

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: АРЦЮХ ДМИТРИЙ ВЛАДИМИРОВИЧ
Должность: Директор Организации
Дата подписания: 01.05.2026 17:38:59
Уникальный программный ключ:
194e9de362a3e118beb1b1af4bc7c1577477a952

**Автономная некоммерческая организация
дополнительного профессионального образования
«Академия Информационных Технологий»**

ОГРН: 1230600003457, ИНН/КПП: 0600010064/060001001
Российская Федерация, 386001, Республика Ингушетия, городской округ Магас,
город Магас, улица Н.С. Хрущева, дом 10

УТВЕРЖДЕНО
Приказом № 1-ОБР от «12» апреля 2024г.

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«Основы IT-инфраструктуры»**

Магас 2024

1. Пояснительная записка

Программа «Основы IT-инфраструктуры» направлена на формирование у слушателей базовых инженерных и программных компетенций, необходимых для понимания устройства и работы современной IT-инфраструктуры. Это практико-ориентированный курс, в котором участники последовательно осваивают ключевые инструменты и технологии, лежащие в основе разработки и эксплуатации программного обеспечения: от работы с операционной системой Linux и системой контроля версий Git до основ программирования на языке C, включая управление памятью, структуры данных и файловый ввод/вывод.

В рамках курса слушатель:

получает целостное представление об экосистеме разработки программного обеспечения и принципах функционирования IT-инфраструктуры;

осваивает базовые инструменты, используемые в системном и прикладном программировании;

формирует фундамент для дальнейшего изучения низкоуровневого программирования и инженерных дисциплин.

Уровень освоения программы: базовый.

Направленность (профиль) программы – техническая

Настоящая программа реализуется в очно-заочной форме исключительно с использованием дистанционных образовательных технологий.

Формат реализации асинхронный. Используются образовательные видеоролики, текстовые уроки, ссылки на дополнительные материалы, практические упражнения и подробные технические инструкции.

Платформа для обучения: GetCourse.

Технические требования для слушателя:

ПК/ноутбук с доступом в Интернет

Веб-браузер (Google Chrome, Яндекс.Браузер).

Программы для просмотра PDF-файлов и работы с офисными документами.

Онлайн-компилятор языка Си

Командный ubuntu-emulator

Условия обучения:

Доступ к учебным материалам, заданиям и тестам предоставляется в личном кабинете на платформе GetCourse 24/7.

1.1. Актуальность Программы

Актуальность программы обусловлена стремительным развитием IT-

отрасли и растущей потребностью в квалифицированных инженерных кадрах, способных работать с низкоуровневыми компонентами программного обеспечения и инфраструктурными решениями. Начинающие разработчики часто испытывают трудности при переходе от изучения синтаксиса языков высокого уровня к пониманию системных процессов, управлению памятью и работе в среде командной строки. Это создает барьер для входа в такие востребованные направления, как системное программирование, встраиваемые системы, DevOPS и администрирование.

Программа «Основы IT-инфраструктуры» отвечает на этот запрос, предоставляя структурированный практический маршрут освоения ключевых инженерных инструментов и языков. Курс закладывает прочный фундамент, позволяя слушателям сформировать целостное понимание экосистемы разработки и приобрести навыки, непосредственно применимые в реальных проектах, тем самым сокращая путь от теоретического обучения к полноценной инженерной практике.

1.2. Категории обучающихся

Взрослые (от 18 лет). Программа рассчитана на начинающих специалистов, планирующих карьеру в сфере информационных технологий: выпускников вузов и курсов по программированию, веб-разработке, аналитике данных, дизайну и смежным IT-направлениям, а также на специалистов из других отраслей, желающих перейти в IT.

1.3. Цель и задачи программы

Цель Программы – формирование у слушателей инженерных компетенций и практических навыков работы с базовыми инструментами IT-инфраструктуры, включая операционную систему Linux, систему контроля версий Git и программирование на языке C, для дальнейшего профессионального развития в сфере информационных технологий.

