Федеральное государственное автономное образовательное учреждение высшего образования
«Санкт-Петербургский политехнический университет Петра Великого»

УТВЕРЖДАЮ
Директор ИКНК
_____ Д.П. Зегжда
«17» июня 2024 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

«Низкоуровневое программирование»

Разработчик	Высшая школа компьютерных технологий и информационных систем
Направление (специальность) подготовки	09.03.01 Информатика и вычислительная техника
Наименование ООП	09.03.01_01 Разработка компьютерных систем
Квалификация (степень) выпускника	**бакалавр**
Образовательный стандарт	**СУОС**
Форма обучения	**Очная**

СОГЛАСОВАНО
Руководитель ОП
_____ Р.В. Цветков
«26» марта 2024 г.

Соответствует СУОС
Утверждена протоколом заседания высшей школы "ВШКТиИС"
от «26» марта 2024 г. № 1

РПД разработал:
Доцент, к.т.н., доц. В.А. Сушников

1. Цели и планируемые результаты изучения дисциплины

Цели освоения дисциплины

1. Получение фундаментальных сведений о принципах функционирования и основах программирования (универсальных) ЭВМ.
2. Получение сведений об организации инструментов разработки программного обеспечения (препроцессоры, компиляторы, ассемблеры и дизассемблеры, библиотекари, компоновщики, отладчики, утилиты анализа бинарных файлов, интегрированные среды), навыков их использования.
3. Освоение языка программирования Си и стандартной библиотеки.

Результаты обучения выпускника

Код	Результат обучения (компетенция) выпускника ООП
ОПК-10	**Проверяет работоспособность и осуществляет рефакторинг кода программного обеспечения**
ИД-1 ОПК-10	Разрабатывает процедуры проверки работоспособности и измерения характеристик программного обеспечения
ОПК-4	**Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью**
ИД-2 ОПК-4	Оформляет и документирует программный код в соотвествии с принятыми в организации стандартами
ОПК-9	**Способен осваивать методики использования программных средств для решения практических задач**
ИД-2 ОПК-9	Выполняет написание программного кода с использованием языков программирования, определения и манипулирования данными, используя выбранную систему контроля версий и инструментальные программные средства

Планируемые результаты изучения дисциплины

знания:

- Знает инструментарий для создания текстов программ; методы повышения читаемости программного кода; нормативные документы, определяющие требования к оформлению программного кода
- порядок ручного и автоматизированного тестирования программного обеспечения
- языки и инструментальные средства программирования сценариев администрирования операционных систем

умения:

- определять порядок тестирования для различных программ
- выполнять программирование, отладку и сопровождение сценариев администрирования операционных систем

навыки:

- Владеет навыками создания, структурирования и форматирования исходного программного кода
- создание автоматизированных тестов в соответствующих библиотеках
- программирование сценариев администрирования операционных систем на языках их командных оболочек

2. Место дисциплины в структуре ООП

В учебном плане дисциплина «Низкоуровневое программирование» относится к модулю «Модуль цифровых компетенций».

Изучение дисциплины базируется на результатах освоения следующих дисциплин:

- Алгоритмизация и программирование

3. Распределение трудоёмкости освоения дисциплины по видам учебной работы и формы текущего контроля и промежуточной аттестации

3.1. Виды учебной работы

Виды учебной работы	Трудоемкость по семестрам
	Очная форма
Лекционные занятия	30
Лабораторные занятия	6
Самостоятельная работа	58
Промежуточная аттестация (экзамен)	0
Промежуточная аттестация (зачет с оценкой)	6
Курсовое проектирование	8
Общая трудоемкость освоения дисциплины	108, ач
	3, зет

3.2. Формы текущего контроля и промежуточной аттестации

Формы текущего контроля и промежуточной аттестации	Количество по семестрам
	Очная форма
Текущий контроль	
Контрольные, шт.	1
Курсовые проекты, шт.	1
Промежуточная аттестация	
Зачеты с оценкой, шт.	1

4. Содержание и результаты обучения

4.1 Разделы дисциплины и виды учебной работы

№ раздела	Разделы дисциплины, мероприятия текущего контроля	Очная форма		
		Лек, ач	Лаб, ач	СР, ач

