

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«УНИВЕРСИТЕТ УПРАВЛЕНИЯ «ТИСБИ»**

Кафедра информационных технологий

Утверждаю
Зав. кафедрой
О.В.Федорова
Протокол заседания
кафедры № 10
от 06.04.2026



Рабочая программа дисциплины

Наименование дисциплины	Методы трансляции
Направление подготовки	09.03.04 Программная инженерия
Профиль подготовки	Программное обеспечение информационных систем
Год набора	2023, 2024, 2025, 2026

Составитель:

Ст. преп., Ахтямов Р.Р.

Казань

Содержание

1. Цели и задачи учебной дисциплины	3
2. Место дисциплины в структуре ОПОП	3
3. Требования к результатам освоения дисциплины	4
4. Структура и содержание дисциплины	5
4.1 Модульно-тематический план и пояснительная записка с указанием этапов формирования компетенций	5
4.2 Содержание дисциплины по темам (разделам)	7
4.3 Планы практических занятий	10
4.4 Планы практической подготовки/лабораторных занятий	12
5. Учебно-методическое обеспечение самостоятельной работы студентов	13
6. Учебно-методическое и информационное обеспечение дисциплины	14
7. Материально-техническое обеспечение дисциплины	14
8. Оценка компетенций по изучаемой дисциплине	15
Приложение 1. Методические указания для обучающихся по освоению дисциплины	17
Приложение 2. Фонд оценочных средств для проведения текущей и промежуточной аттестации по дисциплине	

1. Цели и задачи учебной дисциплины

Дисциплина является одной из основных при подготовке бакалавров по направлению 09.03.04 «Программная инженерия».

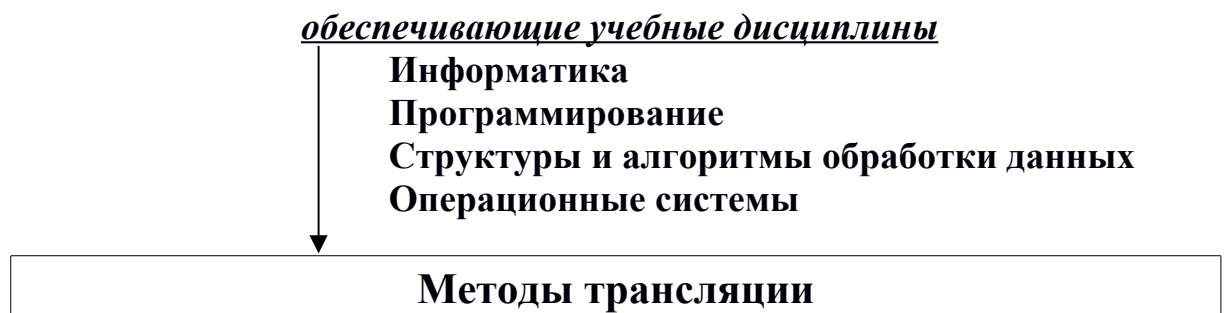
Целью дисциплины является изучение основных компонентов системного программного обеспечения, таких как низкоуровневые языки программирования и трансляторы для языков высокого уровня. В первой части дается введение в языки ассемблерного типа и принципы построения ассемблирующих программ как простейших трансляторов. Во второй части на основе минимально необходимых сведений из области формального описания языков программирования дается описание всех основных этапов работы трансляторов – лексического, синтаксического и семантического анализа, а также генерации выходного кода.

Задачи дисциплины:

Изучив данный курс, студенты должны иметь представление об ассемблерах и ассемблирующих программах и основах разработки трансляторов для языков высокого уровня; знать методы реализации основных этапов работы трансляторов и уметь применять их в практических целях.

2. Место дисциплины в структуре ОПОП

Данная дисциплина относится к группе дисциплин по выбору в части, формируемой участниками образовательного процесса блока 1 учебного плана подготовки бакалавра по направлению 09.03.04. «Программная инженерия» и находится во взаимосвязи с дисциплинами согласно схеме:



Производственная практика (преддипломная практика)

До начала изучения дисциплины у студента должны быть сформированы компоненты компетенций, полученные в результате изучения дисциплин Информатика, Программирование, Структуры и алгоритмы обработки данных, Операционные системы.

3. Требования к результатам освоения содержания дисциплины

Дисциплина «Методы трансляции» участвует в формировании следующих компетенций в соответствии с ФГОС ВО по направлению «Программная инженерия»:

ПК-4 «Способен разрабатывать компоненты системного программного обеспечения»

После освоения дисциплины студент должен получить следующие образовательные результаты, соотнесенные с индикаторами достижения компетенций

Декомпозиция компетенций

Индикаторы	Результаты обучения по дисциплине
ПК-4.1. Разрабатывает компоненты системного программного обеспечения	ПК-4.1.3.1. Знает элементы системного программного обеспечения: языки ассемблерного типа, структуру и разновидности программ- трансляторов ПК-4.1.У.1. Умеет разрабатывать компоненты программ-трансляторов ПК-4.1.В.1. Владеет навыками разработки простых ассемблирующих программ, лексических и синтаксических анализаторов

4. Структура и содержание дисциплины

4.1. Модульно - тематический план и пояснительная записка с указанием этапов формирования компетенций

Общая трудоемкость дисциплины составляет 3 зачетных единицы (108 академических часов)

Модульная разбивка учебной дисциплины						
Направление «Программная инженерия»						
Дисциплина « Системное программное обеспечение»						
Наименование модулей	Количество ауд. часов		Самостоятельная работа Очное/ заочное	Всего часов		Формируемые индикаторы
	Лекции Очное/ заочное	Практики Очное/ заочное				
Модуль 1: «Введение в языки ассемблерного уровня»						
Тема 1: Особенности и структура ассемблерных программ	2/2		1/8	3/10	ПК-4.1	
Тема 2: Простейшие команды (перемещение, арифметика)	2		8/10	10/10		
Тема 3: Переходы, циклы, ветвления	2		10/10	12/10		

Тема 4: Обработка массивов	2		9/12	11/12	
Тема 5: Стек и подпрограммы	2		9/11	11/11	
Тема 6: Основы разработки ассемблирующих программ-трансляторов	4/2	4/2	6/6	14/10	
Модуль 2: «Введение в разработку программ-трансляторов»					
Тема 1. Назначение и структура трансляторов	2/2		5/5	7/7	ПК-4.1
Тема 2. Основы формального описания языков программирования	2/2		5/9	7/11	
Тема 3. Лексический анализ	2	1/1	5/7	8/8	
Тема 4: Синтаксический анализ	2	3/1	5/5	10/6	
Тема 5: Основы генерации выходного кода	2		5/5	7/5	
Подготовка к зачету			8/8	8/8	
Всего	24/8	8/4	76/96	108/108	

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Принцип и обоснование деления дисциплины на модули

Данный модульный курс состоит из двух модулей, порядок освоения которых выстраивает траекторию и этапы формирования заявленных компетенций или их составляющих.

Модуль 1 «Введение в языки ассемблерного уровня» включает в себя 6 учебных тем.

В результате прохождения модуля студент должен:

- знать особенности оформления программ на языке ассемблера, основные команды для простейших вычислений, организации ветвлений и циклов, реализацию подпрограмм, а также принципы генерации кода для перечисленных конструкций
- уметь разрабатывать основные модули ассемблирующих программ для генерации машинного кода
- иметь навыки разработки простейших ассемблирующих программ-трансляторов.

Модуль 2 «Введение в разработку программ-трансляторов» включает в себя 5 учебных тем.

В результате прохождения первого модуля студент должен:

- знать структуру и функции программ-трансляторов, основы формального описания языков программирования, назначение и принципы реализации программ лексического и синтаксического анализа;
- уметь выполнять лексический анализ исходного текста, строить деревья грамматического разбора в качестве основы синтаксического анализа.
- иметь навыки разработки лексических и синтаксических анализаторов.

Уровень освоения полученных знаний, умений и навыков проверяется тестовым опросом, выполнением типовых практических заданий и ответами на теоретические и практические вопросы.

Данное деление дисциплины на модули активизирует самостоятельную работу студентов, повышает интенсивность и системность учебной работы, регулирует контроль учебной деятельности студентов в течении семестров, усиливает мотивацию студентов к изучению учебного материала.

