Лабораторная работа №4

Продвинутое использование git

Буллер Татьяна Александровна

Содержание

# 1 Цель работы

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 Выполнение лабораторной работы

## 3.1 Добавление общепринятых коммитов.

С помощью утилиты pnpm добавляем две программы: standard changelog и commitizen.



Рис. 1: Добавление программ с помощью pnpm

Программы добавлены. В дальнейшем они будут использоваться для добавления коммитов и создания лога изменений.

## 3.2 Создание первого коммита в репозитории.

Репозиторий создаем через сайт github, после чего добавляем его на виртуальную машину. Для первого коммита создадим README-файл и добавим его:

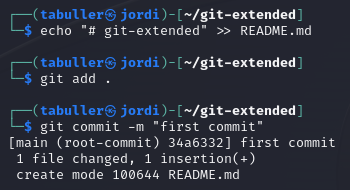


Рис. 2: Первый коммит в тестовый репозиторий

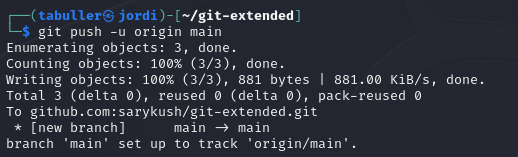


Рис. 3: Отправление первого коммита

Файл отправлен с первым коммитом, работу можно продолжать дальше.

## 3.3 Конфигурация общепринятых коммитов.

Введем команду pnpm init. Вывод команды демонстрирует содержание файла и его место в системе.

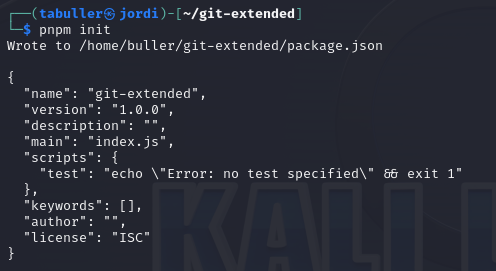


Рис. 4: pnpm init

Откроем файл, адрес которого видим в выводе команды, и отредактируем его так, как указано в задании лабораторной работы:

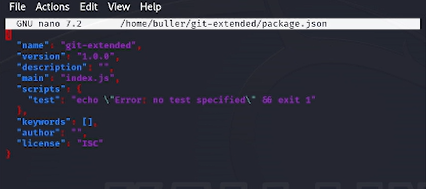


Рис. 5: Изначальный файл package.json

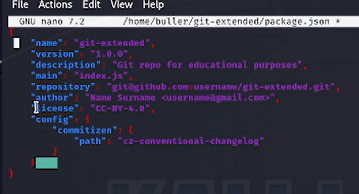


Рис. 6: Изначальный файл package.json

После редактирования файла необходимо добавить файлы в репозиторий и выполнить коммит с помощью установленного ранее скрипта.

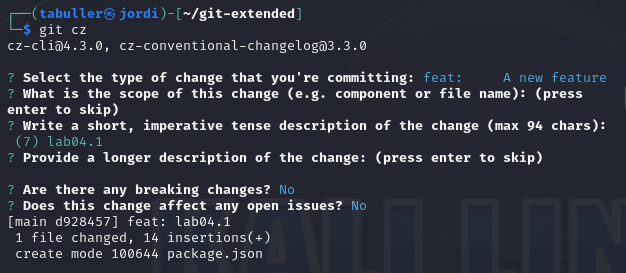


Рис. 7: Использование скрипта cz для коммита

Коммит успешно настроен и отправлен.