1.	Вычислимость			
1.1.	Понятие эффективной процедуры. Машина Тьюринга, универсальная машина Тьюринга, проблема останова.	2	1	2
1.2.	Мю-рекурсивные функции Клини, эквивалентность формализмов. Тезис Черча-Тьюринга.	2	1	2
2.	Основы организации и программирования ЭВМ			
2.1.	История ЭВМ, принципы фон Неймана, основы организации и программирования ЭВМ на уровне машинных кодов (на примере ЭВМ первого поколения EDSAC).	3	0	3
2.2.	Организация сложных программ (на примере EDSAC): макросы и подпрограммы; передача управления при вызове и возврате из подпрограмм, передача аргументов и возврат результатов; библиотеки подпрограмм.	3	0	3
2.3.	Развитие архитектуры системы команд в ЭВМ второго и третьего поколений. Индексные регистры, регистры общего назначения, режимы адресации данных. Система команд гипотетической машины MIX (Д. Кнут). Косвенные и относительные переходы.	2	0	2
2.4.	Рекурсивный вызов подпрограмм. Использование стека для организации подпрограмм.	2	0	2
2.5.	Архитектура системы команд современного процессора (на примере RISC-V).	2	0	2
3.	Средства разработки программного обеспечения для современных ЭВМ			
3.1.	Компилятор, ассемблер, ABI для современного процессора (на примере системы gcc для архитектуры RISC-V).	2	1	2
3.2.	Организация раздельной компиляции модулей, редактор связей и загрузчик, статические и динамические библиотеки (на примере RISC-V).	2	1	2
3.3.	Системы автоматической сборки (на примере make). Отладчик. Интегрированные системы разработки.	2	0	2
4.	Язык программирования Си			
4.1.	Введение. Обзор действующего стандарта языка. Препроцессор языка Си, организация исходного текста модуля.	2	1	2
4.2.	Типы данных. Выражения. Операторы.	2	0	3
4.3.	Указатели и массивы, строки. Динамическое выделение памяти.	2	0	2

4.4.	Стандартная библиотека.	2	1	2
Итого по видам учебной работы:		30	6	58
Зачеты с оценкой, ач				0
Часы на контроль, ач				0
Курсовое проектирование		8		
Промежуточная аттестация (зачет с оценкой)		6		
Общая трудоёмкость освоения: ач / зет		108 / 3		

4.2. Содержание разделов и результаты изучения дисциплины

Раздел дисциплины	Содержание
1. Вычислимость	
1.1. Понятие эффективной процедуры. Машина Тьюринга, универсальная машина Тьюринга, проблема останова.	Идея эффективной процедуры на примере алгоритма Евклида (вычисления) и кодирования МТК-2 (обработка текста). Необходимость в строгом определении эффективной процедуры (проблема разрешимости предикатного исчисления). Определение машины Тьюринга, кодирование ленты, машина Тьюринга, реализующая алгоритм Евклида. Кодирование описания машины, универсальная машина Тьюринга. Неразрешимость проблемы останова и связанных проблем.
1.2. Мю-рекурсивные функции Клини, эквивалентность формализмов. Тезис Черча-Тьюринга.	Примитивно рекурсивные функции. Примеры примитивно рекурсивных определений: арифметические операции (сложения, вычитание, умножение), логические операции (И, ИЛИ, НЕ), определения вариантами. Язык LOOP. Оператор минимизации. Понятие об арифметизации. Эквивалентность формализмов. Тезис Черча-Тьюринга, примеры его применения.
2. Основы организации и программирования ЭВМ	
2.1. История ЭВМ, принципы фон Неймана, основы организации и программирования ЭВМ на уровне машинных кодов (на примере ЭВМ первого поколения EDSAC).	Предыстория ЭВМ: механические счетные машины, аналитическая машина Беббиджа, дифференциальный анализатор Буша и другие аналоговые машины, перфокарты и перфоленты. История ЭВМ: электронная машина Атанасова, ЭВМ ENIAC, принципы фон Неймана, ЭВМ EDVAC, ЭВМ IAS. Организация и архитектура системы команд ЭВМ EDSAC. Загрузчик Initial Orders 1. Использование симулятора EDSAC. Пример программы для EDSAC, модификация адресов в инструкциях чтения и записи данных.
2.2. Организация сложных программ (на примере EDSAC): макросы и подпрограммы; передача управления при вызове и возврате из подпрограмм, передача аргументов и возврат результатов; библиотеки подпрограмм.	Необходимость в повторном использовании кода. Открытые подпрограммы (макросы), необходимость модификации адресов, поддержка подпрограмм загрузчиком Initial Orders 2. Закрытые подпрограммы. Организация передачи управления, модификация адресов в инструкциях перехода для возврата из подпрограммы. Способы передачи аргументов и возврата результата. Идея библиотеки подпрограмм, обзор библиотеки подпрограмм EDSAC.

