Лабораторная работа №4

Операционные системы

Тойчубекова Асель Нуралановна

Содержание

# 1 Цель работы

Целью данной лабораторной работы является получение навыков правильной работы с репозиториями git.

# 2 Задание

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

# 3 Теоретическое введение

**GitFlow**- это определенная надстройка над моделью ветвление Git, которая включает в себя использование фича веток и несколько основных веток, для удобства работы над тем или иным проектом. Gitflow Workflow опубликована и популяризована Винсентом Дриссеном. Gitflow Workflow предполагает выстраивание строгой модели ветвления с учётом выпуска проекта. Данная модель отлично подходит для организации рабочего процесса на основе релизов.

**Последовательность действий при работе по модели Gitflow:** - Из ветки master создаётся ветка develop.  
- Из ветки develop создаётся ветка release.  
- Из ветки develop создаются ветки feature.  
- Когда работа над веткой feature завершена, она сливается с веткой develop.  
- Когда работа над веткой релиза release завершена, она сливается в ветки develop и master.  
- Если в master обнаружена проблема, из master создаётся ветка hotfix.  
- Когда работа над веткой исправления hotfix завершена, она сливается в ветки develop и master.

# 4 Выполнение лабораторной работы

## 4.1 Установка gitflow

Устанавливаю gitflow из коллекции репозиториев Copr (рис. 1 и рис. 2 ).

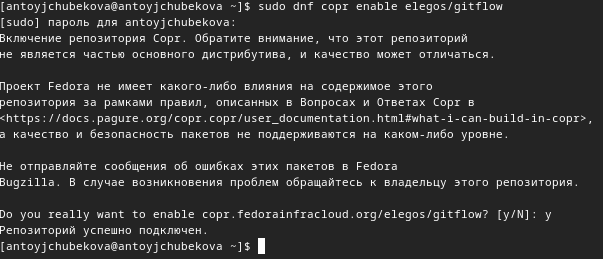


Рис. 1: Установка gitflow

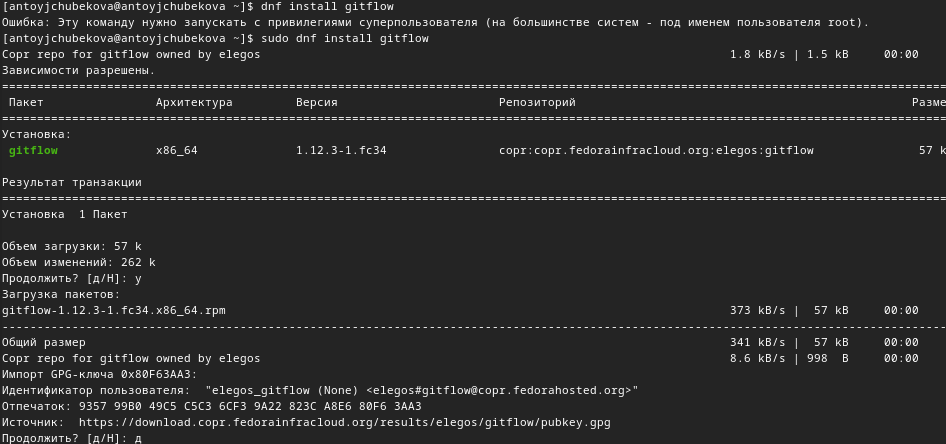


Рис. 2: Установка gitflow

## 4.2 Установка Node.js

Устанавливаю Node.js. pnpm скачиваю с интернета и запускаю с помощью команды dnf install. рис. 3 и рис. 4 ).

