Лабораторная работа №4

Отчёт

Сергеев Даниил Олегович

Содержание

# 1 Цель работы

Получение навыков правильной работы с репозиториями git. [1]

# 2 Задание

* Выполнить работу для тестового репозитория.
* Преобразовать рабочий репозиторий в репозиторий с git-flow и conventional commits.

# 3 Ход выполнения лабораторной работы

## 3.1 Установка необходимого ПО

Для начала поставим git-flow. Включим репозиторий copr и установим необходимую библиотеку. (рис. 2)

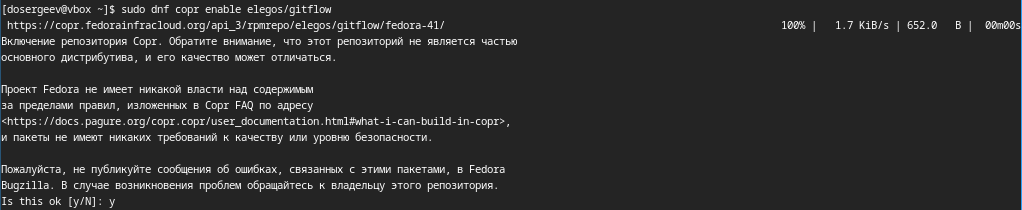


Рис. 1: Включение copr.

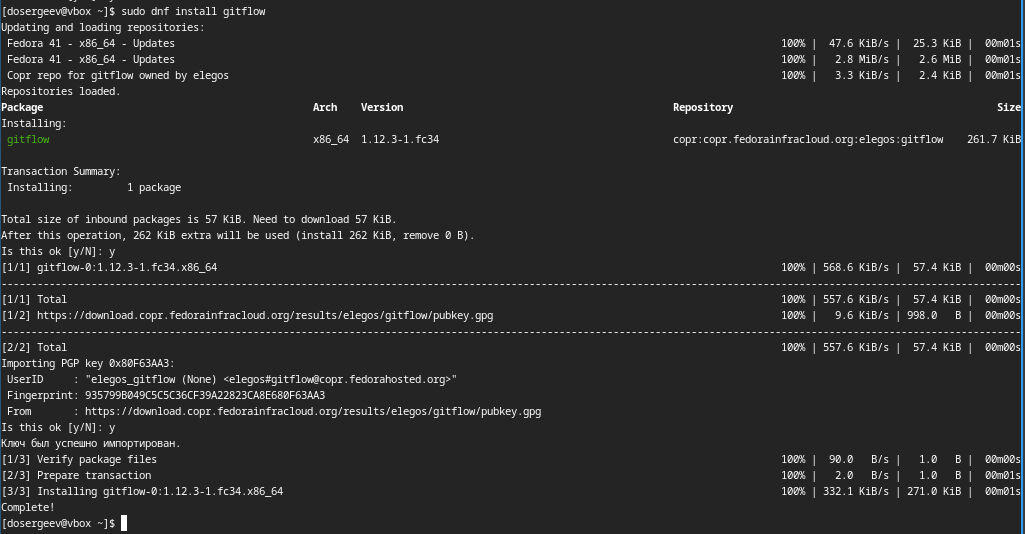


Рис. 2: Установка git-flow.

Теперь установим ПО для семантического версионирования и общепринятых коммитов. Поставим Node.js и pnpm.

