

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет прикладной математики и информатики

“УТВЕРЖДАЮ”

Декан ФПМИ

профессор, д.т.н. Лемешко
Борис Юрьевич

“ ____ ” _____ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Теория вычислительных процессов и структур

ООП: специальность 010503.65 Математическое обеспечение и администрирование информационных систем

Шифр по учебному плану: СД.Ф.1

Факультет: прикладной математики и информатики очная форма обучения

Курс: 3, семестр: 5 6

Лекции: 52

Практические работы: - Лабораторные работы: 34

Курсовой проект: - Курсовая работа: - РГЗ: -

Самостоятельная работа: 89

Экзамен: - Зачет: 5 6

Всего: 175

Новосибирск

2011

Рабочая программа составлена на основании Государственного образовательного стандарта высшего профессионального образования по направлению (специальности): 351500 Математическое обеспечение и администрирование информационных систем.(№ 72 мжд/сп от 10.03.2000)

СД.Ф.1, дисциплины федерального компонента

Рабочая программа обсуждена на заседании кафедры Прикладная математика протокол № 4 от 28.06.2011

Программу разработал

доцент, к.т.н.

Еланцева Ирина Леонидовна

Заведующий кафедрой

профессор, д.т.н.

Соловейчик Юрий Григорьевич

Ответственный за основную образовательную программу

профессор, д.т.н.

Попов Александр Александрович

1. Внешние требования

Таблица 1.1

Шифр дисциплины	Содержание учебной дисциплины	Часы
СД.Ф.1	ТЕОРИЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ И СТРУКТУР Теория формальных языков и трансляций: Математическое моделирование языков. Синтаксис и семантика. Метаязыки. Нормальные формы Бекуса-Наура (БФА). Формальные грамматики. Языки, порождаемые грамматиками; Классы формальных грамматик; проблема распознавания языков. Автоматы: конечные автоматы, анализаторы и преобразователи. Анализаторы контекстно-свободных языков. Трансляторы: схема компилятора; методы построения; схематическая теория программ; способы оптимизации кода. Семантическая теория программ; схемы программ, методы формальной спецификации и верификации; модели вычислительных процессов; взаимодействие процессов; протоколы и интерфейсы; асинхронные процессы; сети Петри: принципы построения, алгоритмы поведения, способы реализации, области применения; принципы и способы технической реализации моделей процессов и структур.	175

2. Особенности (принципы) построения дисциплины

Таблица 2.1

Особенности (принципы) построения дисциплины

Особенность (принцип)	Содержание
Основания для введения дисциплины в учебный план по направлению или специальности	Стандарт специальности 351500
Адресат курса	Студенты специальности 010503.65 - "Математическое обеспечение и администрирование информационных систем".
Основная цель (цели) дисциплины	Освоение синтаксиса и семантики языка ассемблера. Освоение основ синтаксически-управляемых методов обработки языков на примере процесса трансляции с алгоритмических языков высокого уровня. Классификация и сравнительный анализ языков программирования.
Ядро дисциплины	Синтаксис и семантика языков программирования. Формальные языки и грамматики. Автоматы. Трансляторы, методы трансляции.
Связи с другими учебными дисциплинами основной образовательной	Программирование, Информатика. Выполнение производственной практики и выпускной работы.

программы	
Требования к первоначальному уровню подготовки обучающихся	Для успешного изучения курса студенту необходимо знать теорию множеств, архитектуру ЭВМ, основы программирования на языках высокого уровня
Особенности организации учебного процесса по дисциплине	Промежуточный контроль - защита лабораторных работ. Итоговый контроль - зачет в каждом из семестров.

