

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания по выполнению лабораторных работ по дисциплине
Машинно-ориентированные языки программирования

Направление подготовки

01.04.02 - Прикладная математика и информатика

Профиль подготовки

Математическое моделирование

Квалификация выпускника

магистр

Учебный план 2026 г. Изучается в 1 семестре

Ставрополь, 2026

Печатается по решению
редакционно-издательского совета
Северо-Кавказского федерального
университета

Ионисян А.С. Машинно-ориентированные языки программирования. Методические указания по выполнению лабораторных работ. – Ставрополь: Издательство Северо-Кавказского федерального университета, 2024.– ____ с.

Указания подготовлены в соответствии с Федеральным государственным образовательным стандартом высшего образования, раскрывает основные особенности изучения машинно-ориентированных языков программирования.

В указаниях изложены основные положения машинно-ориентированных языков программирования, вводятся и иллюстрируются примерами основные методы работы с ЭВМ. Уделено внимание типам данных, определяемым программистом, показаны особенности программной реализации решений.

Методические указания по дисциплине «Машинно-ориентированные языки программирования» одобрены на заседании кафедры математического моделирования «__» _____ 2024 г., протокол № ____.

Указания предназначены для организации и проведения лабораторных занятий по дисциплине «Машинно-ориентированные языки программирования» для направления 01.04.02 «Прикладная математика и информатика» (Магистр).

© ФГАОУ ВО «Северо-Кавказский
федеральный университет», 202__

Содержание

		Стр.
1.	Введение	
2.	Лабораторная работа №1 «Структура программы, основные мнемоники, директивы и типы данных компилятора NASM»	
3.	Лабораторная работа №2 «Структурное программирование на ассемблере»	
4.	Лабораторная работа №3 «Обработка массивов данных на ассемблере»	
5.	Литература	
6.	Приложение 1. Оборудование, материалы и программное обеспечение, используемые для организации и проведения лабораторных работ	
7.	Приложение 2. Содержимое отчета по лабораторной работе ...	
8.	Приложение 3. Правила поведения в компьютерном классе (аудитории, лаборатории)	

Введение

Дисциплина «Машинно-ориентированные языки программирования» относится к обязательной части учебного плана 2024 года и изучается студентами магистратуры специальности 01.04.02 направления «Прикладная математика и информатика» в 1 семестре.

Целями освоения дисциплины «Машинно-ориентированные языки программирования» являются:

1. Формирование представлений и фундаментальных знаний об архитектуре современной микропроцессорной техники;
2. Формирование представлений и фундаментальных знаний о современных методах низкоуровневого программирования;
3. Реализация математических алгоритмов на ЭВМ при помощи низкоуровневого языка программирования Netwide Assembler.

Основными задачами изучения дисциплины являются:

1. Изучение принципов создания быстродействующего программного обеспечения.
2. Изучение языков программирования низкого уровня.
3. Реализация на языке ассемблера базовых алгоритмических структур,
4. Реализация на языке ассемблера типовых алгоритмов прикладной математики.

Формы отчетности: экзамен - 1 семестр. Курсовая работа 1 семестр.

Содержание дисциплины: архитектура современного микропроцессора, основные мнемоники, директивы и типы данных компилятора NASM, структура программы на языке NASM, методы компиляции и сборки программы на языке NASM, технология структурного программирования на ассемблере, техника реализации базовых алгоритмических структур на ассемблере, методика работы с математическим сопроцессором на ассемблере, методика обработки массивов данных на ассемблере, методика обработка файлов на ассемблере..

Реализуемые компетенции:

ОПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ОПК-4 - «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

Результаты освоения дисциплины.

Знать: архитектуру современного микропроцессора, основные мнемоники, директивы и типы данных компилятора NASM, структуру программы на языке NASM, методы компиляции и сборки программы на языке NASM, технологию структурного программирования на ассемблере, технику реализации базовых алгоритмических структур на ассемблере, методику работы с математическим сопроцессором на ассемблере, методику обработки массивов данных на ассемблере, методику обработка файлов на ассемблере.

Уметь: создавать вычислительно эффективные программные проекты на языке NASM с графическим пользовательским интерфейсом для решения

задач прикладной математики и информатики (решение задачи внутренней и внешней сортировки на ассемблере, расчет сплайнов на ассемблере, выполнение операции свертки на ассемблере, фурье-анализ, фурье-синтез, быстрое преобразование фурье на ассемблере, вейвлет-преобразования на ассемблере, численное интегрирование на ассемблере).

Владеть: методами проектирования вычислительно эффективного программного обеспечения, методами разработки быстродействующих алгоритмов на низкоуровневых языках программирования.

Рекомендуемое программное обеспечение:

Компилятор: NASM (версия не ниже 2.0),

Сборщик: GCC (версия не ниже 3.1),

Текстовый процессор: LibreOffice Writer (версия не ниже 5)

Просмотр документации: Foxit Reader или Adobe Reader или встроенное в ОС средство просмотра содержимого файлов формата PDF.

Оболочка навигации в файловой системе ОС: Far manager 3.0 (Windows), Midnight commander 4.5 (Linux)

Оболочка навигации в файловой системе ОС: Far manager 3.0 (Windows), Midnight commander 4.5 (Linux)

Операционная система: Windows (версия не ниже 7) или Linux Mint (версия не ниже 17.2 Cinnamon).

Рекомендуемое материально-техническое обеспечение: компьютерный класс (лаборатория, аудитория) с числом исправных ЭВМ по числу студентов группы плюс рабочее место преподавателя и производительностью, достаточной для освоения дисциплины. На всех ЭВМ класса (лаборатории, аудитории) программное обеспечение, выход в интернет для изучения интернет-источников дисциплины (со скоростью доступа, достаточной для комфортной работы).

Лабораторная работа №1

Тема: «Структура программы, основные мнемоники, директивы и типы данных компилятора NASM»

Цель работы: «закрепление знаний по теоретическим основам структуры программы, основным мнемоникам, директивам и типам данных компилятора NASM».

Формируемые компетенции (формируется часть компетенций):

ОПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ОПК-4 - «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

Ход работы

1. Орг. момент.
2. Выбор задания в соответствии с вариантом.
3. Изучение теоретической части (литературы) по теме работы.
4. Изучение техники безопасности (правил поведения в компьютерной лаборатории).
5. Выполнение работы.
 - загрузить шаблон программы.
 - внести в шаблон необходимые изменения.
 - отладить полученную компьютерную программу.
6. Оформление результатов работы.
7. Выводы по проделанной работе.
8. Защита работы.

Теоретическая часть

См. список рекомендуемой литературы.

Оборудование и материалы: см. приложение 1.

Техника безопасности: см. приложение 3.

