Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий  
Кафедра «Информатика и информационные технологии»

Направление подготовки/специальность: Программирование электронных устройств и систем

ОТЧЕТ

по проектной практике

Студенты:

Саранчин Александр Павлович, группа 241-329

Еськин Мирослав Сергеевич, группа 241-329

Давыдов Матвей Николаевич, группа 241-329

Место прохождения практики:

Московский Политех, кафедра «Информатика и информационные технологии»

Отчет принят с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики: Привалов Вячеслав Анатольевич

Москва 2025

Оглавление

[Введение 3](#_Toc198752540)

[Общая информация о проекте 3](#_Toc198752541)

[Описание проекта 3](#_Toc198752542)

[Первая часть практики: 4](#_Toc198752543)

[Вторая часть практики: 7](#_Toc198752544)

[Достигнутые результаты 9](#_Toc198752545)

[Заключение 10](#_Toc198752546)

# **Введение**

В рамках прохождения проектной практики перед студентами была поставлена задача реализации комплексного проекта, включающего две взаимосвязанные части. Первая часть заключалась в создании полноценного веб-сайта, предназначенного для представления результатов выполненной проектной деятельности. Этот сайт должен не только отражать структуру и содержание проделанной работы, но и служить удобным средством навигации по материалам проекта.  
Вторая часть проекта предусматривала разработку собственной оболочки командной строки (Shell) на языке программирования C. Основная цель данной части заключалась в углублённом освоении принципов работы системного программного обеспечения, а также в практическом применении знаний в области взаимодействия с операционной системой на низком уровне.

# **Общая информация о проекте**

**Цель проекта:** создание демонстрационной платформы, включающей сайт и собственную shell-оболочку.

**Задачи проекта:**

Изучение и использование Hugo как генератора статических сайтов.

Разработка шаблона сайта и его наполнения.

Реализация консольной оболочки на языке C.

Документирование процесса создания и использования каждой части проекта.

# 

# **Описание проекта**

## Мы использовали следующие технологии для реализации 1 части практики:

* Hugo
* PaperMod (стиль для оформления сайта)
* Markdown
* GitHub Pages

На начальном этапе работы была произведена настройка системы контроля версий Git. В соответствии с требованиями, был создан персональный или групповой репозиторий на платформе GitHub, используя предоставленный шаблон проекта. В процессе работы участники освоили базовые команды Git, такие как клонирование репозитория, создание коммитов, отправка изменений на удалённый сервер и работа с ветками.

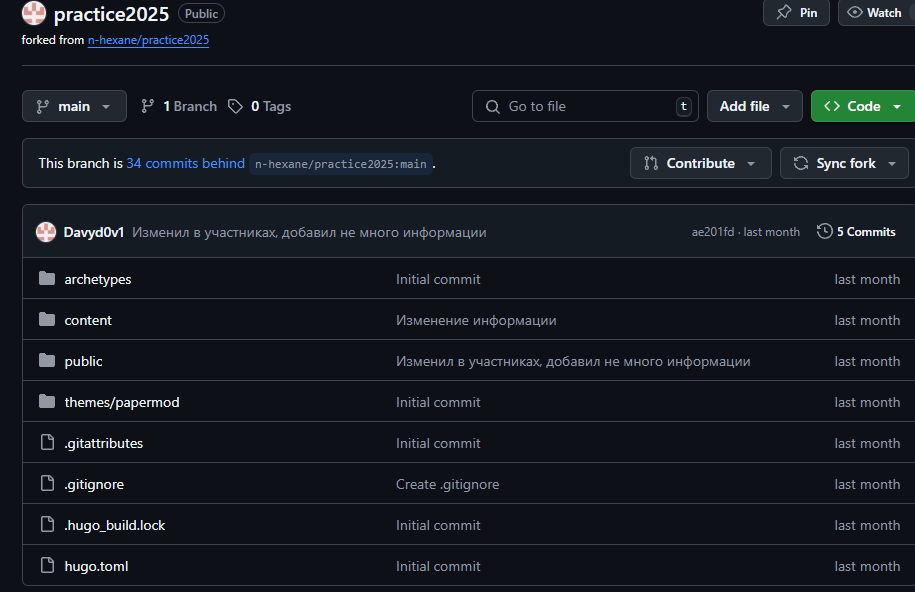


Рисунок 1 – репозиторий сайта на GitHub

Особое внимание уделялось регулярному сохранению изменений с подробными и понятными комментариями к коммитам. Это обеспечило прозрачность совместной работы и позволило контролировать ход выполнения проекта.