2.3. Развитие архитектуры системы команд в ЭВМ второго и третьего поколений. Индексные регистры, регистры общего назначения, режимы адресации данных. Система команд гипотетической машины MIX (Д. Кнут). Косвенные и относительные переходы.	Использование индексных регистров для доступа к данным без модификации инструкций, добавление B-line в архитектуру EDSAC. Основные режимы адресации: прямая, непосредственная, неявная, регистровая, индексная, косвенная. Иерархия памяти, "узкое горлышко" фон Неймана. Регистры общего назначения, выгрузка регистров. Относительные переходы, понятие PIC-кода. Косвенные переходы. Устранение необходимости в модификации инструкций для возврата из подпрограмм. Типичная система команд ЭВМ 60-х - 70-х годов XX века на примере гипотетической машины MIX ("Искусство программирования" Д.Кнута).
2.4. Рекурсивный вызов подпрограмм. Использование стека для организации подпрограмм.	Рекурсия, пример рекурсивного алгоритма: решение задачи о ханойских башнях. Стек возвратов (А.Тьюринг). Использование стека для вычисления выражений (Самельсон-Бауэр). Использование стека для вызова подпрограмм (Э.Дейкстра). Передача аргументов через стек. Размещение в стеке локальных переменных. Терминология. Хвостовая рекурсия. Организация адресного пространства программы. Стеки в многопоточных приложениях, split-стек, оценка и контроль размера стека.
2.5. Архитектура системы команд современного процессора (на примере RISC-V).	Открытая архитектура RISC-V. Спецификации RISC-V. Разрядность, регистры, представление целых чисел. Форматы инструкций. Инструкции регистр-регистр, инструкции с непосредственным операндом. Инструкции загрузки и сохранения данных. Инструкции безусловной передачи управления, косвенные переходы. Инструкции условной передачи управления, относительные переходы. Инструкции LUI, AUIPC. Адресация байтов: порядок байтов, выравнивание.
3. Средства разработки программного обеспечения для современных ЭВМ	
3.1. Компилятор, ассемблер, ABI для современного процессора (на примере системы gcc для архитектуры RISC-V).	Ассемблер и ABI RISC-V. Функции регистров. Организация стека. Разбор примеров фрагментов программ на языке Си и результатов их компиляции в ассемблерный код.

