

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Галунин Сергей Александрович  
Должность: проректор по учебной работе  
Дата подписания: 26.11.2024 14:26:37  
Уникальный программный ключ:  
08ef34338325bdb0ac5a47baa5472ce36cc3fc3b

Приложение к ОПОП  
«Организация и программирова-  
ние интеллектуальных 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

**«ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ АССЕМБЛЕРА»**

для подготовки бакалавров

по направлению

09.03.01 «Информатика и вычислительная техника»

по профилю

**«Организация и программирование интеллектуальных систем»**

Санкт-Петербург

2024

## **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  
17.01.2024, протокол № 1

Рабочая программа рассмотрена и одобрена учебно-методической комиссией  
ФКТИ, 24.01.2024, 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	3

## **Виды занятий**

Лекции (академ. часов)	34
Практические занятия (академ. часов)	34
Иная контактная работа (академ. часов)	1
Все контактные часы (академ. часов)	69
Самостоятельная работа, включая часы на контроль (академ. часов)	39
Всего (академ. часов)	108

## Вид промежуточной аттестации

Лифф зачет (курс) 2

## **2 АННОТАЦИЯ ДИСЦИПЛИНЫ**

### **«ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ АССЕМБЛЕРА»**

Учебная дисциплина «Основы программирования на языке Ассемблера» формирует знания, умения и навыки, необходимые для разработки программного обеспечения на ассемблере — языке программирования низкого уровня. В рамках учебной дисциплины «Основы программирования на языке Ассемблера» изучаются основные инструкции ассемблера. Практическая часть курса в виде практических работ нацелена на приобретение и закрепление умений и навыков разрабатывать, отлаживать, проверять работоспособность, модифицировать программное обеспечение на ассемблере.

### **SUBJECT SUMMARY**

### **«BASICS OF ASSEMBLY LANGUAGE PROGRAMMING»**

The academic discipline "Basics of Assembly language programming" forms the knowledge, skills and abilities necessary for developing software in assembly language — a low-level programming language. As part of the discipline "Basics of Assembly language programming", the basic instructions of the assembler 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 operability, modify the software in assembly language.

## **3 ОБЩИЕ ПОЛОЖЕНИЯ**

### **3.1 Цели и задачи дисциплины**

1. Целями изучения учебной дисциплины является приобретение теоретических знаний конструкций ассемблера — языка программирования низкого уровня и формирование практических умений и навыков использовать изученные конструкции ассемблера для реализации компьютерных программ (приложений). Теоретический базис учебной дисциплины основывается на знаниях, полученных при изучении программирования, информатики и математики.
2. Задачами изучения учебной дисциплины является формирование следующих профессиональных компетенций: способность создавать программы на языке низкого уровня, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ.
3. Учебная дисциплина обеспечивает получение знаний основных инструкций процессоров архитектур x86 (IA-32) и x86-64 (AMD64), форматов инструкций процессоров архитектур x86 (IA-32) и x86-64 (AMD64).
4. Учебная дисциплина вырабатывает умения реализовывать алгоритмы на ассемблере, изучать исходный код программ (приложений), реализованных на ассемблере, а также изучать код дизассемблированных программ (приложений).
5. Результатом освоения учебной дисциплины является приобретение практических навыков использования инструментальных средств разработки программного обеспечения на ассемблере, отладки программного обеспечения.

### **3.2 Место дисциплины в структуре ОПОП**

Дисциплина изучается на основе ранее освоенных дисциплин учебного плана:

1. «Информатика»
2. «Программирование»
3. «Информационные технологии»

и обеспечивает изучение последующих дисциплин:

1. «Операционные системы»
2. «Защита компьютерной информации»
3. «Микропроцессорные системы»