3. Цели учебной дисциплины

Таблица 3.1

После изучения дисциплины студент будет

иметь представление	
1	о месте языка ассемблера среди других языков программирования
2	о задачах, решаемых с использованием языка ассемблера
3	о роли транслятора в системе программного обеспечения
4	о различиях в подходах к проектированию трансляторов
5	о множестве задач, решаемых на каждом этапе проектирования транслятора
6	об истории языков программирования
7	о влиянии машинной архитектуры на развитие языков программирования
8	Об инкапсуляции, наследовании и управлении последовательностью действий в различных языках программирования
знать	
9	назначение регистров процессора
10	механизм адресации и модели памяти
11	структуру программы на языке ассемблера
12	этапы процесса разработки программы на языке ассемблера
13	команды, директивы и операции языка ассемблера
14	основы взаимодействия языка ассемблера и языка C++
15	средства операционной системы для ввода-вывода информации в консольных приложениях
16	основы работы на уровне макроязыка ассемблера
17	задачи и функционирование блоков транслятора (блоки лексического, синтаксического анализов и блок генерации кода)
18	особенности нисходящего и восходящего синтаксического анализа
19	методы построения блока лексического анализа (сканера)
20	методы построения синтаксического анализатора, работающего по принципу "сверху-вниз"
21	методы построения синтаксического анализатора, работающего по принципу "снизу-вверх"
22	методы генерации кода
23	основы автоматизации конструирования синтаксических и лексических анализаторов
24	классификацию языков программирования
25	реализацию элементарных типов данных в основных языках программирования
26	реализацию сложных типов данных в основных языках программирования
27	реализацию управления последовательностью действий в основных языках программирования
уметь	
28	пользоваться инструментальными средствами разработки программ на языке ассемблера
29	использовать при программировании команды, директивы и операции языка ассемблера
30	пользоваться средствами операционной системы для ввода-вывода

	информации в консольных приложениях
31	организовать интерфейс между модулями на языке ассемблера и языке C++
32	классифицировать грамматику исходного языка
33	выбрать структуру транслятора, наиболее подходящую для конкретной задачи
34	выбрать структуру и реализовать блок лексического анализа (с возможной диагностикой ошибок исходной программы)
35	выбрать стратегию синтаксического анализа
36	разработать и реализовать блок синтаксического анализа(с возможной диагностикой и исправлением ошибок исходной программы)
37	преобразовать исходное предложение в постфиксную запись
38	разработать и реализовать блок генерации кода
иметь опыт (владеть)	
39	разработки программ на языке ассемблера и организации межмодульного интерфейса для языков ассемблера и C++
40	проектирования и разработки трансляторов

4. Содержание и структура учебной дисциплины

Лекционные занятия

Таблица 4.1

(Модуль), дидактическая единица, тема	Часы	Ссылки на цели
Семестр: 5		
Модуль: Программирование на языке ассемблера		
Дидактическая единица: Теория формальных языков и трансляций: Математическое моделирование языков. Синтаксис и семантика. Метаязыки		
Язык ассемблера, задачи, решаемые с его помощью. Основы программирования на языке ассемблера. Регистры процессора. Процесс разработки программы. Структура программы. Сегменты. Понятия команд, директив и операций. Формат записи команд. Директивы segment, proc, assume, end, директивы определения данных. Инициализация программы.	3	1, 11, 12, 13, 15, 2, 9
Основные команды языка ассемблера. Команды пересылки данных: mov, push, pop. Режимы адресации: регистровая, непосредственная, прямая, косвенная регистровая, по базе, прямая с индексированием, по базе с индексированием. Команды передачи управления: call, ret, безусловной и условной передачи управления, управления циклами.	3	10, 13
Цепочечные команды. Адресация строки-приемника и строки-источника, команда lea, префиксы повторения, команды пересылки, сравнения, сканирования, загрузки и сохранения цепочек.	2	13