Задания для лабораторной работы

Ввести с клавиатуры значения целых переменных a и b и вывести ответ по образцу:

Практическая работа N1

1ПМИ (магистратура), ФИО

Вариант N<номер варианта>

a= <значение a>

b= <значение b>

Ответ: c= <выражение> = <вычисленное значение>

Для выхода из программы нажмите Enter

<задержка выхода из программы>

Формулу для вычисления c взять в соответствии с вариантом

1. $c = (a + b)^2$

2. $c = a^2 + 2 * a * b - b^2$
3. $c = a + b + 2 * a * b$
4. $c = a - b + a^2$
5. $c = (a + 2 * b)^2$
6. $c = (a - 2 * b)^2$
7. $c = a^2 - b^2$
8. $c = (a + b) * (a - b)$
9. $c = (a - b)^2$
10. $c = b^2 + 2 * a * b - a^2$
11. $c = a + b - 2 * a * b$
12. $c = b - a + b^2$
13. $c = (2 * a + b)^2$
14. $c = (2 * a - b)^2$
15. $c = a^2 + b^2$

Контрольные вопросы

1. Архитектура современного микропроцессора, исполнение машинного кода микропроцессором x86.
2. Реальный и защищенный режимы работы микропроцессора x86, способы адресация памяти.
3. Регистры микропроцессора x86 и их назначение.
4. Прерывания и исключения микропроцессора x86.
5. Математический сопроцессор, технология MMX, технология SSE.
6. Назначение языков ассемблера.
7. Основные мнемоники языка NASM (мнемоники пересылки данных, мнемоники двоичной и десятичной арифметики, мнемоники логических операций, мнемоники передачи управления, мнемоники математического сопроцессора).
8. Основные директивы языка NASM (директивы определения сегментов данных, стека и кода, директивы подключения дополнительных файлов, директивы задания глобальных имен, директивы задания экспортируемых и импортируемых символов)
9. Основные типы данных языка NASM, правила объявления целых и вещественных чисел, символов, цепочек символов, команды заполнения памяти по шаблону и создания структур.
10. Структура программы на языке NASM.
11. Общая схема компиляции и сборки программы на языке ассемблера (назначение компилятора NASM и способы его запуска, назначение сборщика GCC и способы его запуска, автоматизация процесса компиляции и сборки программы, статическое и динамическое связывание).

Список литературы, рекомендуемый к использованию по данной теме

1. Гук М., Юров В. Процессоры Pentium 4, Athlon и Duron. - СПб.: Питер, 2006.

2. Королев Л.Н. Архитектура ЭВМ: Учеб. Пособие для вузов. - М.: «Научный мир», 2005.
3. Пирогов В.Ю. Ассемблер для Windows. - СПб.: БХВ-Петербург, 2008.
4. Ляхов П.А. Организация ЭВМ и систем. – М. 2006г.
5. Юров В.И. Assembler. Учебник для Вузов (2-е издание) - М.: Питер, 2006.

Дополнительная литература

1. Гук М. Процессоры Intel от 8086 до Pentium II. - СПб.: Питер, 1998.
2. Петзолд Ч. Программирование для Windows 95; в двух томах. пер. с англ. - СПб.: BHV - Санкт-Петербург, 1997.
3. Пирогов В.Ю. Assembler. Учебный курс. - М.: Нолидж, 2001.
4. Шилдт Герберт. Программирование на С и С++ для Windows 95. - Киев, 1996.

Интернет-ресурсы

1. www.osr.ru (издат. Открытые системы)
2. www.compres.ru (журнал компьютер пресс)
3. www.ibxt.ru (новости вычислительной техники)
4. <http://ru.wikipedia.org> (интернет-энциклопедия Википедия)
5. <http://www.intel.com>, <http://www.intel.ru> (на русском языке) Фирма Intel:

Лабораторная работа №2

Тема: «Структурное программирование на ассемблере»

Цель работы: «закрепление знаний по теоретическим основам структурного программирования на ассемблере».

Формируемые компетенции (формируется часть компетенций):

ОПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ОПК-4 - «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

Ход работы

1. Орг. момент.
2. Выбор задания в соответствии с вариантом.
3. Изучение теоретической части (литературы) по теме работы.
4. Изучение техники безопасности (правил поведения в компьютерной лаборатории).
5. Выполнение работы.
 - загрузить шаблон программы.
 - внести в шаблон необходимые изменения.
 - отладить полученную компьютерную программу.
6. Оформление результатов работы.
7. Выводы по проделанной работе.
8. Защита работы.

Теоретическая часть

См. список рекомендуемой литературы.

Оборудование и материалы: см. приложение 1.

Техника безопасности: см. приложение 3.

Задания для лабораторной работы

Ввести с клавиатуры значения вещественной переменной x и натуральной n и вывести ответ по образцу:

Практическая работа N2

1ПМИ (магистратура), ФИО

Вариант N<номер варианта>

$x =$ <значение a >

$n =$ <значение b >

Ответ: $f(x) =$ <вычисленное значение>

Для выхода из программы нажмите Enter

<задержка выхода из программы>

Формулу для вычисления $f(x)$ взять в соответствии с вариантом (математические упрощения формулы не проводить, повторяющиеся части формул вынести в вспомогательные подпрограммы, обязательно использование математического сопроцессора ЭВМ)

1. $f(x) = \sum_{i=1}^n \frac{(-1)^i + x}{i!}$
2. $f(x) = \sum_{i=1}^n \frac{x+1}{(-2)^i}$
3. $f(x) = \sum_{i=1}^n \frac{x+2^i}{x^i}$
4. $f(x) = \sum_{i=1}^n \frac{1}{x^i + i^2}$
5. $f(x) = \sum_{i=1}^n \frac{i}{3^i + x^i}$
6. $f(x) = \sum_{i=1}^n \frac{x^{2i}}{i + x^{2i}}$
7. $f(x) = \sum_{i=1}^n \frac{i^2}{x^{i+1}}$
8. $f(x) = \sum_{i=1}^n \frac{x^i}{3^i + x^i}$
9. $f(x) = \sum_{i=1}^n \frac{3}{2^i + (-x)^i}$
10. $f(x) = \sum_{i=1}^n \frac{1}{3^i + x^i}$
11. $f(x) = \sum_{i=1}^n \frac{x^2}{3^i + x^i}$
12. $f(x) = \sum_{i=1}^n \frac{x^3}{2^i + (-1)^i}$
13. $f(x) = \sum_{i=1}^n \frac{i^2}{(-3)^i + (-x)^i}$
14. $f(x) = \sum_{i=1}^n \frac{2-i}{2^i + x^i}$
15. $f(x) = \sum_{i=1}^n \frac{i}{3^i - x^i}$

Контрольные вопросы

1. Архитектура современного микропроцессора, исполнение машинного кода микропроцессором x86.
2. Реальный и защищенный режимы работы микропроцессора x86, способы адресации памяти.
3. Регистры микропроцессора x86 и их назначение.
4. Прерывания и исключения микропроцессора x86.
5. Математический сопроцессор, технология MMX, технология SSE.
6. Назначение языков ассемблера.
7. Основные мнемоники языка NASM (мнемоники пересылки данных, мнемоники двоичной и десятичной арифметики, мнемоники логических операций, мнемоники передачи управления, мнемоники математического сопроцессора).

8. Основные директивы языка NASM (директивы определения сегментов данных, стека и кода, директивы подключения дополнительных файлов, директивы задания глобальных имен, директивы задания экспортируемых и импортируемых символов)
9. Основные типы данных языка NASM, правила объявления целых и вещественных чисел, символов, цепочек символов, команды заполнения памяти по шаблону и создания структур.
10. Структура программы на языке NASM.
11. Общая схема компиляции и сборки программы на языке ассемблера (назначение компилятора NASM и способы его запуска, назначение сборщика GCC и способы его запуска, автоматизация процесса компиляции и сборки программы, статическое и динамическое связывание).
12. Структурное программирование на ассемблере (назначение вспомогательных алгоритмов, нисходящая и восходящая технологии структурного программирования, корректный вход в подпрограмму на языке NASM, корректный выход из подпрограммы на языке NASM, рекурсия)
13. Передача параметров подпрограмме на языке NASM через регистры и через стек, способы передачи вычисленного подпрограммой результата в основную программу.
14. Стандартная библиотека LIBC.
15. Базовые алгоритмические структуры (следование, ветвление, цикл) и их запись на языке ассемблера.

