы. Основные подсистемы UNIX-подобной операционной системы.
9	Системные вызовы в UNIX-подобной операционной системе. Способы обращения к системным вызовам в UNIX-подобной операционной системе.
10	Понятие исполняемого файла. Понятие объектного файла.
11	Основные форматы объектных файлов. Основные форматы исполняемых файлов.
12	Сходства и различия между форматами исполняемых файлов DOS, Portable Executable (PE), Executable and Linkable Format (ELF).
13	Понятие дизассемблирования. Понятие декомпиляции.
14	Сходства и различия между дизассемблированием и декомпиляцией. Сферы применения дизассемблирования и декомпиляции.
15	Понятие статического анализа кода. Понятие динамического анализа кода.
16	Сходства и различия между статическим и динамическим анализом кода. Сферы применения статического и динамического анализа кода.
17	Понятие антиотладочного приема. Способы обнаружения отладчика и противодействия обнаружению отладчика.
18	Понятие обфускации кода. Понятие деобфускации кода.
19	Тенденции и перспективы развития инструментальных средств низкоуровневого программирования.
20	Тенденции и перспективы развития инструментальных средств реверс-инжиниринга.

Форма билета

Министерство науки и высшего образования Российской Федерации
 ФГАОУ ВО «Санкт-Петербургский государственный электротехнический
 университет «ЛЭТИ» имени В. И. Ульянова (Ленина)»

БИЛЕТ № 1

Дисциплина **Низкоуровневое программирование ФКТИ**

1. Сходства и различия между статическим и динамическим анализом кода.
2. Понятие исполняемого файла.
3. Задача.

УТВЕРЖДАЮ

Заведующий кафедрой

Образцы задач (заданий) для контрольных (проверочных) работ

Вариант задания для контрольной работы № 1:

Опишите основные элементы, в совокупности образующие соглашение о вызове функции, на примере наиболее распространенных соглашений о вызове функции.

Вариант задания для контрольной работы № 2:

Перечислите основные способы обращения к системным вызовам в операционной системе Microsoft Windows, с описанием преимуществ и недостатков каждого способа.

Вариант задания для контрольной работы № 3:

Перечислите и кратко опишите основные форматы объектных и исполняемых файлов, с указанием на то, в каких операционных системах используются перечисленные форматы объектных и исполняемых файлов.

Вариант задания для контрольной работы № 4:

Перечислите основные инструментальные средства реверс-инжиниринга, с описанием их назначения и функциональности.

Примеры контрольных вопросов для защиты практической работы:

1. Какие системные вызовы или иные низкоуровневые обращения использовались в программе, реализованной или изученной в рамках выполнения практической работы?
2. Почему использовались именно эти системные вызовы или низкоуровневые обращения?
3. Можно ли было реализовать программу с использованием других системных вызовов или низкоуровневых обращений?
4. В какой части кода программы происходит низкоуровневое обращение

для чтения входных данных?

5. В какой части кода программы происходит низкоуровневое обращение для записи выходных данных?

Весь комплект контрольно-измерительных материалов для проверки сформированности компетенции (индикатора компетенции) размещен в закрытой части по адресу, указанному в п. 5.3

6.3 График текущего контроля успеваемости

Неделя	Темы занятий	Вид контроля
6	Низкоуровневые интерфейсы взаимодействия	
7		Контрольная работа
10	Низкоуровневое программирование приложений для операционных систем Microsoft Windows	
11		Контрольная работа
14	Форматы объектных и исполняемых файлов для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	
15		Контрольная работа
16	Технологии реверс-инжиниринга	
17		Контрольная работа

6.4 Методика текущего контроля

На лекционных занятиях:

Текущий контроль включает в себя контроль посещаемости (не менее 80% дистанционных занятий), по результатам которого студент получает допуск к зачету с оценкой.

При выполнении практической работы:

— Порядок выполнения практических работ, подготовки отчетов и их защиты

В процессе обучения по учебной дисциплине «Низкоуровневое программирование» студент обязан выполнить все практические работы. Под выполнением практической работы подразумевается получение задания на практическую работу, подготовка к выполнению практической работы, выполнение практической работы в соответствии с полученным заданием, подготовка отчета по выполненной практической работе и ее защита. Выполнение практических работ студентами осуществляется индивидуально. Оформление отчета студентами осуществляется индивидуально, в соответствии с требованиями, изложенными в:

1. ГОСТ Р 7.0.97-2016 «Национальный стандарт Российской Федерации. Система стандартов по информации, библиотечному и издательскому делу.

Организационно-распорядительная документация. Требования к оформлению документов»;

2. ГОСТ 7.32-2017 «Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления»;
3. Требования к оформлению научно-технических отчетов, принятые в СПбГ-ЭТУ «ЛЭТИ» им. В. И. Ульянова (Ленина).