Взаимодействие языков C++ и ассемблера. Внутренняя структура программы на языке C++. Использование функций на языке ассемблера в программах на языке C++: основы взаимодействия языков C++ и ассемблера, передача управления в подпрограмму и обратно, использование глобальных данных, использование аргументов функции, возвращение значения. Вызов функций на языке C++ из программ на языке ассемблера. Использование альтернативных моделей распределения памяти.	4	13, 14
Арифметические команды. Форматы хранения десятичных чисел, команды сложения, вычитания, умножения, деления, коррекция результата для BCD-форматов, команды приращения и уменьшения значения приемника на единицу, команды обращения и расширения знака.	2	13
Директивы и операции. Директивы определения идентификаторов. Условные директивы. Операции: арифметические, логические, отношения, возвращающие значения, присваивания атрибутов.	2	13, 16
Макроопределения. Сравнение макроопределений и процедур; состав макроопределений; директивы макроассемблера: повторения, условные, local; операции в макроопределениях; задание макроопределений в исходных программах; использование библиотеки макроопределений; рекомендации по разработке макроопределений; считывание библиотеки макроопределений в программу; удаление макроопределений.	2	13, 16
Семестр: 6		
Модуль: Формальные языки, грамматики, методы трансляции		
Дидактическая единица: Трансляторы: схема компилятора; методы построения; схематическая теория программ; способы оптимизации кода. Семантическая теория программ; схемы программ.		
Структуры трансляторов и методы их реализации	1	3, 4, 5
Генерация кода	2	17, 22, 23, 37, 38
Дидактическая единица: Теория формальных языков и трансляций: Математическое моделирование языков. Синтаксис и семантика. Метаязыки		
Синтаксический анализ	1	17, 18, 23, 3, 32, 33, 35, 36, 4, 5
Дидактическая единица: Автоматы: конечные автоматы, анализаторы и преобразователи. Анализаторы контекстно-свободных языков.		
Лексический анализатор	2	17, 19, 32, 33, 5
Дидактическая единица: Формальные грамматики.		

Языки, порождаемые грамматиками; Классы формальных грамматик; проблема распознавания языков.		
Основная идея алгоритма нисходящего синтаксического анализа	6	17, 18, 20, 3, 32, 33, 34, 35, 36, 4, 5
Основные идеи восходящих методов синтаксического анализа.	5	17, 18, 21, 3, 32, 33, 34, 35, 36, 4, 5
Модуль: Сравнительный анализ и классификация языков программирования		
Дидактическая единица: Семантическая теория программ; схемы программ, методы формальной спецификации и верификации; модели вычислительных процессов; взаимодействие процессов; протоколы и интерфейсы; асинхронные процессы;		
Основные понятия языков программирования	6	24, 6, 7
Типы данных	4	24, 25, 26, 6, 7
Структурированные типы данных	4	24, 25, 26, 6, 8
Дидактическая единица: сети Петри: принципы построения, алгоритмы по-ведения, способы реализации, области применения; принципы и способы технической реализации мо-делей процессов и структур		
Управление последовательностью выполнения программы	3	24, 27, 6, 8

Лабораторная работа

Таблица 4.2

(Модуль), дидактическая единица, тема	Учебная деятельность	Часы	Ссылки на цели
Семестр: 5			
Модуль: Программирование на языке ассемблера			
Дидактическая единица: Теория формальных языков и трансляций: Математическое моделирование языков. Синтаксис и семантика. Метаязыки			
Основные элементы языка ассемблера	Изучить и приобрести практические навыки работы с основными командами языка ассемблера, функциями ввода-вывода, регистрами и символьными данными.	6	10, 11, 12, 13, 15, 28, 29, 30, 39, 9
Цепочечные команды	Изучить цепочечные команды (команды	6	10, 11, 12, 13, 15, 28,

	обработки строк); разработать программу на языке ассемблера, которая использует команды обработки строк и выполняет ввод-вывод информации с использованием средств операционной системы.		29, 30, 39, 9
Интерфейс с языком Си++	Изучить правила взаимодействия модулей на языках C++ и ассемблера; разработать программу, которая включает вызов подпрограммы на языке ассемблера из функции на языке C++.	6	10, 11, 12, 13, 14, 28, 29, 31, 39, 9
Семестр: 6			
Модуль: Формальные языки, грамматики, методы трансляции			
Дидактическая единица: Трансляторы: схема компилятора; методы построения; схематическая теория программ; способы оптимизации кода. Семантическая теория программ; схемы программ.			
Проектирование и реализация таблиц, используемых в трансляторе	сформулировать цели и задачи проекта; определить вид, объем и источник исходных данных; выбрать структуру таблиц; реализовать таблицы; оформить отчет и защитить работу.	4	17, 19, 33, 40
Разработка и реализация блока генерации кода	сформулировать цели и задачи проекта; определить вид, структуру входных и выходных данных; разработать и реализовать генератор на языке высокого уровня; оформить отчет и защитить работу	4	17, 22, 37, 38, 40

Дидактическая единица: Автоматы: конечные автоматы, анализаторы и преобразователи. Анализаторы контекстно- свободных языков.			
Разработка и реализация блока лексического анализа (сканер)	сформулировать цели и задачи проекта; определить вид, структуру входных и выходных данных; разработать и реализовать сканер на языке высокого уровня; оформить отчет и защитить работу	4	17, 19, 3, 32, 33, 4, 40, 5
Дидактическая единица: Формальные грамматики. Языки, порождаемые грамматиками; Классы формальных грамматик; проблема распознавания языков.			
Разработка и реализация блока синтаксического анализа	сформулировать цели и задачи проекта; определить вид, структуру входных и выходных данных; классифицировать входной язык, определить необходимость преобразования языка к требуемому виду, при необходимости выполнить эти преобразования; выбрать стратегию анализа разработать и реализовать анализатор на языке высокого уровня; оформить отчет и защитить работу	4	17, 18, 20, 21, 3, 32, 33, 34, 35, 36, 4, 40, 5

5. Самостоятельная работа студентов

Семестр- 5, Подготовка к зачету

Подготовка с использованием конспекта лекций, а также рекомендуемой литературы - 15 часов.

Семестр- 5, Подготовка к занятиям

Подготовка с использованием конспекта лекций, методических указаний, а также рекомендуемой литературы - по 20 часов на каждую лабораторную работу.

Семестр- 6, Подготовка к зачету

Подготовка с использованием конспекта лекций, а также рекомендуемой литературы. На подготовку к зачету отводится 2 часа.

Семестр- 6, Подготовка к занятиям

Подготовка к лабораторным занятиям с использованием конспекта лекций, методических указаний (см. пункт 8.1), а также рекомендуемой литературы.

На подготовку к лекционным занятиям в семестре отводится 2 часа.

На подготовку к лабораторным занятиям в семестре отводится 10 часов.

Занятие 1. Студент должен ознакомиться с темой и содержанием лабораторной работы 1 (по методическим указаниям к лабораторным работам), повторить темы проектирование структур данных "Таблицы", операции работы с таблицами, разработка классов, постоянные и переменные таблицы транслятора. На подготовку к занятию отводится 1 час.

Занятие 2. Студент должен подготовить отчет и подготовить ответы на контрольные вопросы по лабораторной работе 1, в соответствии с требованиями, приведенными в методических указаниях. На подготовку к занятию отводится 1 час.

Занятие 3. Студент должен ознакомиться с темой и содержанием лабораторной работы 2 (по методическим указаниям к лабораторным работам), повторить темы: формальные языки, регулярные выражения, автоматные грамматики, детерминированные конечные автоматы, лексический анализ. На подготовку к занятию отводится 1 час.

Занятие 4. Студент должен подготовить отчет и подготовить ответы на контрольные вопросы по лабораторной работе 2, в соответствии с требованиями, приведенными в методических указаниях. На подготовку к занятию отводится 2 часа.

Занятие 5. Студент должен ознакомиться с темой и содержанием лабораторной работы 3 (по методическим указаниям к лабораторным работам), повторить темы: формальные языки, LL1-грамматики, нисходящий разбор предложений, синтаксический анализ. 1 часа.

Занятие 6. Студент должен написать грамматику транслируемой программы (в соответствии с полученным вариантом), привести грамматику к LL1-виду, составить схему разбора и таблицу разбора. На подготовку к занятию отводится 1 час.

Занятие 7. Студент должен подготовить отчет и подготовить ответы на контрольные вопросы по лабораторной работе 3, в соответствии с требованиями, приведенными в методических указаниях. На подготовку к занятию отводится 2 часа.

Занятие 8. Студент должен ознакомиться с темой и содержанием лабораторной работы 4 (по методическим указаниям к лабораторным работам), повторить темы: формы промежуточной записи (четверки), генерация кода, оптимизация кода, вид программы на языке Ассемблер. На подготовку к занятию отводится. Студент должен подготовить отчет и подготовить ответы на контрольные вопросы по лабораторной работе 4, в соответствии с требованиями, приведенными в методических указаниях. На подготовку к занятию отводится 1 час.

6. Правила аттестации студентов по учебной дисциплине

В соответствии с ООП итоговая аттестация включает: зачет в пятом и шестом семестрах. Знания и умения студента по каждому из модулей оцениваются как в ходе лабораторных работ, так и при защите индивидуального задания и на зачете.

Текущая аттестация студента в пятом семестре проводится по результатам выполнения им лабораторных работ и дополнительного учебного задания.

Баллы за каждую из трех лабораторных работ выставляются по правилам, приведенным в таблице 6.1.

Таблица 6.1.

Вид деятельности	Минимальное количество баллов	Максимальное количество баллов
Выполнение лабораторной работы (оценивается полнота выполнения задания, работоспособность разработанного программного обеспечения, полнота набора тестов, полнота отчета, его соответствие требованиям оформления, своевременность* выполнения лабораторной работы)	7	11
Защита (оценивается полнота и глубина теоретических знаний, умение применить знания на практике, своевременность* защиты)	8	16(15)**
Всего за лабораторную работу	15	27(26)**

* график выполнения и защиты лабораторных работ приведен ниже

** в скобках – баллы за лабораторную работу №3

Минимальное допустимое количество баллов за лабораторную работу получает студент, выполнивший задание не полно, оформивший отчет и показавший на защите некоторое владение теоретическими и практическими навыками. Студент, не выполнивший и не защитивший лабораторную работу, не допускается к выполнению следующей лабораторной работы.

График выполнения и защиты лабораторных работ:

- 1 – 7 неделя – 1-я лабораторная работа;
- 8 – 12 неделя – 2-я лабораторная работа;
- 13 – 18 неделя – 3-я лабораторная работа.

Студент, выполнивший и защитивший лабораторные работы досрочно (не позднее 14-й недели) и набравший за их выполнение не менее 65 баллов, по согласованию с преподавателем может выполнить дополнительное учебное задание (как самостоятельное углубленное освоение отдельных тем дисциплины). Выполнение задания оценивается по тем же критериям, что и лабораторные работы, при этом минимальное количество баллов составляет 10 (5 за выполнение и 5 за защиту), максимальное количество баллов – 20 (8 за выполнение и 12 за защиту).

Оценка за контрольные недели выставляется в соответствии с таблицей 6.2.

Таблица 6.2.

Номер недели	7 контрольная неделя			12 контрольная неделя		
Оценка за контрольную неделю	0	1	2	0	1	2
Текущий рейтинг студента	Менее 7	От 7 до 11	15 и более	Менее 22	От 22 до 38	30 и более
Количество зачетных лабораторных заданий	0	1*	1	1	2*	2

* лабораторная работа выполнена, но не защищена

Зачет в пятом семестре сдается письменно по билетам, каждый из которых включает два теоретических вопроса. К зачету допускаются студенты, выполнившие и защитившие все 3 лабораторные работы и набравшие не менее 45 баллов. Максимальное количество баллов за теоретический зачет – 20 баллов, минимальное – 5. Оценивается правильность ответа, последовательность и полнота ответа, владение профессиональной терминологией. Минимальное количество баллов за теоретический зачет получает студент, частично ответивший на один вопрос билета.

Текущая аттестация студента в шестом семестре проводится по результатам выполнения им лабораторных работ. Баллы за каждую из четырех лабораторных работ выставляются по правилам, приведенным в таблице 6.3.

Таблица 6.3

Вид деятельности	Максимальное количество баллов	Минимальное количество баллов
Выполнение лабораторной работы (оценивается сложность задания, полнота выполнения задания, работоспособность разработанного программного обеспечения, самостоятельность выполнения)	7	4
Отчет (оценивается полнота отчета, соответствие требованиям оформления, своевременность* и самостоятельность написания отчета)	6	3
Защита (оценивается полнота и глубина теоретических знаний, умение применить знания на практике, своевременность* защиты)	7	3
Всего за лабораторную работу	20	10

* - на выполнение и защиту каждой лабораторной работы отводится два занятия.

Максимальное количество баллов за лабораторную работу – 20 баллов. Минимальное допустимое количество баллов за лабораторную работу – 10 баллов – получает студент, выполнивший задание минимальной сложности, оформивший отчет и показавший на защите некоторое владение теоретическими и практическими навыками. Студент, не набравший минимально допустимого количества баллов за лабораторную работу, не допускается к выполнению следующей лабораторной работы.

Оценка за контрольные недели выставляются в соответствии с таблицей 6.4

Таблица 6.4

Номер недели	7 контрольная неделя			12 контрольная неделя		
Оценка за контрольную неделю	0	1	2	0	1	2
Текущий рейтинг студента	Менее 8	От 8 до 16	16 и более	Менее 16	От 16 до 30	30 и более
Количество зачтенных лабораторных заданий	0	1	2	Менее 2	2	3

Максимальное количество баллов за работу студента в семестре – 80 балла.

К зачету в шестом семестре допускаются студенты, выполнившие все лабораторные работы и набравшие не менее 40 баллов. Студент сдает зачет в письменной форме в виде ответов на вопросы билета. Билет содержит 20 вопросов. За правильный и полный ответ на каждый вопрос студент получает 1 балл. Максимальное количество баллов за зачет – 20 баллов. Зачет сдан, если студент набрал не менее 10 баллов.

Итоговый рейтинг студента по дисциплине в шестом семестре рассчитывается по формуле:

$$R = (R_1 + R_2) * 0.5,$$

где R_1 – итоговый рейтинг студента в пятом семестре;

R_2 – итоговый рейтинг студента в шестом семестре.

7. Список литературы

7.1 Основная литература

В печатном виде

1. Еланцева И. Л. Языки программирования и методы трансляции. Раздел "Методы трансляции" : конспект лекций / И. Л. Еланцева, И. А. Полетаева ; Новосиб. гос. техн. ун-т. - Новосибирск, 2008. - 97, [2] с. : ил., табл.
2. Юров В. И. Assembler : учебное пособие для вузов по направлению подготовки специалистов " Информатика и вычислительная техника" / В. И. Юров. - СПб. [и др.], 2007. - 636 с. : ил., табл. - Рекомендовано МО.
3. Павловская Т. А. C/C++. Программирование на языке высокого уровня : учебник для вузов / Т. А. Павловская. - СПб., 2007. - 460 с. : ил.. - На тит. л.: Издательская программа 300 лучших учебников для высшей школы в честь 300-летия Санкт-Петербурга. - Рекомендовано МО.

В электронном виде

1. Еланцева И. Л. Языки программирования и методы трансляции. Раздел "Методы трансляции" : конспект лекций / И. Л. Еланцева, И. А. Полетаева ; Новосиб. гос. техн. ун-т. - Новосибирск, 2008. - 97, [2] с. : ил., табл.. - Режим доступа:
<http://www.library.nstu.ru/fulltext/metodics/2008/elan.rar>

7.2 Дополнительная литература

В печатном виде

1. Лебедев В. Н. Введение в системы программирования. - М., 1975. - 311 с. : ил.
2. Ахо А. В. Компиляторы : Принципы, технологии, инструменты / А. Ахо, Р. Сети, Д. Ульман. - М., 2001. - 766 с.
3. Хантер Р. Проектирование и конструирование компиляторов / Р. Хантер; пер. с англ. С. М. Круговой под ред. В. М. Савинкова. - М., 1984. - 231, [1] с. : ил.
4. Юров В. И. Assembler. Практикум : учебное пособие для студентов вузов, обучающихся по направлению подготовки дипломированных специалистов "Информатика и вычислительная техника" / В. Юров. - СПб.[и др.], 2003. - 395 с. : ил. + 1 дискета. - Рекомендовано МО.
5. Юров В. И. Assembler : [учебник] / В. И. Юров. - СПб. [и др], 2002. - 622, [1] с. : ил.
6. Юров В. И. Assembler : учебник / В. Юров. - СПб. и [и др.], 2001. - 622, [1] с. : ил. + 1 дискета.
7. Юров В. И. Assembler : учебное пособие для вузов по направлению "Информатика и вычислительная техника" / В. И. Юров. - СПб., 2003. - 636 с. : ил. - Рекомендовано МО.
8. Юров В. И. Assembler : [учебное пособие для вузов по направлению "Информатика и вычислительная техника"] / В. И. Юров. - СПб. [и др.], 2004. - 636 с. : ил. - Рекомендовано МО.
9. Юров В. И. Assembler : [учебное пособие для вузов по направлению "Информатика и вычислительная техника"] / В. И. Юров. - СПб. [и др.], 2005. - 636 : ил. - Рекомендовано МО.
10. Юров В. И. Assembler : Учебный курс. - СПб., 2000. - 672 с. : ил.. - В прилож.: дискета.
11. Абель П. Язык Ассемблера для IBM PC и программирования : Пер. с англ.. - М., 1992. - 447 с.
12. Голубь Н. Г. Искусство программирования на Ассемблере / Надежда Голубь. - СПб., 2006. - 819 с. : ил.

13. Голубь Н. Г. Искусство программирования на Ассемблере : лекции и упражнения / Голубь Н. Г. - СПб., 2002. - 644 с. : ил.
14. Абель П. Ассемблер. Язык и программирование для IBM PC / пер. с англ. под ред. С.М. Молявко. - Киев [и др.], 2003. - 734 с. : табл.
15. Зубков С. В. Assembler для DOS, Windows и Unix. - М., 1999. - 637 с.
16. Пильщиков В. Н. Программирование на языке ассемблера IBM PC. - М., 1996. - 286 с.
17. Трой Д. А. Программирование на языке Си для персонального компьютера IBM PC : Пер. с англ. / Под ред. И. В. Емелина. - М., 1991. - 429 с.
18. Скэнлон Л. Персональные ЭВМ IBM PC и XT. Программирование на языке ассемблера / Скэнлон Л. ; пер. с англ. Емелина И. В. - М., 1991. - 335, [1] с.
19. Финогенов К. Г. Использование языка Ассемблера : учебное пособие для вузов по специальности 351400 "Прикладная информатика (по областям)" / К. Г. Финогенов. - М., 2004. - 438 с. : ил. - Рекомендовано УМО.
20. Финогенов К. Г. Основы языка Ассемблера : Учебный курс. - М., 1999. - 288 с. : ил.
21. Программирование на языке СИ. Ч. 3. Программирование на языке Ассемблера. 4 : Практикум на ЭВМ : метод. указания к лаб. работам для 1-2 курса ФПМИ (спец. 01. 02) дневного отд-ния / сост.: Полетаева И. А., Стасышин В. М. - Новосибирск, 1995. - 86 с.
22. Пратт М. Языки программирования: разработка и реализация : Пер с англ. / Под ред. Баяковского Ю. М. - М., 1979. - 574 с.

8. Методическое и программное обеспечение

8.1 Методическое обеспечение

В печатном виде

1. Языки программирования и методы трансляции : методические указания к лаб. работам для 3 курса ФПМИ направлений 510200, 351500 / Новосиб. гос. техн. ун-т ; [сост.: И. Л. Еланцева, И. А. Полетаева]. - Новосибирск, 2005. - 24 с.

В электронном виде

1. Лисицин Д. В. Программирование на языке Ассемблера [Электронный ресурс] : учебно-методическое пособие / Д. В. Лисицин, Р. В. Петров, И. А. Полетаева ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://ciu.nstu.ru/fulltext/unofficial/2011/lib_842_1320653325.doc. - Загл. с экрана.

8.2 Программное обеспечение

1. Microsoft, eMbedded Visual C++, Интегрированная среда разработки

9. Контролирующие материалы для аттестации студентов по дисциплине

Вопросы для теоретического зачета пятого семестра

1. Особенности языка Ассемблера и задачи, решаемые с его использованием
2. Директивы SEGMENT, PROC, ASSUME, END
3. Команды пересылки данных (MOV, PUSH, POP)
4. Команды CALL, RET, JMP
5. Команды управления циклами
6. Внутренние регистры
7. Директивы определения данных и идентификаторов
8. Команды пересылки строк, префиксы повторения
9. Команды сравнения, сканирования, загрузки и сохранения строк
10. Регистровая, непосредственная и прямая адресация
11. Косвенная регистровая адресация и адресация по базе
12. Прямая адресация с индексированием, адресация по базе с индексированием
13. Команды сложения, коррекция результата сложения
14. Команды вычитания, коррекция результата вычитания
15. Команды умножения и деления, коррекция результатов умножения и деления
16. Команды сдвига и циклического сдвига
17. Условные директивы
18. Сравнение макроопределений и процедур, состав макроопределений
19. Макроассемблер: директива LOCAL, директивы повторения
20. Макроассемблер: условные директивы
21. Операции в макроопределениях
22. Внутренняя структура программы на языке C++
23. Основы взаимодействия языков C++ и ассемблера. Передача управления в подпрограмму и обратно
24. Передача данных из функции на языке C++ в подпрограмму на языке ассемблера
25. Возвращение данных из подпрограммы на языке ассемблера в функцию на языке C++
26. Вызов функций на языке C++ из подпрограммы на языке ассемблера
27. Ввод-вывод информации с использованием средств операционной системы
28. Команды условной передачи управления

Пример варианта билета для зачёта в шестом семестре

1. Перечислите основные этапы процесса трансляции.
2. Укажите отличия транслятора и компилятора.
3. Напишите автоматную грамматику для формального языка идентификаторов.
4. Напишите детерминированный конечный автомат для формального языка идентификаторов.
5. Охарактеризуйте нисходящую стратегию разбора.
6. Напишите постфиксную запись для выражения $a = (x - y) - 2 * c$.
7. Приведите любой метод оптимизации кода.
8. Одним из критериев сравнения языков является критерий ясности и единообразия понятий языка. Что это за критерий? Приведите примеры удачных и неудачных конструкций языка C++ или Fortran по этому критерию.
9. Одним из критериев сравнения языков является критерий стоимости использования. Что это за критерий? Что такое стоимость вычислений?
10. Языки программирования бывают императивные, аппликативные, основанные на системе правил и объектно-ориентированные. Определите понятие "императивный язык".
11. Что такое раздельная компиляция? Почему для раздельной компиляции желательно описание подпрограмм?

12. Что такое связывание? Нужно ли связывание для интерпретируемого языка? Ответ обосновать.
13. Для чего в программировании требуется определять константы? Приведите пример. Как описать кон-станту в языке C++?
14. Приведите, если это возможно, примеры явного и контекстуального определения типа на языке Fortran.
15. Поясните все преобразования типов при вычислении выражения в приведённом фрагменте программы на языке C++ и определите результат вычислений:
float a=10.,b=0.1; int c; double d=6;
c=a*b*d;
16. В языке PL1 определён тип FIXED DECIMAL. При описании этого типа указывается число хранимых десятичных знаков. Определить результат вычисления выражения в приведённом фрагменте программы:
FIXED DECIMAL(3) A,B,C,D;
A=9;B=10;C=3;
D=A+B/C;
17. Как хранится строковый тип в памяти, если в языке строки имеют строго фиксированную длину.
18. При условии, что длина слова компьютера равна 4 байта, определите число байт, занимаемое структу-рой, описанной на языке C++ следующим образом:
struct
{
char a;
long b;
union
{
double c;
char d[7];
}
}
19. Какие ошибки компиляции будут выданы для приведённого фрагмента описаний классов? Исправьте описание класса C, оставив множественное наследование, так, чтобы связанная с этим ошибка пропала:
class A
{
virtual sub1(){cout << "sub1A" <<endl;}
};
class B
{
virtual sub1(){cout << "sub1B" <<endl;}
};
class C: virtual public A, virtual public B
{
};
20. Пусть A=10, B=0.2, C=5. Будет ли вызвана функция Sub при вычислении выражения на языке C++: A-2*C>0 || Sub(B)<0. Ответ обосновать.