Список литературы, рекомендуемый к использованию по данной теме

1. Гук М., Юров В. Процессоры Pentium 4, Athlon и Duron. - СПб.: Питер, 2006.
2. Королев Л.Н. Архитектура ЭВМ: Учеб. Пособие для вузов. - М.: «Научный мир», 2005.
3. Пирогов В.Ю. Ассемблер для Windows. - СПб.: БХВ-Петербург, 2008.
4. Ляхов П.А. Организация ЭВМ и систем. – М. 2006г.
5. Юров В.И. Assembler. Учебник для Вузов (2-е издание) - М.: Питер, 2006.

Дополнительная литература

1. Гук М. Процессоры Intel от 8086 до Pentium II. - СПб.: Питер, 1998.
2. Петзолд Ч. Программирование для Windows 95; в двух томах. пер. с англ. - СПб.: BHV - Санкт-Петербург, 1997.
3. Пирогов В.Ю. Assembler. Учебный курс. - М.: Нолидж, 2001.
4. Шилдт Герберт. Программирование на С и С++ для Windows 95. - Киев, 1996.

Интернет-ресурсы

1. www.osp.ru (издат. Открытые системы)
2. www.compres.ru (журнал компьютер пресс)
3. www.ibxt.ru (новости вычислительной техники)
4. <http://ru.wikipedia.org> (интернет-энциклопедия Википедия)
5. <http://www.intel.com>, <http://www.intel.ru> (на русском языке) Фирма Intel:

Лабораторная работа №3

Тема: «Обработка массивов данных на ассемблере».

Цель работы: «закрепление знаний по теоретическим основам обработки массивов данных на ассемблере».

Формируемые компетенции (формируется часть компетенций):

ОПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ОПК-4 - «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

Ход работы

1. Орг. момент.
2. Выбор задания в соответствие с вариантом.
3. Изучение теоретической части (литературы) по теме работы.
4. Изучение техники безопасности (правил поведения в компьютерной лаборатории).
5. Выполнение работы.
 - загрузить шаблон программы.
 - внести в шаблон необходимые изменения.
 - отладить полученную компьютерную программу.
6. Оформление результатов работы.
7. Выводы по проделанной работе.
8. Защита работы.

Теоретическая часть

См. список рекомендуемой литературы.

Оборудование и материалы: см. приложение 1.

Техника безопасности: см. приложение 3.

Задания для лабораторной работы

Ввести с клавиатуры размер массива n и элементы массива $A[n]$ по образцу.

Практическая работа N3

1ПМИ (магистратура), ФИО

Вариант N<номер варианта>

$n =$ <значение n >

$A[1] =$ <значение $A[1]$ >

...

$A[n] =$ <значение $A[n]$ >

Ответ:

<результат работы программы>

Для выхода из программы нажмите Enter

<задержка выхода из программы>

Вывести ответ с пояснениями согласно задания (n заранее не известно).

1. Упорядочить элементы массива $A[n]$ по возрастанию (алгоритм пузырьковой сортировки)
2. Упорядочить элементы массива $A[n]$ по убыванию (алгоритм пузырьковой сортировки)
3. Упорядочить элементы массива $A[n]$ по возрастанию (алгоритм сортировки выбором)
4. Упорядочить элементы массива $A[n]$ по убыванию (алгоритм сортировки выбором)
5. Упорядочить элементы массива $A[n]$ по возрастанию (алгоритм быстрой сортировки)
6. Упорядочить элементы массива $A[n]$ по убыванию (алгоритм быстрой сортировки)
7. Упорядочить элементы массива $A[n]$ по возрастанию (алгоритм пирамидальной сортировки)
8. Упорядочить элементы массива $A[n]$ по убыванию (алгоритм пирамидальной сортировки)
9. Подсчитать в массиве $A[n]$ количество различных элементов
10. Найти в массиве $A[n]$ k -й по величине элемент
11. Найти сумму первых по величине k элементов массива $A[n]$
12. Найти произведение первых по величине k элементов массива $A[n]$
13. Упорядочить элементы массива $A[n]$ на четных индексах по возрастанию, а на нечетных по убыванию
14. Упорядочить элементы массива $A[n]$ на четных индексах по убыванию, а на нечетных по возрастанию
15. Найти в массиве $A[n]$ наиболее длинную цепочку одинаковых подряд стоящих элементов

Контрольные вопросы

1. Архитектура современного микропроцессора, исполнение машинного кода микропроцессором x86.
2. Реальный и защищенный режимы работы микропроцессора x86, способы адресации памяти.
3. Регистры микропроцессора x86 и их назначение.
4. Прерывания и исключения микропроцессора x86.
5. Математический сопроцессор, технология MMX, технология SSE.
6. Назначение языков ассемблера.
7. Основные мнемоники языка NASM (мнемоники пересылки данных, мнемоники двоичной и десятичной арифметики, мнемоники логических операций, мнемоники передачи управления, мнемоники математического сопроцессора).
8. Основные директивы языка NASM (директивы определения сегментов данных, стека и кода, директивы подключения дополнительных файлов, директивы задания глобальных имен, директивы задания экспортируемых и импортируемых символов)

9. Основные типы данных языка NASM, правила объявления целых и вещественных чисел, символов, цепочек символов, команды заполнения памяти по шаблону и создания структур.
10. Структура программы на языке NASM.
11. Общая схема компиляции и сборки программы на языке ассемблера (назначение компилятора NASM и способы его запуска, назначение сборщика GCC и способы его запуска, автоматизация процесса компиляции и сборки программы, статическое и динамическое связывание).
12. Структурное программирование на ассемблере (назначение вспомогательных алгоритмов, нисходящая и восходящая технологии структурного программирования, корректный вход в подпрограмму на языке NASM, корректный выход из подпрограммы на языке NASM, рекурсия)
13. Передача параметров подпрограмме на языке NASM через регистры и через стек, способы передачи вычисленного подпрограммой результата в основную программу.
14. Стандартная библиотека LIBC.
15. Событийно-ориентированная технология программирования, назначение цикла сообщений, оконной процедуры, точки входа в программу, основные сообщения, посылаемые от операционной системы к программе, типовые классы окон.
16. Базовые алгоритмические структуры (следование, ветвление, цикл) и их запись на языке ассемблера.
17. Вычислительно сложные математические функции и методы расчета их значений на ассемблере.
18. Функции динамического выделения и высвобождения памяти из стандартной библиотеки.
19. Обработка файлов на ассемблере (назначение файлового дескриптора, API открытия и закрытия файлов, чтение и запись информации в файлы).

Список литературы, рекомендуемый к использованию по данной теме

1. Гук М., Юров В. Процессоры Pentium 4, Athlon и Duron. - СПб.: Питер, 2006.
2. Королев Л.Н. Архитектура ЭВМ: Учеб. Пособие для вузов. - М.: «Научный мир», 2005.
3. Пирогов В.Ю. Ассемблер для Windows. - СПб.: БХВ-Петербург, 2008.
4. Ляхов П.А. Организация ЭВМ и систем. – М. 2006г.
5. Юров В.И. Assembler. Учебник для Вузов (2-е издание) - М.: Питер, 2006.

Дополнительная литература

1. Гук М. Процессоры Intel от 8086 до Pentium II. - СПб.: Питер, 1998.
2. Петзолд Ч. Программирование для Windows 95; в двух томах. пер. с англ. - СПб.: BHV - Санкт-Петербург, 1997.
3. Пирогов В.Ю. Assembler. Учебный курс. - М.: Нолидж, 2001.
4. Шилдт Герберт. Программирование на С и С++ для Windows 95. - Киев, 1996.

