

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТ-
ВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ

Н.Г. ЧЕРНЫШЕВСКОГО»

Факультет компьютерных наук и информационных технологий



УТВЕРЖДАЮ
декан факультета компьютерных
наук и информационных технологий
/Миронов С.В.
"13" 09 2021 г.

Рабочая программа дисциплины
Машинно-зависимые языки программирования

Направление подготовки бакалавриата
02.03.02 Фундаментальная информатика и информационные технологии

Профиль подготовки бакалавриата
Информатика и компьютерные науки

Квалификация (степень) выпускника

Бакалавр

Форма обучения
очная

Саратов,
2021

Статус	ФИО	Подпись	Дата
Преподаватель-разработчик	Федорова Антонина Гавриловна	<i>Федорова</i>	13.09.2021
Председатель НМК	Кондратова Юлия Николаевна	<i>Кондратова</i>	13.09.2021
Заведующий кафедрой	Огнева Марина Валентиновна	<i>Огнева</i>	13.09.2021
Специалист Учебно-го управления			

1. Цели освоения дисциплины

Целями освоения дисциплины Машинно-зависимые языки программирования являются:

- знакомство с архитектурой персональных ЭВМ магистрально-модульного принципа построения;
- изучение архитектуры микропроцессоров семейства Intel x86;
- изучение языка программирования Ассемблер для персональных ЭВМ, построенных на базе процессоров семейства Intel;
- овладение умениями создавать программы на языке программирования Ассемблер для персональных ЭВМ, построенных на базе процессоров семейства Intel;
- развитие алгоритмического мышления, способностей к формализации, элементов системного мышления;
- воспитание культуры проектной деятельности, в том числе умения планировать, работать в коллективе; чувства ответственности за результаты своего труда, используемые другими людьми; установки на позитивную социальную деятельность в информационном обществе, недопустимости действий, нарушающих правовые и этические нормы работы с информацией;

2. Место дисциплины в структуре ООП

Дисциплина «Машинно-зависимые языки программирования» относится к блоку «Дисциплины», части, формируемой участниками образовательных отношений.

Для изучения дисциплины необходимы компетенции, сформированные в результате изучения дисциплин: теоретическая информатика, информационные технологии и программирование. Компетенции, сформированные при изучении данной дисциплины, используются при изучении следующих дисциплин: структуры данных и алгоритмы, языки программирования, технологии программирования. Компетенции, сформированные при изучении данной дисциплины, могут быть полезны при изучении следующих дисциплин: информационная безопасность и защита информации, параллельное и распределенное программирование, тестирование программного обеспечения. Программирование на машинно-зависимых языках – это создание системного программного обеспечения, то есть написание таких программ, которые в совокупности составляют некоторую программную систему либо набор элементов для построения других программных систем. Программирование на машинно-зависимых языках – это, программирование для какой-то аппаратной системы. Наиболее распространенными системами на сегодняшний день (с точки зрения обучения и повседневного использования) являются персональные компьютеры на основе процессора фирмы Intel. Этот курс для студентов фа-

культета компьютерных наук и информационных технологий обеспечивается кафедрой информатики и программирования.

3. Результаты обучения по дисциплине

Код и наименование компетенции	Код и наименование индикатора (индикаторов) достижения компетенции	Результаты обучения
<p>ПК-4. Способен собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственнотехнологической деятельности; способен к разработке новых алгоритмических, методических и технологических решений для конкретной сферы профессиональной деятельности</p>	<p>ПК-4.1. Знает основы проектирования и элементы архитектурных решений информационных систем. ПК-4.2. Умеет применять в практической деятельности профессиональные стандарты в области информационных технологий. ПК-4.3. Имеет опыт составления технического задания на разработку информационной системы</p>	<p>Знает основные принципы построения архитектуры компьютеров, математические методы и правила анализа сложности программ, методы и средства повышения эффективности программ на машинно-зависимых языках.</p> <p>Умеет использовать современные среды разработки программных продуктов, для получения качественных и эффективных решений поставленных перед ним конкретных задач на машинно-зависимом языке программирования, применять в практической деятельности профессиональные стандарты.</p> <p>Владеет навыками использования современного системного программного обеспечения в том числе и отечественного при решении конкретных стандартных задач.</p>
<p>ПК-5. Способен применять в профессиональной деятельности современные технологии программирования, методы обработки и анализа больших данных, операционные системы, системы управления базами данных, сетевые технологии</p>	<p>ПК-5.1. Умеет создавать программный код в соответствии с техническим заданием (готовыми спецификациями) ПК-5.2. Умеет создавать варианты архитектуры программного средства в зависимости от спецификации окружения программного средства ПК-5.3. Владеет навыками описания алгоритмов компонентов, включая методы и схемы ПК-5.4.</p>	<p>Знает особенности разработки программ на машинно-зависимом языке программирования в среде ДОС и Windows в соответствии с техническим заданием.</p> <p>Умеет решать стандартные задачи с использованием различных структур данных, подпрограмм, макросом и директив условной генерации средствами машинно-зависимого языка программирования, используя современные информационные технологии.</p> <p>Владеет навыками описания</p>