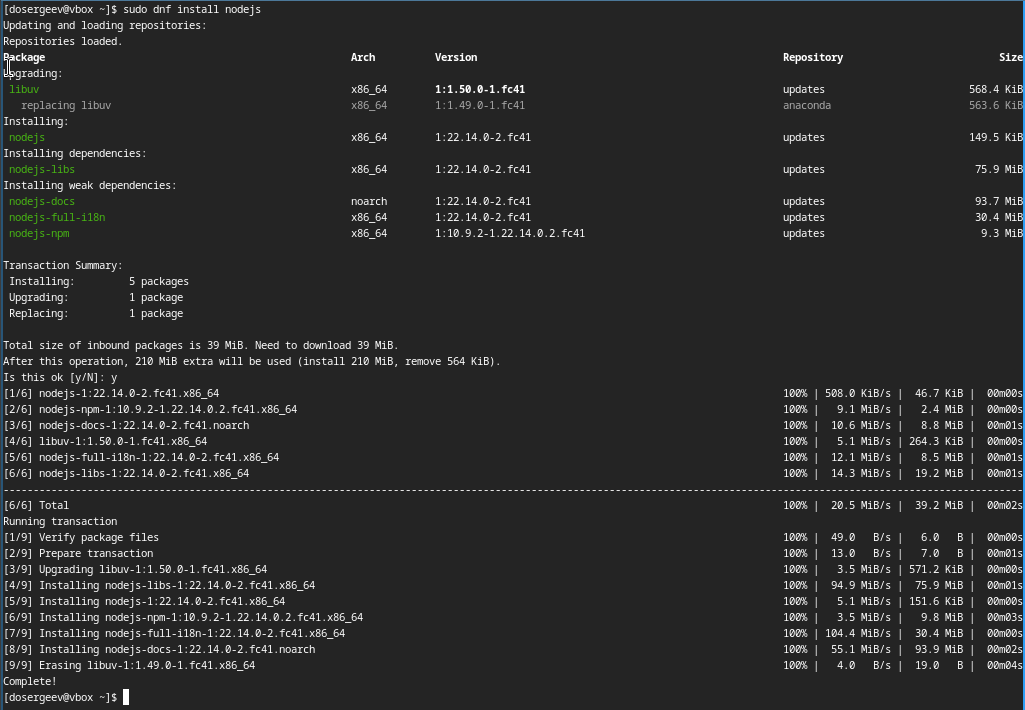


Рис. 3: Установка Node.js.

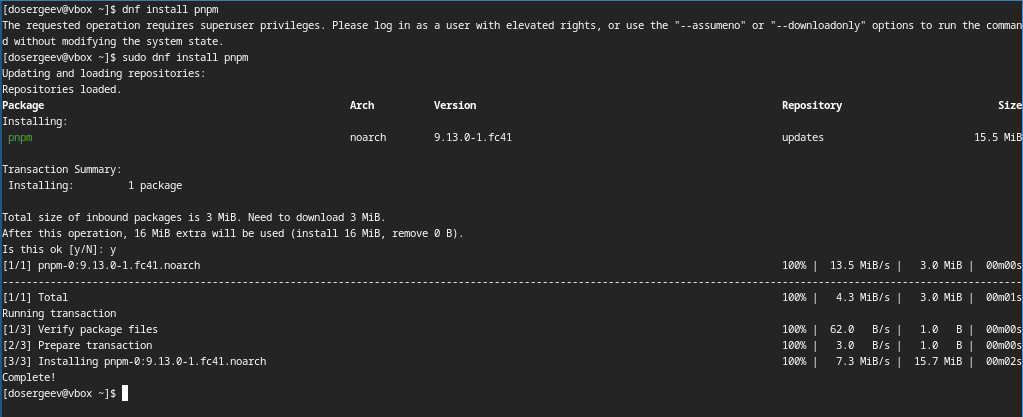


Рис. 4: Установка pnpm.

Запустим pnpm setup. В результате выходит ошибка (рис. 5): нет требуемой библиотеки sqlite3, установим её.

Попытка запустить pnpm.

Рис. 5: Попытка запустить pnpm.