Интернет-ресурсы

1. www.osp.ru (издат. Открытые системы)
2. www.compres.ru (журнал компьютер пресс)
3. www.ibxt.ru (новости вычислительной техники)
4. <http://ru.wikipedia.org> (интернет-энциклопедия Википедия)
5. <http://www.intel.com>, <http://www.intel.ru> (на русском языке) Фирма Intel:

Литература

Основная литература:

1. Гук М., Юров В. Процессоры Pentium 4, Athlon и Duron. - СПб.: Питер, 2006.
2. Королев Л.Н. Архитектура ЭВМ: Учеб. Пособие для вузов. - М.: «Научный мир», 2005.
3. Пирогов В.Ю. Ассемблер для Windows. - СПб.: БХВ-Петербург, 2008.
4. Ляхов П.А. Организация ЭВМ и систем. – М. 2006г.
5. Юров В.И. Assembler. Учебник для Вузов (2-е издание) - М.: Питер, 2006.

Дополнительная литература

1. Гук М. Процессоры Intel от 8086 до Pentium II. - СПб.: Питер, 1998.
2. Петзолд Ч. Программирование для Windows 95; в двух томах. пер. с англ. - СПб.: BHV - Санкт-Петербург, 1997.
3. Пирогов В.Ю. Assembler. Учебный курс. - М.: Нолидж, 2001.
4. Шилдт Герберт. Программирование на С и С++ для Windows 95. - Киев, 1996.

Интернет-ресурсы

1. www.osr.ru (издат. Открытые системы)
2. www.compres.ru (журнал компьютер пресс)
3. www.ibxt.ru (новости вычислительной техники)
4. <http://ru.wikipedia.org> (интернет-энциклопедия Википедия)
5. <http://www.intel.com>, <http://www.intel.ru> (на русском языке) Фирма Intel:

**Оборудование, материалы и программное обеспечение, используемые
для организации и проведения лабораторных работ**

Оборудование:

Компьютерный класс (лаборатория, аудитория) с числом исправных ЭВМ по числу студентов группы плюс рабочее место преподавателя и производительностью, достаточной для освоения дисциплины.

Программное обеспечение:

1. Компилятор: NASM (версия не ниже 2.0),
2. Сборщик: GCC (версия не ниже 3.1),
3. Текстовый процессор: LibreOffice Writer (версия не ниже 5)
4. Просмотр документации: Foxit Reader или Adobe Reader или встроенное в ОС средство просмотра содержимого файлов формата PDF.
5. Оболочка навигации в файловой системе ОС: Far manager 3.0 (Windows), Midnight commander 4.5 (Linux)
6. Оболочка навигации в файловой системе ОС: Far manager 3.0 (Windows), Midnight commander 4.5 (Linux)
7. Операционная система: Windows (версия не ниже 7) или Linux Mint (версия не ниже 17.2 Cinnamon).

Содержимое отчета по лабораторной работе

Лабораторная работа считается выполненной студентом и засчитывается преподавателем (с зачислением баллов пропорционально выполненным пунктам отчета, т. е. необязательно, чтобы отчет содержал все пункты) при предоставлении в электронном виде к защите (бумажная распечатка не требуется!!!) аккуратно оформленного файла-отчета в формате .odt (OpenOffice, LibreOffice) или .doc (Microsoft Office), именованный по шаблону (при несоответствии работа не допускается к защите):

ASM_MAG_LR(номер работы)_V(номер варианта)_(фамилия).odt

Файл-отчет должен иметь следующие элементы в указанном порядке (с сохранением их номера в случае пропуска некоторых пунктов):

0. Отметка об ознакомлении с техникой безопасности (правилами поведения в компьютерном классе (аудитории, лаборатории)).

В данном пункте необходимо наличие записи «С правилами техники безопасности при выполнении лабораторной работы и поведения в компьютерном классе (аудитории, лаборатории) ознакомлен, обязуюсь выполнять _____ (ФИО), _____ дата».

1. Постановка задачи (полное условие в соответствии с выбранным вариантом).

2. Подробная математическая (совокупность формул и поясняющего формулы текста) или информационной (словесное описание алгоритма) модели решения задачи. В данном пункте допустимо наличие экранных снимков расчетов в системах компьютерной алгебры (SciLab, Octave, Maxima и т.п.) или профессиональных графических системах. Использование проприетарных (коммерческих) систем (MathCad, MatLab и т.п.) не рекомендуется.

3. Тестовые наборы исходных данных и соответствующих им правильных результатов для проверки работоспособности программы (в данном пункте **должно быть ручное решение задачи с промежуточными выкладками**). В данном пункте допустимо наличие экранных снимков расчетов в системах компьютерной алгебры (SciLab, Octave, Maxima и т.п.) или профессиональных графических системах. Использование проприетарных (коммерческих) систем (MathCad, MatLab и т.п.) не рекомендуется.

4. Указание имен, типов и назначения всех переменных и сигналов, входящих в математическую или информационную модель.

5. Основной и вспомогательные (если есть) алгоритмы решения задачи (допустимо описание алгоритма на алгоритмическом языке, например, Pascal).

6. Запись полных имен файлов, образующих проект с указанием назначения каждого файла (минимум указать имена и содержимое файлов с расширением .asm, .s и т.д. (какие подпрограммы содержат, что делают).

7. Полные исходные тексты компьютерных программ (в каждом файле обязательно наличие информации о разработчике — ФИО, курс, группа, специальность, университет).

8. Экранный снимок или распечатка результата работы компьютерной программы при введенных тестовых данных из пункта 3 отчета. Конкретно в данном пункте необходимо сравнить полученные результаты с эталонным тестовым решением и написать «результат совпадает с эталонным решением».

9. Исследование проекта.

В данном пункте требуется записать несколько наборов входных данных (определить самостоятельно) и соответствующих им результатов. Оценить примерное число шагов и время работы алгоритма решения задачи в общем случае.

10. Выводы о проделанной работе.

Указать возможности, достоинства и недостатки, привести примеры использования.

Правила поведения в компьютерном классе (аудитории, лаборатории)

1. К работе в компьютерном классе допускаются лица, ознакомленные с инструкцией по технике безопасности и охране труда, с правилами поведения и размещения информационных ресурсов.
2. Работа студентов в компьютерном классе разрешается только в присутствии преподавателя (инженера, лаборанта).
3. Во время групповых занятий посторонние лица могут находиться в классе только с разрешения преподавателя.
4. **Перед началом работы необходимо:**
 - убедиться в отсутствии видимых повреждений на рабочем месте;
 - разместить на столе тетради, учебные пособия так, чтобы они не мешали работе на компьютере;
 - принять правильную рабочую позу;
 - если сеанс работы предыдущего пользователя не был завершен, завершить его;
 - ввести регистрационную информацию (при необходимости).
5. **При работе в компьютерном классе категорически запрещается:**
 - находиться в классе в верхней одежде;
 - размещать одежду и сумки на рабочих местах;
 - находиться в классе с едой и напитками;
 - класть книги, тетради и т.п. на клавиатуру;
 - располагаться сбоку или сзади от включенного монитора;
 - присоединять или отсоединять кабели, трогать разъемы, провода и розетки;
 - передвигать компьютеры;
 - открывать системный блок;
 - пытаться самостоятельно устранять неисправности в работе аппаратуры;
 - перекрывать вентиляционные отверстия на системном блоке и мониторе;
 - ударять по клавиатуре, нажимать бесцельно на клавиши;
 - удалять или перемещать чужие файлы;
 - устанавливать и запускать компьютерные игры;
 - использовать Интернет-ресурсы неучебного назначения.
6. **Находясь в компьютерном классе, необходимо:**
 - соблюдать тишину и порядок, выключать мобильные телефоны от громкой связи;
 - выполнять все требования преподавателя, инженера и лаборанта;
 - работать только под своим именем и паролем;
 - соблюдать режим работы (продолжительность непрерывной работы за компьютером не более двух часов с обязательным 10-минутным перерывом и

гимнастикой для глаз; продолжительность интенсивной работы с клавиатурой не более 30 минут с последующей гимнастикой для рук; общая продолжительность работы не более 4 часов в день);

– при появлении рези в глазах, резком ухудшении видимости, невозможности сфокусировать взгляд или навести его на резкость, появлении боли в пальцах и кистях рук, усилении сердцебиения немедленно покинуть рабочее место, сообщить о происшедшем преподавателю и обратиться к врачу;

– после окончания работы завершить все активные программы и корректно завершить сеанс;

– оставить рабочее место чистым.