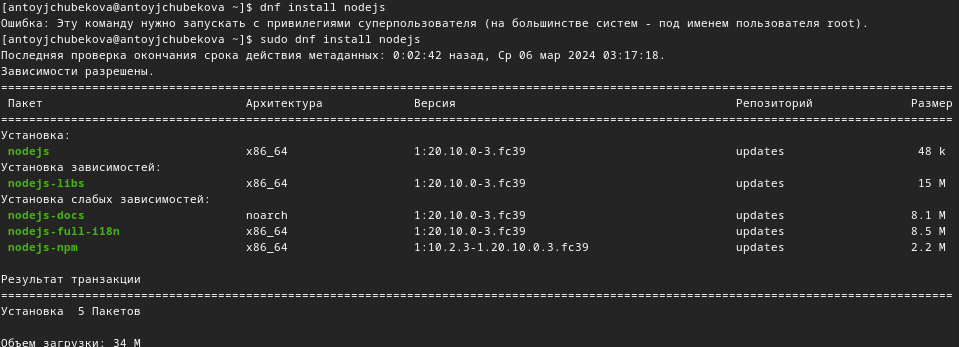


Рис. 3: Установка Node.js.

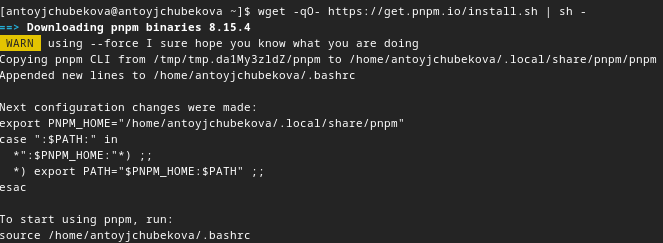


Рис. 4: Установка Node.js.

## 4.3 Настройка Node.js

Запускаю pnpm. (рис. 5 ).

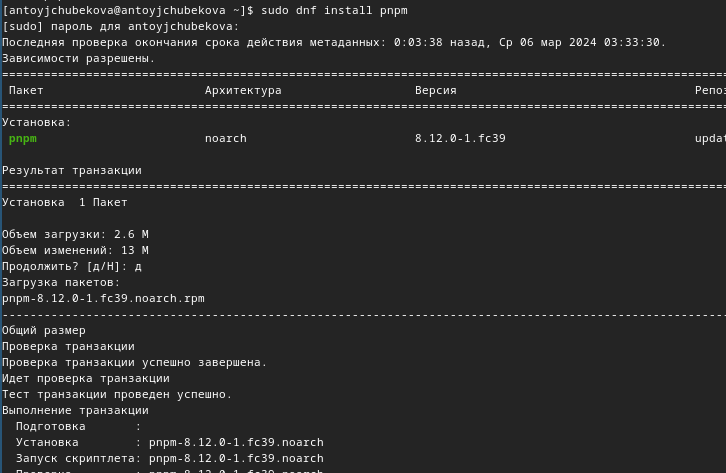


Рис. 5: Запуск pnpm

Перелогиниваюсь, выполняя команду source ~/.bashrc. (рис. 6 ).

Выпонение команды source ~/.bashrc

Рис. 6: Выпонение команды source ~/.bashrc

## 4.4 Общепринятые коммиты

Устанавливаю программу, используемая для помощи в формировании коммитов. При этом устанавливвается скрипт git-cz, который мы и будем использовать для коммитов. (рис. 7 ).

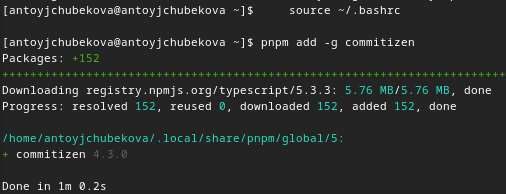


Рис. 7: Установка программы для коммитов

Устанавливаю программу для помощи в создании логов. (рис. 8 ).

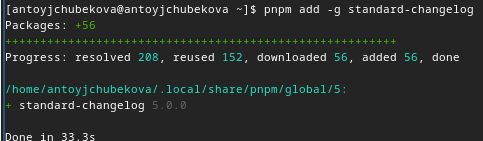


Рис. 8: Установка программы для создании логов

Создаю репозиторий git-extended. (рис. 9 ).

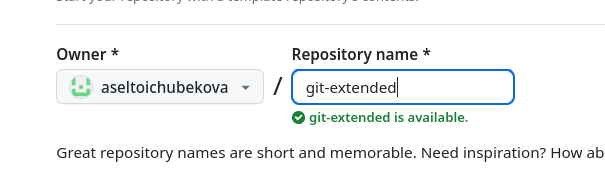


Рис. 9: Создание репозитория

Клонирую репозторий. (рис. 10 ).