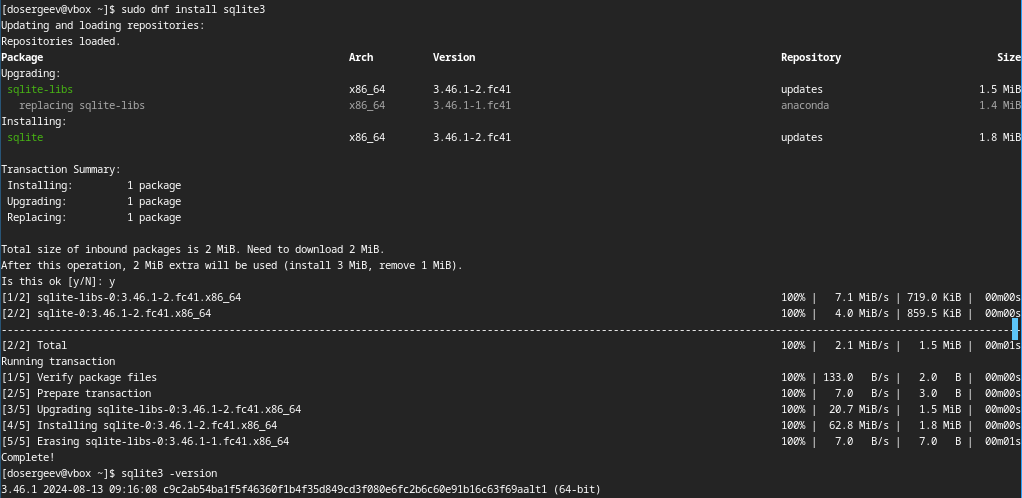


Рис. 6: Установка sqlite3.

Теперь повторим запуск pnpm и выполним команду source ~/.bashrc для начала использования pnpm.

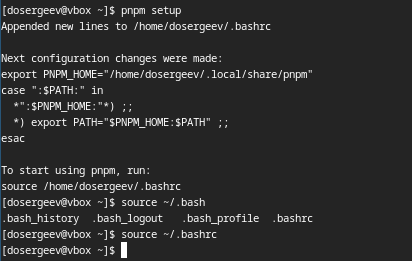


Рис. 7: Повторный запуск pnpm.

##Настройка общепринятых коммитов

Установим программу commitizen и скрипт git-cz для коммитов.

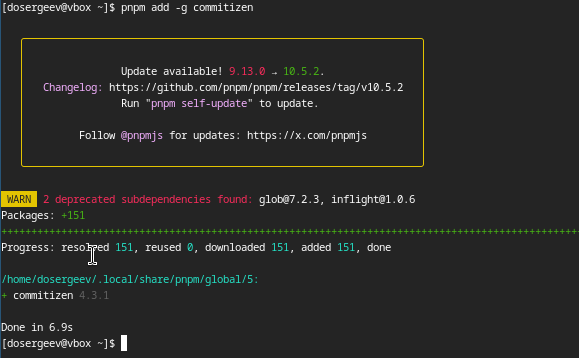


Рис. 8: Установка commitizen.

Также установим standard-changelog для помощи в создании логов.

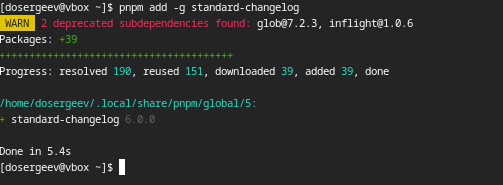


Рис. 9: Установка standard-changelog.

## 3.2 Практическое использование git

Для получения опыта практического использования git выполним несколько действий. Первым делом создадим репозиторий git с именем git-extended. (рис. 11)

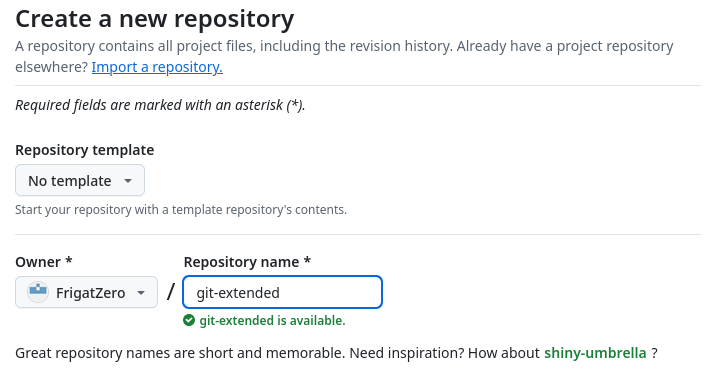


Рис. 10: Процесс создания репозитория.

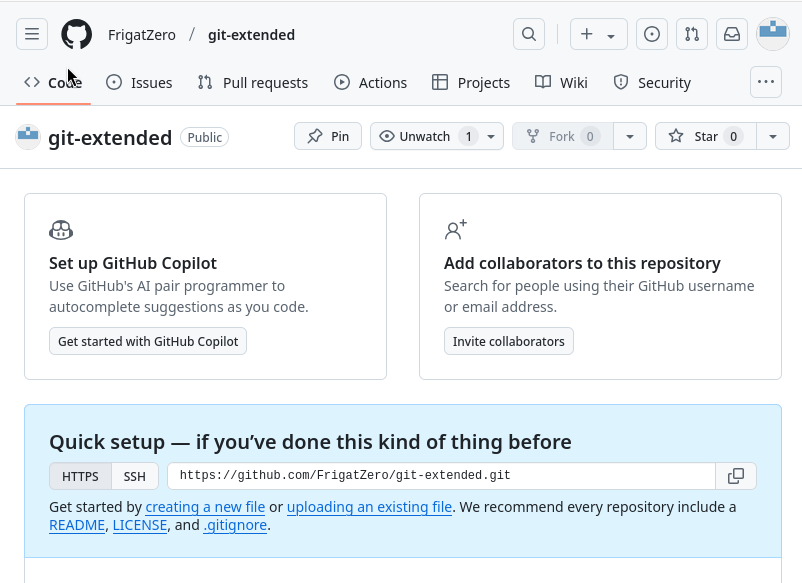


Рис. 11: Созданный репозиторий.

Чтобы подключить локальный репозиторий с сервером, создаем каталог git-extended и инициализируем в нем пустой git репозиторий. Создадим в нем файл test.txt и что-нибудь в него запишем. После добавим его в индекс, коммитим, и подключаем наш локальный репозиторий к серверному с помощью ссылки и команды git remote add origin. Выкладываем изменения на github.

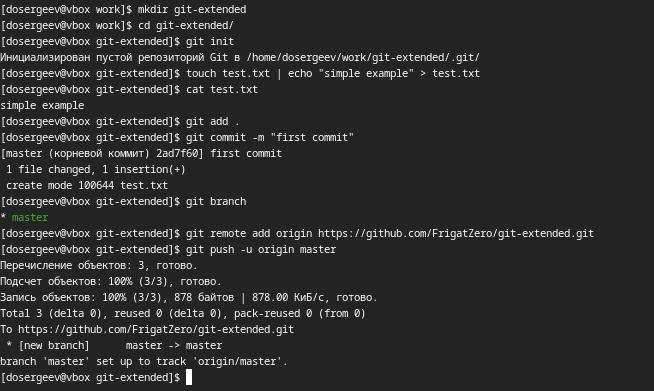


Рис. 12: Процесс выкладывания на github.

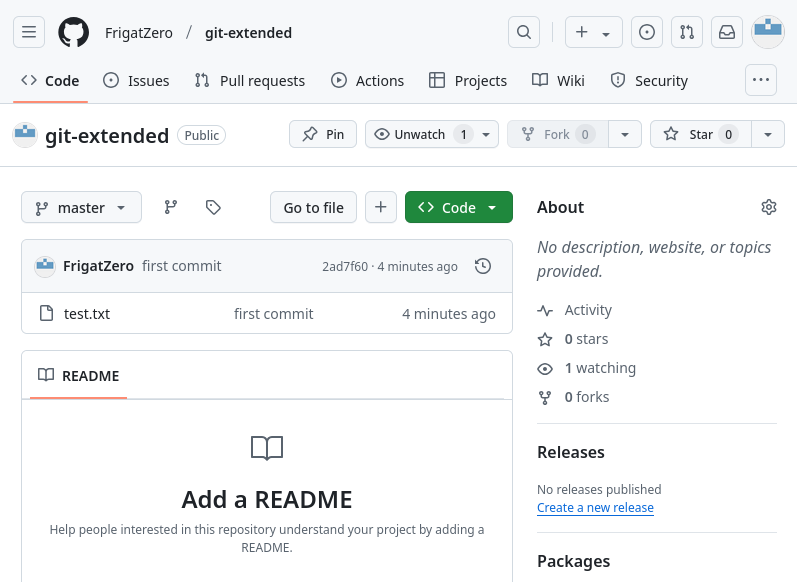


Рис. 13: Изменения на github.

Настроим общепринятые коммиты. Инициализируем конфигурацию для пакетов Node.js. Заполняем её рекомендованными параметрами: - Лицензия CC-BY-4.0 - Меняем формат коммитов

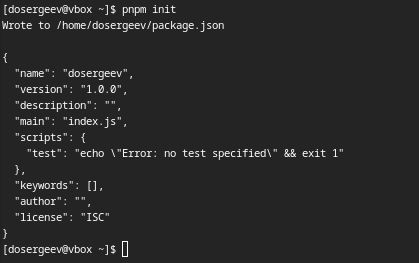


Рис. 14: Инициализация package.json.

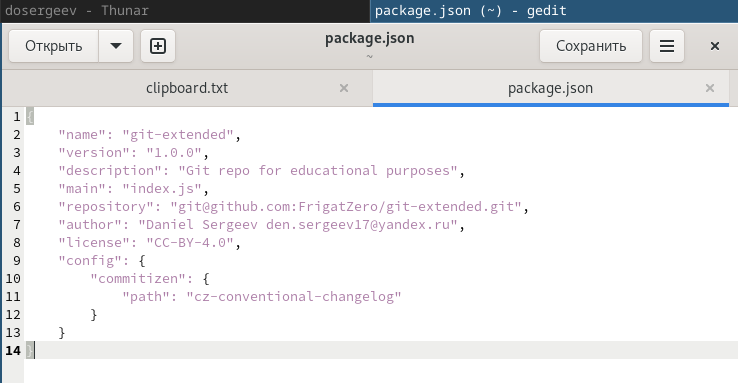


Рис. 15: Настройка package.json.

Добавим новый файл для отправки на github (рис. 16). Назовем его anotherone.txt.

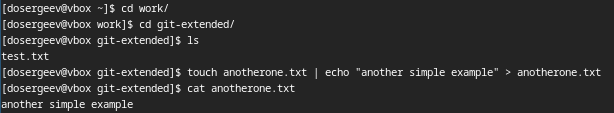


Рис. 16: Создание нового файла.

Проиндексируем все изменения и выполним коммит командой git cz: теперь можно настроить коммит в редакторе. Завершаем коммит и отправляем изменения на github.

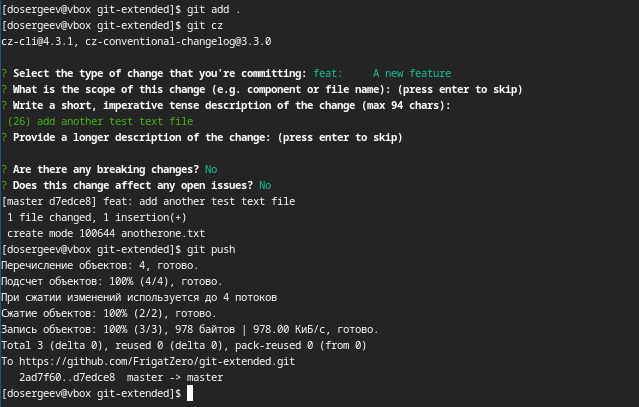


Рис. 17: Проверка общепринятых коммитов.

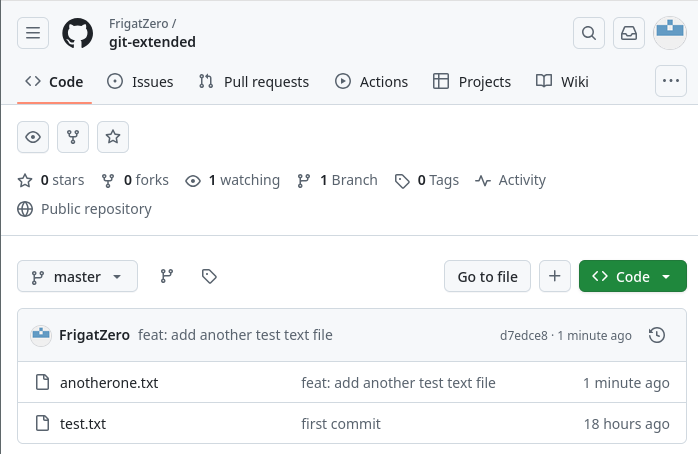


Рис. 18: Изменения github.

## 3.3 Конфигурация git-flow

Инициализируем git-flow. Для каждого ярлыка установим префикс v.

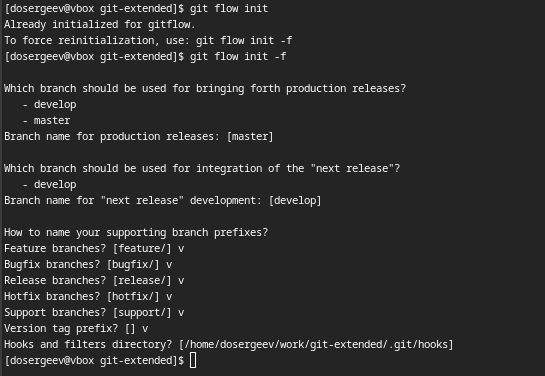


Рис. 19: Инициализация git-flow

Проверим текущую ветку. Мы находимся на develop, тогда загрузим весь репозиторий в хранилище.

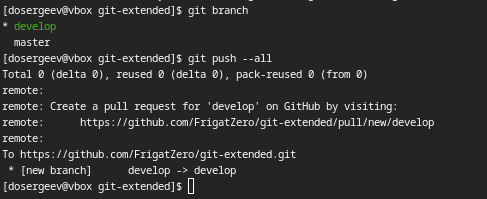


Рис. 20: Загрузка репозитория в develop.

Установим внешнюю ветку как вышестоящую для ветки develop.

Настройка внешней ветки.

Рис. 21: Настройка внешней ветки.

Создадим первый релиз с версией 1.0.0.

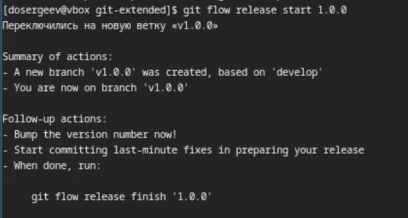


Рис. 22: Создание первого релиза.

Создадим журнал изменений.

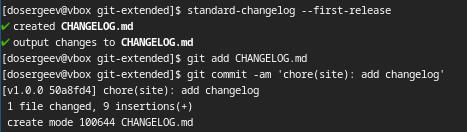


Рис. 23: Создание журнала изменений.

Проиндексируем журнал изменений и проведем коммит.

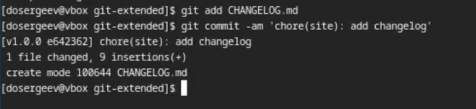


Рис. 24: Индексирование журнала изменений.

Зальём релизную ветку в основную ветку и отправим данные на github.

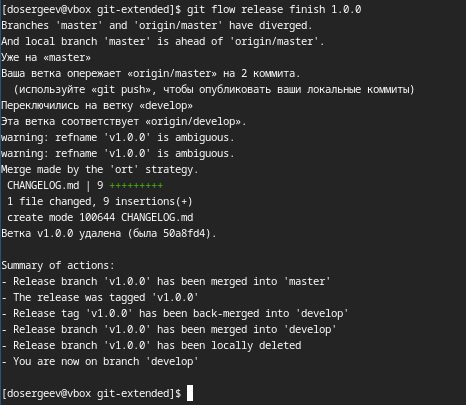


Рис. 25: Слияние релизной ветки в основную.

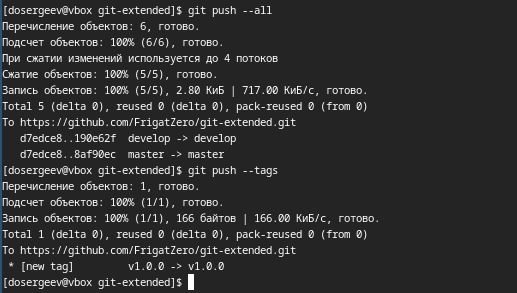


Рис. 26: Отправка данных на github.

Создадим релиз на github с помощью утилиты работы с github: gh.

Создание релиза через терминал.

Рис. 27: Создание релиза через терминал.

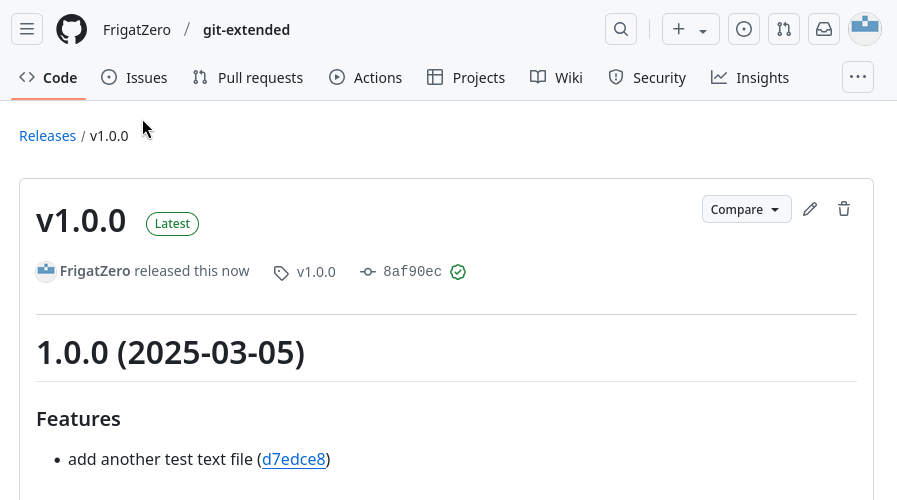


Рис. 28: Первый релиз на сайте.

## 3.4 Работа с репозиторием git

Создадим ветку для новой функциональности и в ней добавим новый файл. Следующим шагом объединяем ветку feature\_branch с develop.

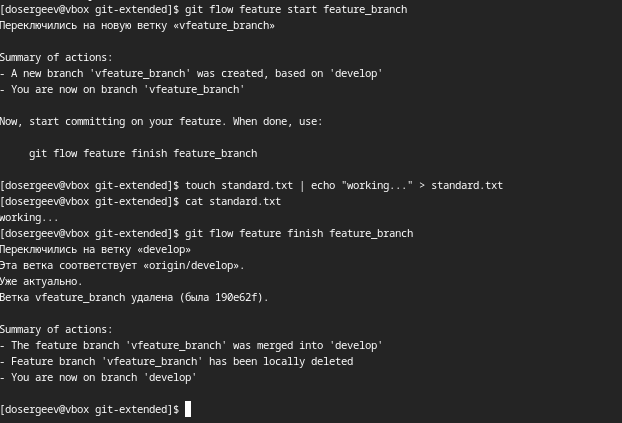


Рис. 29: Создание ветки новой функциональности и её слияние.

Создадим релиз с версией 1.2.3. Поменяем номер версии в файле package.json.

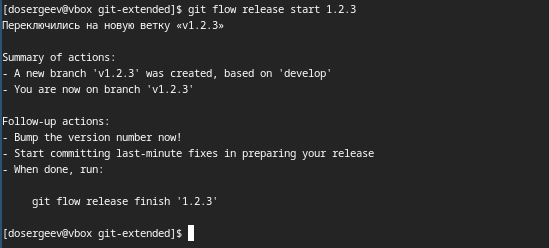


Рис. 30: Создание релиза 1.2.3.

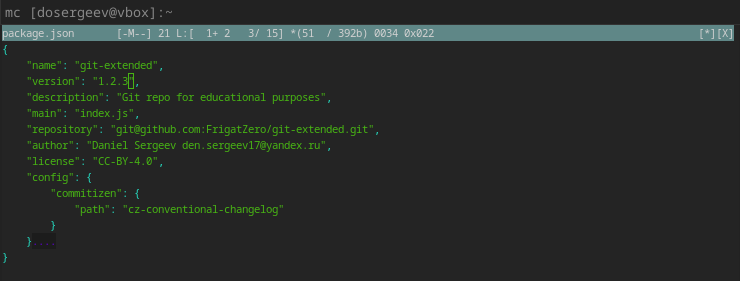


Рис. 31: Изменение версии в package.json.

Создадим журнал изменений, добавим его в индекс и зальём релизную ветку в основную (рис. 32).

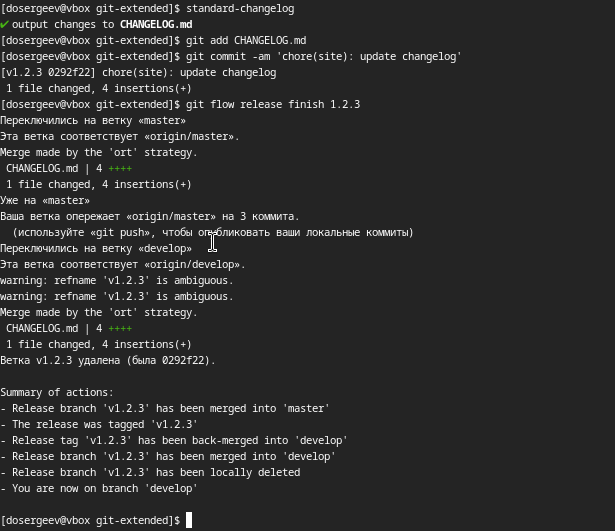


Рис. 32: Работа над релизом 1.2.3.

Отправим данные на github.

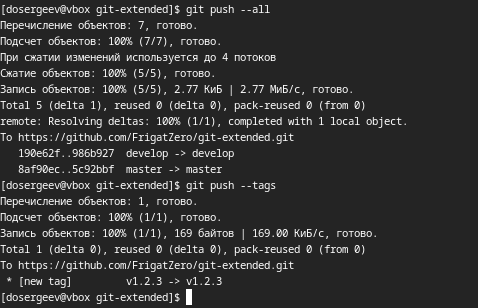


Рис. 33: Отправляем данные релиза 1.2.3 на github.

Создадим релиз на github с комментарием из журнала изменений.

Отправляем релиз на github.

Рис. 34: Отправляем релиз на github.

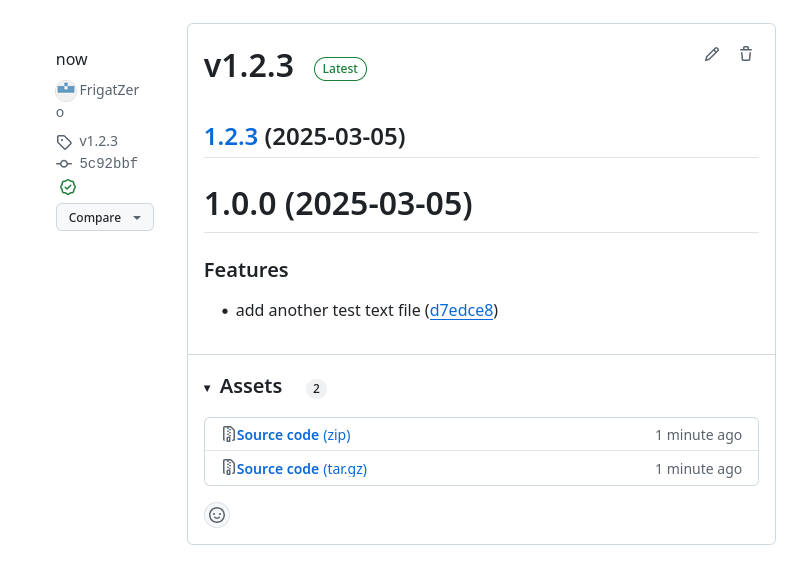


Рис. 35: Проверка релиза 1.2.3 на сайте.

# 4 Вывод

В результате выполнения лабораторной работы я получил навыки и умения качественной работы с репозиториями git и с github.

# Список литературы

1. Kulyabov. [Лабораторная работа №4. Продвинутое использование git](https://esystem.rudn.ru/mod/page/view.php?id=1224099). RUDN.