7. Работая за компьютером, необходимо соблюдать правильную позу:

– расстояние от экрана до глаз 70-80 см (расстояние вытянутой руки);

– вертикально прямая спина;

– плечи опущены и расслаблены;

– ноги на полу и не скрещены;

– локти, запястья и кисти рук на одном уровне;

– локтевые, тазобедренные, коленные, голеностопные суставы под прямым углом.

8. При появлении программных ошибок или сбоях оборудования студент обязан немедленно обратиться к преподавателю (инженеру, лаборанту).

9. В случае порчи или выхода из строя оборудования компьютерного класса по вине пользователя ремонт или замена оборудования производится за счет пользователя.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические рекомендации для студентов
по организации самостоятельной работы по дисциплине
Машинно-ориентированные языки программирования
Направление подготовки
01.04.02 - Прикладная математика и информатика
Профиль подготовки
Математическое моделирование
Квалификация выпускника
магистр
Учебный план 2024 г. Изучается в 1 семестре

Ставрополь, 2024

Печатается по решению
редакционно-издательского совета
Северо-Кавказского федерального
университета

Ионисян А.С. Машинно-ориентированные языки программирования. Методические рекомендации для студентов по организации самостоятельной работы. – Ставрополь: Издательство Северо-Кавказского федерального университета, 2024.– ____ с.

Рекомендации подготовлены в соответствии с Федеральным государственным образовательным стандартом высшего образования, раскрывает основные особенности изучения машинно-ориентированных языков программирования.

В рекомендациях изложены основные положения машинно-ориентированных языков программирования, вводятся и иллюстрируются примерами основные методы работы с ЭВМ. Уделено внимание типам данных, определяемым программистом, показаны особенности программной реализации решений.

Методические рекомендации по дисциплине «Машинно-ориентированные языки программирования» одобрены на заседании кафедры прикладной математики и математического моделирования «__» _____ 2024 г., протокол № ____.

Рекомендации предназначены для организации самостоятельной работы студентов по дисциплине «Машинно-ориентированные языки программирования» для направления 01.04.02 «Прикладная математика и информатика» (Магистр).

© ФГАОУ ВПО «Северо-Кавказский
федеральный университет», 202____

Содержание

		Стр.
1.	Введение	
2	Общая характеристика самостоятельной работы студента при изучении дисциплины	
3.	План-график выполнения самостоятельной работы	
4.	Методические рекомендации по изучению теоретического материала	
5	Методические указания (по видам работ, предусмотренных рабочей программой дисциплины)	
6.	Методические указания по подготовке к экзамену	
7.	Список рекомендуемой литературы	
8.	Методические указания по выполнению курсовых работ	

Введение

Дисциплина «Машинно-ориентированные языки программирования» относится к обязательной части учебного плана 2024 года и изучается студентами магистратуры специальности 01.04.02 направления «Прикладная математика и информатика» в 1 семестре.

Целями освоения дисциплины «Машинно-ориентированные языки программирования» являются:

1. Формирование представлений и фундаментальных знаний об архитектуре современной микропроцессорной техники;
2. Формирование представлений и фундаментальных знаний о современных методах низкоуровневого программирования;
3. Реализация математических алгоритмов на ЭВМ при помощи низкоуровневого языка программирования Netwide Assembler.

Основными задачами изучения дисциплины являются:

1. Изучение принципов создания быстродействующего программного обеспечения.
2. Изучение языков программирования низкого уровня.
3. Реализация на языке ассемблера базовых алгоритмических структур,
4. Реализация на языке ассемблера типовых алгоритмов прикладной математики.

Формы отчетности: экзамен - 1 семестр.

Содержание дисциплины: архитектура современного микропроцессора, основные мнемоники, директивы и типы данных компилятора NASM, структура программы на языке NASM, методы компиляции и сборки программы на языке NASM, технология структурного программирования на ассемблере, техника реализации базовых алгоритмических структур на ассемблере, методика работы с математическим сопроцессором на ассемблере, методика обработки массивов данных на ассемблере, методика обработка файлов на ассемблере..

Реализуемые компетенции:

ОПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ОПК-4 - «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

Результаты освоения дисциплины.

Знать: архитектуру современного микропроцессора, основные мнемоники, директивы и типы данных компилятора NASM, структуру программы на языке NASM, методы компиляции и сборки программы на языке NASM, технологию структурного программирования на ассемблере, технику реализации базовых алгоритмических структур на ассемблере, методику работы с математическим сопроцессором на ассемблере, методику обработки массивов данных на ассемблере, методику обработка файлов на ассемблере.

Уметь: создавать вычислительно эффективные программные проекты на языке NASM с графическим пользовательским интерфейсом для решения

задач прикладной математики и информатики (решение задачи внутренней и внешней сортировки на ассемблере, расчет сплайнов на ассемблере, выполнение операции свертки на ассемблере, фурье-анализ, фурье-синтез, быстрое преобразование фурье на ассемблере, вейвлет-преобразования на ассемблере, численное интегрирование на ассемблере).

Владеть: методами проектирования вычислительно эффективного программного обеспечения, методами разработки быстродействующих алгоритмов на низкоуровневых языках программирования.

Рекомендуемое программное обеспечение:

Компилятор: NASM (версия не ниже 2.0),

Сборщик: GCC (версия не ниже 3.1),

Текстовый процессор: LibreOffice Writer (версия не ниже 5)

Просмотр документации: Foxit Reader или Adobe Reader или встроенное в ОС средство просмотра содержимого файлов формата PDF.

Оболочка навигации в файловой системе ОС: Far manager 3.0 (Windows), Midnight commander 4.5 (Linux)

Оболочка навигации в файловой системе ОС: Far manager 3.0 (Windows), Midnight commander 4.5 (Linux)

Операционная система: Windows (версия не ниже 7) или Linux Mint (версия не ниже 17.2 Cinnamon).

Рекомендуемое материально-техническое обеспечение: компьютерный класс (лаборатория, аудитория) с числом исправных ЭВМ по числу студентов группы плюс рабочее место преподавателя и производительностью, достаточной для освоения дисциплины. На всех ЭВМ класса (лаборатории, аудитории) программное обеспечение, выход в интернет для изучения интернет-источников дисциплины (со скоростью доступа, достаточной для комфортной работы).

Общая характеристика самостоятельной работы студента при изучении дисциплины «Машинно-ориентированные языки программирования»

Лабораторная работа №1

Тема: «Структура программы, основные мнемоники, директивы и типы данных компилятора NASM»

Цель работы: «закрепление знаний по теоретическим основам структуры программы, основным мнемоникам, директивам и типам данных компилятора NASM».