4.2. Содержание дисциплины по темам (разделам)

Модуль 1. «Введение в языки ассемблерного уровня»

Тема 1. Особенности и структура ассемблерных программ

Место языков ассемблерного уровня в общей структуре программного обеспечения. Наиболее распространенные семейства процессоров: семейство Intel и семейство ARM. Программная модель процессоров семейства Intel.

Особенности оформления ассемблерных программ. Раздел данных и раздел команд. Директивы описания данных и примеры их использования. Структура строк в разделе описания команд: метки, мнемокоды, операнды.

Тема 2. Простейшие команды

Необходимость перемещения данных. Формат команды MOV. Правила использования. Примеры использования команд перемещения. Команды сложения и вычитания целых чисел. Правила использования. Возможные особые ситуации и их обработка. Команды умножения и деления целых чисел. Правила использования. Особые ситуации.

Тема 3. Переходы, циклы, ветвления

Изменение порядка выполнения команд, регистр Instruction Pointer (IP). Безусловные и условные переходы. Команда сравнения. Варианты реализации условных переходов. Моделирование условного оператора.

Организация циклов в ассемблерных программах. Реализация циклов

с предусловием и постусловием. Реализация цикла, управляемого счетчиком (два способа). Организация вложенных циклов.

Тема 4. Обработка массивов

Особенности обработки массивов в ассемблерных программах. Понятие об индексной адресации. Назначение и использование индексных регистров. Примеры обработки массивов с использованием индексной адресации. Особенности обработки двумерных массивов.

Понятие о косвенной адресации. Команда загрузки адреса и ее использование. Обработка массивов с использованием косвенной адресации. Примеры программ с косвенной адресацией.

Тема 5. Стек и подпрограммы

Назначение стековой памяти и ее организация. Команды занесения данных в стек и извлечения из стека, особенности выполнения команд. Примеры использования стековой памяти.

Правила объявления подпрограмм в языке ассемблера. Вызов подпрограммы и выполняемые при этом действия. Возврат из подпрограммы и выполняемые действия. Реализация вложенных вызовов.

Передача параметров через регистры. Соглашения о регистрах. Передача параметров через стек. Основные правила взаимодействия подпрограмм.

Тема 6. Основы разработки ассемблирующих программ-трансляторов

Основные функции ассемблирующих программ. Обработка раздела описания данных. Таблица имен областей памяти и Счетчик Адреса. Алгоритм генерации машинного представления данных.

Обработка раздела команд. Необходимые таблицы: Таблица Команд, Таблица Регистров, Таблица Имен. Алгоритм генерации кода для простейших программ с линейной структурой.

Особенности генерации кода для программ с нелинейной структурой. Таблица Меток. Проблема обработки ссылок вперед. Алгоритм генерации кода для программ с командами перехода.

Модуль 2 . «Введение в разработку программ-трансляторов»

Тема 1. Назначение и структура трансляторов

Краткая история развития языков программирования и средств трансляции.

Трансляторы, компиляторы, интерпретаторы. Двухэтапная схема трансляции. Общее описание основных фаз работы трансляторов: лексический анализ, синтаксический и семантический анализ, генерация и оптимизация программного кода.

Тема 2. Основы формального описания языков программирования

Алфавит, лексемы, грамматические конструкции и правила вывода. Метаязык-нотация Бэкуса-Наура: назначение, основные метасимволы. Пример формального описания языка для арифметических выражений. Язык микро-Паскаль и его формализованное описание.

Тема 3. Лексический анализ

Назначение и основные задачи лексического анализа. Типы лексем. Необходимые таблицы и их использование. Алгоритм преобразования потока символов в поток лексем. Основы программной реализации лексических анализаторов.

Тема 4. Синтаксический анализ

Задачи синтаксического анализа. Понятие о дереве грамматического разбора. Примеры построения деревьев разбора для основных грамматических конструкций. Нисходящий и восходящий синтаксический анализ.

Метод рекурсивного спуска как набор подпрограмм выделения и обработки отдельных конструкций. Правила оформления подпрограмм на основе формального описания грамматических конструкций. Необходимые вспомогательные подпрограммы и таблицы.

Тема 5. Основы генерации выходного кода

Основные задачи, возникающие при генерации кода. Преимущества использования ассемблера в качестве выходного языка. Встраивание генерации фрагментов ассемблерного кода в подпрограммы метода рекурсивного спуска. Примеры генерации для основных грамматических конструкций.

4.3. Планы практических занятий

Тема 1. Разработка модуля обработки раздела описания данных в ассемблерных программах

Цель: получение практических навыков по преобразованию символьного представления данных в двоичное представление.

Основные этапы выполнения:

- изучение правил оформления раздела описания данных
- изучение алгоритма преобразования исходных строк
- программная реализация модуля в виде консольного или оконного приложения
- отладка приложения на тестовых примерах

Тема 2. Разработка модуля генерации машинных команд для ассемблерных программ с линейной структурой

Цель: получение практических навыков по преобразованию символьного представления команд в двоичное представление.

Основные этапы выполнения:

- изучение правил оформления раздела описания команд
- изучение особенностей упрощенного учебного языка ассемблера
- изучение используемых структур – Таблицы Команд и Таблицы Регистров
- изучение алгоритма преобразования исходных строк
- программная реализация модуля в виде консольного или оконного приложения
- отладка приложения на тестовых примерах

Тема 3. Разработка модуля генерации машинных команд для ассемблерных программ с командами перехода

Цель: получение практических навыков по преобразованию символьного представления команд в двоичное представление.

Основные этапы выполнения:

- изучение особенностей алгоритма преобразования исходных строк при наличии команд перехода
- программная реализация первой версии модуля (без использования ссылок вперед) в виде консольного или оконного приложения
- отладка первой версии на тестовых примерах
- программная реализация и отладка второй версии модуля

Тема 4: Программная реализация лексического анализатора

Цель: получение практических навыков по реализации лексического анализа для простого подмножества языка программирования

Основные этапы выполнения:

- изучение формального описания заданного подмножества языка программирования и формирование набора лексем
- изучение алгоритма работы лексического анализатора и выбор способа реализации таблицы ключевых слов
- программная реализация анализатора в виде консольного приложения с выводом в каждой строке исходного текста кодов распознанных лексем
- программная реализация сканера в виде оконного приложения

Тема 5: Синтаксический анализ и метод рекурсивного спуска

Цель: получение практических навыков по реализации синтаксического анализа программ для заданного подмножества языка программирования

Основные этапы выполнения:

- изучение основных конструкций заданного подмножества языка программирования и соответствующих правил вывода
- «ручное» построение деревьев грамматического разбора для основных конструкций учебного языка программирования
- разработка набора подпрограмм метода рекурсивного спуска с

- выбором способа реализации и структуры таблицы имен
- программная реализация метода рекурсивного спуска в виде консольного приложения
- программная реализация метода рекурсивного спуска в виде оконного приложения

4.4. Планы практической подготовки/лабораторных занятий

Практическая подготовка/лабораторные занятия не предусмотрены учебным планом.

5. Учебно-методическое обеспечение самостоятельной работы студентов

Самостоятельная работа является важным элементом освоения основной программы курса. В соответствии со спецификой предмета, самостоятельная работа включает в себя:

- самостоятельное изучение материала по заданным преподавателем темам;
- подготовку к практическим занятиям по темам;
- работу с литературой и учебно-методическими пособиями

Самостоятельная подготовка к практическим занятиям выполняется студентом в свободное от занятий время и включает в себя:

- проработку лекционного материала по указанной теме;
- подготовку алгоритмов решения задач по теме;
- написание программ в соответствии с составленным алгоритмом;
- проверку работоспособности программ, как в домашних условиях, так и в компьютерных классах института;
- предоставление результатов самостоятельной работы преподавателю.

Темы для самостоятельного изучения:

- изучение и анализ формализованного описания одного из базовых языков программирования (C, C++, Java, C#, Pascal и др.)
- изучение структуры промежуточного байт-кода для языка Java
- изучение структуры промежуточного представления для языков платформы .NET (язык MSIL)
- практическая реализация методов улучшения генерируемого кода
- реализация модульного программирования на языке ассемблера
- практическое освоение программ-отладчиков

6. Учебно-методическое и информационное обеспечение *Основная литература*

1. Романенко, В. В. Теория языков программирования и методы трансляции : учебно-методическое пособие по лабораторным работам, практическим занятиям и самостоятельной работе студентов / В. В. Романенко, В. Т. Калайда. — Томск : Томский государственный университет систем управления и радиоэлектроники, 2024. — 124 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/152863.html>. — Режим доступа: для авторизир. Пользователей

2. Захаров, Ф. Н. Системы беспроводной связи : учебно-методическое пособие для практических и лабораторных работ студентов всех форм обучения, обучающихся по техническим направлениям / Ф. Н. Захаров, К. А. Ярков. — Томск : Томский государственный университет систем управления и радиоэлектроники, 2024. — 69 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/152851.html>. — Режим доступа: для авторизир. пользователей

Дополнительная

1. Залогова, Л. А. Разработка Паскаль-компилятора / Л. А. Залогова. — 5-е изд. — Москва : Лаборатория знаний, 2021. — 184 с. — ISBN 978-5-00101-110-1. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/109435.html>

Интернет-ресурсы и перечень ежегодно обновляемых современных профессиональных баз данных и информационных справочных систем:

www.iprbookshop.ru – Электронно-библиотечная система «IPRbooks»
<http://citforum.ru/>

Национальный открытый университет ИНТУИТ <https://intuit.ru/>
Образовательная платформа и конструктор онлайн-курсов Stepik
<https://stepik.org/>

7. Материально-техническое обеспечение дисциплины

В процессе изучения данной дисциплины используется учебная аудитория, кабинет для самостоятельной работы студентов, читальный зал, видеопроекторное оборудование, компьютер, оснащенный типовым пакетом системного и офисного ПО, в соответствии с Реестром материально-технического обеспечения аудиторного фонда Университета управления «ТИСБИ».

Пакет лицензионного системного и офисного ПО:

Операционная система Microsoft Windows 10 Pro
Microsoft Office 2013

Программное обеспечение, входящее в пакет, получает обновление в автоматическом, установленном разработчиком (компанией Microsoft) порядке, посредством сети Интернет.

Открытая среда разработки программного обеспечения - Lazarus.
Delphi 10.1 Berlin Professional Academic Concurrent License. License
Certificate Number: 457049

8. Оценка компетенций по изучаемой дисциплине

Для оценки компетентности рекомендуется использовать рейтинговую оценку знаний, умений и навыков студента по окончанию изучения каждого Модуля в соответствии с Положением о модульно-рейтинговой системе организации образовательного процесса. Итоговая оценка (в баллах) складывается из баллов, набранных по каждому Модулю (семестровая оценка) и баллов, набранных, непосредственно на экзамене (зачете).

Расчет набранных баллов по дисциплине осуществляется в следующей последовательности:

, где M – количество баллов по модулю; n – количество модулей

, где K - количество баллов на экзамене (зачете);

, где Π – поощрительные баллы (от 1 до 5).

Уровень сформированности компетенций и их основные признаки оцениваются по таблицам.

Оценка сформированности компетенции ПК-4 «Способен разрабатывать компоненты системного программного обеспечения» в части дисциплины «Методы трансляции»

№ п/п	Уровни сформированности компетенции	Основные признаки уровня	Инструменты оценки сформированности уровня
1	2	3	4
1	Пороговый уровень (как минимально допустимый) (от 60 до 70 баллов)	- знает особенности языков-ассемблеров, назначение и структуру программ-трансляторов; - умеет оформлять простейшие программы на языке ассемблера, строить деревья грамматического разбора для простейших конструкций языков программирования - владеет навыками разработки простейших ассемблирующих программ	Практические задания Тестовый опрос Зачет
2	Базовый уровень (относительно порогового уровня) (От 71 до 85 баллов)	- знает возможности ассемблеров по реализации циклов, ветвлений и обработке массивов; особенности реализации лексического и синтаксического анализа; - умеет разрабатывать программы лексического анализа - владеет навыками разработки ассемблеров для программ с нелинейной структурой	Практические задания Тестовый опрос Зачет
3	Повышенный уровень (относительно порогового уровня) (От 86 до 100 баллов)	- знает особенности реализации подпрограмм на языке ассемблера и способы реализации всех функций трансляторов - умеет выполнять программную реализацию метода рекурсивного спуска - владеет навыками разработки всех основных модулей трансляторов, включая генерацию выходного кода	Практические задания Тестовый опрос Зачет