Министерство науки и высшего образования Российской Федерации

Муромский институт (филиал)

федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(МИ ВлГУ)

Кафедра *ИС*

«УТВЕРЖДАЮ»
Заместитель директора по УР
_____Д.Е. Андрианов
19.05.2026

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Архитектура микропроцессоров и язык Ассемблера

Направление подготовки *09.03.02 Информационные системы и технологии*

Профиль подготовки *Информационные системы и технологии*

Семестр	Трудоемкость, час./зач. ед.	Лекции, час.	Практические занятия, час.	Лабораторные работы, час.	Консультация, час.	Контроль, час.	Всего (контактная работа), час.	СРС, час.	Форма промежуточного контроля (экз., зач., зач. с оц.)
5	**108 / 3**	**28**		**16**	**2,8**	**0,25**	**47,05**	**60,95**	**Зач. с оц.**
Итого	**108 / 3**	**28**		**16**	**2,8**	**0,25**	**47,05**	**60,95**	

Муром, 2026 г.

1. Цель освоения дисциплины

Целью изучения дисциплины является приобретение знаний и навыков в области проектирования и разработки вычислительных систем с учетом архитектуры микропроцессора, а также знакомство с возможностями оптимизации программного кода за счет использования низкоуровневого языка программирования.

Задачами дисциплины являются: изучение архитектуры микропроцессоров Intel; изучение основ языка ассемблера для микропроцессоров Intel; изучение методов программного взаимодействия с аппаратной частью ЭВМ.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина базируется на знаниях, полученных студентами в процессе изучения дисциплин: информатика; технологии программирования. Дальнейшее освоение материала осуществляется в рамках дисциплин: интерфейсы информационных систем.

3. Планируемые результаты обучения по дисциплине

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции	Результаты обучения по дисциплине	
ПК-5 Способность выполнять работы по созданию (модификации) и сопровождению информационных систем	ПК-5.1 Применяет современные средства создания, модификации и сопровождения информационных систем	Знать современные технологии создания, модификации и сопровождения информационных систем (ПК-5.1) Уметь применять современные технологии создания, модификации и сопровождения информационных систем (ПК-5.1) Владеть современными технологиями создания, модификации и сопровождения информационных систем (ПК-5.1)	Вопросы к устному опросу, задания к лабораторным работам
ПК-1 Способность проводить исследования на всех этапах жизненного цикла программных средств	ПК-1.1 Применяет основные подходы исследования на всех этапах жизненного цикла программых средств	Знать методологию и основные подходы исследования на всех этапах жизненного цикла программных средств (ПК-1.1) Уметь применять методологию и основные подходы исследования на всех этапах жизненного цикла программных средств (ПК-1.1) Владеть методологией и основными подходами исследования на всех этапах жизненного цикла программных средств (ПК-1.1)	Вопросы к устному опросу, задания к лабораторным работам

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 3 зачетные единицы, 108 часов.

4.1. Форма обучения: очная

Уровень базового образования: среднее общее.
Срок обучения 4г.

4.1.1. Структура дисциплины

№ п\п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Архитектура IA-32	5	14		8						Устный опрос, лабораторная работа
2	MIPS	5	14		8					60,95	Устный опрос, лабораторная работа
Всего за семестр		108	28		16			2,8	0,25	60,95	Зач. с оц.
Итого		108	28		16			2,8	0,25	60,95	

4.1.2. Содержание дисциплины

4.1.2.1. Перечень лекций

Семестр 5

Раздел 1. Архитектура IA-32

Лекция 1.

Основы системы команд i386, операнды, целочисленные команды (2 часа).

Лекция 2.

Условный и безусловный переход, циклы, побитовые и строковые операции (2 часа).

Лекция 3.

Стек, подпрограмма, рекурсия (2 часа).

Лекция 4.

Основные особенности ассемблера NASM, макросредства и макропроцессор (2 часа).

Лекция 5.

Мультизадачность и ее аппаратная поддержка, взаимодействие с операционной системой (2 часа).

Лекция 6.

Раздельная трансляция (2 часа).

Лекция 7.

Устройство математического сопроцессора, команды математического сопроцессора (2 часа).

Раздел 2. MIPS

Лекция 8.

Введение в MIPS. Инструкции, операнды и регистры (2 часа).

Лекция 9.

Память, константы и непосредственные операнды (2 часа).

Лекция 10.

Машинный язык (2 часа).

Лекция 11.

Арифметические и логические инструкции, переходы (2 часа).

Лекция 12.

Условия, выбор и циклы. Массивы (2 часа).

Лекция 13.

Вызовы функций и стек, Рекурсивные вызовы (2 часа).

Лекция 14.

Режимы адресации (2 часа).

4.1.2.2. Перечень практических занятий

Не планируется.

4.1.2.3. Перечень лабораторных работ

Семестр 5

Раздел 1. Архитектура IA-32

Лабораторная 1.

Основы NASM. Отладка кода, операторы условия, циклы, арифметические операции (4 часа).

Лабораторная 2.

Подпрограммы и взаимодействие с ОС (4 часа).

Раздел 2. MIPS

Лабораторная 3.

Изучения эмулятора MARS на примере простейших программ (4 часа).

Лабораторная 4.

Подпрограммы MIPS (4 часа).