	Умеет осуществлять выбор модели обеспечения необходимого уровня производительности компонент, включая вопросы балансировки нагрузки	алгоритмов, постановки и решения стандартных задач с помощью языка программирования Ассемблер для процессора фирмы Intel.
--	---	---

4. Структура и содержание дисциплины «Машинно-зависимые языки программирования»

Общая трудоемкость дисциплины составляет 4 зачетные единицы
144 часа.

№ п/п	Раздел дисциплины	С е м е с т р	Не де ля се ме ст ра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Формы текущего контроля успеваемости (по неделям семестра)		Формы промежуточной аттестации (по семестрам)
				Все го ча- сов	Л е к ц и и	Лабо- раторные		С Р		
						общая трудо- емк.	из них – прак- тиче- ская подго- товка			
1	Введение: цели и задачи, краткая характеристика курса. Общие принципы организации ЭВМ на примере ЭВМ семейства IBM PC	3	1-3	13	6	4	1	3	Опрос, проверка индивидуальных заданий, работы с пособием на сайте course.sgu.ru	
2	Основные элементы программирования на Ассемблере.	3	4-7	22	8	8	2	6	Опрос, проверка индивидуальных заданий, работы с пособием на сайте course.sgu.ru	
3	Сложные типы данных в Ассемблере: массивы, строки, структуры и за- писи	3	8-11	22	8	8	2	6	Опрос, проверка индивидуальных заданий, работы с пособием на сайте course.sgu.ru	
4	Процедуры в ассембле- ре, особенности их ис- пользования. Многомо-	3	12-14	18	6	6	3	6	Опрос, проверка индивидуальных заданий, работы с	

	дульные программы.								пособием на сайте course.sgu.ru
5	Макросредства в языке Ассемблер	3	15-16	15	4	6	1	5	Опрос, проверка индивидуальных заданий, работы с пособием на сайте course.sgu.ru
6	Работа с файлами и директориями в Ассемблере Организация ввода вывода. Программирование в среде операционной системы Windows	3	17-18	18	4	4	1	10	Опрос, проверка индивидуальных заданий, работы с пособием на сайте course.sgu.ru
	Промежуточная аттестация – 36ч.								Экзамен, Контрольная работа
	ИТОГО за 3 семестр			144	36	36	36	36	36

Содержание учебной дисциплины

Введение: цели и задачи, краткая характеристика курса. Общие принципы организации ЭВМ на примере ЭВМ семейства IBM PC. Области применения языков программирования низкого уровня. Поколения ПК IBM PC. Основные факторы, влияющие на рост производительности ВС. Архитектурные особенности современных ПК. Базовая архитектура ПК IBM PC, процессор с точки зрения программиста, регистры общего назначения, регистр флагов. Представление данных и команд, форматы команд, способы адресации операндов. Организация памяти, режимы работы процессора. Организация памяти в реальном режиме работы, сегментные регистры, понятие исполняемого и физического адреса.

Основные элементы программирования на Ассемблере.

Структура программы на Ассемблере, модели памяти, команды, директивы и комментарии. Алфавит, слова, константы, выражения, переменные. Стандартные директивы сегментации и упрощенные, (точечные), организация СОМ-файлов. Директивы определения данных и памяти. Команды пересылки безусловной и условной, команды загрузки адреса. Сегмент стека, организация работы со стекком, команды для работы со стекком, команды прерывания. Команды двоичной арифметики. Программирование линейных, разветвляющихся и циклических алгоритмов.

Сложные типы данных в Ассемблере: массивы, строки, структуры и записи. Массивы, выделение памяти, работа с одномерными и двумерными массивами. Команды для работы со строками, организация работы со строками переменной длины. Структуры в Ассемблере, их описание и использо-

вание. Команды побитовой обработки данных. Записи в Ассемблере, их описание и использование, команды для работы с записями.

Процедуры в ассемблере, особенности их использования. Многомодульные программы. Работа с подпрограммами в Ассемблере, способы передачи параметров. Передача параметров через стек, локальные параметры в процедуре, организация рекурсивных процедур. Этапы решения задач на Ассемблере в среде операционной системы MS-DOS. Многомодульные программы в Ассемблере, директивы для организации межмодульных связей. Трансляция программ на Ассемблере, компоновка программ, отладка программ

Макросредства в языке Ассемблер. Блоки повторения, макросы, Команды условной генерации в Ассемблере. Макрооператоры. Примеры использования блоков повторений, макросов и команд условной генерации.

Работа с файлами и директориями в Ассемблере. Программирование в среде операционной системы Windows. Дескриптор файла. Создать файл, открыть, закрыть, удалить файл. Чтение из файла и запись в файл, установка указателя файла, поиск файла, переименование файла. Работа с директориями в DOS: создать, удалить директорию, определить текущую, сменить текущую директорию, ошибки при работе с файлами. Пример простого алгоритма шифрования файлов.

Организация ввода вывода на высоком среднем и низком уровне в Ассемблере. Функции DOS для вывода на экран, пример. Функции BIOS для вывода на экран, пример. Возможности вывода на экран, вывод информации с помощью прямой записи в видеобуфер. Организация ввода с клавиатуры, функции DOS.

Основы работы в FASM, разбор готовой программы, создание учебной программы с использованием пособия на сайте course.sgu.ru.

План лабораторных занятий

План проведения лабораторных занятий рассчитан на 36 аудиторных часов и самостоятельную работу студентов и поддерживается учебно-методическими разработками.

<i>№ занятия</i>	<i>Содержание занятия</i>	<i>Упражнение</i>
1	Знакомство со средой программирования, разбор готовой программы на Ассемблере, этапы выполнения программы на компьютере, анализ результата работы программы. Модификация программы.	
2- 3	Структура программы на Ассемблере с использованием стандартных директив сегментации для создания EXE и COM-файла, разработка и отладка программ.	Задания из пособия.
4 - 6	Функции DOS для ввода символа с клавиатуры и вывода символов и строк на экран. Структура программы на Ассемблере с использованием точечных директив для EXE-файла и COM-файла. Создание Bat-файла для автоматизации процесса обработки программы (ASM-файл) на компьютере. Решение задач.	Задания из пособия.
7 - 9	Директивы описания данных и памяти. Описание массива	Задания из посо-

	в Ассемблере. Ввод строки символов, использование команд с различными способами адресации операндов.	бия.
10 - 12.	Понятие процедуры в Ассемблере, описание и использование. Реализация циклических и разветвляющихся алгоритмов в Ассемблере	Задания из пособия.
13	Макросредства в языке Ассемблер, использование макросов и директив условной генерации в программах.	Задания из пособия.
14 - 15	Знакомство с программированием на Ассемблере под Windows по руководству FASM. Установка и проверка работы компилятора. Работа с готовой программой (FASM/examples), модификация программы.	Руководство FASM], пособие на course.sgu.ru
16- 17	Изучение примеров Hello, PEDemo, Dialog (FASM/examples). Написание собственного приложения формата 32-bit PE с использованием секций sections (code, data, idata) и макросов invoke, library, import. Вызов функций Win32 API MessageBox, ExitProcess. Возможность в секции данных менять заголовок и сообщение окна.	Руководство FASM], пособие на course.sgu.ru
18	Итоговое, отчетное занятие	Контрольная работа

Лабораторные занятия проводятся с использованием Интернет-пособий, созданных в среде Moodle и расположенных на сервере <http://course.sgu.ru>. Отчет по лабораторным работам оформляется студентом в произвольной форме. В отчете необходимо указать:

1. титульный лист,
1. постановку задания,
2. алгоритм решения задачи,
3. исходный текст программы с комментариями,
4. описание средств языка, использованных для решения задачи,
5. результат выполнения программы,
6. дату и подпись.

5. Образовательные технологии, применяемые при освоении дисциплины

При проведении занятий по курсу используются как лекционные, так и лабораторные занятия и самостоятельная работа с использованием систем дистанционной поддержки занятий, разработанных сотрудниками факультета компьютерных наук и информационных технологий: Центра олимпиадной подготовки программистов, имени Н.Л. Андреевой, Центра непрерывной подготовки IT-специалистов. Эти системы позволяют проводить аудиторные занятия (практические и лабораторные) в интерактивной форме. В рамках лекционных и лабораторных занятий предусмотрены встречи с представителями организаций – разработчиков программного обеспечения, с которыми у факультета КНИИТ имеются договорные отношения, а также с сотрудниками

ПРЦ НИТ СГУ, обеспечивающих работу локальной сети университета и работу университетского кластера.

Предусматривается широкое использование в учебном процессе активных и интерактивных форм проведения занятий, в том числе: технология модульно-рейтингового обучения; информационные технологии, включая технологии дистанционного обучения; технологии организации группового взаимодействия.

В рамках практической подготовки по данной дисциплине используются технологии оценки достижений и самоконтроля, анализ конкретных ситуаций и технология развития критического мышления при защите рефератов: 2, 3 студента должны сделать замечания по теме, отметить достоинства работы и, возможно, высказать предложения по улучшению доклада и работы в целом, высказать свою оценку.

Удельный вес занятий, проводимых в активных и интерактивных формах, составляет 60% аудиторных занятий.

При обучении лиц с ограниченными возможностями и инвалидов используются подходы, способствующие созданию безбарьерной образовательной среды: технологии дифференциации и индивидуализации обучения, сопровождение тьюторами в образовательном пространстве; увеличивается время на самостоятельное освоение материала. Используется сочетание разных форм и способов передачи учебной информации: вербальный, невербальный, с использованием средств визуализации информации (презентации) и разных способов отчетности (письменно, устно, с использованием электронных дистанционных технологий). Занятия лекционного типа для соответствующих групп студентов составляют менее 60% аудиторных занятий.

6. Учебно-методическое обеспечение самостоятельной работы студентов. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

В качестве самостоятельной работы студенты должны изучить дополнительную литературу по соответствующим разделам рабочей программы, выполнить самостоятельно задания из пособий по языку программирования с использованием сайта course.sgu.ru, написать отчет.

Фонд оценочных средств дисциплины включает в себя:

- разноуровневые задания: репродуктивного уровня, позволяющие оценивать и диагностировать знание фактического материала; реконструктивного уровня, позволяющие оценивать и диагностировать умения синтезировать, анализировать, обобщать фактический и теоретический материал; творческого уровня, позволяющие оценивать и диагностировать умения, интегрировать знания, разрабатывать эффективные алгоритмы,
- собеседование,
- контрольные вопросы,

- доклад по теме самостоятельной работы,

Рекомендуемые темы для самостоятельной работы:

1. Обратить внимание на области применения языков программирования низкого уровня, применение Ассемблеров для решения задач реального времени, задач, требующих обработка большого объема информации.
2. Рассмотреть понятие сопроцессора, его назначение и использование при программировании на Ассемблере.
3. Изучить работу с числами с плавающей точкой в Ассемблере. Литература:
4. Изучить сложные структуры и типы данных в Ассемблере – самостоятельно рассмотреть организацию работы с динамическими данными в Ассембле.
5. Макросредства в языке Ассемблер, рассмотреть примеры макросов с использованием директив условной генерации, примеры применения макросов для обращения к подпрограммам.
6. Углубленно рассмотреть этапы решения задач на Ассемблере в среде операционной системы MS-DOS. Организацию многомодульных программ. Самостоятельно изучить этапы решения задач на Ассемблере в среде операционной системы Windows.
7. Рассмотреть примеры работы с файлами и директориями в Ассемблере с использованием всех функций, рассмотренных в лекциях; изучить работу с файлами и директориями с длинными именами.
8. Изучить организацию ввода и вывода на высоком среднем и низком уровне в Ассемблере, рассмотреть самостоятельно примеры работы с портами ввода-вывода: клавиатура, последовательный и параллельный порт, видеоадаптер и таймер.
9. Рассмотреть прерывания, классификацию прерываний, контроллеры прерываний, углубленно рассмотреть перехват прерываний и обработчики прерываний.
10. Резидентные программы, их структура и использование: рассмотреть примеры резидентных и полурезидентных программ и способы запуска и обращения к ним.
11. Дополнительно к практическим занятиям рассмотреть программирование в среде операционной системы Windows.

Вопросы к экзамену:

- 1) Области применения языков программирования низкого уровня. Поколения ПК IBM PC.
- 2) Архитектурные особенности современных ПК. Основные факторы, влияющие на рост производительности ВС.
- 3) Базовая архитектура ПК IBM PC , процессор с точки зрения программиста, регистры общего назначения, регистр флагов.
- 4) Организация памяти в реальном режиме работы, сегментные регистры, понятие исполняемого и физического адреса.

- 5) Форматы команд, способы адресации операндов, примеры команд с использованием различных способов адресации.
- 6) Структура программы на Ассемблере, модели памяти, организация СОМ-файлов.
- 7) Команды пересылки безусловной и условной, команды загрузки адреса.
- 8) Сегмент стека, организация работы со стеком, команды для работы со стеком, команды прерывания.
- 9) Команды двоичной арифметики: сложение, вычитание, умножение и деление.
- 10) Команды безусловной передачи управления, обращения к подпрограмме и возврат из подпрограммы, директивы для организации межмодульных связей.
- 11) Команды условной передачи управления, организация циклов в Ассемблере, работа с массивами.
- 12) Структуры в Ассемблере, их описание и использование.
- 13) Команды побитовой обработки данных : логические операции, операции сдвига.
- 14) Записи в Ассемблере, их описание и использование.
- 15) Работа с подпрограммами в Ассемблере, способы передачи параметров.
- 16) Передача параметров через стек, локальные параметры в процедуре, организация рекурсивных процедур.
- 17) Команды для работы со строками, организация работы со строками.
- 18) Понятие списка, реализация списков в языке Ассемблер.
- 19) Организация динамической памяти в Ассемблере, создание и удаление динамических переменных.
- 20) Макросредства в языке Ассемблере: блоки повторения.
- 21) Модели памяти, организация программы с помощью точечных директив.
- 22) Макросы в Ассемблере, их описание и использование.
- 23) Исполняемые СОМ-файлы, их отличие от EXE-файлов, примеры.
- 24) Команды условной генерации в Ассемблере.
- 25) Работа со строками переменной длины в Ассемблере.
- 26) Многомодульные программы в Ассемблере, директивы для организации межмодульных связей.
- 27) Передача параметров в подпрограммы по ссылке и по значению, организация рекурсивных подпрограмм.
- 28) Работа с файлами. Программирование на высоком, среднем и низком уровне. Дескриптор файла. Создать файл, открыть, закрыть, удалить файл.
- 29) Работа с файлами. Чтение из файла и запись в файл, установка указателя файла, поиск файла, переименование файла.

- 30) Работа с директориями в DOS: создать, удалить директорию, определить текущую, сменить текущую директорию, ошибки при работе с файлами.
- 31) Простой алгоритм шифрования файлов. Пример.
- 32) Функции DOS для вывода на экран, пример.
- 33) Функции BIOS для вывода на экран, пример.
- 34) Возможности вывода на экран, вывод информации с помощью прямой записи в видеобуфер.
- 35) Организация ввода с клавиатуры, функции DOS, примеры.

Темы рефератов:

1. Классификации ЭВМ.
2. Архитектура микропроцессора семейства Intel.
3. Системы команд машин различных поколений, адресация памяти.
4. История суперкомпьютеров.
5. Компьютеры будущего.
6. Квантовые компьютеры.
7. Нейронные сети и искусственный интеллект.
8. Сопроцессор, его назначение.
9. Работа с числами с плавающей точкой в Ассемблере.
10. Классификация прерываний, обработка прерываний.
11. Резидентные программы, их назначение и использование.
12. Полурезидентные программы, особенности их использования.
13. Битовые операции в Ассемблере, примеры их использования.
14. Контроллеры прерываний и обработчики прерываний.
15. Применение макросов для обращения к подпрограммам, примеры.

Примеры заданий для контрольной работы содержатся в ФОС.

Вопросы для текущего контроля

1. Почему машинно-зависимые языки называют языками низкого уровня?
2. Что такое разрядность процессора?
3. Что такое регистр, назначение регистров?
4. Как регистр флагов определяет текущее состояние программы и процессора?
5. Что такое прерывание в Ассемблере?
6. Какие типы прерываний существуют?
7. Как определяется физический адрес памяти в реальном режиме процессора?
8. Понятие исполняемого адреса для команды и для операнда.
9. Чем отличается регистровая адресация от косвенно-регистровой?
10. Какая адресация используется для работы с массивами и структурами?
11. Виды исполняемых файлов в Ассемблере.

12. Чем отличается исполняемый COM-файл от EXE-файла?
13. Чем отличается использование стандартных директив сегментации от упрощенных?
14. Какие модели памяти существуют при использовании точечных директив сегментации?
15. Чем отличается процедура дальнего типа вызова от процедуры ближнего типа вызова?
16. Каким образом можно определить переменную в Ассемблере?
17. Как загрузить адрес переменной в регистр?
18. Как определить тип операнда в команде?
19. Как вывести символ на экран?
20. Как вывести строку на экран?
21. В каком регистре по умолчанию находится второй операнд в командах умножения и деления?
22. Что значит короткая безусловная передача?
23. Как реализовать длинную условную передачу?
24. Как реализовать цикл с предусловием и постусловием в Ассемблере?
25. В каком регистре должно находиться количество повторений цикла?
26. Особенности организации циклов с помощью специальных команд?
27. Почему второй операнд в логических командах называется маской?
28. Чем отличается арифметический сдвиг от логического сдвига?
29. Чем отличается структурный тип данных в Ассемблере от структурного типа в языках высокого уровня?
30. Что такое запись в Ассемблере?
31. Как можно работать с полями записи?
32. Где хранится операнд источник и операнд приемник для строковых команд?
33. Как установить направление обработки строки в сторону уменьшения адресов?
34. Как установить направление обработки строки в сторону увеличения адресов?
35. Как осуществляется переход от одного элемента строки к другому?
36. Какой регистр по умолчанию используется вторым операндом в командах загрузки памяти и считывания из памяти?
37. Как загрузить оба операнда строковой команды в один сегмент данных?
38. Какие способы передачи параметров в подпрограмму возможны в Ассемблере?
39. Чем отличается передача параметров по значению от передачи параметров по ссылке?
40. Где могут размещаться параметры подпрограммы?
41. Особенности использования регистров для передачи параметров в подпрограмму?
42. В чем заключаются особенности передачи параметров через стек?
43. Как реализовать рекурсивную подпрограмму в Ассемблере?

44. Что такое макросредства?
45. Блоки повторений, что это, для чего используются?
46. Что такое макроопределение и макрорасширение?
47. Как обратиться к макросу?
48. Что общего у макроса и подпрограммы?
49. Чем отличается макрос от подпрограммы?
50. Для чего используются директивы условной генерации?

7. Данные для учета успеваемости студентов в БАРС

Таблица 1. Таблица максимальных баллов по видам учебной деятельности.

1	2	3	4	5	6	7	8	9
Семестр	Лек-ции	Лабораторные занятия	Практические занятия	Самостоятельная работа	Автоматизированное тестирование	Другие виды учебной деятельности	Промежуточная аттестация	Итого
4	10	35	0	10	0	10	35	100

Программа оценивания учебной деятельности студента 4 семестр

Лекции посещаемость, активность, опрос от 0 до 10 баллов

Лабораторные занятия

Контроль выполнения лабораторных заданий в течение одного семестра (выполнение заданий, контрольная работа) – от 1 до 35 баллов.

Практические занятия не предусмотрены.

Самостоятельная работа (подготовка к опросу, выполнение домашних заданий) – от 1 до 10 баллов

Автоматизированное тестирование - не предусмотрено

Другие виды учебной деятельности

Виды учебной деятельности, не вошедшие в предыдущие колонки таблицы (блиц-опрос, своевременное выполнение текущих заданий) – от 1 до 10 баллов.

Промежуточная аттестация

26-35 баллов – ответ на «отлично»

16-25 баллов – ответ на «хорошо»

6-15 баллов – ответ на «удовлетворительно»

0-5 баллов – неудовлетворительный ответ.

Таким образом, максимально возможная сумма баллов за все виды учебной деятельности студента за один семестр по дисциплине «Машинно-зависимые языки программирования» составляет 100 баллов.

Таблица 2. Таблица пересчета полученной студентом суммы баллов по дисциплине «Машинно-зависимые языки программирования» в оценку:

91 - 100 баллов и более	«отлично»
76 – 90 баллов	«хорошо»
65 – 75 баллов	«удовлетворительно»
0 – 64 баллов	«неудовлетворительно»

8. Учебно-методическое и информационное обеспечение дисциплины «Машинно-зависимые языки программирования»

а) литература:

1. Колдаев Виктор Дмитриевич. Архитектура ЭВМ [Текст] : Учебное пособие / Виктор Дмитриевич Колдаев, Сергей Андреевич Лупин. - Москва : Издательский Дом "ФОРУМ"; Москва : ООО "Научно-издательский центр ИНФРА-М", 2013. - 384 с. - ISBN 978-5-8199-0373-5: Б. ц
2. Калашников, О.А. Ассемблер — это просто. Учимся программировать. 2 изд. [Электронный ресурс] / О. Калашников. - Санкт-Петербург: БХВ-Петербург, 2014. - 336 с.: ил. - ISBN 978-5-9775-2170-3 : Б. ц, <http://ibooks.ru/reading.php?productid=22669>
3. Кирнос В. Н. Введение в вычислительную технику. Основы организации ЭВМ и программирование на Ассемблере [Электронный ресурс] : учебное пособие / Кирнос В. Н. - Томск :Томский государственный университет систем управления и радиоэлектроники, 2011. - 172 с. - ISBN 978-5-4332-0019-7 : Б. ц., ЭБС IPRbooks.
4. Таненбаум, Эндрю С. Архитектура компьютера [Текст] / Э. С. Таненбаум ; пер. с англ. Ю. Гороховского, Д. Шинтякова. - 5-е изд. - Москва ; Санкт-Петербург [и др.] : Питер, 2010. - 843, [5] с. + 1 эл. опт. диск (CD-ROM).- (Классика computer science). - Алф. указ.: с. 825-843. - ISBN 978-5-469-01274-0 (в пер.). - ISBN 0-13-148521-0 (англ.) : 433.30 р.
5. Бройдо, Владимир Львович. Архитектура ЭВМ и систем [Текст] : учеб. для вузов / В. Л. Бройдо, О. П. Ильина. - 2-е изд. - Москва ; Санкт-Петербург [и др.] : Питер, 2009. - 720 с. : ил. - (Серия "Учебник для вузов"). - Библиогр.: с. 717-720 (70 назв.).- ISBN 978-5-388-00384-3(в пер.): 267.60 р.

2) программное обеспечение и Интернет-ресурсы:

- Программное обеспечение: Internet Explorer, текстовый редактор, среда программирования TASM и FASM.
- Интернет-пособия для проведения практических и самостоятельных занятий на сайте <http://coors.sgu.ru>
- Образовательный сайт course.sgu.ru.

- <http://www.kalashnikoff.ru/Assembler>

9. Материально-техническое обеспечение дисциплины

Необходимыми средствами обучения является:

1. лекционная аудитория с мультимедийным оборудованием с выходом в Интернет;
2. компьютерные классы с программным обеспечением, рассчитанные на обучение группы студентов из 8 – 12 человек, удовлетворяющие санитарно-гигиеническим требованиям под управлением операционной системы Microsoft Windows Vista, 7, XP, 10 с подключением к Интернет;
3. программное обеспечение: Internet Explorer, текстовый редактор, среда программирования TASM и FASM.

Реализация практической подготовки в рамках учебных занятий запланирована на базе кафедры информатики и программирования, а также Центра олимпиадной подготовки программистов имени Н.Л. Андреевой.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению и профилю подготовки 02.03.02 Фундаментальная информатика и информационные технологии», профиль «Информатика и компьютерные науки».

Автор

доцент, к.ф.-м.н.

А.Г. Федорова

Программа одобрена на заседании кафедры информатики и программирования от 13 мая 2019 года, протокол № 17.

Программа актуализирована в 2021 году (одобрена на заседании кафедры информатики и программирования от 13 сентября 2021 года, протокол № 2)