3.2. Организация раздельной компиляции модулей, редактор связей и загрузчик, статические и динамические библиотеки (на примере RISC-V).	Необходимость в раздельной компиляции. Разделение задач между ассемблером и редактором связей. Объектные файлы. Секции, таблица символов, таблица перемещений. Статические библиотеки. Алгоритм поиска объектных файлов редактором связей. Организация динамических библиотек, преимущества и недостатки.
3.3. Системы автоматической сборки (на примере make). Отладчик. Интегрированные системы разработки.	Необходимость в системах автоматической сборки. Основные сведения и пример использования make. Принципы работы отладчиков. Отладочная информация в объектных и исполнимых файлах. Понятие об интегрированных системах разработки.
4. Язык программирования Си	
4.1. Введение. Обзор действующего стандарта языка. Препроцессор языка Си, организация исходного текста модуля.	История языка Си и операционной системы Unix. Диалект K&R. Стандарты языка. Язык C++. Обзор действующего стандарта: понятия единицы трансляции, среды трансляции, среды исполнения; разделы стандарта; нотация описания синтаксиса языка в стандарте; лексические принципы языка. Препроцессор языка Си. Директивы включения и условной компиляции. Макросы, макросы с параметрами. Организация исходного текста программного модуля.
4.2. Типы данных. Выражения. Операторы.	Базовые типы языка, представления типов. Обычные арифметические преобразования типов. Перечисления. Выражения, приоритет и ассоциативность операций. Операторы (statements) языка. Структурные типы данных, объединения.
4.3. Указатели и массивы, строки. Динамическое выделение памяти.	Понятие указателя, передача аргументов по адресу. Квалификатор const. Динамическое выделение памяти, интерфейс менеджера памяти. Строки в языке Си. Указатели на функции. Функции обратного вызова. Реализация интерфейсов на основе указателей на функции.
4.4. Стандартная библиотека.	Функции с переменным числом аргументов. Стандартный ввод-вывод, файловый ввод-вывод. Строковые функции. Библиотека математических функций.

5. Образовательные технологии

1. Для реализации курса используются традиционные образовательные технологии: лекции в сочетании с лабораторными работами.
2. Самостоятельное изучение методических материалов и рекомендуемой профессиональной литературы
3. Выполнение курсового проекта по индивидуальному заданию

6. Лабораторный практикум

№ раздела	Наименование лабораторных работ	Трудоемкость, ач
		Очная форма
1.	Реализация машины Тьюринга, решающей поставленную задачу (варианты).	1
2.	Разработка программы для EDSAC (варианты).	2
3.	Разработка и согласование технического задания на курсовой проект (постановка задачи, форматы входных и выходных файлов, ограничения).	1
4.	Изучение инструментов разработки пакета gcc и системы автоматической сборки make.	2
	Итого часов	6

7. Практические занятия

Не предусмотрено

8. Организация и учебно-методическое обеспечение самостоятельной работы

Примерное распределение времени самостоятельной работы студентов

Вид самостоятельной работы	Примерная трудоемкость, ач
	Очная форма
Текущая СР	
работа с лекционным материалом, с учебной литературой	9
опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	0
самостоятельное изучение разделов дисциплины	6
выполнение домашних заданий, домашних контрольных работ	0
подготовка к лабораторным работам, к практическим и семинарским занятиям	0
подготовка к контрольным работам, коллоквиумам	0
Итого текущей СР:	15
Творческая проблемно-ориентированная СР	
выполнение расчётно-графических работ	0
выполнение курсового проекта или курсовой работы	16
поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	0
работа над междисциплинарным проектом	0
исследовательская работа, участие в конференциях, семинарах, олимпиадах	0
анализ данных по заданной теме, выполнение расчётов, составление схем и моделей на основе собранных данных	0
Итого творческой СР:	16
Общая трудоемкость СР:	58

9. Учебно-методическое обеспечение дисциплины

9.1. Адрес сайта курса

http://kspt.icc.spbstu.ru/course/lowlevelprog

9.2. Рекомендуемая литература

Основная литература

№	Автор, название, место издания, издательство, год (годы) издания	Год изд.	Источник
1	Керниган Б., Ритчи Д. Язык программирования Си: Санкт-Петербург: Невский Диалект, 2001.	2001	ИБК СПбПУ
2	Паттерсон Д., Хеннесси Д., Вильчинский Н. Архитектура компьютера и проектирование компьютерных систем: М. [и др.]: Питер, 2012.	2012	ИБК СПбПУ

Дополнительная литература

№	Автор, название, место издания, издательство, год (годы) издания	Год изд.	Источник
1	Минский М., Овсиевич Б.Л., Розенблюм Л.Я. Вычисления и автоматы: Москва: Мир, 1971.	1971	ИБК СПбПУ
2	Таненбаум Э. Архитектура компьютера: Санкт-Петербург: Питер, 2002. URL: http://www.piter.com	2002	ИБК СПбПУ
3	Кнут Д. Искусство программирования для ЭВМ. Т. 1 Основные алгоритмы: Москва: Мир, 1976.	1976	ИБК СПбПУ

Ресурсы Интернета

1. Ссылки и материалы даны на сайте курса: http://kspt.icc.spbstu.ru/course/lowlevelprog

9.3. Технические средства обеспечения дисциплины

Симулятор машин Тьюринга (http://matinf.igpu.ru/simulator/tm.html).