4.1.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Виды архитектур микропроцессоров.
2. Однотактный процессор.
3. Многотактный процессор.
4. Конвейерный процессор.
5. Улучшенные процессоры.
6. Архитектура процессора Pentium.
7. Форматы представления чисел.
8. Арифметический сопроцессор.
9. Расширение MMX.
10. Инструменты для отладки и дизассемблирования.
11. Отладчики программного кода.
12. Дизассемблеры.
13. Методы и приемы отладки кода.
14. Типы и форматы сообщений об ошибках, предупреждений.
15. Трансляция и компоновка программного кода. Сборка программ из нескольких объектных модулей.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.1.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

Не планируется.

4.1.2.6. Примерный перечень тем курсовых работ (проектов)

Не планируется.

4.2 Форма обучения: заочная

Уровень базового образования: среднее профессиональное.
Срок обучения 3г 6м.

Семестр	Трудоемкость, час./ зач. ед.	Лекции, час.	Практические занятия, час.	Лабораторные работы, час.	Консультация, час.	Контроль,час.	Всего (контактная работа), час.	СРС, час.	Форма промежуточного контроля (экз., зач., зач. с оц.)
6	**108 / 3**	**4**		**8**	**2**	**0,5**	**14,5**	**89,75**	**Зач. с оц.(3,75)**
Итого	**108 / 3**	**4**		**8**	**2**	**0,5**	**14,5**	**89,75**	**3,75**

4.2.1. Структура дисциплины

№ п\п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Архитектура IA-32	6	4		8					0	Устный опрос, лабораторная работа
2	MIPS	6								89,75	Устный опрос
Всего за семестр		108	4		8	+		2	0,5	89,75	Зач. с оц.(3,75)
Итого		108	4		8			2	0,5	89,75	3,75

4.2.2. Содержание дисциплины

4.2.2.1. Перечень лекций

Семестр 6

Раздел 1. Архитектура IA-32

Лекция 1.

Основы системы команд i386, операнды, целочисленные команды (2 часа).

Лекция 2.

Условный и безусловный переход, циклы, побитовые и строковые операции (2 часа).

4.2.2.2. Перечень практических занятий

Не планируется.

4.2.2.3. Перечень лабораторных работ

Семестр 6

Раздел 1. Архитектура IA-32

Лабораторная 1.

Основы NASM. Отладка кода, операторы условия, циклы, арифметические операции (4 часа).

Лабораторная 2.

Подпрограммы и взаимодействие с ОС (4 часа).

4.2.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Виды архитектур микропроцессоров.
2. Однотактный процессор.
3. Многотактный процессор.
4. Конвейерный процессор.
5. Улучшенные процессоры.
6. Архитектура процессора Pentium.
7. Форматы представления чисел.
8. Арифметический сопроцессор.
9. Расширение MMX.
10. Инструменты для отладки и дизассемблирования.
11. Отладчики программного кода.
12. Дизассемблеры.
13. Методы и приемы отладки кода.
14. Типы и форматы сообщений об ошибках, предупреждений.
15. Трансляция и компоновка программного кода. Сборка программ из нескольких объектных модулей.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.2.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

1. Принципы фон Неймана и Гарвардская архитектура: Сравнение, преимущества, области применения.
2. Системы счисления и представление данных: Перевод из 2 в 16, прямой, обратный и дополнительный коды.
3. АЛУ (Арифметико-логическое устройство): Функции, принцип работы сумматора.
4. Регистровая модель процессора: Назначение регистров (R0..R15, PC, SP, Flag Register).
5. Организация памяти: Стек, куча, сегменты кода и данных (адресное пространство).
6. Цикл выполнения команды: Fetch (выборка), Decode (декодирование), Execute (исполнение), Write Back.
7. Конвейеризация (Pipeline): Понятие такта, стадии конвейера, проблема структурных конфликтов.
8. CISC против RISC: Ключевые различия, примеры архитектур (x86 vs ARM).
9. Система прерываний: Векторные прерывания, маскируемые и немаскируемые прерывания (NMI).
10. Кэш-память: Принцип работы, отображение (прямое, ассоциативное), понятие промаха (Miss) и попадания (Hit).
11. Режимы адресации данных: Непосредственная, прямая, регистровая, косвенная.
12. Сложные режимы адресации x86: Базирование, индексирование, масштабирование (LEA).
13. Команды пересылки данных (MOV, PUSH, POP): Работа со стеком и сегментами.

14. Арифметические команды: ADD, SUB, MUL, DIV, ADC (сложение с переносом), обработка флагов.

15. Логические команды и битовые операции: AND, OR, XOR, NOT, SHL, SHR, ROR.

16. Команды сравнения и условных переходов: CMP, TEST, JZ, JNZ, JG, JL.

17. Безусловные переходы и вызовы: JMP, CALL, RET. Механизм возврата из подпрограммы.

18. Обработка строк (String Instructions): MOVS, STOS, LODS, CMPS, префикс REP.

19. Директивы ассемблера: DB, DW, DD, RESB, EQU. Разница между командой и директивой.

20. Структура программы на TASM/NASM: Сегменты .data, .bss, .text, точка входа (start/_start).

21. Макросы в ассемблере: Определение (%macro), расширение макросов, отличие от процедуры.

22. Стек фреймов (Stack Frames): пролог функции (PUSH EBP, MOV EBP, ESP) и эпилог (LEAVE, RET).

23. Соглашения вызовов (Calling Conventions): cdecl, stdcall, fastcall (передача параметров через стек/регистры).