Отчет оформляется после выполнения практической работы и представляется преподавателю на проверку. После проверки отчет по практической работе либо возвращается (при наличии замечаний) на доработку, либо подписывается к защите. На защите практической работы студент должен иметь при себе распечатанный отчет. По усмотрению преподавателя, принимающего защиту практической работы, вместо наличия распечатанного отчета допускается возможность предварительной (до защиты) загрузки электронной версии отчета в ЭИОС, в форматах PDF и OpenDocument/Microsoft Office. Практические работы защищаются студентами индивидуально. Каждый студент получает контрольные вопросы по теоретической части и/или по практической части, после чего ему предоставляется время для подготовки ответов. При обсуждении ответов преподаватель может задать студенту несколько уточняющих вопросов. В случае, если в ответах на вопросы студент показал достаточное знание материала, практическая работа считается защищенной. На защите практической работы студент должен показать:

1. Понимание методики исследования и знание особенностей ее применения;
2. Понимание и умение объяснять особенности применяемых методов, возможные области их применения и т. д.;
3. Умение давать качественную и количественную оценку полученных результатов и прогнозировать реакции исследуемого объекта на различные воздействия;

4. Навыки и умения, приобретенные при выполнении практической работы. Примеры контрольных вопросов приведены в п.6.2.

Текущий контроль включает в себя выполнение, сдачу в установленные сроки всех отчетов по практическим работам и их защиту, по результатам которой студент получает допуск к зачету с оценкой.

На практических (семинарских) занятиях:

Текущий контроль включает в себя контроль посещаемости (не менее 80% дистанционных занятий), по результатам которого студент получает допуск к зачету с оценкой. В ходе проведения семинарских и практических занятий целесообразно привлечение студентов к как можно более активному участию в дискуссиях, решении задач, обсуждениях и т. д. при этом активность студентов также может учитываться преподавателем, как один из способов текущего контроля на практических занятиях.

По самостоятельной работе студентов:

Контроль самостоятельной работы студентов осуществляется на лекционных и практических занятиях студентов по методикам, описанным выше.

При выполнении контрольной работы:

В процессе обучения по учебной дисциплине «Низкоуровневое программирование» студент обязан выполнить все контрольные работы.

Под выполнением контрольной работы подразумевается получение задания на контрольную работу, выполнение контрольной работы в соответствии с полученным заданием. Выполнение контрольных работ студентами осуществляется индивидуально. В случае, если при выполнении контрольной работы студент показал достаточное знание материала, контрольная работа считается выполненной. При выполнении контрольной работы студент должен показать:

1. Понимание теоретических и практических положений, затрагиваемых в задании на контрольную работу;

2. Навыки и умения, приобретенные при выполнении контрольной работы. Примеры заданий для контрольных работ приведены в разделе с критериями оценивания и оценочными материалами.

Текущий контроль включает в себя выполнение по графику текущего контроля успеваемости всех контрольных работ, по результатам выполнения которых студент получает допуск к зачету с оценкой.

Критерии оценивания контрольной работы:

- «Неудовлетворительно» — выставляется студенту, не выполнившему контрольную работу или при выполнении контрольной работы показавшему существенные пробелы в знании основных теоретических и практических положений, затрагиваемых в задании на контрольную работу;
- «Удовлетворительно» — выставляется студенту, при выполнении контрольной работы показавшему знание основных теоретических и практических положений, затрагиваемых в задании на контрольную работу;
- «Хорошо» — выставляется студенту, при выполнении контрольной работы показавшему знание большинства теоретических и практических положений, затрагиваемых в задании на контрольную работу;
- «Отлично» — выставляется студенту, при выполнении контрольной работы показавшему знание всех теоретических и практических положений, затрагиваемых в задании на контрольную работу.

7 Описание информационных технологий и материально-технической базы

Тип занятий	Тип помещения	Требования к помещению	Требования к программному обеспечению
Лекция	Лекционная аудитория	Количество посадочных мест в соответствии с контингентом, рабочее место преподавателя (стол, кресло, компьютер с доступом в электронную информационно-образовательную среду университета (ЭИОС) и сеть «Интернет»), проектор, экран, микрофон, меловая или маркерная доска.	Программное обеспечение для демонстрации презентаций в формате PDF или в формате Microsoft PowerPoint, операционная система Microsoft Windows или Linux.
Практические занятия	Аудитория	Количество посадочных мест в соответствии с контингентом, компьютеры в соответствии с контингентом, рабочее место преподавателя (стол, кресло, компьютер с доступом в электронную информационно-образовательную среду университета (ЭИОС) и сеть «Интернет»), меловая или маркерная доска.	Операционная система Microsoft Windows или Linux, офисный пакет (Microsoft Office или LibreOffice), инструментальные средства разработки и отладки программного обеспечения (интегрированные среды разработки (IDE) Code::Blocks и Fresh, компилятор ассемблера FASM, hex-редактор HT Editor, отладчик x64dbg, дизассемблер Ghidra).

Самостоятельная работа	Помещение для самостоятельной работы	Оснащено компьютерами с доступом в электронную информационно-образовательную среду университета (ЭИОС) и сеть «Интернет».	Операционная система Microsoft Windows или Linux, офисный пакет (Microsoft Office или LibreOffice), инструментальные средства разработки и отладки программного обеспечения (интегрированные среды разработки (IDE) Code::Blocks и Fresh, компилятор ассемблера FASM, hex-редактор HT Editor, отладчик x64dbg, дизассемблер Ghidra).
------------------------	--------------------------------------	---	--

8 Адаптация рабочей программы для лиц с ОВЗ

Адаптированная программа разрабатывается при наличии заявления со стороны обучающегося (родителей, законных представителей) и медицинских показаний (рекомендациями психолого-медико-педагогической комиссии). Для инвалидов адаптированная образовательная программа разрабатывается в соответствии с индивидуальной программой реабилитации.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Дата	Изменение	Дата и номер протокола заседания УМК	Автор	Начальник ОМОЛА