Задачи:

1. Изучение принципов создания быстродействующего программного обеспечения.
2. Изучение языков программирования низкого уровня.
3. Реализация на языке ассемблера базовых алгоритмических структур,
4. Реализация на языке ассемблера типовых алгоритмов прикладной математики.

Формируемые компетенции (формируется часть компетенций):

ОПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ОПК-4 - «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

Лабораторная работа №2

Тема: «Структурное программирование на ассемблере»

Цель работы: «закрепление знаний по теоретическим основам структурного программирования на ассемблере».

Задачи:

1. Изучение принципов создания быстродействующего программного обеспечения.
2. Изучение языков программирования низкого уровня.
3. Реализация на языке ассемблера базовых алгоритмических структур,
4. Реализация на языке ассемблера типовых алгоритмов прикладной математики.

Формируемые компетенции (формируется часть компетенций):

ОПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ОПК-4 - «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

Лабораторная работа №3

Тема: «Обработка массивов данных на ассемблере».

Цель работы: «закрепление знаний по теоретическим основам обработки массивов данных на ассемблере».

Задачи:

1. Изучение принципов создания быстродействующего программного обеспечения.
2. Изучение языков программирования низкого уровня.
3. Реализация на языке ассемблера базовых алгоритмических структур,
4. Реализация на языке ассемблера типовых алгоритмов прикладной математики.

Формируемые компетенции (формируется часть компетенций):

ОПК-2 – «Способность разрабатывать, отлаживать, модифицировать и поддерживать системное программное обеспечение».

ОПК-4 - «Способность осуществлять планирование организации разработки и интеграции системного программного обеспечения».

План-график выполнения самостоятельной работы

В рамках рейтинговой системы успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации.

№ п/п	Вид деятельности студентов	Сроки выполнения	Количество баллов
1	Подготовка к лабораторной работе №1	5	15
3	Подготовка к лабораторной работе №2	11	20
3	Подготовка к лабораторной работе №3	15	20
	Итого за 2 семестр		55
	Итого		55

Оценка 5 (отлично) выставляется студентам, своевременно выполнившим лабораторную работу, не имеющим пропусков лекционных и практических занятий.

Оценка 4 (хорошо) выставляется студентам, выполнившим лабораторную работу, возможно с задержкой (до наступления срока защиты следующей лабораторной работы) и имеющим до 30% пропусков лекционных и лабораторных занятий.

Оценка 3 (удовлетворительно) выставляется студентам, выполнившим лабораторную работу с существенной задержкой и имеющим большое (свыше 30%) количество пропусков лекционных и практических занятий.

Оценка 2 (неудовлетворительно) выставляется студентам, не защитившим лабораторную работу до экзамена, либо вообще потерявшим связь с университетом.

Методические рекомендации по изучению теоретического материала

Лабораторная работа №1

Тема: «Структура программы, основные мнемоники, директивы и типы данных компилятора NASM»

Контрольные вопросы

1. Архитектура современного микропроцессора, исполнение машинного кода микропроцессором x86.
2. Реальный и защищенный режимы работы микропроцессора x86, способы адресации памяти.
3. Регистры микропроцессора x86 и их назначение.
4. Прерывания и исключения микропроцессора x86.
5. Математический сопроцессор, технология MMX, технология SSE.
6. Назначение языков ассемблера.
7. Основные мнемоники языка NASM (мнемоники пересылки данных, мнемоники двоичной и десятичной арифметики, мнемоники логических операций, мнемоники передачи управления, мнемоники математического сопроцессора).
8. Основные директивы языка NASM (директивы определения сегментов данных, стека и кода, директивы подключения дополнительных файлов, директивы задания глобальных имен, директивы задания экспортируемых и импортируемых символов)
9. Основные типы данных языка NASM, правила объявления целых и вещественных чисел, символов, цепочек символов, команды заполнения памяти по шаблону и создания структур.
10. Структура программы на языке NASM.
11. Общая схема компиляции и сборки программы на языке ассемблера (назначение компилятора NASM и способы его запуска, назначение сборщика GCC и способы его запуска, автоматизация процесса компиляции и сборки программы, статическое и динамическое связывание).

Лабораторная работа №2

Тема: «Структурное программирование на ассемблере»

Контрольные вопросы

1. Архитектура современного микропроцессора, исполнение машинного кода микропроцессором x86.
2. Реальный и защищенный режимы работы микропроцессора x86, способы адресации памяти.
3. Регистры микропроцессора x86 и их назначение.
4. Прерывания и исключения микропроцессора x86.
5. Математический сопроцессор, технология MMX, технология SSE.
6. Назначение языков ассемблера.
7. Основные мнемоники языка NASM (мнемоники пересылки данных, мнемоники двоичной и десятичной арифметики, мнемоники логических

операций, мнемоники передачи управления, мнемоники математического сопроцессора).

8. Основные директивы языка NASM (директивы определения сегментов данных, стека и кода, директивы подключения дополнительных файлов, директивы задания глобальных имен, директивы задания экспортируемых и импортируемых символов)

9. Основные типы данных языка NASM, правила объявления целых и вещественных чисел, символов, цепочек символов, команды заполнения памяти по шаблону и создания структур.

10. Структура программы на языке NASM.

11. Общая схема компиляции и сборки программы на языке ассемблера (назначение компилятора NASM и способы его запуска, назначение сборщика GCC и способы его запуска, автоматизация процесса компиляции и сборки программы, статическое и динамическое связывание).

12. Структурное программирование на ассемблере (назначение вспомогательных алгоритмов, нисходящая и восходящая технологии структурного программирования, корректный вход в подпрограмму на языке NASM, корректный выход из подпрограммы на языке NASM, рекурсия)

13. Передача параметров подпрограмме на языке NASM через регистры и через стек, способы передачи вычисленного подпрограммой результата в основную программу.

14. Стандартная библиотека LIBC.

15. Базовые алгоритмические структуры (следование, ветвление, цикл) и их запись на языке ассемблера.

Лабораторная работа №3

Тема: «Обработка массивов данных на ассемблере».

Контрольные вопросы

1. Архитектура современного микропроцессора, исполнение машинного кода микропроцессором x86.

2. Реальный и защищенный режимы работы микропроцессора x86, способы адресация памяти.

3. Регистры микропроцессора x86 и их назначение.

4. Прерывания и исключения микропроцессора x86.

5. Математический сопроцессор, технология MMX, технология SSE.

6. Назначение языков ассемблера.

7. Основные мнемоники языка NASM (мнемоники пересылки данных, мнемоники двоичной и десятичной арифметики, мнемоники логических операций, мнемоники передачи управления, мнемоники математического сопроцессора).