24. Листинг-файлы и объектные файлы: Процесс компиляции (ASM -> OBJ -> EXE).

25. Работа с дизассемблером: Восстановление исходного кода из машинного кода.

26. Inline-ассемблер: Вставки кода в C/C++ (__asm {}), ограничения.

27. Умножение без использования MUL: Реализация через сдвиги и сложения.

28. Деление и взятие остатка: Алгоритм деления столбиком на ассемблере.

29. Преобразование чисел в ASCII (itoa): Алгоритм перевода Hex в десятичную строку.

30. Сортировка пузырьком (Bubble Sort): Реализация вложенных циклов с условиями переходов.

31. Поиск минимума/максимума в массиве: Обработка знаковых и беззнаковых сравнений.

32. Реализация конечного автомата: Обработка последовательности символов (токенов).

33. Копирование блоков памяти (memcpy): Оптимизация с использованием MOVSD/REP MOVSD.

34. Заполнение памяти паттерном (memset): Использование STOS и префикса REP.

35. Работа с двусвязным списком: Вставка/удаление элементов вручную через указатели.

36. Реализация рекурсии (Факториал/Фибоначчи): Управление стеком вызовов сохранением EBP.

37. Вывод шестнадцатеричного дампа: Поразрядная обработка HEX в текстовую строку.

38. Эмуляция оператора SWITCH (Jump Table): Использование косвенного JMP по адресу из таблицы.

39. Защищенный режим (Protected Mode) против Реального (Real Mode): Сегментация, GDT (Global Descriptor Table), ограничения.

40. Страничная организация памяти (Paging): Линейные и физические адреса, таблицы страниц (PDE, PTE).

41. SIMD-инструкции (MMX/SSE/AVX): Регистры XMM/YMM, команды ADDPS, MULPD (векторная обработка).

42. Потоки и синхронизация: Использование LOCK PREFIX, инструкции XCHG, CAS (Compare-And-Swap).

43. Системные вызовы (Syscalls): INT 0x80 в Linux vs SYSENTER/SYSCALL.

44. Управление кэшем: Инструкции prefetch, clflush, non-temporal hints (MOVNT).

45. Программируемый контроллер прерываний (PIC - 8259): Настройка IRQ, маски прерываний.

46. Прямой доступ к памяти (DMA): Режимы работы, инициация передачи.

47. Работа с видео-памятью в текстовом режиме: Прямая запись в B8000h (атрибуты символов).

48. Микропрограммирование (на примере процессора 8080 или учебного). Микро-операции, управляющая память.

49. Таймер и системное время (PIT - 8253/8254): Подсчет тактов, генерация задержек на ассемблере.

50. Эмуляция ввода с клавиатуры: Обработка скан-кодов через порты 0x60/0x64.

4.2.2.6. Примерный перечень тем курсовых работ (проектов)

Не планируется.

5. Образовательные технологии

В процессе изучения дисциплины применяется контактная технология преподавания (за исключением самостоятельно изучаемых студентами вопросов). При проведении лабораторных работ применяется имитационный или симуляционный подход, когда преподавателем разбирается на конкретном примере проблемная ситуация, все шаги решения задачи студентам демонстрируются при помощи мультимедийной техники. Затем студенты самостоятельно решают аналогичные задания.

Во время выполнения лабораторных работ каждому студенту выдается конкретное задание, тем самым формируется способность обучающихся к самостоятельной работе при решений определенных задач, связанных с изучением конкретных видов ПО.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Фонды оценочных материалов (средств) приведены в приложении.

7. Учебно-методическое и информационное обеспечение дисциплины.

7.1. Основная учебно-методическая литература по дисциплине

1. Гагарина Л.Г. Архитектура вычислительных систем и Ассемблер с приложением методических указаний к лабораторным работам [Электронный ресурс]: учебное пособие/ Гагарина Л.Г., Кононова А.И.— Электрон. текстовые данные.— Москва: СОЛОН-Пресс, 2019.— 368 c.— Режим доступа: http://www.iprbookshop.ru/94943.html.— ЭБС «IPRbooks» - http://www.iprbookshop.ru/94943.html

2. Аблязов Р.З. Программирование на ассемблере на платформе x86-64 [Электронный ресурс]/ Аблязов Р.З.— Электрон. текстовые данные.— Саратов: Профобразование, 2019.— 301 c.— Режим доступа: http://www.iprbookshop.ru/88005.html.— ЭБС «IPRbooks» - http://www.iprbookshop.ru/88005.html

3. Гуров В.В. Архитектура микропроцессоров [Электронный ресурс]: учебное пособие/ Гуров В.В.— Электрон. текстовые данные.— Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020.— 326 c.— Режим доступа: http://www.iprbookshop.ru/89419.html.— ЭБС «IPRbooks» - http://www.iprbookshop.ru/89419.html

7.2. Дополнительная учебно-методическая литература по дисциплине

1. Кирнос В.Н. Введение в вычислительную технику. Основы организации ЭВМ и программирование на Ассемблере: учебное пособие/ Кирнос В.Н.— Электрон. текстовые данные.— Томск: Томский государственный университет систем управления и радиоэлектроники, Эль Контент, 2011.— 172 c. - http://www.iprbookshop.ru/13921.html

2. Программные продукты и системы : международный научно-практический журнал. - Тверь: Научно-исследовательский институт "Центрпрограммсистем" - http://swsys.ru/index.php?page=10&lang=