Симулятор EDSAC (http://www.dcs.warwick.ac.uk/~edsac/).

Пакет средств разработки gcc.

10. Материально-техническое обеспечение дисциплины

Компьютерный класс, оснащенный компьютерами (процессор x64, не менее 2 ГБ ОЗУ, не менее 32 ГБ дискового пространства) с 64-разрядной ОС Windows не ниже версии Windows 7, с установленным симулятором EDSAC (http://www.dcs.warwick.ac.uk/~edsac/), пакетом средств разрабоки mingw, доступом к сети Интернет.

11. Критерии оценивания и оценочные средства

11.1. Критерии оценивания

Для дисциплины «Низкоуровневое программирование» формой аттестации является зачёт с оценкой. Дисциплина реализуется с применением системы индивидуальных достижений.

Текущий контроль успеваемости

Максимальное значение персонального суммарного результата обучения (ПСРО) по приведенной шкале - 100 баллов
Максимальное количество баллов приведенной шкалы по результатам прохождения двух точек контроля - 80 баллов.

Подробное описание правил проведения текущего контроля с указанием баллов по каждому контрольному мероприятию и критериев выставления оценки размещается в СДО в навигационном курсе дисциплины.

Промежуточная аттестация по дисциплине

Максимальное количество баллов по результатам проведения аттестационного испытания в период промежуточной аттестации – 20 баллов приведенной шкалы.

Промежуточная аттестация по дисциплине проводится в соответствии с расписанием.

Экзамен состоит в выполнении практического задания (написание текста простой программы на языке Си, определение результата выполнения программы, поиск ошибок во фрагменте программы и пр.) и устном ответе на вопросы билета (по 1 вопросу из разделов 1-3 курса).

При выполнении практического задания допускается обращаться к стандарту языка (но не другой литературе). При проверке выполненного задания контролируется способность студента обосновывать свою позицию ссылками на стандарт.

Результаты промежуточной аттестации, определяются на основе баллов, набранных в рамках применения, СИД

Баллы по приведенной шкале в рамках применения СИД (ПСРО+ ПА)	**Оценка по результатам промежуточной аттестации**
	Экзамен/диф.зачет/зачет
0 - 60 баллов	Неудовлетворительно/не зачтено

Баллы по приведенной шкале в рамках применения СИД (ПСРО+ ПА)	Оценка по результатам промежуточной аттестации
	Экзамен/диф.зачет/зачет
61 - 75 баллов	Удовлетворительно/зачтено
76 - 89 баллов	Хорошо/зачтено
90 и более	Отлично/зачтено

11.2. Оценочные средства

Оценочные средства по дисциплине представлены в фонде оценочных средств, который является неотъемлемой частью основной образовательной программы и размещается в электронной информационно-образовательной среде СПбПУ на портале etk.spbstu.ru.

12. Методические рекомендации по организации изучения дисциплины

При подготовке лекционного материала особое внимание следует уделить интеграции сведений, данных в разделах курса. Так, при изложении принципов фон Неймана следует обратить внимание на связь с идеями Тьюринга (Davis "Engines of logic"), при рассмотрении подпрограмм (в особенности, открытых) - к практике построения машин Тьюринга из логически завершенных "фрагментов", при рассмотрении рекурсивных подпрограмм указывать на связь с мю-рекурсивными функциями, при рассмотрении перемещений (relocation) необходимо указать на связь с системой "плавающих адресов" EDSAC и принципами фон Неймана, при рассмотрении вопросов использования неинициализированных локальных переменных Си - на подлежащий механизм стека и т.д.

В лекционном материале примеры (фрагменты) программ на языке Си используются, начиная с раздела 2 курса. В разделе 4 курса упор должен быть сделан на использование стандарта языка для разрешения возникающих на практике вопросов.