8. Основные директивы языка NASM (директивы определения сегментов данных, стека и кода, директивы подключения дополнительных файлов, директивы задания глобальных имен, директивы задания экспортируемых и импортируемых символов)

9. Основные типы данных языка NASM, правила объявления целых и вещественных чисел, символов, цепочек символов, команды заполнения памяти по шаблону и создания структур.
10. Структура программы на языке NASM.
11. Общая схема компиляции и сборки программы на языке ассемблера (назначение компилятора NASM и способы его запуска, назначение сборщика GCC и способы его запуска, автоматизация процесса компиляции и сборки программы, статическое и динамическое связывание).
12. Структурное программирование на ассемблере (назначение вспомогательных алгоритмов, нисходящая и восходящая технологии структурного программирования, корректный вход в подпрограмму на языке NASM, корректный выход из подпрограммы на языке NASM, рекурсия)
13. Передача параметров подпрограмме на языке NASM через регистры и через стек, способы передачи вычисленного подпрограммой результата в основную программу.
14. Стандартная библиотека LIBC.
15. Событийно-ориентированная технология программирования, назначение цикла сообщений, оконной процедуры, точки входа в программу, основные сообщения, посылаемые от операционной системы к программе, типовые классы окон.
16. Базовые алгоритмические структуры (следование, ветвление, цикл) и их запись на языке ассемблера.
17. Вычислительно сложные математические функции и методы расчета их значений на ассемблере.
18. Функции динамического выделения и высвобождения памяти из стандартной библиотеки.
19. Обработка файлов на ассемблере (назначение файлового дескриптора, API открытия и закрытия файлов, чтение и запись информации в файлы).

Методические указания (по видам работ, предусмотренных рабочей программой дисциплины)

Текущая аттестация студентов проводится преподавателями, ведущими лабораторные работы по дисциплине.

При проверке лабораторной работы, оцениваются правильность ответа, последовательность и рациональность выполнения, точность расчетов.

Максимальное количество баллов студент получает, если оформление отчета соответствует установленным требованиям, а отчет полностью раскрывает суть работы. Основанием для снижения оценки являются:

- незначительные нарушения правил оформления отчета;
- затруднения студентов при ответе на некоторые контрольные вопросы;

Отчет может быть отправлен на доработку в следующих случаях:

- отсутствия каких-либо элементов отчета.
- абсолютное не соответствие требованиям к оформлению.

Лабораторная работа считается выполненной студентом и засчитывается преподавателем (с зачислением баллов пропорционально выполненным пунктам отчета, т. е. необязательно, чтобы отчет содержал все пункты) при предоставлении в электронном виде к защите (бумажная распечатка не требуется!!!) аккуратно оформленного файла-отчета в формате .odt (OpenOffice, LibreOffice) или .doc (Microsoft Office), именованный по шаблону (при несоответствии работа не допускается к защите):

ASM_MAG_LR(номер работы)_V(номер варианта)_(фамилия).odt

Файл-отчет должен иметь следующие элементы в указанном порядке (с сохранением их номера в случае пропуска некоторых пунктов):

0. Отметка об ознакомлении с техникой безопасности (правилами поведения в компьютерном классе (аудитории, лаборатории)).

В данном пункте необходимо наличие записи «С правилами техники безопасности при выполнении лабораторной работы и поведения в компьютерном классе (аудитории, лаборатории) ознакомлен, обязуюсь выполнять _____ (ФИО), _____ дата».

1. Постановка задачи (полное условие в соответствии с выбранным вариантом).

2. Подробная математическая (совокупность формул и поясняющего формулы текста) или информационной (словесное описание алгоритма) модели решения задачи. В данном пункте допустимо наличие экранных снимков расчетов в системах компьютерной алгебры (SciLab, Octave, Maxima и т.п.) или профессиональных графических системах. Использование проприетарных (коммерческих) систем (MathCad, MatLab и т.п.) не рекомендуется.

3. Тестовые наборы исходных данных и соответствующих им правильных результатов для проверки работоспособности программы (в данном пункте **должно быть ручное решение задачи с промежуточными выкладками**). В данном пункте допустимо наличие экранных снимков расчетов в системах

компьютерной алгебры (SciLab, Octave, Maxima и т.п.) или профессиональных графических системах. Использование проприетарных (коммерческих) систем (MathCad, MatLab и т.п.) не рекомендуется.

4. Указание имен, типов и назначения всех переменных и сигналов, входящих в математическую или информационную модель.

5. Основной и вспомогательные (если есть) алгоритмы решения задачи (допустимо описание алгоритма на алгоритмическом языке, например, Pascal).

6. Запись полных имен файлов, образующих проект с указанием назначения каждого файла (минимум указать имена и содержимое файлов с расширением .asm, .s и т.д. (какие подпрограммы содержат, что делают).

7. Полные исходные тексты компьютерных программ (в каждом файле обязательно наличие информации о разработчике — ФИО, курс, группа, специальность, университет).

8. Экранный снимок или распечатка результата работы компьютерной программы при введенных тестовых данных из пункта 3 отчета. Конкретно в данном пункте необходимо сравнить полученные результаты с эталонным тестовым решением и написать «результат совпадает с эталонным решением».

9. Исследование проекта.

В данном пункте требуется записать несколько наборов входных данных (определить самостоятельно) и соответствующих им результатов. Оценить примерное число шагов и время работы алгоритма решения задачи в общем случае.

10. Выводы о проделанной работе.

Указать возможности, достоинства и недостатки, привести примеры использования.

Методические указания по подготовке к экзамену

Процедура проведения экзамена осуществляется в соответствии с Положением о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования в СКФУ.

В экзаменационный билет включаются 2 теоретических вопроса и задача.

Для подготовки по билету отводится 40 минут

При подготовке к ответу студенту предоставляется право пользования справочными таблицами, калькулятором, персональной ЭВМ с установленным на ней программным обеспечением дисциплины.

Оценка «отлично» выставляется студенту, если:

он безукоризненно изложил теоретический материал без наводящих вопросов преподавателя и решил предложенную задачу и разработал алгоритм на языке высокого уровня. Возможны одна - две неточности при освещении второстепенных вопросов или выкладках, которые студент легко исправил после замечания преподавателя

Оценка «хорошо» выставляется студенту, если:

он показал систематический характер знаний по дисциплине и способность к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности; если допущены один - два недочёта при изложении основного содержания ответа и освещении второстепенных вопросов или выкладках, легко исправленные после замечания преподавателя.

Оценка «удовлетворительно» выставляется студенту, если:

он допустил погрешности в ответе и при выполнении заданий, недостаточно правильно формулировал основные положения программного материала, нарушал последовательность в его изложении, показал недостаточную сформированность основных умений и навыков при выполнении практического задания, но обладающим необходимыми знаниями для их устранения под руководством преподавателя

Оценка «неудовлетворительно» выставляется студенту, если:

обнаружившему пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой практических знаний.

Вопросы к экзамену

1. Архитектура современного микропроцессора, исполнение машинного кода микропроцессором x86.
2. Реальный и защищенный режимы работы микропроцессора x86, способы адресация памяти.
3. Регистры микропроцессора x86 и их назначение.
4. Прерывания и исключения микропроцессора x86.
5. Математический сопроцессор, технология MMX, технология SSE.
6. Назначение языков ассемблера.

7. Основные мнемоники языка NASM (мнемоники пересылки данных, мнемоники двоичной и десятичной арифметики, мнемоники логических операций, мнемоники передачи управления, мнемоники математического сопроцессора).
8. Основные директивы языка NASM (директивы определения сегментов данных, стека и кода, директивы подключения дополнительных файлов, директивы задания глобальных имен, директивы задания экспортируемых и импортируемых символов)
9. Основные типы данных языка NASM, правила объявления целых и вещественных чисел, символов, цепочек символов, команды заполнения памяти по шаблону и создания структур.
10. Структура программы на языке NASM.
11. Общая схема компиляции и сборки программы на языке ассемблера (назначение компилятора NASM и способы его запуска, назначение сборщика GCC и способы его запуска, автоматизация процесса компиляции и сборки программы, статическое и динамическое связывание).
12. Структурное программирование на ассемблере (назначение вспомогательных алгоритмов, нисходящая и восходящая технологии структурного программирования, корректный вход в подпрограмму на языке NASM, корректный выход из подпрограммы на языке NASM, рекурсия)
13. Передача параметров подпрограмме на языке NASM через регистры и через стек, способы передачи вычисленного подпрограммой результата в основную программу.
14. Стандартная библиотека LIBC.
15. Событийно-ориентированная технология программирования, назначение цикла сообщений, оконной процедуры, точки входа в программу, основные сообщения, посылаемые от операционной системы к программе, типовые классы окон.
16. Базовые алгоритмические структуры (следование, ветвление, цикл) и их запись на языке ассемблера.
17. Вычислительно сложные математические функции и методы расчета их значений на ассемблере.
18. Функции динамического выделения и высвобождения памяти из стандартной библиотеки.
19. Обработка файлов на ассемблере (назначение файлового дескриптора, API открытия и закрытия файлов, чтение и запись информации в файлы).