3. Гагарина Л.Г. Архитектура вычислительных систем и Ассемблер с приложением методических указаний к лабораторным работам [Электронный ресурс]: учебное пособие/ Гагарина Л.Г., Кононова А.И.— Электрон. текстовые данные.— Москва: СОЛОН-Пресс, 2019.— 368 c.— Режим доступа: http://www.iprbookshop.ru/94943.html.— ЭБС «IPRbooks» - http://www.iprbookshop.ru/94943.html

7.3. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

В образовательном процессе используются информационные технологии, реализованные на основе информационно-образовательного портала института (www.mivlgu.ru/iop), и инфокоммуникационной сети института:

- предоставление учебно-методических материалов в электронном виде;
- взаимодействие участников образовательного процесса через локальную сеть института и Интернет;
- предоставление сведений о результатах учебной деятельности в электронном личном кабинете обучающегося.

Информационные справочные системы:

электронная библиотечная системы "IPRBooks" (http://www.iprbookshop.ru/);

центр информационных технологий (http://citforum.ru).

Программное обеспечение:

РЕД ОС (Соглашение №140/05-21У от 18.05.2021 года о сотрудничестве в области науки, развития инновационной деятельности)

nasm (LGPL)

GCC (GNU GPL)

7.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

iprbookshop.ru

swsys.ru

citforum.ru).

mivlgu.ru/iop

8. Материально-техническое обеспечение дисциплины

Лаборатория ГИС и САПР

Сервер; 12 персональных компьютеров; проектор Sanyo PDG-DSU20; экран настенный Drapper Apex Star

9. Методические указания по освоению дисциплины

Для успешного освоения теоретического материала обучающийся: знакомится со списком рекомендуемой основной и дополнительной литературы; уточняет у преподавателя, каким дополнительным пособиям следует отдать предпочтение; ведет конспект лекций и прорабатывает лекционный материал, пользуясь как конспектом, так и учебными пособиями.

До выполнения лабораторных работ обучающийся изучает соответствующий раздел теории. Перед занятием студент знакомится с описанием заданий для выполнения работы, внимательно изучает содержание и порядок проведения лабораторной работы. Лабораторная работа проводятся в компьютерном классе. Обучающиеся выполняют индивидуальную задачу компьютерного моделирования в соответствии с заданием на лабораторную работу. Полученные результаты исследований сводятся в отчет и защищаются по традиционной методике в классе на следующем лабораторном занятии. Необходимый теоретический материал, индивидуальное задание, шаги выполнения лабораторной работы и требование к

отчету приведены в методических указаниях, размещенных на информационно-образовательном портале института.

Самостоятельная работа оказывает важное влияние на формирование личности будущего специалиста, она планируется обучающимся самостоятельно. Каждый обучающийся самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием дисциплины. Он выполняет внеаудиторную работу и изучение разделов, выносимых на самостоятельную работу, по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

Форма заключительного контроля при промежуточной аттестации – зачет с оценкой. Для проведения промежуточной аттестации по дисциплине разработаны фонд оценочных средств и балльно-рейтинговая система оценки учебной деятельности студентов. Оценка по дисциплине выставляется в информационной системе и носит интегрированный характер, учитывающий результаты оценивания участия студентов в аудиторных занятиях, качества и своевременности выполнения заданий в ходе изучения дисциплины и промежуточной аттестации.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению *09.03.02 Информационные системы и технологии* и профилю подготовки *Информационные системы и технологии*
Рабочую программу составил *к.т.н., доцент Симаков Р.А.*_____

Программа рассмотрена и одобрена на заседании кафедры *ИС*

протокол № 21 от 05.05.2026 года.
Заведующий кафедрой *ИС* _____*Андрианов Д.Е.*
(Подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии факультета

протокол № 9 от 12.05.2026 года.
Председатель комиссии ФИТР _____ *Кутарова Е.И.*
(Подпись) (Ф.И.О.)

Приложение

Фонд оценочных материалов (средств) по дисциплине
Архитектура микропроцессоров и язык Ассемблера

1. Оценочные материалы для проведения текущего контроля успеваемости по дисциплине

1. Темы для устного опроса:
- Характеристики микропроцессоров.
- Структура микропроцессора Intel 8086.
- Состав и назначение регистров микропроцессора Intel 8086.
- Основные типы программных сегментов.
- Алгоритм вычисления абсолютного адреса.
- Виды прерываний.
- Последовательность обработки прерываний.
- Виды режимов адресации памяти.
- Особенности архитектуры IA-32.
- Реальный режим работы микропроцессора.
- Защищенный режим работы микропроцессора.
- Принцип конвейерной обработки.
- Ассемблер. Идентификаторы, переменные, метки, имена.
- Ассемблер. Типы данных.
- Ассемблер. Операторы.
- Ассемблер. Директивы определения данных.
- Ассемблер. Структуры.
- Ассемблер. Записи.
- Ассемблер. Структура программы.
- Ассемблер. Команды пересылки.
- Ассемблер. Арифметические команды.
- Ассемблер. Команды побитовой обработки.
- Ассемблер. Команды сравнения и передачи управления.
- Ассемблер. Подпрограммы.
- Ассемблер. Команды работы со строками.
- Ассемблер. Команды управления процессором.
- Ассемблер. Структуры данных.
- Ассемблер. Условное ассемблирование.
- Ассемблер. Макросредства.
- Ассемблер. Ассемблерные вставки.