Задания на лабораторные работы по EDSAC должны быть связаны с обработкой массивов данных в памяти машины, так что для их выполнения необходимо использовать модификацию (адресной части) инструкций. Использование этого механизма на практике подготавливает студента к восприятию режимов индексной и косвенной адресации, механизмов редактирования связей и механизма указателей.

Темы курсовых работ должны быть связаны с реализацией симуляторов моделей вычислимости, архитектур ЭВМ или построением интерпретаторов или компиляторов. Желательно обеспечить самостоятельный выбор студентом темы курсовой работы (обеспечив необходимое число вариаций предложенных заданий). Во всех курсовых проектах должен использоваться

файловый ввод-вывод и использование динамически выделяемой памяти. Сборка программ должна осуществляться автоматически с помощью make или аналогичных систем. Разработка должна вестись с использованием системы управления версиями.

Примерный перечень вопросов по разделу 1 курса:

1.1) Понятие эффективной процедуры. Машина Тьюринга, решающая заданную задачу (варианты).

1.2) Универсальная машина Тьюринга (с использованием методических материалов). Тезис Черча-Тьюринга.

1.3) Проблема останова и связанные проблемы. Связь с оператором минимизации.

1.4) Понятие эффективной процедуры. Мю-рекурсивные функции Клини.

1.5) Элементарные арифметические и логические примитивно рекурсивные функции.

1.6) Эквивалентность формализмов машин Тьюринга и мю-рекурсивных функций. Тезис Черча-Тьюринга.

Примерный перечень вопросов по разделу 2 курса:

2.1) Принципы фон Неймана. Узкое горлышко фон Неймана.

2.2) Система команд EDSAC (с использованием методических материалов).

2.3) Оперирование массивами данных в EDSAC. Индексный режим адресации и B-line.

2.4) Организация открытых подпрограмм в EDSAC. Аналоги открытых подпрограмм в машинах Тьюринга и языке Си.

2.5) Организация закрытых подпрограмм в EDSAC.

2.6) Система команд EDSAC (с использованием методических материалов).

2.7) Изменения архитектуры системы команд, устраняющие необходимость в модификации инструкций в прикладных программах.

2.8) Рекурсивные подпрограммы, рекурсивное решение заданной задачи (варианты).

2.9) Рекурсивный алгоритм разбора/вычисления значения выражения.

2.10) Использование стека для вызова подпрограмм. Стек возвратов, хвостовая рекурсия, размещение локальных переменных в стеке.

2.11) Использование стека для вызова подпрограмм. Передача аргументов и возврат значений, размещение локальных переменных в стеке.

2.12) Архитектуры системы команд с регистрами общего назначения. Использование регистров при организации подпрограмм.

2.13) Система команд RISC-V (с использованием методических материалов). Форматы инструкций.

2.14) Система команд RISC-V (с использованием методических материалов). Арифметические и логические инструкции.

2.15) Система команд RISC-V (с использованием методических материалов). Команды загрузки и сохранения.

2.16) Система команд RISC-V (с использованием методических материалов). Команды перехода.

2.17) Порядок байтов (endianness). Выравнивание.

Примерный перечень вопросов по разделу 3 курса:

3.1) ABI RISC-V (с использованием методических материалов).

3.2) Разбор результата компиляции программы на Си в ассемблер RISC-V ((с использованием методических материалов; варианты).

3.3) Раздельная компиляция. Таблица символов, таблица перемещений, функции редактора связей.

3.4) Раздельная компиляция. Статические библиотеки.

3.5) Раздельная компиляция. Динамические библиотеки.

3.6) Раздельная компиляция. Системы автоматической сборки.

3.7) Раздельная компиляции на примере языка Си. Проблема контроля типов. Организация исходного текста программного модуля.

3.8) Основные инструменты разработки ПО. Состав, функции и связи.

Примерный перечень задач по разделу 4 курса: соответствует задачнику "The C Puzzle Book" (A.Feuer), при адаптации задач для действующего стандарта языка, а также онлайн-сборнику задач "C Puzzles" (chortle.ccsu.edu/CPuzzles/CPuzzlesMain.html).

13. 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.