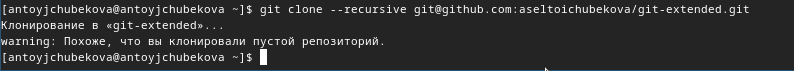


Рис. 10: Клонирование репозитория

Далее я перехожу в этот репозиторий и создаю файл README.md, затем записываю в этот файл некоторый текст и сохраняю его. (рис. 11 и рис. 12 ).

Создание файла

Рис. 11: Создание файла

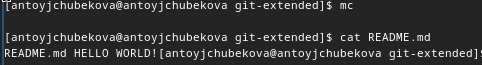


Рис. 12: Редактирование файла

Делаю первый коссит и выкладываю на github. (рис. 13 ) Захожу на github и видим, что успешно закоммитолось. (рис. 14 )

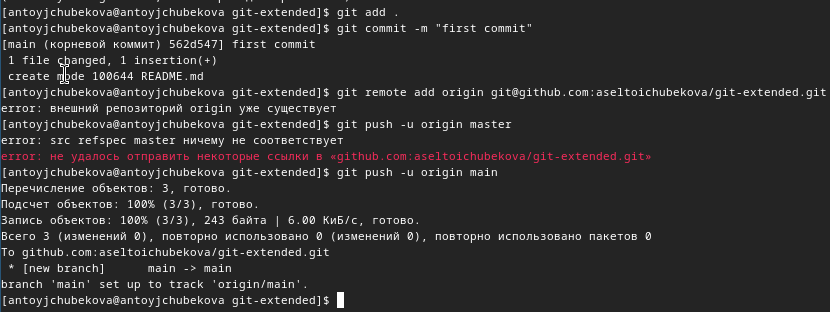


Рис. 13: Первый коммит

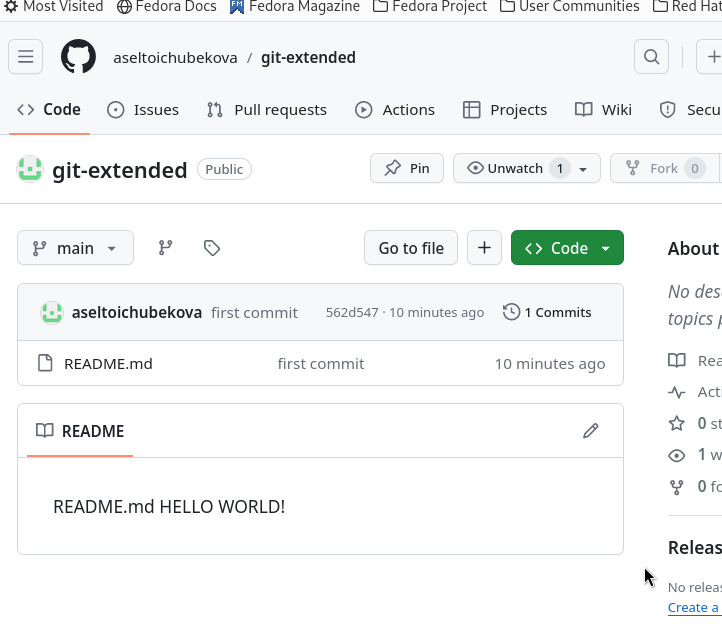


Рис. 14: На github

Выполняю конфигурацию для пакетов Node.js. (рис. 15 ).

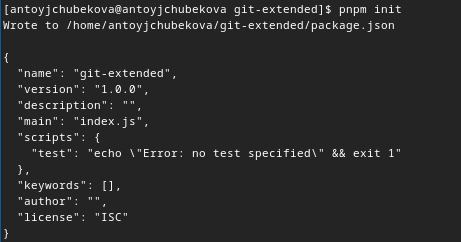


Рис. 15: Конфигурация для пакетов Node.js

Открыв файл package.json с помщью команды mc, редактирую его заполняя некоторыые параметры пакета, также сконфигурирую формат коммитов. (рис. 16 ).

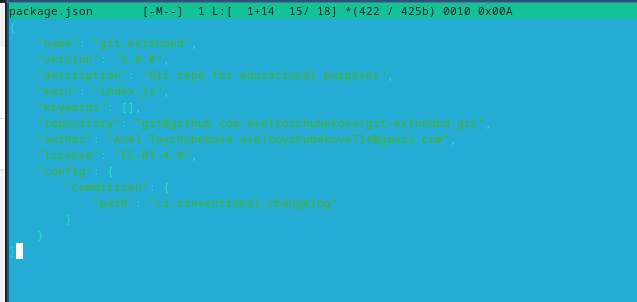


Рис. 16: Редактирование файла package.json

Добавляю новые файлы и выполняю коммит, затем отправляю на github. (рис. 17 ).

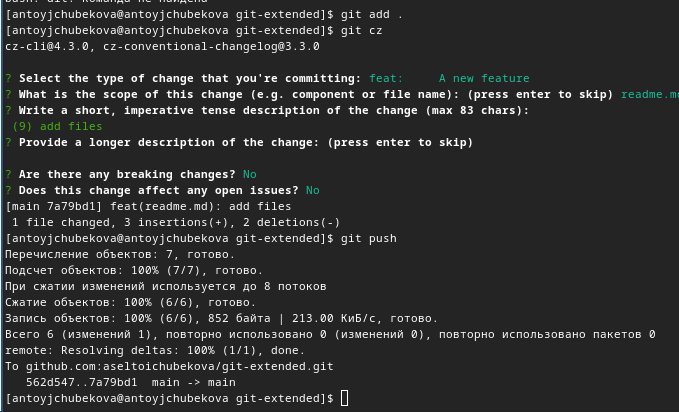


Рис. 17: Выполнение коммита

Инициализирую gitflow. (рис. 18 ).

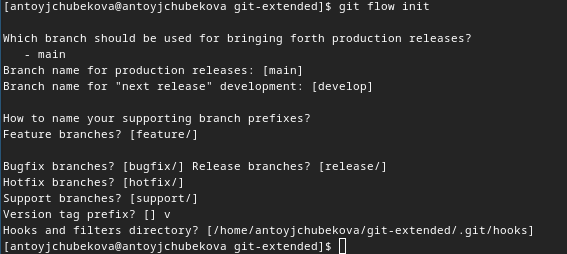


Рис. 18: Инициализация gitflow

Проверяю, что я нахожусь в ветке develop. Мы видим, что все верно и мы находимся в этой ветке. (рис. 19 ).

Нахождение в ветке develop

Рис. 19: Нахождение в ветке develop

Загружаю весь репозиторий в хранилище. (рис. 20 ).

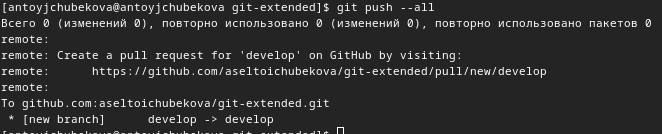


Рис. 20: Загрузка репозитория

Устанавливаю внешнюю ветку как вышестоящую для этой ветки. (рис. 21 ).

Установка внешней ветки

Рис. 21: Установка внешней ветки

Создаю релиз с версией 1.0.0 (рис. 22 ).

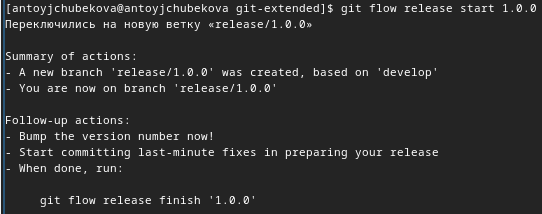


Рис. 22: Создание релиза

Создаю журнал изменений (рис. 23 ).

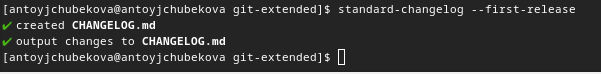


Рис. 23: Создание журнала изменений

Добавляю журнал изменений в индекс. (рис. 24 ).

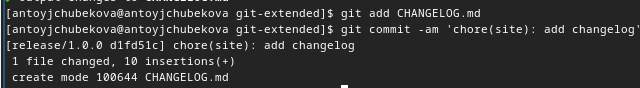


Рис. 24: Добавление журнала изменений в индекс

Заливаю релизную ветку в основную ветку. (рис. 25 и рис. 26 ).