2. Задания для выполнения лабораторных работ:

- Записать в файле код программы из справочной информации. Эта программа печатает заданное слово, меняем местами в нем буквы и выводит получившиеся слово. Используются транслитерация букв. Скомпилировать и слинковать код в программу. Запустить программу и посмотреть результат. Запустить программу под отладчиком, выполнить по шагам просматривая значения регистров и памяти, используемой в текущей строчке кода. Согласно своему варианту (по последней цифре в номере журнала) составить свою программу преобразований слов, отладить ее и показать.

- Написать программу, которая выводит на экран предопределенную в коде строку, содержащую разные типы символов (заглавные и прописные буквы, цифры, знаки). Составить программу печати целого числа. Согласно своему варианту добавить преобразование этой строки в другую и вывести на экран результат.

- Составить простейшую подпрограмму для вывода на экран заданной строки. Реализовать подпрограмму согласно своему варианту и ввод-вывод параметров для нее с помощью библиотек языка C. Использовать соглашение CDECL для организации подпрограммы. Реализовать программу вывода файла на экран (аналог программы cat).

- Составить программу, которая запрашивает у пользователя входные параметры для расчета выражения. Согласно своего варианта, составить программу вычисления выражения. Вывести результат вычисления на экран. Аналогично составить программу на C++ и сравнить полученные результаты.

- Установить и запустить эмулятор MARS. Записать в файле код программы из справочной информации. Эта программа запрашивает два числа, сохраняет их в память, складывает и выводит. Выполнить программу полностью. Выполнить программу по шагам, наблюдая за изменением регистров и памяти. Согласно своему варианту (по последней цифре в номере журнала, для 0 вариант 10) составить свою программу вычисления выражения используя целочисленные переменные типа .word, отладить ее и показать. Переделать программу под использование переменных типа .float. Учесть что для ввода, вывода и вычислений используются другие регистры и номера системных вызовов. В отчете сравнить точность вычислений для типов .word и .float.

- Написать программу, которая запрашивает у пользователя строку не более 63 символов. Согласно своему варианту добавить преобразование этой строки в другую и вывести на экран результат.

- Реализовать подпрограмму согласно своему варианту (остаток от деления номера в журнале на 4). Программа должна соблюдать соглашения о вызовах MIPS.

Общее распределение баллов текущего контроля по видам учебных работ для студентов

Рейтинг-контроль 1	Устный опрос (2 вопроса)	30
Рейтинг-контроль 2	Устный опрос (2 вопроса)	30
Рейтинг-контроль 3	Устный опрос (2 вопроса)	20
Посещение занятий студентом	Отметка в журнале посещений	5
Дополнительные баллы (бонусы)		10
Выполнение семестрового плана самостоятельной работы	Защита лабораторных работ	5

2. Промежуточная аттестация по дисциплине

Перечень вопросов к экзамену / зачету / зачету с оценкой. Перечень практических задач / заданий к экзамену / зачету / зачету с оценкой (при наличии)

Перечень вопросов для проведения тестирования:
ПК-1, ПК-5:

1. Расположите в правильной последовательности этапы создания программы на языке ассемблера.

написание программного кода
трансляция
компоновка
выполнение

2. Установите соответствие между сегментами программы и сегментными регистрами по умолчанию.

сегмент кода
сегмент данных
сегмент стека
дополнительный сегмент

3. В каком режиме адресации в качестве операнда используется адрес, вычисляемый как сумма значения смещения и содержимого регистра SI или DI?

прямая
косвенная
базовая
индексная

4. В каком режиме адресации в качестве операнда используется адрес, вычисляемый как сумма значения смещения и содержимого регистра BP или BX?

базово-индексная
косвенная
базовая
индексная

5. Какие директивы служат для задания размеров, содержимого и местоположения полей данных?

директивы определения данных
директивы сегментации
директивы эквивалентности

6. Какие идентификаторы относятся к ключевым словам?

директивы ассемблера
инструкции процессора
имена регистров
переменные

7. В каком режиме адресации в качестве операнда используется адрес, вычисляемый как сумма значения смещения и содержимого регистра BP или BX и регистра SI или DI?

базово-индексная
косвенная
базовая
индексная

8. В каком режиме адресации в качестве операнда используется адрес памяти?

регистровая
прямая
базовая
индексная

9. В каком режиме адресации в качестве операнда используется адрес, формируемый исходя из сегментного адреса в одном из сегментных регистров и смещения в регистре для адресации?

регистровая
косвенная
базовая
индексная

10. В каком режиме адресации в качестве операнда используется непосредственное значение?

регистровая
прямая
базовая
непосредственная

11. В каком режиме адресации в качестве операнда используется регистр процессора?
- регистровая
- прямая
- базовая
- индексная

12. Какой вид памяти ЭВМ является энергозависимым?
- ОЗУ
- ПЗУ
- дисковая
- flash

13. Какие предложения языка ассемблера действуют лишь в период компиляции программы?
- директивы ассемблера
- инструкции процессора
- команды управления процессором
- команды пересылки данных

14. Какие предложения языка ассемблера представляют собой мнемоническую форму записи машинных команд?
- директивы ассемблера
- инструкции процессора
- ограничители
- операторы выражений

15. Какие регистры не относятся к регистрам адресации?
- DX
- BP
- SP
- IP

16. Какие регистры не относятся к регистрам общего назначения?
- DI
- DX
- CX
- BP

17. Какие регистры не относятся к сегментным регистрам?
- SI
- DS
- SS
- SP

18. Какие регистры относятся к регистрам адресации?
- BX
- DX
- DI
- BP

19. Какие регистры относятся к регистрам общего назначения?
- AX
- CX

DX
DI

20. Какие регистры относятся к сегментным регистрам?
CS
DS
SS
SP

21. Какие устройства входят в блок интерфейса шин микропроцессора Intel 8086?
блок очереди команд
устройство управления шинами
сегментные регистры
устройство управления

22. Какие устройства входят в блок исполнения микропроцессора Intel 8086?
арифметическо-логическое устройство
устройство управления
устройство управления шинами
блок очереди команд

23. Как называется адрес ячейки памяти внутри сегмента памяти?
смещение
абсолютный адрес
исполнительный адрес
базовый адрес

24. Как называется блок непосредственно адресуемой памяти размером 64К в архитектуре Intel 8086?
ячейка
сегмент
параграф
кластер

25. Как называется объект программы, представляющий собой набор полей байтов, объединенных одним именем?
запись
структура
массив
строка

26. Как называется объект программы, представляющий собой набор полей бит, объединенных одним именем?
запись
структура
массив
строка

27. Какой латинской буквой принято сопровождать восьмеричные числа при написании программ?
B
H
D
O

28. Какой латинской буквой принято сопровождать двоичные числа при написании программ?

B
H
D
O

29. Какой латинской буквой принято сопровождать десятичные числа при написании программ?

B
H
D
O

30. Какой латинской буквой принято сопровождать шестнадцатеричные числа при написании программ?

B
H
D
O

31. Какой максимальный размер может иметь сегмент памяти в архитектуре Intel 8086?

16 Байт
64 Байт
16 Килобайт
64 Килобайт

32. Какой объект программы идентифицирует область памяти, содержащей машинный код?

переменная
метка
имя
ограничитель

33. Какой объект программы идентифицирует хранящиеся в памяти данные?

переменная
метка
имя
ограничитель

34. Какой объект программы соответствует константе?

переменная
метка
имя
ограничитель

35. Какой регистр определяет смещение текущей вершины стека?

SP
SS
BP
CS

36. Какой регистр хранит адрес ячейки памяти, содержащей начало следующей команды?

IP
SI
DI
AX

37. Какой сегментный регистр используется по умолчанию для указания на адреса из сегмента данных?

CS
DS
SS
ES

38. Какой сегментный регистр используется по умолчанию для указания на адреса из сегмента кода?

CS
DS
SS
ES

39. Какой сегментный регистр используется по умолчанию для указания на адреса из сегмента стека?

CS
DS
SS
ES

40. Какой сегментный регистр не инициализируется адресом конкретного программного сегмента в процессе загрузки программы в память?

CS
DS
SS
ES

41. Какой тип меток должен располагаться в том же сегменте, что и оператор перехода на нее?

NEAR
SHORT
FAR

42. Какой тип меток должен располагаться не далее чем на 128 байт от места вызова?

NEAR
SHORT
FAR

43. Какой тип меток может располагаться в другом сегменте памяти по отношению к расположению оператора перехода на нее?

NEAR
SHORT
FAR

44. Какой флаг регистра флагов определяет знак результата?

OF

DF
SF
CF

45. Какой флаг регистра флагов определяет направление обработки строк?
DF
DF
SF
TF

46. Какой флаг регистра флагов определяет четность результата?
PF
DF
SF
TF

47. Какой флаг регистра флагов разрешает прерывания?
OF
IF
SF
TF

48. Какой флаг регистра флагов сигнализирует об арифметическом переполнении?
OF
DF
SF
TF

49. Какой флаг регистра флагов указывает на наличие переноса или заема при выполнении арифметических операций?
OF
DF
CF
TF

50. Какой флаг регистра флагов указывает на нулевой результат?
OF
DF
SF
ZF

51. По какому принципу организован стек памяти в архитектуре Intel 8086?
первый пришел - первый ушел
последний пришел - первый ушел

52. Сколько программных сегментов содержат программы COM формата?
1
2
3
4

ПК-1, ПК-5:
1. Запишите команду арифметического вычитания

2. Запишите команду пересылки операнда

3. Запишите команду арифметического вычитания с заемом

4. Запишите команду арифметического сдвига влево

5. Запишите команду арифметического сдвига вправо

6. Запишите команду арифметического сложения

7. Запишите команду арифметического сложения с переносом

8. Запишите команду безусловного перехода

9. Запишите команду ввода операнда из порта

10. Запишите команду вывода операнда в порт

11. Запишите команду вызова подпрограммы

12. Запишите команду выхода из подпрограммы

13. Запишите команду декремента

14. Запишите команду деления чисел без знака

15. Запишите команду деления чисел со знаком

16. Запишите команду загрузки полного указателя в регистр DS

17. Запишите команду загрузки полного указателя в регистр ES

18. Запишите команду загрузки строки байт

19. Запишите команду замены содержимого регистра AL байтом из таблицы перекодировки

20. Запишите команду извлечения операнда из стека

21. Запишите команду извлечения регистра флагов из стека

22. Запишите команду извлечения регистров общего назначения из стека

23. Запишите команду изменения значения флага переноса

24. Запишите команду инкремента

25. Запишите команду логического "И"

26. Запишите команду логического "ИЛИ"

27. Запишите команду логического исключающего "ИЛИ"

28. Запишите команду логического "НЕ"

29. Запишите команду логического сдвига влево

30. Запишите команду логического сдвига вправо

31. Запишите команду обмена содержимого двух операндов

32. Запишите команду пересылки строки байт

33. Запишите команду размещения операнда в стеке

34. Запишите команду размещения регистра флагов в стеке

35. Запишите команду размещения регистров общего назначения в стеке

36. Запишите команду сброса флага направления

37. Запишите команду сброса флага переноса

38. Запишите команду сканирования строки байт

39. Запишите команду смены знака

40. Запишите команду сохранения строки байт

41. Запишите команду сравнения строк байт

42. Запишите команду умножения чисел без знака

43. Запишите команду умножения чисел со знаков

44. Запишите команду управления циклом (повторения команд тела цикла)

45. Запишите команду установки флага направления

46. Запишите команду установки флага переноса

47. Запишите команду циклического сдвига влево

48. Запишите команду циклического сдвига влево с переносом

49. Запишите команду циклического сдвига вправо

50. Запишите команду циклического сдвига вправо с переносом

51. Чему равно значение в регистре AX после выполнения участка кода?

```
mov ah, 00000011b
mov al, 00000000b
xchg ah, al
```

52. Чему равен флаг CF после выполнения участка кода?

```
mov al, 0FFh
add al, 1
```

53. Чему равен флаг ZF после выполнения участка кода?

```
mov al, 11b
sub al, 2
```

54. Чему равен флаг ZF после выполнения участка кода?

```
mov al, 11b
sub al, 3
```

55. Чему равно значение в регистре AL после выполнения участка кода?

```
mas db 10 dup (?)
...
mov al, mas5
inc al
```

56. Чему равно значение в регистре AL после выполнения участка кода?

```
mas db 1, 2, 3, 4, 5
...
mov al, mas2
```

57. Чему равно значение в регистре AL после выполнения участка кода?

```
mas db 1, 2, 3, 4, 5
...
mov bx, offset mas
mov si, 2
mov [bxsi], 111b
mov al, mas2
```

58. Чему равно значение в регистре AL после выполнения участка кода?

```
mas db 1, 2, 3, 4, 5
...
mov si, 2
mov al, mas[si1]
```

59. Чему равно значение в регистре AL после выполнения участка кода?

```
mas dw 1, 2, 3, 4, 5
...
mov bx, offset mas
mov si, 4
mov al, [bxsi]
```

60. Чему равно значение в регистре AL после выполнения участка кода?

```
mov al, 10h
add al, 10b
```

61. Чему равно значение в регистре AL после выполнения участка кода?

```
mov al, 11110000b
not al
```

62. Чему равно значение в регистре AL после выполнения участка кода?

```
mov al, 111b
dec al
dec al
```

63. Чему равно значение в регистре AL после выполнения участка кода?

```
mov al, 111b
shr al, 2
```

64. Чему равно значение в регистре AL после выполнения участка кода?

```
mov al, 1
cmp al, 2
jb m1
        add al, 1
m1:
xor al, al
```

65. Чему равно значение в регистре AL после выполнения участка кода?

```
mov al, 1h
stc
rcl al, 1
```

66. Чему равно значение в регистре AL после выполнения участка кода?

```
mov al, 1
shl al, 2
```

67. Чему равно значение в регистре AL после выполнения участка кода?

```
mov al, 2
cmp al, 2
jb m1
        add al, 1
        jmp m2
m1:
xor al, al
m2:
```

68. Чему равно значение в регистре AL после выполнения участка кода?

```
mov al, 3
and al, 00000010b
```

69. Чему равно значение в регистре AL после выполнения участка кода?

```
mov al, 3
or al, 00000100b
```

70. Чему равно значение в регистре AX после выполнения участка кода?

```
mas dw 1, 2, 3, 4, 5
...
mov bx, offset mas
mov si, 2
mov ax, [bx][si]
```

71. Чему равно значение в регистре AX после выполнения участка кода?

```
mov al, 1b
clc
jc m1
        add al, 1
        jmp m2
m1:
sub al, 1
m2:
```

72. Чему равно значение в регистре AX после выполнения участка кода?

```
mov al, 1h
stc
jc m1
        add al, 1
m1:
sub al, 1
```

73. Чему равно значение в регистре AX после выполнения участка кода?

```
mov al, 2
mov bl, 3
mov cl, 4
mov dl, 5
mul bl
```

74. Чему равно значение в регистре AX после выполнения участка кода?

```
mov ax, 10h
mov bx, 10b
mov cx, 10q
mov dx, 10d
add ax, bx
add ax, cx
sub ax, dx
```

75. Чему равно значение в регистре AX после выполнения участка кода?

```
mov ax, 3h
mov bx, 11b
cmp ax, bx
jnae m1
        mov ax, 5
m1:
```

76. Чему равно значение в регистре AX после выполнения участка кода?

```
mov ax, 4h
sub ax, 11b
jz m1
        inc ax
m1:
```

77. Чему равно значение в регистре AX после выполнения участка кода?

```
xor ah, ah
mov al, 101b
add al, 2
```

78. Чему равно значение в регистре BL после выполнения участка кода?

```
mov al, 2
mov bl, 3
mov cl, 4
mov dl, 5
mul bl
```

79. Чему равно значение в регистре CX после выполнения участка кода?

```
mov cx, 5q
```

```
m1:
loop m1
```

80. Чему равно значение в регистре SI после выполнения участка кода?

```
mov si, 10h
mov cx, 10
m1:
        inc si
loop m1
```

Методические материалы, характеризующие процедуры оценивания

На основе перечня вопросов к тестированию программным комплексом информационно-образовательного портала МИ ВлГУ формируются в автоматическом режиме тестовые задания для студентов: 10 вопросов из блока 1 и 10 вопросов из блока 2. Программный комплекс формирует индивидуальные задания для каждого зарегистрированного в системе студента и устанавливает время прохождения тестирования. Результатом тестирования является балл, рассчитанный на основе количества правильных ответов. С учетом индивидуального семестрового рейтинга студента формируется итоговый балл по курсу.

Максимальная сумма баллов, набираемая студентом по дисциплине равна 100.

Оценка в баллах	Оценка по шкале	Обоснование	***Уровень сформированности компетенций***
Более 80	«Отлично»	Содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	***Высокий уровень***
66-80	«Хорошо»	Содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	***Продвинутый уровень***

50-65	«Удовлетворительно»	Содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки	***Пороговый уровень***
Менее 50	«Неудовлетворительно»	Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки	***Компетенции не сформированы***

3. Задания в тестовой форме по дисциплине

Примеры заданий:

1. Темы для устного опроса:

- Характеристики микропроцессоров.
- Структура микропроцессора Intel 8086.
- Состав и назначение регистров микропроцессора Intel 8086.
- Основные типы программных сегментов.
- Алгоритм вычисления абсолютного адреса.
- Виды прерываний.
- Последовательность обработки прерываний.
- Виды режимов адресации памяти.
- Особенности архитектуры IA-32.
- Реальный режим работы микропроцессора.
- Защищенный режим работы микропроцессора.
- Принцип конвейерной обработки.
- Ассемблер. Идентификаторы, переменные, метки, имена.
- Ассемблер. Типы данных.
- Ассемблер. Операторы.
- Ассемблер. Директивы определения данных.
- Ассемблер. Структуры.
- Ассемблер. Записи.
- Ассемблер. Структура программы.
- Ассемблер. Команды пересылки.
- Ассемблер. Арифметические команды.
- Ассемблер. Команды побитовой обработки.
- Ассемблер. Команды сравнения и передачи управления.
- Ассемблер. Подпрограммы.
- Ассемблер. Команды работы со строками.
- Ассемблер. Команды управления процессором.
- Ассемблер. Структуры данных.
- Ассемблер. Условное ассемблирование.
- Ассемблер. Макросредства.
- Ассемблер. Ассемблерные вставки.

2. Задания для выполнения лабораторных работ:

- Записать в файле код программы из справочной информации. Эта программа печатает заданное слово, меняем местами в нем буквы и выводит получившиеся слово. Используются

транслитерация букв. Скомпилировать и слинковать код в программу. Запустить программу и посмотреть результат. Запустить программу под отладчиком, выполнить по шагам просматривая значения регистров и памяти, используемой в текущей строчке кода. Согласно своему варианту (по последней цифре в номере журнала) составить свою программу преобразований слов, отладить ее и показать.

- Написать программу, которая выводит на экран предопределенную в коде строку, содержащую разные типы символов (заглавные и прописные буквы, цифры, знаки). Составить программу печати целого числа. Согласно своему варианту добавить преобразование этой строки в другую и вывести на экран результат.

- Составить простейшую подпрограмму для вывода на экран заданной строки. Реализовать подпрограмму согласно своему варианту и ввод-вывод параметров для нее с помощью библиотек языка С. Использовать соглашение CDECL для организации подпрограммы. Реализовать программу вывода файла на экран (аналог программы cat).

- Составить программу, которая запрашивает у пользователя входные параметры для расчета выражения. Согласно своего варианта, составить программу вычисления выражения. Вывести результат вычисления на экран. Аналогично составить программу на C++ и сравнить полученные результаты.

- Установить и запустить эмулятор MARS. Записать в файле код программы из справочной информации. Эта программа запрашивает два числа, сохраняет их в память, складывает и выводит. Выполнить программу полностью. Выполнить программу по шагам, наблюдая за изменением регистров и памяти. Согласно своему варианту (по последней цифре в номере журнала, для 0 вариант 10) составить свою программу вычисления выражения используя целочисленные переменные типа .word, отладить ее и показать. Переделать программу под использование переменных типа .float. Учесть что для ввода, вывода и вычислений используются другие регистры и номера системных вызовов. В отчете сравнить точность вычислений для типов .word и .float.

- Написать программу, которая запрашивает у пользователя строку не более 63 символов. Согласно своему варианту добавить преобразование этой строки в другую и вывести на экран результат.

- Реализовать подпрограмму согласно своему варианту (остаток от деления номера в журнале на 4). Программа должна соблюдать соглашения о вызовах MIPS.

Полный перечень тестовых заданий с указанием правильных ответов, размещен в банке вопросов на информационно-образовательном портале института по ссылке https://www.mivlgu.ru/iop/question/edit.php?courseid=482

Оценка рассчитывается как процент правильно выполненных тестовых заданий из их общего числа.