Список рекомендуемой литературы

Основная литература:

1. Гук М., Юров В. Процессоры Pentium 4, Athlon и Duron. - СПб.: Питер, 2006.
2. Королев Л.Н. Архитектура ЭВМ: Учеб. Пособие для вузов. - М.: «Научный мир», 2005.
3. Пирогов В.Ю. Ассемблер для Windows. - СПб.: БХВ-Петербург, 2008.
4. Червяков Н.И. Организация ЭВМ и систем. – М. 2006г.
5. Юров В.И. Assembler. Учебник для Вузов (2-е издание) - М.: Питер, 2006.

Дополнительная литература

1. Гук М. Процессоры Intel от 8086 до Pentium II. - СПб.: Питер, 1998.
2. Петзолд Ч. Программирование для Windows 95; в двух томах. пер. с англ. - СПб.: BHV - Санкт-Петербург, 1997.
3. Пирогов В.Ю. Assembler. Учебный курс. - М.: Нолидж, 2001.
4. Шилдт Герберт. Программирование на С и С++ для Windows 95. - Киев, 1996.

Интернет-ресурсы

1. www.osr.ru (издат. Открытые системы)
2. www.compres.ru (журнал компьютер пресс)
3. www.ibxt.ru (новости вычислительной техники)
4. <http://ru.wikipedia.org> (интернет-энциклопедия Википедия)
5. <http://www.intel.com>, <http://www.intel.ru> (на русском языке) Фирма Intel:

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания по выполнению курсовых работ
по дисциплине: Машинно-ориентированные языки
программирования

Направление подготовки
01.04.02 - Прикладная математика и информатика
Профиль подготовки
Математическое моделирование
Квалификация выпускника
магистр
Учебный план 2024 г. Изучается в 1 семестре

Тематика курсовых работ

по дисциплине: Машинно-ориентированные языки программирования

1. Реализация методов оптимизации для нейронных сетей на машинно-ориентированных языках программирования
2. Реализация цифровых фильтров с вычислениями по методу Винограда на машинно-ориентированных языках программирования
3. Исследование способов исполнения машинного кода микропроцессором x86
4. Исследование реального и защищенного режимов работы микропроцессора x86
5. Исследование способов адресации памяти микропроцессора x86
6. Исследование регистров микропроцессора x86
7. Исследование прерываний и исключений микропроцессора x86
8. Исследование математического сопроцессора x86
9. Исследование технологии MMX
10. Исследование технологии SSE
11. Вычислительно сложные математические функции и методы расчета их значений на ассемблере
12. Исследование методов обработки файлов в машинно-ориентированных языках программирования
13. Общая схема компиляции и сборки программы в машинно-ориентированных языках программирования
14. Исследование приемов структурного программирования в машинно-ориентированных языках программирования
15. Исследование способов передачи параметров подпрограммам в машинно-ориентированных языках программирования
16. Исследование стандартной библиотеки LIBC для написания программ в машинно-ориентированных языках программирования
17. Функции динамического выделения и высвобождения памяти в машинно-ориентированных языках программирования
18. Исследование базовых алгоритмических структур (следование, ветвление, цикл) на языке ассемблера
19. Исследование методов сортировки массивов на ассемблере
20. Исследование методов генерации случайных и псевдослучайных чисел на ассемблере
21. Исследование методов перемножения матриц на ассемблере
22. Исследование методов расчета быстрого преобразования Фурье на ассемблере
23. Исследование методов расчета обратной матрицы на ассемблере
24. Исследование методов решения систем линейных алгебраических уравнений на ассемблере
25. Исследование алгоритмов деления сверхдлинных целых чисел на ассемблере

26. Исследование алгоритмов перемножения сверхдлинных целых чисел на ассемблере
27. Исследование методов поиска корней полинома на ассемблере
28. Исследование методов свертки арифметических векторов на ассемблере
29. Исследование методов расчета вейвлет-преобразования на ассемблере
30. Исследование алгоритмов дискретного косинусного преобразования на ассемблере
31. Исследование алгоритмов симметричного шифрования данных на ассемблере
32. Исследование алгоритмов асимметричного шифрования данных на ассемблере
33. Исследование алгоритмов электронной цифровой подписи документов на ассемблере
34. Исследование методов ускорения криптоанализа данных с использованием языка ассемблера
35. Исследование методов поиска собственных чисел матрицы на языке ассемблера

Курсовая работа должна содержать:

1. Титульный лист
2. Содержание (с номерами страниц)
3. Введение
 - актуальность темы работы;
 - цель и задачи работы;
 - использованные методы;
 - практическая значимость курсовой работы.
4. Теоретическая часть курсовой работы
 - общая характеристика темы работы;
 - обзор или анализ существующих решений задачи;
 - краткое описание методов разработки предлагаемого решения задачи.
5. Практическая часть курсовой работы
 - математическая модель решаемой задачи и ее описание;
 - информационная модель решаемой задачи (алгоритмы, диаграммы UML и т.п. с описанием);
 - компьютерная программа, моделирующая исследуемые алгоритмы и методы описанием;
 - пример сеанса работы с разработанной компьютерной программой.
6. Заключение
 - перечислить основные результаты работы;
 - сделать вывод по проделанной работе;
 - написать возможные перспективы применения результата работы.
7. Список использованных источников

Список рекомендуемой литературы

Основная литература:

6. Гук М., Юров В. Процессоры Pentium 4, Athlon и Duron. - СПб.: Питер, 2006.
7. Королев Л.Н. Архитектура ЭВМ: Учеб. Пособие для вузов. - М.: «Научный мир», 2005.
8. Пирогов В.Ю. Ассемблер для Windows. - СПб.: БХВ-Петербург, 2008.
9. Червяков Н.И. Организация ЭВМ и систем. – М. 2006г.
10. Юров В.И. Assembler. Учебник для Вузов (2-е издание) - М.: Питер, 2006.

Дополнительная литература

5. Гук М. Процессоры Intel от 8086 до Pentium II. - СПб.: Питер, 1998.
6. Петзолд Ч. Программирование для Windows 95; в двух томах. пер. с англ. - СПб.: BHV - Санкт-Петербург, 1997.
7. Пирогов В.Ю. Assembler. Учебный курс. - М.: Нолидж, 2001.
8. Шилдт Герберт. Программирование на С и С++ для Windows 95. - Киев, 1996.

Интернет-ресурсы

6. www.osp.ru (издат. Открытые системы)
7. www.compres.ru (журнал компьютер пресс)
8. www.ibxt.ru (новости вычислительной техники)
9. <http://ru.wikipedia.org> (интернет-энциклопедия Википедия)
10. <http://www.intel.com>, <http://www.intel.ru> (на русском языке) Фирма Intel:

