

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Такаев Сергей Валерьевич
Должность: Ректор
Дата подписания: 15.06.2026 12:30:22
Уникальный программный ключ:
04c19ed8bfb98f3b6cb77a486b9a8788b8322525



МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Математический факультет Кафедра компьютерной безопасности и прикладной алгебры			
Фонд оценочных средств по дисциплине «Языки Ассемблера» по специальности 10.05.01 Компьютерная безопасность специализации № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»			
Версия документа - 1	стр. 1	Первый экземпляр _____	КОПИЯ № _____

**Фонд оценочных средств
для промежуточной аттестации
по дисциплине
Языки Ассемблера**

Направление подготовки (специальность)
10.05.01 Компьютерная безопасность


Направленность (профиль)
специализация № 6 «Информационно-аналитическая и техническая
экспертиза компьютерных систем»

Присваиваемая квалификация
специалист по защите информации

Форма обучения
очная

Год набора 2026

Челябинск 2026 г.

	МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Математический факультет Кафедра компьютерной безопасности и прикладной алгебры		
	Фонд оценочных средств по дисциплине «Язык Ассемблера» по специальности 10.05.01 Компьютерная безопасность специализации № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»		
Версия документа - 1	стр. 2	Первый экземпляр _____	КОПИЯ № _____

Содержание

1. Паспорт фонда оценочных средств
2. Перечень формируемых компетенций
 - 2.1. Компетенции, закреплённые за дисциплиной
3. Содержание оценочных средств по дисциплине
 - 3.1. Виды оценочных средств
 - 3.2. Содержание оценочных средств
4. Порядок проведения и критерии оценивания промежуточной аттестации
 - 4.1. Порядок проведения промежуточной аттестации
 - 4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств
 - 4.3. Результаты промежуточной аттестации и уровни сформированности компетенций



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра компьютерной безопасности и прикладной алгебры

Фонд оценочных средств по дисциплине «Языки Ассемблера»
по специальности 10.05.01 Компьютерная безопасность
специализации № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»

Версия документа - 1

стр. 3

Первый экземпляр _____

КОПИЯ № _____

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

Специальность 10.05.01 Компьютерная безопасность.

Специализация № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем».

Дисциплина: **Языки Ассемблера.**

Семестр изучения: 4, 5 семестры.

Формы промежуточной аттестации: зачет (4 семестр), экзамен (5 семестр).


Используется балльно-рейтинговая система для оценивания результатов.

2. ПЕРЕЧЕНЬ ФОРМИРУЕМЫХ КОМПЕТЕНЦИЙ

2.1. Компетенции, закреплённые за дисциплиной

Изучение дисциплины «Языки Ассемблера» направлено на формирование следующих компетенций:

Коды компетенции согласно ФГОС (ОПОП ВО)	Содержание компетенций согласно ФГОС (ОПОП ВО)	Индикаторы достижения компетенции согласно ОПОП	Перечень планируемых результатов обучения по дисциплине
1	2	3	4
ОПК-7	Способен создавать программы на языках высокого и низкого уровня, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ	ОПК-7.1 Знает общие принципы построения, области и особенности применения языков программирования высокого и низкого уровня; язык программирования высокого и низкого уровня (объектно-ориентированное программирование); знает язык ассемблера персонального компьютера. ОПК-7.2 Умеет работать с интегрированной средой разработки программного обеспечения; разрабатывать и реализовывать на языке высокого и низкого уровня алгоритмы решения типовых профессиональных задач; ОПК-7.3 Владеет навыками разработки, документирования, тестирования и отладки программ.	Знать: – специфику создания низкоуровневого кода под современные процессоры. – синтаксис языков ассемблера для современных процессоров. Уметь: – проектировать программное обеспечение с учётом низкоуровневой специфики архитектуры современных процессоров. – создавать код любой сложности под современные процессоры. Владеть: – навыки сопряжения низкоуровневого кода с другими программными системами. – навыками использования инструментальных средств создания кода под современные процессоры.

 <p>МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Математический факультет Кафедра компьютерной безопасности и прикладной алгебры</p>			
<p>Фонд оценочных средств по дисциплине «Языки Ассемблера» по специальности 10.05.01 Компьютерная безопасность специализации № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»</p>			
Версия документа - 1	стр. 4	Первый экземпляр _____	КОПИЯ № _____

3. СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

3.1 Виды оценочных средств

№ п/п	Код компетенции / планируемые результаты обучения	Контролируемые темы/ разделы	Наименование оценочного средства для текущего контроля	Наименование оценочного средства на промежуточной аттестации/№ задания
1.	ОПК-7	Раздел 1. Понятие об архитектуре вычислительной машины. Архитектура фон Неймана.	Зачетные задания. Домашние задания. Аудиторные задания.	Вопросы на зачёте. Задания на зачёте. Вопросы на экзамене. Задания на экзамене.
2.	ОПК-7	Раздел 2. Типы данных используемых в IBM PC.	Зачетные задания. Домашние задания. Аудиторные задания.	Вопросы на зачёте. Задания на зачёте. Вопросы на экзамене. Задания на экзамене.
3.	ОПК-7	Раздел 3. Режимы адресации в системе команд IBM PC	Зачетные задания. Домашние задания. Аудиторные задания.	Вопросы на зачёте. Задания на зачёте. Вопросы на экзамене. Задания на экзамене.
4.	ОПК-7	Раздел 4. Машинные форматы команд IBM PC	Зачетные задания. Домашние задания. Аудиторные задания.	Вопросы на зачёте. Задания на зачёте. Вопросы на экзамене. Задания на экзамене.
5.	ОПК-7	Раздел 5. Система команд IBM PC	Зачетные задания. Домашние задания. Аудиторные задания.	Вопросы на зачёте. Задания на зачёте. Вопросы на экзамене. Задания на экзамене.
6.	ОПК-7	Раздел 6. Основные сведения об ассемблере MASM86	Зачетные задания. Домашние задания. Аудиторные задания.	Вопросы на зачёте. Задания на зачёте. Вопросы на экзамене. Задания на экзамене.
7.	ОПК-7	Раздел 7. Данные в текстах на Ассемблере MASM86	Зачетные задания. Домашние задания. Аудиторные задания.	Вопросы на зачёте. Задания на зачёте. Вопросы на экзамене. Задания на экзамене.
8.	ОПК-7	Раздел 8. Операторы в текстах на Ассемблере MASM86	Зачетные задания. Домашние задания. Аудиторные задания.	Вопросы на зачёте. Задания на зачёте. Вопросы на экзамене. Задания на экзамене.
9.	ОПК-7	Раздел 9. Процедуры в текстах на Ассемблере MASM86	Зачетные задания. Домашние задания. Аудиторные задания.	Вопросы на зачёте. Задания на зачёте. Вопросы на экзамене. Задания на экзамене.
10.	ОПК-7	Раздел 10. Средства макрогенерации и условной генерации	Зачетные задания. Домашние задания. Аудиторные задания.	Вопросы на зачёте. Задания на зачёте. Вопросы на экзамене.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра компьютерной безопасности и прикладной алгебры

Фонд оценочных средств по дисциплине «Язык Ассемблера»
по специальности 10.05.01 Компьютерная безопасность
специализации № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»

Версия документа - 1


стр. 5

Первый экземпляр _____

КОПИЯ № _____

		MASM86		Задания на экзамене.
11.	ОПК-7	Раздел 12. Процессоры архитектуры IA-32	Зачетные задания. Домашние задания. Аудиторные задания.	Вопросы на зачёте. Задания на зачёте. Вопросы на экзамене. Задания на экзамене.
12.	ОПК-7	Раздел 13. Процессоры архитектуры Intel 64	Зачетные задания. Домашние задания. Аудиторные задания.	Вопросы на зачёте. Задания на зачёте. Вопросы на экзамене. Задания на экзамене.
13.	ОПК-7	Раздел 14. RISC процессоры	Зачетные задания. Домашние задания. Аудиторные задания.	Вопросы на зачёте. Задания на зачёте. Вопросы на экзамене. Задания на экзамене.

Типовые задания, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе дисциплины (модуля). Полные комплекты оценочных средств и контрольно-измерительных материалов хранятся на кафедре.

	МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Математический факультет Кафедра компьютерной безопасности и прикладной алгебры		
	Фонд оценочных средств по дисциплине «Языки Ассемблера» по специальности 10.05.01 Компьютерная безопасность специализации № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»		
Версия документа - 1	стр. 6	Первый экземпляр _____	КОПИЯ № _____

3.2 Содержание оценочных средств


3.2.1 Темы для домашних, аудиторных и зачетных заданий

1. Программный интерфейс DOS.
2. Программный интерфейс BIOS.
3. Ассемблер `masm`.
4. Система команд процессоров архитектуры `x86`.
5. Система команд процессоров архитектуры `IA-32`.
6. Система команд процессоров архитектуры `Intel 64`.
7. Арифметический сопроцессор.
8. `SIMD`-расширения процессоров `Intel`.
9. `WinAPI`.
10. `ABI` операционных систем `Windows NT` и `Linux`.
11. Система команд процессоров архитектуры `ARM`.
12. Система команд процессоров архитектуры `MIPS`.
13. Система команд процессоров архитектуры `AVR`.
14. Работа с динамическими структурами данных на языке ассемблера.

3.2.2 Примеры зачетных заданий

I.

- 1) Описать собственный процессор:
 - архитектура (`CISC`, `RISC`);
 - разрядность;
 - регистры;
 - адресное пространство;
 - режимы адресации;
 - система команд;
 - формат машинных команд;
 - доступ к аппаратному обеспечению (проецирование видеобuffers и кода инициализации в память);
 - обработка прерываний (таймер, клавиатура);
 - состояние процессора (значения регистров) в момент запуска.
- 2) Создать язык ассемблера для этого процессора
 - мнемоники и синтаксис инструкций;
 - имена регистров;
 - механизмы сегментирования (код и данные);
 - выделение и инициализация памяти;
 - директивы ассемблера;

	МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Математический факультет Кафедра компьютерной безопасности и прикладной алгебры		
	Фонд оценочных средств по дисциплине «Языки Ассемблера» по специальности 10.05.01 Компьютерная безопасность специализации № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»		
Версия документа - 1	стр. 7	Первый экземпляр _____	КОПИЯ № _____

- метки;
- числовые типы данных;
- представление строк и символов.

3) Написать ассемблер, реализующий созданный язык, на любом языке программирования.

4) Написать программу-эмулятор для разработанного процессора.

Эмулятор должен эмулировать монитор, доступный через запись в видеобuffer, клавиатуру и системный таймер. В эмуляторе должна запускаться программа, созданная написанным ассемблером. В эмуляторе должна отображаться картинка с виртуального монитора, соответствующая текущему состоянию видеобufferа.

Базовая функциональность (к сдаче не принимаются задания с функциональностью меньше базовой), 10 баллов.

Требования:

1) Система команд должна включать основные инструкции:

- арифметические
- логические
- пересылки данных
- битовые инструкции (побитовые логические, сдвиги, установка/получение битов)
- условные переходы
- безусловного перехода
- вызов функции/возврат из функции

2) Инструкции должны кодироваться без явной избыточности. Например, не должны отводиться байты для кодирования отсутствующих у инструкции операндов.

3) Язык ассемблера должен предоставлять средства для именования адресов (метки, переменные).

4) Система команд должна предоставлять как минимум два способа адресации:


- адрес в регистре (общего назначения)
- непосредственный адрес либо смещение относительно некоторого регистра

5) Написать на созданном ассемблере программу, демонстрирующую все реализованные функции.

Полная функциональность, 25 баллов.

Требования (дополнительно к базовым):

1) Система команд должна предоставлять следующие возможности:

	МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Математический факультет Кафедра компьютерной безопасности и прикладной алгебры		
	Фонд оценочных средств по дисциплине «Языки Ассемблера» по специальности 10.05.01 Компьютерная безопасность специализации № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»		
Версия документа - 1	стр. 8	Первый экземпляр _____	КОПИЯ № _____

- получение нажатых клавиш
- периодические действия с заданным периодом
- получение случайных значений

2) Система команд дополнительно должна предоставлять способы адресации:

- адрес получается как сумма двух регистров (общего назначения)
- адрес получается как сумма двух регистров (общего назначения) плюс абсолютное значение

3) Инструкции должны кодироваться оптимальным способом. Например если в системе команд предусмотрено не более 32 инструкций, то для их кодирования отводится не более 5 бит, а оставшиеся в байте биты используются для кодирования чего-то другого.

4) Эмулятор должен предоставлять основные возможности отладки: точки останова,

трассировку с заходом и без, отображение памяти и регистров.

5) Реализовать простую графическую игру для запуска на эмуляторе. В игре должны быть взаимодействие с пользователем, периодическая функциональность компьютера.

II. Реализовать на языке ассемблера арифметику больших чисел.

Большие числа должны храниться в массиве двойных слов. Данные должны храниться максимально компактно, т.е. желательно чтобы в каждом двойном слове хранились 32 бита большого числа.

Для отрицательных чисел можно использовать дополнительный код или специальное поле в структуре.


Реализовать тестовую программу, демонстрирующую реализованные функции.

Базовая функциональность (к сдаче не принимаются задания с функциональностью меньше базовой), 10 баллов.

Требования:

1) Должны быть реализованы функции:

- инициализации большого числа (числом, строкой шестнадцатиричных символов);
- сложения двух больших чисел;
- вычитания двух больших чисел;
- логических побитовых операций (and, xor, or);
- умножение большого числа на 4-х байтное число;
- вывода большого числа в шестнадцатиричном виде.

	МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Математический факультет Кафедра компьютерной безопасности и прикладной алгебры		
	Фонд оценочных средств по дисциплине «Языки Ассемблера» по специальности 10.05.01 Компьютерная безопасность специализации № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»		
Версия документа - 1	стр. 9	Первый экземпляр _____	КОПИЯ № _____

Полная функциональность, 20 баллов.

Требования (дополнительно к базовым):

- 1) В каждом двойном слове должны храниться 32 бита большого числа.
- 2) Во всех операциях должны поддерживаться отрицательные большие числа.
- 3) Реализовать функцию умножения двух больших чисел.

III. Реализовать интерактивную графическую игру на языке ассемблера x86 с использованием интерфейса WinAPI.

Базовая функциональность (к сдаче не принимаются задания с функциональностью меньше базовой), 10 баллов.

Должен быть реализован базовый игровой функционал:

- схематичная графика;
- реакция на действия пользователя;
- отслеживание соблюдения основных правил игры.

Полная функциональность.

Требования (дополнительно к базовым):

1. Случайные ответные действия компьютера (до 5 баллов).
2. Осмысленные ответные действия компьютера (до 10 баллов).
3. Дополнительные улучшения: уровни игры, улучшенная графика, проверки корректности, подсказки и т.д. (до 10 баллов).

IV. Написать программу под Win64.

Программа должна рисовать график функции на указанном интервале.

Интерфейс программы должен позволять ввести функцию и указать нужный интервал (целочисленные концы интервала).


С помощью интегральных сумм необходимо вычислить интеграл функции на заданном интервале.

Для вычисления интеграла и значения функции в точках необходимо использовать арифметический сопроцессор.

Базовая функциональность (к сдаче не принимаются задания с функциональностью меньше базовой), 15 баллов.

1) Должна быть реализована поддержка следующих функций:

- $\sin x$
- $\cos x$
- $\operatorname{tg} x$
- $\operatorname{ctg} x$
- x в произвольной степени

	МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Математический факультет Кафедра компьютерной безопасности и прикладной алгебры		
	Фонд оценочных средств по дисциплине «Языки Ассемблера» по специальности 10.05.01 Компьютерная безопасность специализации № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»		
Версия документа - 1	стр. 10	Первый экземпляр _____	КОПИЯ № _____

- логарифм (двоичный, натуральный, десятичный)

2) Вычисление интеграла.

Полная функциональность.

Требования (дополнительно к базовым):

- 1) Поддержка произвольного многочлена с произвольными степенями (возможно, отрицательными и вещественными) (5 баллов).
- 2) Масштабирование по оси абсцисс (5 баллов).
- 3) Масштабирование по оси ординат (5 баллов).

V. На языке ассемблера для 32-х разрядных процессоров семейства ARM написать программу, выполняющую простые арифметические действия с двумя числами: сложение, вычитание, умножение, деление.

Программа должна принимать три аргумента: операнд1, операция, операнд2. Если аргументов меньше, то должна из стандартного ввода считывать строку с вычисляемым выражением и разбивать её на операнды и операцию. Дальше строки с операндами необходимо перевести в числа (для этого реализовать свой аналог функции atoi). Над операндами выполнить запрашиваемую операцию и вывести результат.

Базовая функциональность (к сдаче не принимаются задания с функциональностью меньше базовой), 10 баллов.

Требования:

- 1) Принимать данные через аргументы командной строки или из стандартного ввода.
- 2) Выполнять операции: сложение, вычитание, умножение.

Полная функциональность.

Требования (дополнительно к базовым):

- 1) Реализовать свою функцию atoi (до 5 баллов).
- 2) Выполнять операцию деления (реализовать её самостоятельно) (до 5 баллов).

VI. Написать графическую программу под Win64 на языке ассемблера для процессоров архитектуры x64 с использованием арифметического сопроцессора. Программа должна отображать движение по разным круговым (эллиптическим) орбитам некоторых тел (считать их точечными объектами). Можно считать это упрощённой моделью движения планет вокруг звезды. Планеты движутся с разной скоростью по своим орбитам и, очевидно, имеют разную продолжительность полного оборота.

Помимо собственно вычислений следует уделить особое внимание



проектированию абстракций и архитектуры программы.

Базовая функциональность (к сдаче не принимаются задания с функциональностью меньше базовой).

20 баллов.

1) Реализовать упрощенную (по круговым орбитам) модель движения 8 планет Солнечной системы (т.е. взять радиусы орбит и периоды обращения настоящих планет).

2) Реализовать визуализацию (плавного) движения планет по своим орбитам. Интерфейс должен предоставлять возможность указать скорость вращения (например, дней/месяцев/лет в секунду/минуту). Плавность движения должна обеспечиваться изменением положения планет несколько раз (изменяемый параметр) в секунду.

3) Самостоятельно отрисовывать линии орбит.

Полная функциональность.

Требования (дополнительно к базовым):

1) Масштабировать отображение так, чтобы наибольшая орбита занимала всю доступную область окна. И относительно этого масштабировать другие орбиты.

До 5 баллов.

Предусмотреть возможность отображения (в отдельных окнах) отдельных областей общей картины (со своими коэффициентами масштабирования).

До 10 баллов.

2) Интерфейс должен позволять схватить мышкой любое тело и начать перетаскивать его (по его орбите). Соответствующим образом должны двигаться все остальные тела.

До 10 баллов.

3) Сделать модель более правдоподобной, добавив дополнительные параметры/элементы из реальной Солнечной системы. Примеры таких изменений:


- вместо круговых использовать эллиптические орбиты (Солнце в одном из фокусов эллипса), взять эксцентриситеты настоящих планет (для наглядности можно добавить Плутон);

- добавить движение спутников планет;

- вращение тел вокруг своей оси (взять углы наклона планет).

Баллы определяются в зависимости от сложности добавленной функциональности.

4) Сделать модель трёхмерной. Все реализуемые параметры (например,

	МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Математический факультет Кафедра компьютерной безопасности и прикладной алгебры		
	Фонд оценочных средств по дисциплине «Язык Ассемблера» по специальности 10.05.01 Компьютерная безопасность специализации № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»		
Версия документа - 1	стр. 12	Первый экземпляр _____	КОПИЯ № _____

наклоны орбит, наклоны осей) должны соответствовать реальным. Интерфейс должен позволять поворачивать и масштабировать трёхмерную картинку произвольным образом.

До 30 баллов.

3.2.3 Примеры домашних и аудиторных заданий

1. Копирование файлов через интерфейс DOS.
2. Алгоритм рисования линии через видеобuffer.
3. Реализовать эмулятор для собственной простой архитектуры процессора.
4. Реализовать алгоритмы обработки данных (сортировка, поиск, контейнеры и др.) на языке ассемблера.
5. Реализовать на языке ассемблера стандартные функции работы со строками.
6. Реализовать арифметику больших чисел.
7. Реализовать замену символа в строке.
8. Реализовать замену строк.
9. Реализовать сортировку массива строк.
10. Реализовать копирование файла.
11. Реализовать замену строк в файле.
12. Реализовать разбиение строки на подстроки по подстроке.
13. Инвертирование массива чисел.
14. Сложение элементов массивов.
15. Поиск элементов массива.
16. Реализовать стандартные строковые функции с помощью строковых инструкций.
17. Реализовать динамические структуры данных и программы для их тестирования.
18. Реализовать на языке ассемблера программы с графическим интерфейсом.
19. Реализовать вычисление интеграла через интегральные суммы.
20. Реализовать функцию вывода на экран в десятичном виде вещественного числа в 4- и 8-байтовом формате.
21. Реализовать функцию возведения в степень с помощью инструкций `f2xm1`, `fyl2x`, `fscale`.
22. Нарисовать график функции x в произвольной целой степени.
23. Реализовать программу отображения графиков функций с использованием арифметического сопроцессора.



24. Реализовать программу для RISC-процессора разбора арифметических выражений.

25. На языке ассемблера для процессоров ARM32 написать функцию, обрабатывающую массивы указателей.

26. На языке ассемблера для процессоров ARM32 вывести на экран файл, имя которого передано в качестве аргумента программы.


27. На языке ассемблера для процессоров ARM32 реализовать копирование файлов.

3.2.4 Темы заданий на экзамене

1. Система команд 16-ти разрядных процессоров архитектуры x86.
2. Система команд 32-х разрядных процессоров архитектуры x86.
3. Программный интерфейс DOS.
4. Программный интерфейс BIOS.
5. Ассемблер `masm`.

3.2.5 Примеры заданий на экзамене

1. Копирование файлов через интерфейс DOS.
2. Алгоритм рисования линии через видеобuffer.
3. Реализовать алгоритмы обработки данных (сортировка, поиск, контейнеры и др.) на языке ассемблера.
4. Реализовать на языке ассемблера стандартные функции работы со строками.
5. Реализовать арифметику больших чисел.
6. Реализовать замену символа в строке.
7. Реализовать замену строк.
8. Реализовать сортировку массива строк.
9. Реализовать копирование файла.
10. Реализовать замену строк в файле.
11. Реализовать разбиение строки на подстроки по подстроке.
12. Инвертирование массива чисел.
13. Сложение элементов массивов.
14. Поиск элементов массива.
15. Реализовать стандартные строковые функции с помощью строковых инструкций.
16. Реализовать динамические структуры данных и программы для их тестирования.

	МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Математический факультет Кафедра компьютерной безопасности и прикладной алгебры		
	Фонд оценочных средств по дисциплине «Язык Ассемблера» по специальности 10.05.01 Компьютерная безопасность специализации № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»		
Версия документа - 1	стр. 14	Первый экземпляр _____	КОПИЯ № _____

3.2.6 Примеры вопросов на экзамене

1. Программное окружение 16-ти разрядных процессоров архитектуры x86.
2. Система команд 16-ти разрядных процессоров архитектуры x86.
3. Программный интерфейс DOS.
4. Программный интерфейс BIOS.
5. Синтаксис ассемблера `masm`.
6. Макросредства ассемблера `masm`.

3.2.7 Темы заданий на зачёте

1. Программный интерфейс DOS.
2. Программный интерфейс BIOS.
3. Ассемблер `masm`.
4. Система команд процессоров архитектуры x86.
5. Система команд процессоров архитектуры IA-32.
6. Система команд процессоров архитектуры Intel 64.
7. Арифметический сопроцессор.
8. SIMD-расширения процессоров Intel.
9. WinAPI.
10. ABI операционных систем Windows NT и Linux.
11. Система команд процессоров архитектуры ARM.
12. Система команд процессоров архитектуры MIPS.
13. Система команд процессоров архитектуры AVR.
14. Работа с динамическими структурами данных на языке ассемблера.

3.2.8 Примеры заданий на зачёте


1. Реализовать алгоритмы обработки данных (сортировка, поиск, контейнеры и др.) на языке ассемблера.
2. Реализовать на языке ассемблера стандартные функции работы со строками.
3. Реализовать арифметику больших чисел.
4. Реализовать замену символа в строке.
5. Реализовать замену строк.
6. Реализовать сортировку массива строк.
7. Реализовать копирование файла.
8. Реализовать замену строк в файле.
9. Реализовать разбиение строки на подстроки по подстроке.
10. Инвертирование массива чисел.



11. Сложение элементов массивов.
12. Поиск элементов массива.
13. Реализовать стандартные строковые функции с помощью строковых инструкций.
14. Реализовать динамические структуры данных и программы для их тестирования.
15. Реализовать на языке ассемблера программы с графическим интерфейсом.
16. Реализовать вычисление интеграла через интегральные суммы.
17. Реализовать функцию вывода на экран в десятичном виде вещественного числа в 4- и 8-байтовом формате.
18. Реализовать функцию возведения в степень с помощью инструкций `f2xm1`, `fyl2x`, `fscale`.
19. Нарисовать график функции x в произвольной целой степени.
20. Реализовать программу отображения графиков функций с использованием арифметического сопроцессора.
21. Реализовать программу для RISC-процессора разбора арифметических выражений.
22. На языке ассемблера для процессоров ARM32 написать функцию, обрабатывающую массивы указателей.
23. На языке ассемблера для процессоров ARM32 вывести на экран файл, имя которого передано в качестве аргумента программы.
24. На языке ассемблера для процессоров ARM32 реализовать копирование файлов.

3.2.9 Примеры вопросов на зачёте

1. Программный интерфейс процессоров архитектуры IA-32.
2. Система команд процессоров архитектуры IA-32.
3. Программный интерфейс процессоров архитектуры Intel 64.
4. Система команд процессоров архитектуры Intel 64.
5. Программный интерфейс арифметического сопроцессора
6. Расширения MMX, SSE, AVX, AES процессоров Intel.
7. Реализация функций на языке ассемблера. Соглашения о вызовах.
8. Программный интерфейс процессоров архитектуры ARM.
9. Система команд процессоров архитектуры ARM.
10. Программный интерфейс процессоров архитектуры MIPS.
11. Система команд процессоров архитектуры MIPS.
12. Программный интерфейс процессоров архитектуры AVR.

	МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Математический факультет Кафедра компьютерной безопасности и прикладной алгебры		
	Фонд оценочных средств по дисциплине «Язык Ассемблера» по специальности 10.05.01 Компьютерная безопасность специализации № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»		
Версия документа - 1	стр. 16	Первый экземпляр _____	КОПИЯ № _____

13. Система команд процессоров архитектуры AVR.

3.2.10 Пример билета

1. Реализовать сортировку массива строк (10 баллов).
2. Система команд 16-ти разрядных процессоров архитектуры x86 (10 баллов).

4. ПОРЯДОК ПРОВЕДЕНИЯ И КРИТЕРИИ ОЦЕНИВАНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

4.1. Порядок проведения промежуточной аттестации

За своевременное и самостоятельно выполнение учебных работ в течение семестра студент получает рейтинговые баллы. Сумма за выполнение основных заданий в полном объеме - 100. Сверх этой суммы могут начисляться баллы за выполнение дополнительных заданий.


Основные баллы выставляются за выполнение объемных заданий (будем называть их зачётными), которые выполняются дома и сдаются в течение семестра. Сумма в 100 баллов делится между зачётными заданиями (не обязательно равномерно). При выдаче зачётного задания определено сколько баллов выставляется за реализацию определённой функциональности. Для зачётных заданий может быть определена функциональность повышенной сложности, за выполнение которой выставляются дополнительные баллы. Также дополнительные баллы могут быть выставлены по усмотрению преподавателя за особо примечательную реализацию.

По пройденному материалу выдаются небольшие задания для выполнения дома и/или во время семинарских занятий. За эти задания выставляются небольшие дополнительные баллы. Сдавать их можно либо в день выдачи либо на следующем занятии.

Пропуск по неуважительной причине одной пары влечет вычет 1 балла из итоговой суммы за семестр.

При нехватке баллов преподавателем может быть предоставлено дополнительное задание или возможность доделать задание, в котором была оценена не вся функциональность.

Итоговая оценка за дисциплину выставляется по результатам выполнения заданий текущего контроля. При необходимости во время зачёта и экзамена может быть предоставлена возможность получить дополнительные баллы (не более 20), выполнив дополнительные задания и

	МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Математический факультет Кафедра компьютерной безопасности и прикладной алгебры		
	Фонд оценочных средств по дисциплине «Язык Ассемблера» по специальности 10.05.01 Компьютерная безопасность специализации № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»		
Версия документа - 1	стр. 17	Первый экземпляр _____	КОПИЯ № _____

ответив (в устной форме) на вопросы.

4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств.

4.2.1 Критерии оценивания зачётных заданий

Выполнение заданий предполагает некоторую программную реализацию, к которой предъявляются обычные требования по качеству кода. Код должен быть удобочитаемым, хорошо структурированным, расширяемым, удобным в сопровождении, написанным в едином стиле. Должна быть проведена функциональная декомпозиция (на подпрограммы, модули, пакеты и т.д.), реализованы необходимые программные абстракции. Реализации алгоритмов должны быть логичными и понятными. Иначе возможна сбавка до 5 баллов. За программные ошибки, приводящие к неработоспособности кода для некоторых возможных случаев, возможна сбавка до 10 баллов (в зависимости от критичности ошибки). За программные ошибки, приводящие к аварийному некорректному завершению программы, возможна сбавка до 10 баллов (в зависимости от критичности ошибки).


При сдаче зачётного задания производится опрос по техническим деталям реализации и по теории, используемой при выполнении заданий. Неудовлетворительный ответ будет означать несамостоятельность выполнения задания, что влечёт выставление 0 баллов за соответствующую функциональность. Если для одного задания это повторяется более 2 раз, то за всё задание выставляется 0 баллов без возможности повторной сдачи.

4.2.2 Критерии оценивания домашних и аудиторных заданий

Домашние и аудиторные задания – это небольшие задания, за которые обычно выставляется 1-2 балла. Они оцениваются атомарно: либо задание выполнено (выставляется указанное при выдаче задания количество баллов), либо не выполнено (0 баллов).

4.2.3 Критерии оценивания заданий на зачёте и экзамене


Дополнительные задания, выдаваемые на зачёте и экзамене являются относительно объёмными, за них выставляется до 10-15 баллов. Поэтому к ним применимы описанные выше критерии оценивания зачётных заданий с соответствующей корректировкой баллов: сбавка за некорректную работу до 5 баллов, за аварийное завершение – до 5 баллов.

 МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Математический факультет Кафедра компьютерной безопасности и прикладной алгебры			
Фонд оценочных средств по дисциплине «Языки Ассемблера» по специальности 10.05.01 Компьютерная безопасность специализации № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»			
Версия документа - 1	стр. 18	Первый экземпляр _____	КОПИЯ № ____

4.2.4 Критерии оценивания ответа в устной форме

Ответ оценивается по трём параметрам 1) построение ответа (структура ответа, грамотность речи, последовательность и т.д.); 2) фактическая полнота ответа; 3) собственный анализ излагаемого материала его оценка в контексте взаимодействия с другими областями, умение применять на практике.

1	Студент самостоятельно правильно выстраивает структуру ответа, изложение последовательное, речь грамотная без оговорок.	2
	Изложение студента непоследовательное и обрывочное, взаимосвязи частей ответа не всегда прослеживаются. Раскрытие сути ответа невозможно без уточняющих вопросов.	1
	Студент испытывает существенные трудности при самостоятельном построении ответа, способен только давать краткие ответы на конкретные вопросы.	0
2	Студент правильно ответил на теоретический вопрос билета. Показал отличные знания в рамках усвоенного учебного материала. Ответил на все дополнительные вопросы.	4
	Студент ответил на теоретический вопрос билета с небольшими неточностями. Показал хорошие знания в рамках усвоенного учебного материала. Ответил на большинство дополнительных вопросов.	3
	Студент ответил на теоретический вопрос билета с существенными неточностями. Показал удовлетворительные знания в рамках усвоенного учебного материала. При ответах на дополнительные вопросы было допущено много неточностей.	2
	При ответе на теоретический вопрос билета студент продемонстрировал недостаточный уровень знаний. При ответах на дополнительные вопросы было допущено множество неправильных ответов.	1
	Студент не ответил на вопрос.	0
3	Студент ясно осознаёт место излагаемого материала в общей структуре профессионального знания, знает стандартные примеры использования и предлагает свои, даёт собственные компетентные оценки.	4
	Студент осознаёт взаимосвязи и знает стандартные примеры использования. Допускает неточности при самостоятельном анализе.	3
	Студент в общих чертах осознаёт взаимосвязи и знает стандартные примеры использования. При проведении самостоятельного анализа нуждается в уточняющих вопросах, при этом допускает существенные неточности.	2

	МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Математический факультет Кафедра компьютерной безопасности и прикладной алгебры		
	Фонд оценочных средств по дисциплине «Языки Ассемблера» по специальности 10.05.01 Компьютерная безопасность специализации № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»		
Версия документа - 1	стр. 19	Первый экземпляр _____	КОПИЯ № _____

	Студент знает основные стандартные примеры использования. Не осознаёт взаимосвязей с другими областями.	1
	Студент не осознаёт взаимосвязей и практическое приложение излагаемого материала.	0

4.3. Результаты промежуточной аттестации и уровни сформированности компетенций

Баллы, полученные за выполнение заданий текущего контроля и промежуточной аттестации, переводятся в оценки за экзамен и зачет следующим образом:

- 0-60 баллов – неудовлетворительно (2), не зачтено;
- 61-75 баллов – удовлетворительно (3), зачтено;
- 76-90 баллов – хорошо (4), зачтено;
- более 90 – отлично (5), зачтено.

Особенности проведения процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья обозначены в рабочей программе дисциплины (модуля).

Уровни сформированности компетенций определяются следующим образом:

Оценка	Отлично/ зачтено	Хорошо/ зачтено	Удовлетворительно/ зачтено	Неудовлетворительно/ не зачтено
Баллы	более 90 баллов	76-90 баллов	61-75 баллов	0-60 баллов
Уровень освоения проверяемых компетенций	высокий	средний	базовый	недостаточный

Уровни сформированности компетенций определяется следующим образом:

1. Высокий уровень сформированности компетенций соответствует оценке отлично:
 - предполагает формирование компетенций на высоком уровне, готовность к самостоятельной профессиональной деятельности;
 - студент способен заниматься разработкой web-приложений.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра компьютерной безопасности и прикладной алгебры

Фонд оценочных средств по дисциплине «Язык Ассемблера»
по специальности 10.05.01 Компьютерная безопасность
специализации № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»

Версия документа - 1

стр. 20

Первый экземпляр _____

КОПИЯ № _____

2. Средний уровень соответствует оценке хорошо:
 - предполагает формирование компетенций на более высоком уровне: формируется комплексное знание основ web-программирования;
 - студент способен давать развернутые ответы на теоретические вопросы дисциплины на уровне не ниже оценки «хорошо».
3. Базовый уровень соответствует оценке удовлетворительно:
 - предполагает формирование компетенций на начальном уровне: знание основных положений web-программирования;
 - студент способен давать ответы на теоретические и практические вопросы дисциплины на уровне не ниже оценки «удовлетворительно».
4. Низкий уровень соответствует оценке неудовлетворительно.