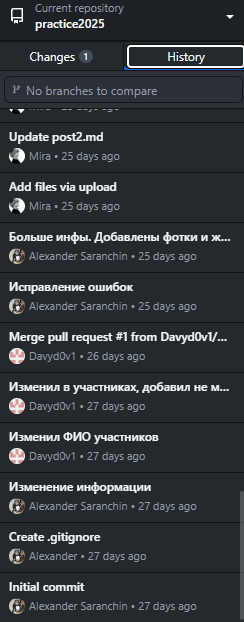


Рисунок 2 – коммиты в репозиторий

Для ведения документации использовался формат Markdown. Все необходимые материалы — описание проекта, журнал прогресса, инструкции — оформлялись именно с помощью этого формата. Участники изучили основные правила синтаксиса Markdown и подготовили структурированные документы, удобные для чтения и публикации. На освоение и оформление документации было затрачено примерно пять часов.

Вторая важная под часть проекта — создание статического веб-сайта, который отражает содержание и результаты выполненной работы. Сайт был разработан с использованием стандартных технологий HTML и CSS, что делает его доступным для студентов с базовыми знаниями веб-разработки.  
Для упрощения процесса и повышения качества работы мы применили генератор статических сайтов Hugo. При выборе этого инструмента мы руководствовались официальным руководством по быстрому старту, что значительно облегчало создание и оформление сайта.

Разработанный сайт посвящён основному проекту дисциплины «Проектная деятельность» и включает в себя следующие разделы:

* главная страница с кратким описанием проекта,
* раздел «О проекте» с детальным изложением целей, задач и этапов работы,
* раздел «Участники», где представлен личный вклад каждого члена команды,
* раздел «Журнал», содержащий не менее трёх публикаций о ходе выполнения проекта,
* раздел «Ресурсы» со ссылками на полезные материалы, включая сайты партнёров и тематические статьи.

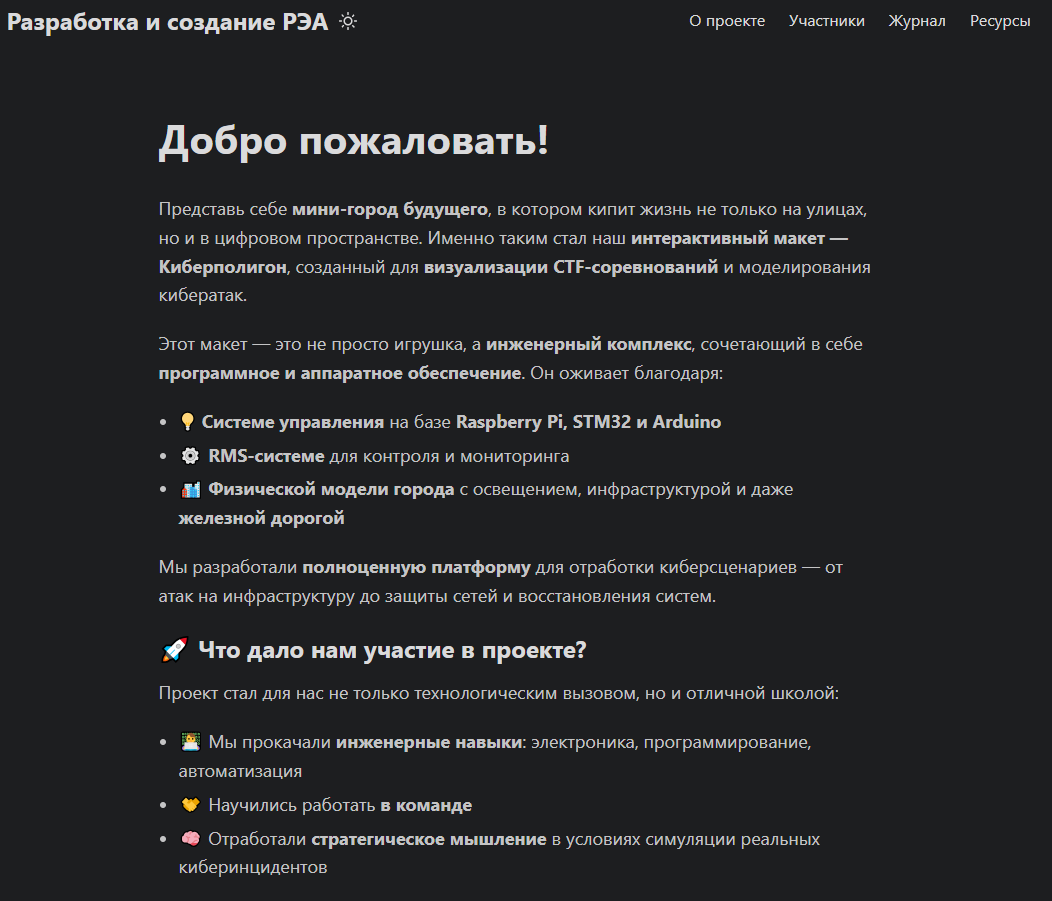


Рисунок 3 – главная страница сайта

Кроме того, для улучшения восприятия информация на сайте была дополнена графическими элементами — фотографиями. Особое внимание уделялось уникальности оформления и содержания сайта — оно должно было отличаться от работ других студентов не менее чем на 50%, что способствовало развитию творческого подхода и оригинальности.

## Мы использовали следующие технологии для реализации 2 части практики:

* ЯП C
* POSIX
* Makefile
* GitHub

Вторая вариативная часть проекта заключалась в выборе интересующей нас теме.

Мы выбрали создание Shell на языке C. Оболочка командной строки — это программа, которая читает команды пользователя, обрабатывает их и запускает выполнение системных программ.

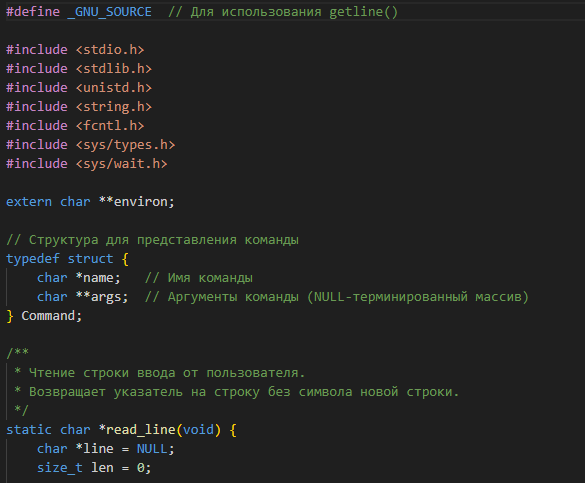


Рисунок 3 – часть программы

Основные функции программы включают:

* Чтение строки команды, вводимой пользователем.
* Разбиение введённой строки на отдельные части — имя команды и её аргументы.
* Обработку встроенных команд, таких как exit для завершения работы оболочки и cd для смены текущего каталога.
* Поддержку перенаправления ввода и вывода с помощью символов <и>.
* Возможность объединения двух команд через пайп |, чтобы вывод одной команды передавался на вход другой.
* Запуск внешних программ через системные вызовы fork и execvp.
* Ожидание завершения выполнения команд перед тем, как принимать новую команду.

Принцип работы программы следующий:

Оболочка выводит приглашение для ввода команды. Программы считывает команду, удаляя символ новой строки. Проверяет, является ли команда встроенной (exit или cd), и выполняет её при необходимости. Если в команде присутствует пайп (символ |), команда разбивается на две части, которые выполняются параллельно с организацией передачи данных между ними. Если пайпа нет, выполняется команда с учётом возможного перенаправления ввода и вывода. После выполнения команды оболочка вновь запрашивает ввод следующей команды. Данная оболочка реализует основные принципы работы командной строки в упрощённом виде и служит хорошей учебной базой для понимания работы процессов, ввода-вывода и взаимодействия между программами в операционных системах семейства Unix.

Пример использования:

* Ввод команды ls -l > files.txt выполнит команду ls -l и запишет результат в файл files.txt вместо вывода на экран.
* Ввод команды cat files.txt | grep main передаст содержимое файла files.txt через пайп в команду grep main, которая отфильтрует строки с текстом main.

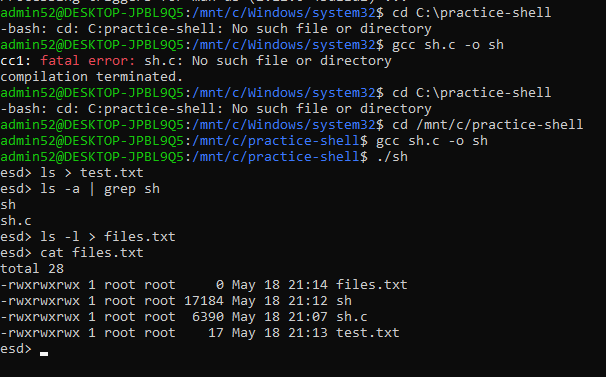


Рисунок 4 – пример использования программы

В результате реализован простой и понятный инструмент, который может быть расширен и дополнен для более сложных задач по управлению процессами и обработке команд пользователя.

# **Достигнутые результаты**

В ходе выполнения практики была настроена система контроля версий Git, создан репозиторий на платформе GitHub с использованием шаблона проекта. Участники освоили основные команды Git: клонирование, создание коммитов с информативными комментариями, работу с ветками и отправку изменений на удалённый сервер, что обеспечило прозрачность и организованность совместной работы. Для ведения документации использовался формат Markdown. Были изучены его основные правила, подготовлены структурированные и удобные для чтения документы, включая описание проекта, журнал прогресса и инструкции. Также разработан статический веб-сайт с помощью генератора Hugo и стиля PaperMod, который отражает содержание и результаты выполненной работы. Сайт содержит главную страницу с описанием проекта, разделы с подробной информацией о целях, задачах, участниках, журналом выполнения и полезными ресурсами. Для улучшения восприятия на сайте использованы графические элементы, а оформление выполнено с учётом оригинальности и творческого подхода.

Кроме того, была создана простая оболочка командной строки (shell) на языке C с применением POSIX API и Makefile для сборки. Реализован функционал чтения и обработки команд пользователя, выполнение встроенных команд (exit, cd), поддержка перенаправления ввода и вывода, работа с пайпами, запуск внешних программ через системные вызовы fork и execvp, а также синхронизация завершения команд перед приёмом новых.

В результате получен готовый репозиторий с организованной системой контроля версий, оформленная документация, полнофункциональный статический сайт и работающая оболочка командной строки, что обеспечивает прочную основу для дальнейшего развития проекта и повышения навыков участников.

# **Заключение**

Практическая работа позволила применить на практике полученные теоретические знания и освоить новые инструменты для совместной разработки и ведения документации. В процессе выполнения проекта были сформированы навыки командной работы, эффективного планирования и управления версиями, а также опыт разработки программных решений для операционных систем. Полученный опыт значительно повысил уровень профессиональной подготовки и заложил основу для дальнейшего развития в выбранной области.

**Список используемой литературы**

1. Документация Hugo - <https://gohugo.io/>
2. Тема PaperMod - <https://github.com/adityatelange/hugo-PaperMod>
3. Учебные материалы по C – <https://cplusplus.com/>
4. POSIX man pages **-** <https://manpages.ubuntu.com>
5. GitHub по практике - <https://github.com/mospol/practice-2025-1>