### **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:

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

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>Лек, ач</b>	<b>Пр, ач</b>	<b>ИКР, ач</b>	<b>СР, ач</b>
1	Введение	2	0		2
2	Инструментальные средства разработки на ассемблере	4	2		3
3	Введение в процессорные архитектуры x86 (IA-32) и x86-64 (AMD64)	4	2		3
4	Структура программы на ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	4	2		3
5	Типы данных, переменные и константы в ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	2	4	0.25	4
6	Инструкции передачи данных в ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	2	4		4
7	Арифметические инструкции, логические инструкции и инструкции сдвига в ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	2	4	0.25	4
8	Условные инструкции и инструкции перехода в ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	2	4		4
9	Циклические и цепочечные инструкции в ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	2	4	0.25	4
10	Функции в ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	4	4		2
11	Формат инструкций процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	4	4	0.25	4
12	Заключение	2	0		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

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>Содержание</b>
1	Введение	Структура и содержание курса. Понятия программирования на ассемблере, языка программирования низкого уровня, машинного кода. История появления и развития языков программирования низкого уровня. Сфера применения языков программирования низкого уровня.
2	Инструментальные средства разработки на ассемблере	Компилятор (транслятор) ассемблера: назначение, обзор распространенных компиляторов (трансляторов) ассемблера. Отладчик: назначение, обзор распространенных отладчиков.
3	Введение в процессорные архитектуры x86 (IA-32) и x86-64 (AMD64)	История появления и развития процессорных архитектур x86 (IA-32) и x86-64 (AMD64). Разрядности и режимы. Регистр: понятие, назначение, обзор основных регистров. Регистр флагов: понятие, назначение, биты регистра флагов. Стек: понятие, назначение. Инструкция: понятие, назначение, наборы инструкций.
4	Структура программы на ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	Директивы и макросы компилятора (транслятора). Объявления формата выходного файла, разрядности, секций (сегментов), меток. Импортирование и экспорт символов. Intel-синтаксис. AT&T-синтаксис.
5	Типы данных, переменные и константы в ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	Типы данных. Объявление переменных и констант. Массивы и строки. Представление данных в оперативной памяти. Порядок байтов: понятие, прямой порядок байтов — Big-Endian (BE), обратный порядок байтов — Little-Endian (LE).
6	Инструкции передачи данных в ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	Основные инструкции передачи данных. Операнды. Виды адресации. Операции со стеком.
7	Арифметические инструкции, логические инструкции и инструкции сдвига в ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	Арифметические инструкции. Логические инструкции. Инструкции сдвига.
8	Условные инструкции и инструкции перехода в ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	Условные инструкции. Инструкции перехода.
9	Циклические и цепочечные инструкции в ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	Циклические инструкции. Цепочечные инструкции.

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>Содержание</b>
10	Функции в ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	Реализация внутренних (собственных) функций. Вызов внутренних (собственных) функций. Импортирование внешних функций. Вызов внешних (импортированных) функций.
11	Формат инструкций процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	Понятие формата инструкции. Префикс инструкции. Опкод инструкции. Байт ModR/M. Байт SIB. Кодирование операндов и режимов адресации.
12	Заключение	Обзор нововведений процессорных архитектур x86 (IA-32) и x86-64 (AMD64). Обзор нововведений в инструментальных средствах разработки на ассемблере. Перспективы развития технологий разработки на ассемблере.

#### **4.2 Перечень лабораторных работ**

Лабораторные работы не предусмотрены.

#### **4.3 Перечень практических занятий**

<b>Наименование практических занятий</b>	<b>Количество ауд. часов</b>
1. Создание проекта в интегрированной среде разработки (IDE) на ассемблере	2
2. Разработка на ассемблере приложения для выполнения операций с переменными	4
3. Отладка приложения, разработанного на ассемблере, в отладчике исполняемого кода	4
4. Разработка на ассемблере приложения для выполнения арифметических операций	4
5. Разработка на ассемблере приложения для работы с массивами и строками	6
6. Реализация внутренней (собственной) функции на ассемблере и ее вызов	4
7. Использование в приложении, реализованном на языке высокого уровня, функции, реализованной на ассемблере	4
8. Кодирование и декодирование инструкций процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	6
<b>Итого</b>	<b>34</b>

#### **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.

<b>Текущая СРС</b>	<b>Примерная трудоемкость, ач</b>
Работа с лекционным материалом, с учебной литературой	12
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	9
<b>ИТОГО СРС</b>	<b>39</b>

## 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	Пирогов В. Ассемблер для Windows. 4 изд. / В. Пирогов, 2007. -896 с. - Текст : электронный.	неогр.
4	Юров, Виктор Иванович. Assembler : учеб. пособие для вузов по направлению подгот. дипломир. специалистов "Информатика и вычисл. техника" / В.И.Юров, 2003. -636 с.	123
5	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. Машино-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 «телеинформатика и телемеханика на железнодорожном транспорте» (специалитет), 2019. -52 с.	неогр.
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	Куссвюром Д. Профессиональное программирование на ассемблере x64 с расширениями AVX, AVX2 и AVX-512 / Д. Куссвюром, 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.

№ п/п	Название, библиографическое описание	К-во экз. в библ.
8	Штеренберг С. И. Ассемблер в задачах защиты информации [Электронный ресурс] : учебное пособие, 2019. -82 с.	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 <a href="https://www.amd.com/content/dam/amd/en/documents/processor-tech-docs/programmer-references/40332.pdf">https://www.amd.com/content/dam/amd/en/documents/processor-tech-docs/programmer-references/40332.pdf</a>
2	AMD Developer Central <a href="https://www.amd.com/en/developer.html">https://www.amd.com/en/developer.html</a>
3	Flat Assembler <a href="https://flat assembler.net/">https://flat assembler.net/</a>
4	Intel® 64 and IA-32 Architectures Software Developer’s Manual Combined Volumes: 1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, and 4 <a href="https://cdrdv2.intel.com/v1/dl/getContent/671200">https://cdrdv2.intel.com/v1/dl/getContent/671200</a>
5	x64dbg <a href="https://github.com/x64dbg/x64dbg">https://github.com/x64dbg/x64dbg</a>
6	X86 Opcode and Instruction Reference <a href="http://ref.x86asm.net/">http://ref.x86asm.net/</a>

## 5.3 Адрес сайта курса

Адрес сайта курса: <https://vec.etu.ru/moodle/course/view.php?id=15072>

## **6 Критерии оценивания и оценочные материалы**

### **6.1 Критерии оценивания**

Для дисциплины «Основы программирования на языке Ассемблера» предусмотрены следующие формы промежуточной аттестации: зачет с оценкой.

#### **Зачет с оценкой**

<b>Оценка</b>	<b>Описание</b>
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 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**

<b>№ п/п</b>	<b>Описание</b>
1	Программирование на ассемблере: понятие. Сходства и различия между программированием на ассемблере и программированием на языках высокого уровня.
2	Особенности программирования на ассемблере. Сфера, в которых применяется программирование на ассемблере.
3	Компилятор (транслятор) ассемблера: понятие, назначение. Отладчик: понятие, назначение.
4	Понятие разрядности процессора. Режим функционирования процессора: понятие, режимы функционирования процессоров архитектур x86 (IA-32) и x86-64 (AMD64).
5	Регистр: понятие, основные регистры процессоров архитектур x86 (IA-32) и x86-64 (AMD64). Инструкция процессора: понятие, основные инструкции процессоров архитектур x86 (IA-32) и x86-64 (AMD64).
6	Директива компилятора (транслятора) ассемблера: понятие, назначение. Макрос компилятора (транслятора) ассемблера: понятие, назначение.

7	Импортирование и экспортование символов: понятие, назначение. Сходства и различия между Intel-синтаксисом и AT&T-синтаксисом.
8	Особенности синтаксиса в компиляторах (трансляторах) ассемблера. Структура программы на ассемблере.
9	Основные типы данных. Примеры объявления переменных и констант.
10	Понятие порядка байтов. Отличие порядка байтов Big-Endian (BE) от порядка байтов Little-Endian (LE).
11	Инструкции процессора для передачи данных. Понятие операнда.
12	Сходства и различия между видами адресации. Стек: понятие, назначение.
13	Инструкции процессора для арифметических операций. Особенности арифметических операций с знаковыми и беззнаковыми значениями.
14	Логические инструкции процессора. Инструкции процессора для сдвига.
15	Инструкции процессора для проверок условий. Инструкции процессора для условных переходов.
16	Инструкции процессора для организации циклов. Инструкции процессора для цепочечной обработки данных в оперативной памяти.
17	Понятие внутренней (собственной) функции. Особенности разработки внутренней (собственной) функции.
18	Понятие внешней (импортированной) функции. Особенности вызовов внутренних (собственных) и внешних (импортированных) функций.
19	Формат инструкции процессора: понятие. Составные элементы инструкций процессорных архитектур x86 (IA-32) и x86-64 (AMD64).
20	Основные нововведения процессорных архитектур x86 (IA-32) и x86-64 (AMD64). Тенденции и перспективы развития инструментальных средств разработки на ассемблере.

## Форма билета

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

---

### БИЛЕТ № 1

Учебная дисциплина **Основы программирования на языке Ассемблера** ФКТИ

1. Директивы и макросы транслятора ассемблера.
2. Условные инструкции. Инструкции перехода.

3. Задача.

УТВЕРЖДАЮ

Заведующий кафедрой

Е.Г. Воробьев

## **Образцы задач (заданий) для контрольных (проверочных) работ**

### **Вариант задания для контрольной работы № 1:**

Опишите основные структурные элементы исходного кода программы на ассемблере, с указанием директив и макросов компилятора (транслятора), используемых в этих элементах.

### **Вариант задания для контрольной работы № 2:**

Перечислите основные инструкции передачи данных, с описанием каждой инструкции и описанием операндов каждой инструкции.

### **Вариант задания для контрольной работы № 3:**

Перечислите основные арифметические инструкции, логические инструкции и инструкции сдвига, с описанием каждой инструкции и описанием операндов каждой инструкции.

### **Вариант задания для контрольной работы № 4:**

Перечислите основные условные инструкции и инструкции перехода, с описанием каждой инструкции и описанием операндов каждой инструкции.

### **Примеры контрольных вопросов для защиты практической работы:**

1. Какие инструкции процессора использовались в программе, разработанной в рамках выполнения практической работы?
2. Почему использовались именно эти инструкции процессора?
3. Можно ли было разработать программу с использованием других инструкций процессора?
4. В какой части кода программы происходит обработка входных данных?
5. В какой части кода программы происходит вывод 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**

<b>Неделя</b>	<b>Темы занятий</b>	<b>Вид контроля</b>
7	Структура программы на ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	Контрольная работа
8		
9	Инструкции передачи данных в ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	Контрольная работа
10		
11	Арифметические инструкции, логические инструкции и инструкции сдвига в ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	Контрольная работа
12		
13	Условные инструкции и инструкции перехода в ассемблере для процессорных архитектур x86 (IA-32) и x86-64 (AMD64)	Контрольная работа
14		

## **6.4 Методика текущего контроля**

### **На лекционных занятиях:**

Текущий контроль включает в себя контроль посещаемости (не менее 80% 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 «Основы программирования на языке Ассемблера» 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 СПбГТУ «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 электронную информационно-образовательную среду университета (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 разделе с критериями оценивания и оценочными материалами.

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% 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 «Основы программирования на языке Ассемблера» 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

<b>Тип занятий</b>	<b>Тип помещения</b>	<b>Требования к помещению</b>	<b>Требования к программному обеспечению</b>
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.	Компьютер с программным обеспечением для демонстрации презентаций в формате PDF или в формате Microsoft PowerPoint.
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.	Офисный пакет, инструментальные средства разработки и отладки программного обеспечения.
Самостоятельная работа	Помещение для самостоятельной работы	Оснащено компьютерами с доступом в электронную информационно-образовательную среду университета (ЭИОС) и сеть «Интернет».	Офисный пакет, инструментальные средства разработки и отладки программного обеспечения.

## **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**

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