Реализация поставленной цели предусматривает решение ряда задач.

Задачи Программы

Сформировать у слушателей целостное понимание экосистемы разработки программного обеспечения, роли операционных систем и принципов функционирования IT-инфраструктуры.

Познакомить с операционной системой Linux: освоить работу с файловой системой, терминалом (Bash), базовыми командами, управлением процессами и установкой программного обеспечения.

Обучить использованию системы контроля версий Git для индивидуальной и командной работы: научить создавать репозитории,

управлять версиями, работать с ветками и удаленными репозиториями на GitHub.

Заложить основы программирования на языке C: освоить синтаксис, типы данных, управляющие конструкции (условия, циклы), функции, а также принципы компиляции с использованием GCC.

Сформировать базовые навыки работы с памятью и указателями в языке C, необходимые для понимания низкоуровневых процессов и эффективного управления ресурсами.

Дать обзорное представление о работе с массивами, строками, структурами данных и файловым вводом/выводом в языке C для формирования насмотренности и подготовки к изучению более сложных тем.

1.4. Сроки реализации

Срок реализации: 3 недели

Трудоемкость: 82 часа

1.5. Планируемые результаты обучения

Навыки:

Работа с операционной системой Linux: навигация по файловой системе, управление файлами и директориями через терминал, выполнение базовых команд (mkdir, ls, cd, sudo, apt, whoami, history, touch, find).

Использование текстовых редакторов (Vim, Nano) для создания и редактирования файлов, написание простых Bash-скриптов.

Управление процессами в Linux: запуск, остановка и мониторинг ресурсов системы.

Работа с системой контроля версий Git: инициализация репозитория, создание коммитов, управление ветками, синхронизация с удаленным репозиторием на GitHub.

Написание, компиляция и запуск программ на языке C с использованием компилятора GCC.

Применение базовых конструкций языка C: переменные, типы данных, арифметические и логические операторы, условные конструкции (if/else, switch) и циклы (for, while, do-while).

Работа с указателями и адресной арифметикой для управления памятью.

Чтение и запись данных, обработка массивов, строк и структур в языке C.

Выполнение операций с текстовыми и бинарными файлами на языке C.

Знания:

История развития, архитектура и область применения ОС Linux, понятие

дистрибутивов и их разновидности.

Структура файловой системы Linux, механизмы прав доступа к файлам и директориям.

Назначение и основные команды командного интерпретатора Bash.

Принципы работы систем контроля версий, различия между Git и GitHub, основные команды Git.

Базовый синтаксис языка программирования C, структура программы, назначение библиотек (stdio.h и др.).

Типы данных в C, их размеры и особенности применения.

Процесс компиляции: этапы преобразования исходного кода в исполняемый файл, роль компилятора GCC.

Понятие указателей и их связь с адресами памяти, основы адресной арифметики.

Принципы работы с массивами, строками и пользовательскими структурами данных (struct).

Особенности работы с текстовыми и бинарными файлами в языке C, режимы открытия файлов.

Совершенствуемые компетенции:

Способность работать в среде командной строки и эффективно использовать терминал для решения инженерных задач.

Понимание низкоуровневых процессов управления памятью и данными при разработке программного обеспечения.

Навыки системного мышления: способность видеть взаимосвязи между компонентами IT-инфраструктуры (ОС, система контроля версий, компиляция, исполнение).

Готовность к использованию профессиональных инструментов разработки, принятых в индустрии (Linux, Git, GCC).

Способность читать и анализировать чужой код, работать с технической документацией.

Базовые навыки отладки и тестирования программ, написанных на языке C.

Фундаментальная подготовка для дальнейшего изучения системного программирования, встраиваемых систем, DevOps и смежных инженерных дисциплин.

1.6. Календарный учебный график

Количество учебных недель – 3.

Количество учебных дней – 82.

Режим занятий: до 28 часов в неделю.

Программа реализуется исключительно с применением ЭО и ДОТ и календарный учебный график формируется по форме календарного рейтинг-плана курса (маршрута обучения), размещаемого в системе дистанционного обучения GetCourse.

2.Содержание программы

2.1. Учебный план

№п/п	Наименование разделов/дисциплин/модулей	Общая трудоемкость	Всего аудиторных часов	в т.ч		Самостоятельная работа	Форма промежуточной аттестации/текущего контроля (при наличии)
				Лекции/теория	Практические занятия		
1	Модуль 1: Основы IT-инфраструктуры	80	68	46	22	12	Тестирование
	Промежуточная аттестация	2					Экзамен
	Итоговая аттестация	0					отсутствует
	Итого часов	82	68	46	22	12	

2.2. Учебно-тематический план

Наименование компонентов программы	Содержание учебного материала и формы организации деятельности слушателей/Рабочие программы модулей	Всего (час.)
<i>Модуль 1. Основы IT-инфраструктуры</i>		82
<i>Тема 1.1. Введение в Linux.</i>	<p><i>Содержание теоретических занятий</i></p> <p>1. Знакомство с ОС Linux. Дистрибутивы. Интерфейс Ubuntu Дистрибутивы: что это такое, какие бывают. Ubuntu и MINT. Интерфейсы для дистрибутивов. Что такое эмулятор. Знакомство с интерфейсом Ubuntu. Как не потеряться в системе. Как подключиться к интернету, открыть проводник, настройки и т.д. Где какие кнопки расположены. Как открыть приложения.</p> <p>2. Работа с файловой системой Понимание структуры файловой системы Linux. Работа с правами доступа к файлам и директориям. Поиск файлов и директорий. Куда сохраняются файлы, где устанавливаются программы.</p> <p>3. Терминал / Bash. Знакомство с командной строкой. Основа работы с Linux. Что такое терминал и зачем он нужен. Bash - командный интерпретатор. Как открыть терминал. Базовые команды в терминале mkdir, ls, cd, sudo, apt install, upgrade, whoami, history, touch, nano, vim, micro, find path -name «name» Работа с файловой системой через терминал. Создание, копирование, перемещение и удаление файлов и директорий.</p> <p>4. Текстовые редакторы. Скрипты Основы работы с Vim и Nano. Создание и редактирование текстовых файлов. Введение в написание простых Bash-скриптов. Переменные, условия, циклы.</p> <p>5. Работа с процессами Запуск, остановка и управление процессами. Мониторинг ресурсов и процессов.</p>	4
	<p><i>Содержание практических занятий</i></p> <p>1. Знакомство с ОС Linux. Дистрибутивы. Интерфейс Ubuntu Запуск эмулятора. Визуальное знакомство с ОС. Развитие базовых навыков работы с Linux</p> <p>2. Магазин Ubuntu Работа с приложениями Ubuntu. Как открыть, найти и скачать приложения (пакеты).</p> <p>3. Работа с файловой системой Создание директорий, операции с ними. Создание файлов.</p>	8

	<p>4. Текстовые редакторы. Скрипты Запуск текстовых редакторов. Освоение простейших операций.</p> <p>5. Работа с процессами Просмотр процессов. Управление процессами.</p>	
	<p><i>СРС</i> Файловая система Linux, работа с простейшими операциями над файлами.</p>	2
<p><i>Тема 1.2.</i> <i>Системы контроля версий Git</i></p>	<p><i>Содержание теоретических занятий</i> 1. Установка Git Git и GitHub: в чем разница. Установка Git. 2. Знакомство с терминалом. Знакомство с терминалом. Директории: создание, перемещение, удаление. Схожесть с Bash. 3. Основы Git. Git Cheat Sheet. Работа с командами (init, add, commit) Коммит. Создание репозитория. Создание файла. Сохранение файла в локальном репозитории. 4. GitHub Регистрация на GitHub. Создание удаленного репозитория. Команды (remote add, push) 5. Ветки Git Что такое ветка, для чего она используется, как её создать и перейти. Pull request 6. Графический интерфейс Git GUI - интерфейс Git.</p>	10
	<p><i>Содержание практических занятий</i> 1. Знакомство с терминалом. Работа в локальном репозитории. Разместить его в сети.</p>	2
	<p><i>СРС</i> Обзорное знакомство слушателя с системой контроля версий Git и возможностями командной работы.</p>	2
<p><i>Тема 1.3.</i> <i>Введение в язык C</i></p>	<p><i>Содержание теоретических занятий</i> 1. Структура программы. Основные компоненты программы на языке C. Синтаксис языка Си. Библиотека. Подключение библиотеки <stdio.h> для ввода/вывода данных. Код начинающего: Привет, мир! Среда разработки (блокнот). Компиляция языка в консоли. Онлайн-компилятор. 2. Функция вывода данных (printf). Общая форма записи функции printf(). Управляющие символы. Форматирование с помощью функции printf 3. Сборка и запуск программ (GCC).</p>	18

	<p>Процесс компиляции и выполнения программ на языке C. Компилятор GCC: установка, запуск, знакомство. Набор компиляторов. Исполняемые файлы. Команды (gcc main.c -o program)</p> <p>4. Типы данных Основные типы данных и их использование. Int, float, double, char.</p> <p>5. Переменные. Типы данных в языке Си. Переменные. Объявление переменных.</p> <p>6. Функция ввода данных (scanf) Функция форматированного ввода данных с клавиатуры scanf()</p> <p>7. Арифметические операторы Основные арифметические операторы языка Си. + оператор сложения - оператор вычитания * оператор умножения % оператор взятия остатка от деления / оператор деления</p> <p>8. Операторы сравнения Равно. Не равно. Больше чем, меньше чем. Больше либо равно, меньше либо равно.</p> <p>9. Логические операторы Логические операторы И/или</p> <p>10. Условные операторы. Последовательность выполнения команд в программе, условные операторы (if else, switch).</p> <p>11. Цикл For Использование цикла for</p> <p>12. Цикл While Циклы while, do while, break-continue.</p> <p>13. Введение в функции языка Си Что такое функция, ее типы. Тело функции. Вызов функции. Возвращаемое значение. Аргументы функции. Системные и пользовательские функции. Рекурсия.</p>	
	<p><i>Содержание практических занятий</i></p> <p>1. Структура программы. Основные компоненты программы на языке C. Запуск онлайн-компилятора. Отработка первых строк кода. Знакомство с синтаксисом.</p> <p>2. Функция вывода данных (printf). Тестирование. Практическая работа в онлайн-компиляторе.</p>	8
	<p><i>СРС</i></p> <p>Запуск онлайн-компилятора. Визуальное знакомство с синтаксисом языка. Выполнение практических заданий.</p>	2
<p>Тема 1.4. Указатель и работа с памятью</p>	<p><i>Содержание теоретических занятий</i></p> <p>1. Понятие указателя. Определение и использование указателей. Память компьютера. Определение указателя. Использование указателей. Разыменование. Операция</p>	4

	<p>получения адреса. Работа с адресами. 2. Адресная арифметика Арифметические действия над указателями. Сравнение. Увеличение/ уменьшение. Сложение, вычитание, индексация. Отличие от арифметических выражений. 3. Массивы Массив: что такое и каких типов бывают. Одномерные/двумерные/многомерные Объявление, инициализация и использование массивов. 4. Алгоритмы с массивами Поиск, сортировка и другие алгоритмы. Сортировка пузырьком.</p>	
	<p><i>Содержание практических занятий</i> 1. Понятие указателя. Определение и использование указателей. Работа в компиляторе.</p>	4
	<p><i>СРС</i> Обзорное знакомство с операциями, которые могут быть применены к массивам.</p>	2
<p><i>Тема 1.5. Работа с текстом и пользовательскими структурами данных</i></p>	<p><i>Содержание теоретических занятий</i> 1. Символы. Работа с символьными данными. Представление символов в Си. Кодовые таблицы. Таблица ASCII. Управляющие символы 2. Строки. Обработка строковых данных. Строка. Строковая константа. Связь строк с указателями, с массивами. Оператор sizeof. 3. Структуры. Определение и использование пользовательских структур данных. Что такое структура. Поля структуры. Struct. Как создавать структуры, объявлять их. Как получать доступ к полям структуры. Определение и использование пользовательских структур данных.</p>	6
	<p><i>Содержание практических занятий</i></p>	0
	<p><i>СРС</i> Представление символов в языке Си, кодовые таблицы, такие как ASCII.</p>	2
<p><i>Тема 1.6. Работа с файлами</i></p>	<p><i>Содержание теоретических занятий</i> 1. Текстовые файлы Тип данных для работы с файлами. Тип File. Как открывать/закрывать файлы. Режимы работы с файлами. Операции с текстовыми файлами: ввод/вывод данных из файла. 2. Бинарные файлы. Работа с бинарными файлами. Что такое бинарные файлы. Чтение и запись бинарных файлов. Функции для работы</p>	4
	<p><i>Содержание практических занятий</i></p>	0

	<i>СРС</i> Знакомство с особенностями работы с файлами в языке Си.	2
Промежуточная аттестация	<i>Экзамен</i>	2

Промежуточная аттестация – экзамен. **Итоговая аттестация** - не предусмотрена

2.3. Оценочные материалы

Промежуточная аттестация проводится в форме экзамена по модулю в виде выполнения практического задания по пройденным темам. Для успешного прохождения ПА слушатель выполняет по одной индивидуальной работе по каждому из модулей и размещает полный комплект материалов в ЛМС, согласно утверждённым критериям. Формат демонстрации не предусмотрен.

Критерии оценивания промежуточной аттестации:

- соответствие заданию модуля и полнота выполнения всех пунктов;
- корректность решений и валидность результатов;
- качество оформления;
- соблюдение академической добросовестности (оригинальность, корректные заимствования, отсутствие совпадений, не объяснённых ссылками);
- соблюдение сроков сдачи в ЛМС.

Шкалы оценивания:

Баллы/Критерии	соответствие заданию модуля и полнота выполнения всех пунктов	корректность решений и валидность результатов;	качество оформления;	соблюдение академической добросовестности	соблюдение сроков сдачи в ЛМС.
5	Выполнены все пункты задания модуля в полном объёме, нет пропусков, работа полностью соответствует заданию	Решения технически/логически верны, результаты воспроизводимы и обоснованы, обработка ошибок корректна	Структурировано, читаемо, единый стиль, код/текст отформатированы, есть комментарии/пояснения, файлы названы логично, материалы упакованы аккуратно	Работа полностью оригинальна (или все заимствования явно указаны ссылками), нет совпадений с другими работами, нет плагиата	--
4	Выполнены все основные пункты, но отсутствует 1 незначительный элемент / незначительная часть	Решения в целом верны, но есть 1–2 незначительные ошибки / неточности, не влияющие на основной результат	Хорошее оформление, но есть мелкие недочёты: местами нарушен стиль, не все файлы подписаны, отсутствуют отдельные комментарии	Единичные совпадения, объяснённые корректными ссылками; общая оригинальность высокая (>80%)	--
3	Выполнено 70–80% задания, отсутствуют 2–3 пункта или один значительный блок	Решения работают, но есть ошибки в крайних случаях / не обработаны исключения / результаты частично неверны	Среднее качество: структура есть, но оформление «грязное» (лишние файлы, разный стиль, мало комментариев)	Есть заимствования без ссылок (10–20% работы), или незначительные совпадения с другими слушателями	Работа сдана строго в установленный срок (включая все материалы)
2	Выполнено 50–69% задания, отсутствует более 3 пунктов или ключевая	Решения содержат системные ошибки, результаты	Оформление плохое: трудно читать, нет структуры, неясно, что к	Значительные заимствования без ссылок (20–40%), или заметные	Незначительная задержка (до 3 календарных дней)

	часть	невалидны или не воспроизводятся	чему относится	совпадения с другими работами	
1	Выполнено менее 50% задания, работа не соответствует заданию	Решения не работают, результаты отсутствуют или противоречивы	Оформление отсутствует (набор файлов без пояснений, нечитаемый код/текст)	Более 40% работы скопировано (без ссылок), или работа несамостоятельна	Задержка от 4 до 7 календарных дней
0	Работа не сдана или не относится к модулю	Решения не представлены или полностью неверны	Материалы не загружены в ЛМС или загружены с нарушением формата (не открываются)	Полный плагиат (работа целиком не своя, или копия другой работы)	Задержка более 7 календарных дней или работа не сдана

Итоговая шкала оценивания

Отлично	21–23 балла
Хорошо	17–20 баллов
Удовлетворительно	14–16 баллов
Неудовлетворительно	менее 14 баллов

Слушателям, получившие по результатам промежуточной аттестации оценку «неудовлетворительно» устанавливаются сроки повторной промежуточной аттестации. Если обучающийся не ликвидировал академическую задолженность при прохождении повторной промежуточной аттестации в первый раз, ему предоставляется возможность пройти повторную промежуточную аттестацию во второй раз.

Первая повторная промежуточная аттестация и вторая повторная промежуточная аттестация проводятся комиссией. Состав комиссии формируется департаментом образовательных программ.

Задание 1. Основы Linux и работа с файловой системой

В домашней директории создайте структуру каталогов и выполните операции с файлами согласно списку. Результат оформите в виде отчёта с командами и скриншотами.

Список действий:

Создать структуру: ~/exam/projects/, ~/exam/backups/, ~/exam/temp/

В projects создать файлы: readme.txt, notes.txt, todo.txt

Скопировать todo.txt в backups

Переместить notes.txt в temp

Установить на readme.txt права 640

Найти все .txt файлы в домашней директории

Эталон выполнения

Отчёт со всеми командами (например, mkdir, touch, cp, mv, chmod, find)

Скриншот вывода ls -la ~/exam

Скриншот вывода find ~ -name "*.txt" 2>/dev/null

Критерии оценки

Что проверяется	Балл
-----------------	------

Все каталоги созданы	1
Все файлы созданы	1
Копирование и перемещение выполнены верно	1
Права 640 установлены	1
Поиск файлов выполнен	1
Итого	5

Задание 2. Bash-скрипт для резервного копирования

Напишите Bash-скрипт `backup.sh`, который принимает два аргумента: исходную директорию и директорию для резервного копирования.

Требования к скрипту:

Проверить, существует ли исходная директория. Если нет — вывести ошибку.

Создать директорию для резервного копирования, если её нет.

Скопировать все файлы с расширением `.txt` из исходной директории в целевую.

При копировании добавить к имени файла текущую дату в формате ГГГГ-ММ-ДД.

Вывести количество скопированных файлов.

Эталон выполнения

Текст скрипта (код)

Скриншот выполнения с примером

Скриншот результата (файлы с датой в имени)

Критерии оценки

Что проверяется	Балл
Проверка исходной директории	1
Создание целевой директории	1
Копирование только <code>.txt</code> файлов	1
Добавление даты к имени файла	1
Подсчёт и вывод количества файлов	1
Итого	5

Задание 3. Git: локальный и удалённый репозиторий

Выполните следующие действия с Git:

Настроить Git (имя и email пользователя)

Создать локальный репозиторий в папке `~/git-exam`

Создать файл `hello.txt` с текстом «Hello, Git!» и сделать первый коммит

Создать ветку `feature`, переключиться на неё

Добавить в `hello.txt` новую строку «Feature branch work» и сделать коммит

Переключиться обратно в ветку `main`

Создать репозиторий на GitHub

Связать локальный репозиторий с удалённым

Отправить ветку main на GitHub

Эталон выполнения

Отчёт со всеми командами

Скриншот вывода `git log --oneline`

Скриншот репозитория на GitHub с коммитом

Ссылка на репозиторий (или скриншот)

Критерии оценки

Что проверяется	Балл
Настройка Git выполнена	1
Локальный репозиторий и первый коммит	1
Ветка feature и коммит в ней	1
Удалённый репозиторий создан и связан	1
Код отправлен (push) на GitHub	1
Итого	5

Задание 4. Программа на C: «Угадай число»

Напишите программу на C, которая реализует игру «Угадай число».

Требования:

Программа загадывает случайное число от 1 до 100

Пользователь вводит числа, программа даёт подсказки: «Больше», «Меньше», «Угадал!»

У пользователя не более 10 попыток

После окончания игры программа спрашивает: «Сыграем ещё? y/n»

При вводе y — новая игра, при n — выход

Эталон выполнения

Исходный код программы (файл guess.c)

Скриншот компиляции (`gcc guess.c -o guess`)

Скриншот выполнения программы (с примером игры)

Критерии оценки

Что проверяется	Балл
Программа компилируется без ошибок	1
Случайное число генерируется	1
Подсказки «Больше»/«Меньше» работают	1
Ограничение на 10 попыток	1
Возможность сыграть ещё	1
Итого	5

Задание 5. Массивы и функции на C

Напишите программу на C, которая работает с массивом целых чисел.

Требования:

Запросить у пользователя размер массива N (не более 20)
 Заполнить массив числами, введёнными с клавиатуры
 Вывести исходный массив на экран
 Найти и вывести минимальный, максимальный элементы и сумму всех элементов (каждое вычисление в отдельной функции)
 Отсортировать массив по возрастанию методом «пузырька»
 Вывести отсортированный массив
 Подсчитать и вывести количество чётных чисел

Эталон выполнения

Исходный код программы (файл array.c)

Скриншот компиляции и выполнения (например, для N=5)

Критерии оценки

Что проверяется	Балл
Ввод N и заполнение массива	1
Функции минимума, максимума, суммы	1
Сортировка «пузырьком»	1
Вывод исходного и отсортированного массивов	1
Подсчёт чётных чисел	1
Итого	5

Задание 6. Работа с файлами на C

Напишите программу file_stats.c, которая анализирует текстовый файл.

Требования:

Программа принимает аргументы командной строки: входной файл и выходной файл

Проверяет, что передано 2 аргумента, иначе выводит инструкцию

Открывает входной файл для чтения, если файл не существует — выводит ошибку

Подсчитывает во входном файле:

количество строк

количество слов (разделители: пробел, табуляция, перевод строки)

количество символов (без учёта перевода строки)

Записывает результаты в выходной файл в формате:

text

Количество строк: X

Количество слов: Y

Количество символов: Z

Закрывает оба файла

Эталон выполнения

Исходный код программы (файл file_stats.c)

Скриншот компиляции

Пример файла input.txt
 Скриншот выполнения программы
 Полученный файл output.txt (содержимое)
 Критерии оценки

Что проверяется	Балл
Проверка аргументов командной строки	1
Проверка существования входного файла	1
Корректный подсчёт строк, слов, символов	1
Запись результатов в выходной файл	1
Программа компилируется и работает	1
Итого	5

3. Организационно-педагогические условия реализации программы

3.1. Материально-технические условия реализации программы

Для осуществления образовательного процесса по программе в дистанционном асинхронном формате необходима следующая материально-техническая база:

Оборудованная платформа дистанционного обучения (LMS), обеспечивающая доступ к учебным материалам, выполнение практических заданий, взаимодействие с преподавателем и отслеживание прогресса.

3.2. Учебно-методическое и информационное обеспечение программы

Для успешного освоения программы необходим комплект методических пособий, включающий:

- пошаговые инструкции к практическим работам по Linux, Git, языку C;
- чек-листы для самопроверки по каждой теме (команды терминала, действия с Git, синтаксис C);
- шаблоны для оформления отчётов о выполнении практических заданий;
- глоссарий основных терминов (Bash, репозиторий, указатель, динамическая память, компиляция и др.);
- набор тестовых файлов и сценариев для отработки навыков работы с файловой системой и файлами в C.

Программное обеспечение и информационные ресурсы

Для полноценного обучения слушателю необходимо использовать следующее ПО и онлайн-сервисы:

Операционная система	Windows 10/11, macOS (любая версия) или Linux (Ubuntu 22.04 /
----------------------	---

	Fedora / Mint)
Эмулятор терминала Linux	Ubuntu-emulator (онлайн) или установка Linux на виртуальную машину (VirtualBox + образ Ubuntu)
Редактор кода	Visual Studio Code, Sublime Text, Geany или любой другой с подсветкой синтаксиса C
Компилятор C	GCC (MinGW для Windows, встроен в Linux/macOS); онлайн-компиляторы (onlinegdb.com , replit.com)
Система контроля версий	Git (официальный дистрибутив), GitHub-аккаунт (регистрация бесплатна)
Работа с виртуальными машинами	VirtualBox, VMware Workstation Player (для установки полноценной ОС Linux)
Текстовые редакторы в Linux	Nano, Vim (предустановлены в большинстве дистрибутивов)
Дополнительные утилиты	make, gdb (отладчик), valgrind (проверка утечек памяти) – устанавливаются через пакетный менеджер
Платформа обучения	GetCourse (личный кабинет, доступ к видеоурокам, тестам, форуму)

3.3. Литература

3.3.1. Основная литература

Шоттс У. – Командная строка Linux. Полное руководство, 2-е изд., БХВ-Петербург.

Барретт Д. Дж. – Linux. Командная строка. Лучшие практики.

Чакон С., Штрауб Б. – Git для профессионального программиста (доступна онлайн бесплатно).

Керниган Б., Ритчи Д. – Язык программирования C (K&R), Вильямс, 2009.

Шилдт Г. – Полный справочник по C.

3.3.2. Дополнительная литература

Бхаргава А. – Грокаем алгоритмы, Питер, 2022.

Седжвик Р. – Фундаментальные алгоритмы на C, ДиаСофтЮП, 2003.

Мартин Р. – Чистый код: создание, анализ и рефакторинг.

Мартин Р. – Идеальный программист. Как стать профессионалом разработки ПО.

3.3.3. Интернет-ресурсы и профессиональные платформы

Официальная документация Bash (GNU)

<https://www.gnu.org/software/bash/manual/>

Официальная документация Git (Pro Git Book)

<https://git-scm.com/book/ru/v2>

Официальная документация GCC (GNU Compiler Collection)

<https://gcc.gnu.org/onlinedocs/>

Справочник по языку C (cppreference.com)

<https://ru.cppreference.com/w/c>

Официальный сайт Git (справочник команд)

<https://git-scm.com/docs>

Онлайн-компилятор C (onlinegdb.com)

<https://www.onlinegdb.com/>

3.4. Кадровое обеспечение программы

№	Модуль/Дисциплина в соответствии с УП	ФИО преподавателя	Уч. степень/Звание	Должность
	Модуль 1: Введение в Linux.	Артеменко Дмитрий	--	ООО «НПП «АИК» Програмист C++
	Модуль 2: Системы контроля версий Git	Зафархужаев Саидазизхон	--	АИКБ "Ипак Йули" Системный администратор начального уровня
	Модуль 3: Введение в язык Си	Артеменко Дмитрий	--	ООО «НПП «АИК» Програмист C++
	Модуль 4: Указатель и работа с памятью	Зафархужаев Саидазизхон	--	АИКБ "Ипак Йули" Системный администратор начального уровня
	Модуль 5: Работа с текстом и пользовательскими структурами данных	Артеменко Дмитрий	--	ООО «НПП «АИК» Програмист C++
	Модуль 6: Работа с файлами	Артеменко Дмитрий	--	ООО «НПП «АИК» Програмист C++