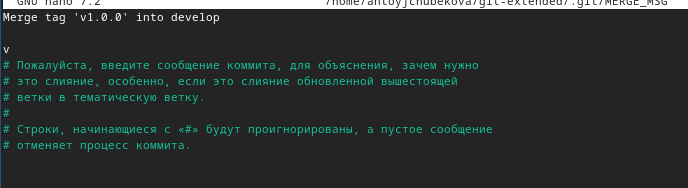


Рис. 25: Заливание релизной ветки в основную

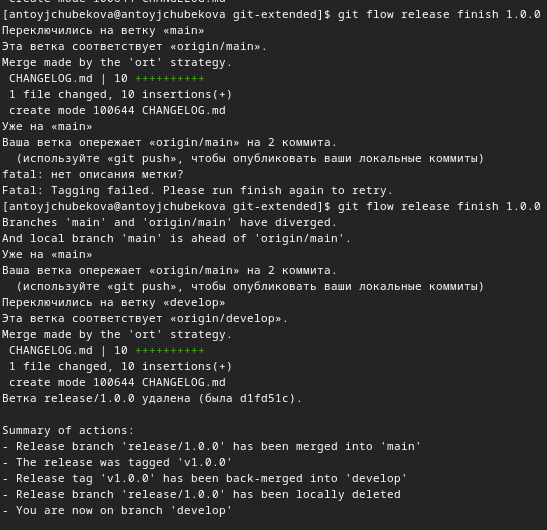


Рис. 26: Заливание релизной ветки в основную

Отправляю данные на github. (рис. 27 ).

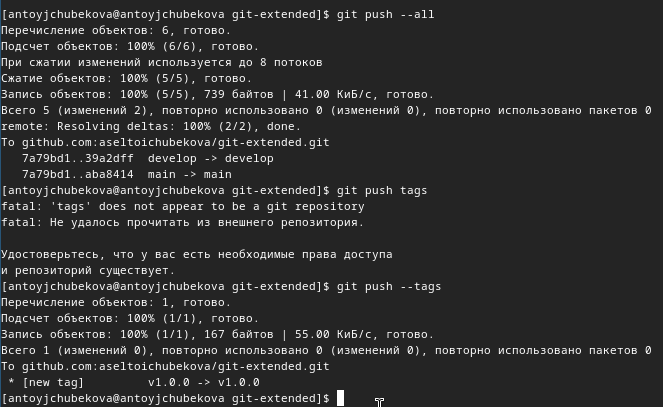


Рис. 27: Отправка данных на гитхаб

Создаю релиз на github. (рис. 28 ).

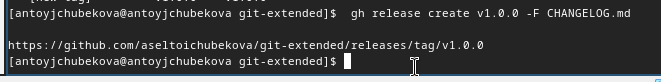


Рис. 28: Создание релиза

Создаю ветку для новой функциональности. (рис. 29 ).

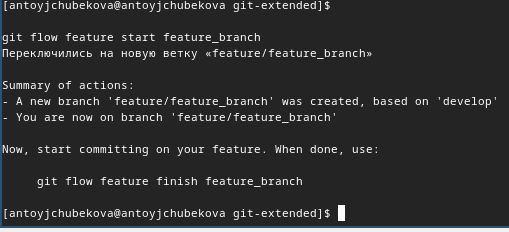


Рис. 29: Создание ветки

Объединяю ветку feature\_branch с develop. (рис. 30 ).

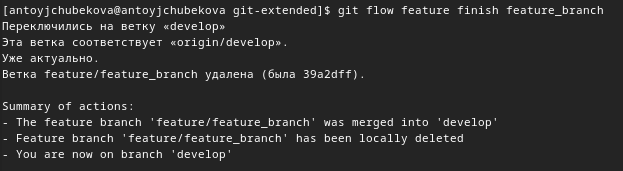


Рис. 30: Объединение ветку feature\_branch с develop

Создадим релиз с версией 1.2.3 (рис. 31 ).

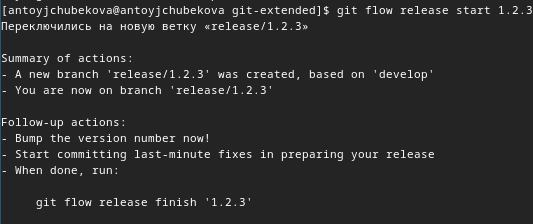


Рис. 31: Создание релиза

Обнавляю номер версии в файле package.json. (рис. 32 ).

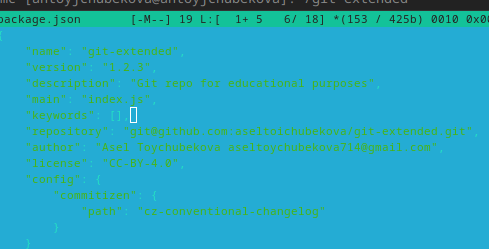


Рис. 32: Обнавление номера версии

Создаю журнал изменений. (рис. 33 ).

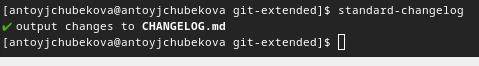


Рис. 33: Создание журнала изменений

Добавляю журнал изменений в индекс. (рис. 34 ).

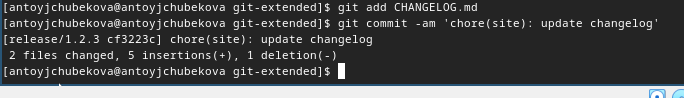


Рис. 34: Добавление журнала изменений в индекс

Заливаю релизную ветку в основную ветку. (рис. 35 и рис. 36 ).

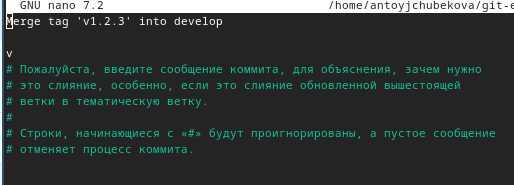


Рис. 35: Заливание релизной ветки в основную

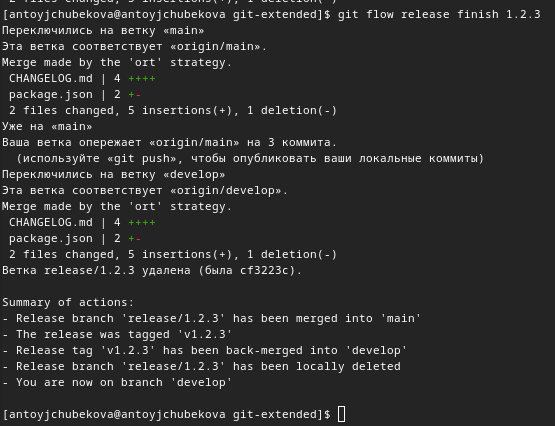


Рис. 36: Заливание релизной ветки в основную

Отправляю данные на github. (рис. 37 ).

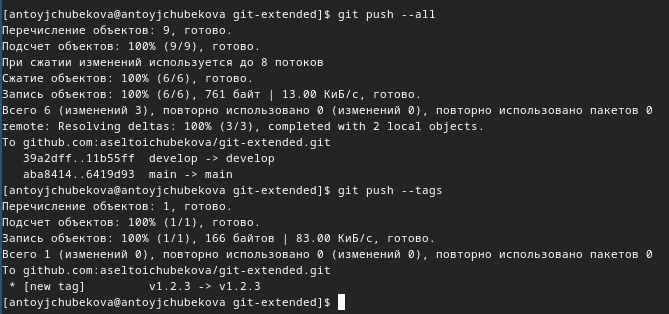


Рис. 37: Отправление данных на гитхаб

Создаю релиз на github c комментарием из журнала изменений. (рис. 38 ).

Создание релиза на гитхаб

Рис. 38: Создание релиза на гитхаб

# 5 Вывод

В ходе выполнения лабораторной работы №4 я получила навыки правильной работы с репозиториями git.

# Список литературы

* https://esystem.rudn.ru/mod/page/view.php?id=1098937.