## 3.4 Конфигурация git-flow.

Инициализируем git-flow и проверим, что мы находимся на нужной ветке (develop):

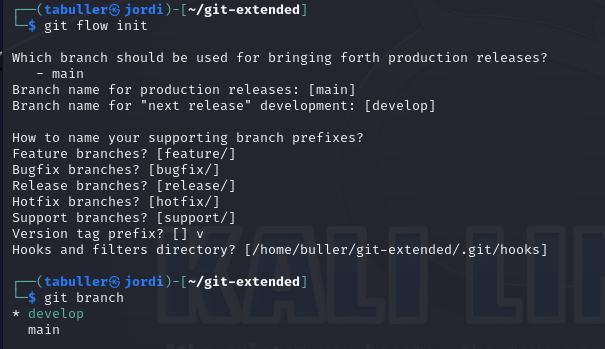


Рис. 8: Инициализация git-flow

Названия веток оставляем по умолчанию, для версий добавляем перфикс v: каждая новая версия будет выглядеть как vNN.NN.NN. Отправляем изменения на гитхаб. Следующее, что нужно сделать - установить внешнюю ветку как вышестоящую для этой ветки.

Установка вышестоящей ветки

Рис. 9: Установка вышестоящей ветки

После того, как работа с ветками закончена, создаем новый релиз: 1.0.0

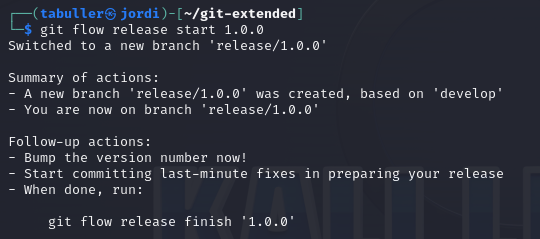


Рис. 10: Создание первого релиза

Далее настраиваем релиз: создаем журнал изменений, добавляем его в индекс и заливаем резиз в основную ветку.

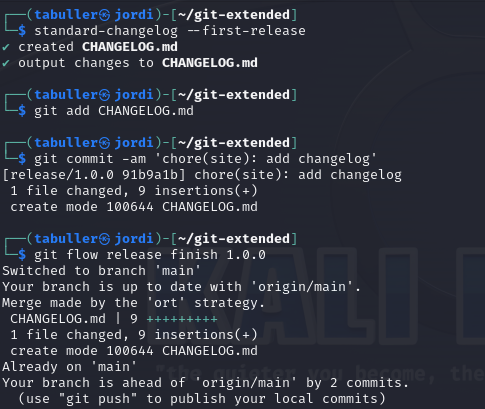


Рис. 11: Работа с релизом

После дого, как отправили данные, создаем релиз на гитхаб и получаем ссылку на него:

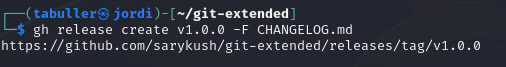


Рис. 12: Создание итогового резиза

## 3.5 Разработка новой функциональности.

Создадим ветку для новой функциональности:

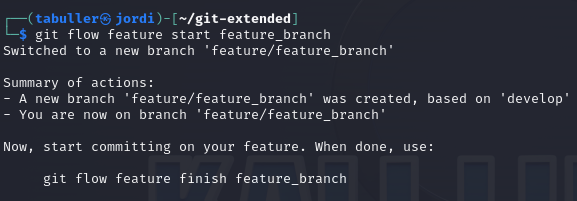


Рис. 13: Создание новой ветки

Далее работа с гит продолжается как обычно. Для тестового репозитория никакую новую функциональность разрабатывать не будем, переходим сразу к следующему шагу:

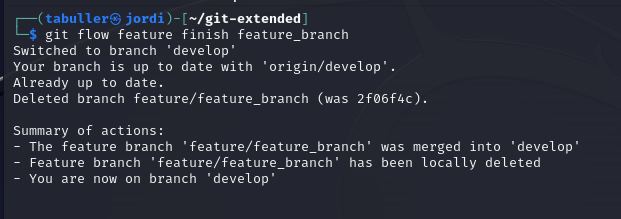


Рис. 14: Закрытие ветки

Созданная нами ветка объединилась с веткой develop.

## 3.6 Создание релиза.

Создадим релиз с версией 1.2.3 и оказываемся на ветке release/1.2.3:

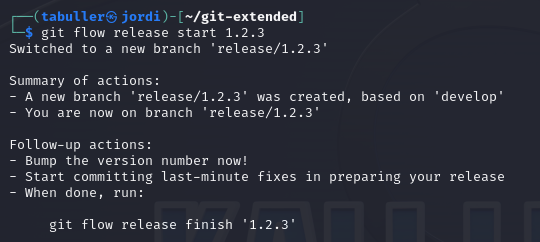


Рис. 15: Создание нового релиза

В файле package.json обновляем номер версии. Этот файл находится прямо в репозитории, в котором мы работаем, и копирует в себе данные, которые мы настроили на первых этапах:

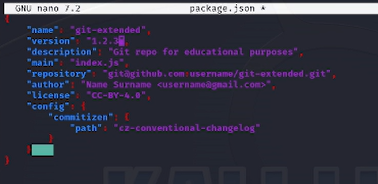


Рис. 16: Редактирование json файла

Когда редактирование завершено, обновляем журнал изменений и добавляем его на гитхаб:

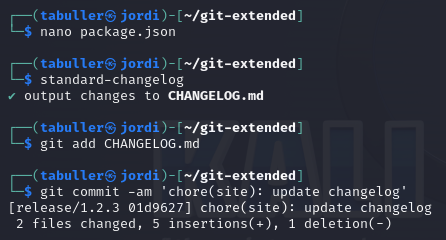


Рис. 17: Обновление журнала изменений

Заливаем релиз в основную ветку, после чего отправляем данные на гитхаб:

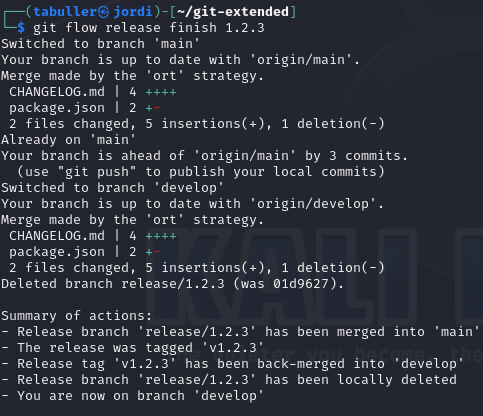


Рис. 18: Завершение работы с релизом

Последним шагом закроем релиз и создадим на гитхаб новую версию - 1.2.3:

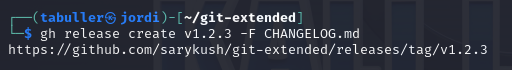


Рис. 19: Завершение работы с релизом

Проверим журнал изменений в репозитории на сайте гитхаб - видим обе версии, следовательно, работа выполнена успешно.

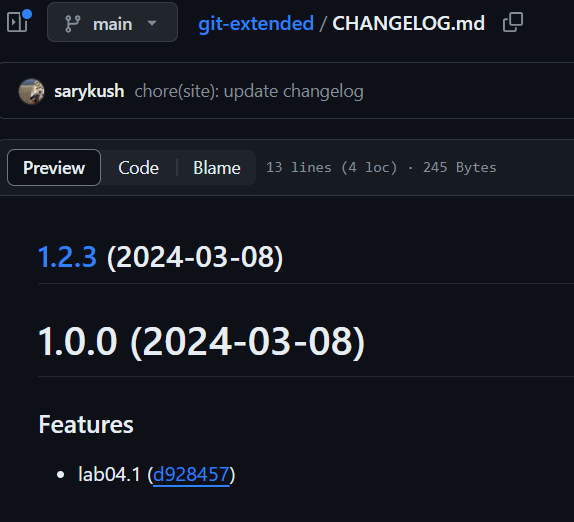


Рис. 20: Журнал версий на github

# 4 Выводы

Получены навыки правильной работы с репозиториями git, выполнена работа для тестового репозитория и дальнейшие преобразования для основного репозитория курса.