Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютеров и операционные системы

Джаллох Ишмаил

Содержание

# 1 Цель работы

Цель данной работы является получением навыков правильной работы с репозиториями git.

# 2 Задание

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

# 3 Выполнение лабораторной работы

## 3.1 Установка git-flow

Gitflow Workflow предполагает выстраивание строгой модели ветвления с учётом выпуска проекта. Сначала я включаю репозиторий copr:

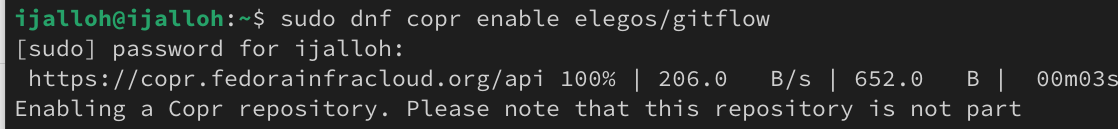


Рис. 1: включение copr

Используя dnf install скачаю gitflow:

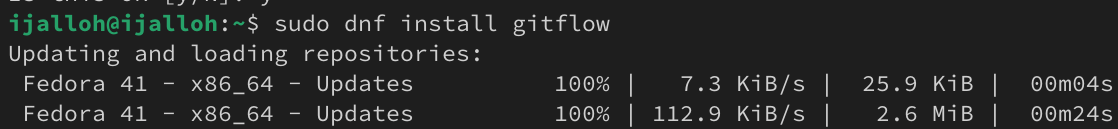


Рис. 2: dnf install gitflow

## 3.2 Установка Node.js

Для семантического версионирования и общепринятых коммитов я устанавливаю Nodejs и pnpm:

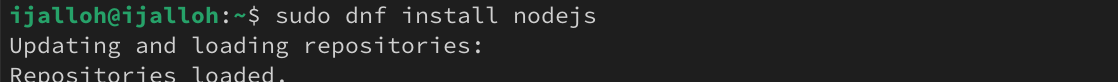


Рис. 3: Установка Nodejs

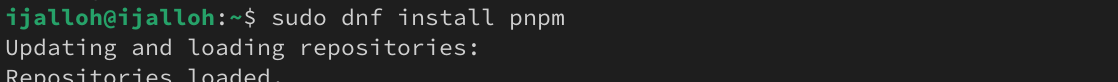


Рис. 4: Установка pnpm

## 3.3 Настройка Node.js

Запуская pnpm setup я добавляю каталог с исполняемыми файлами, устанавливаемыми yarn для работы с Node.js в переменную PATH:

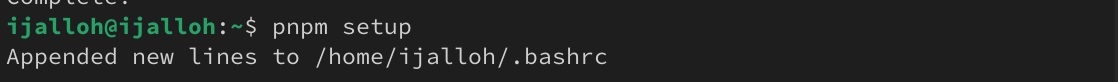


Рис. 5: Запуск pnpm

Далее перелогинуюсь и выполняю source ~/.bashrc:

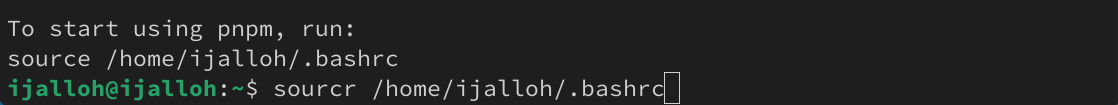


Рис. 6: ~/.bashrc

## 3.4 Общепринятые коммиты

Для помощи в форматировании коммитов добавляю программу commitizen:

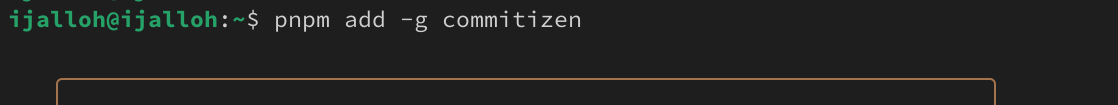


Рис. 7: добавление commitizen

Добавляю standard-changelog для помощи в создании логов:

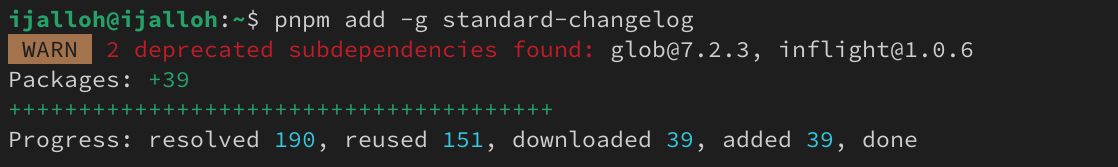


Рис. 8: добавление standard-changelog

Создаю репозиторий на GitHub назову его git-extended:

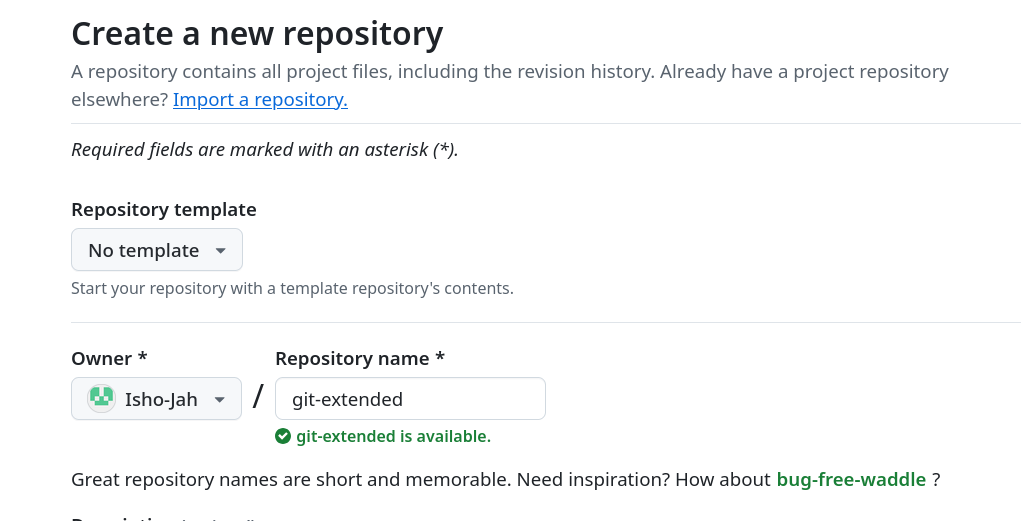


Рис. 9: Создание git-extended

Я клонирую его в каталог git-extended:

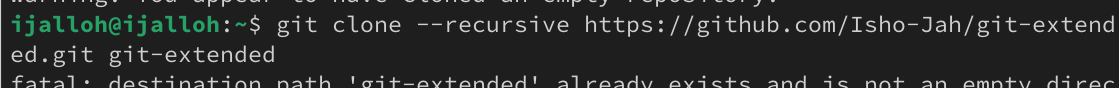


Рис. 10: клонирование git-extended

Создаю и добавляю файл README.md:

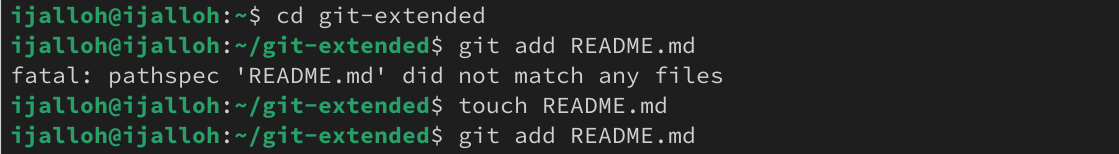


Рис. 11: Добавление readme.md

Делаю первый коммит и выкладываю на github:

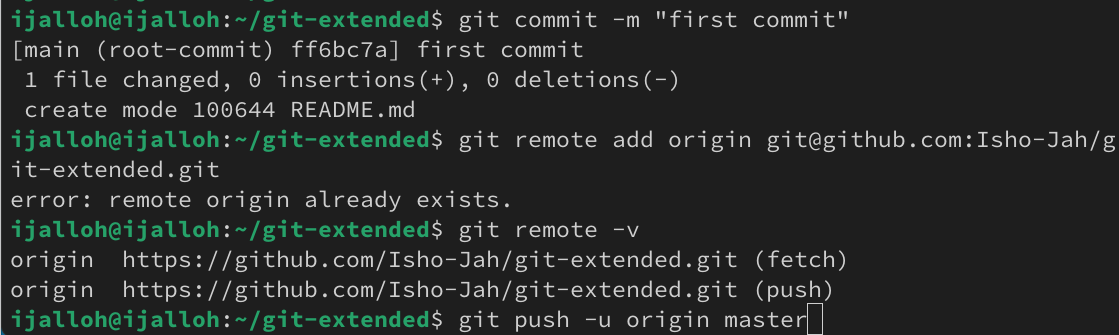


Рис. 12: Первый коммит

Я инициализирую pnpm:

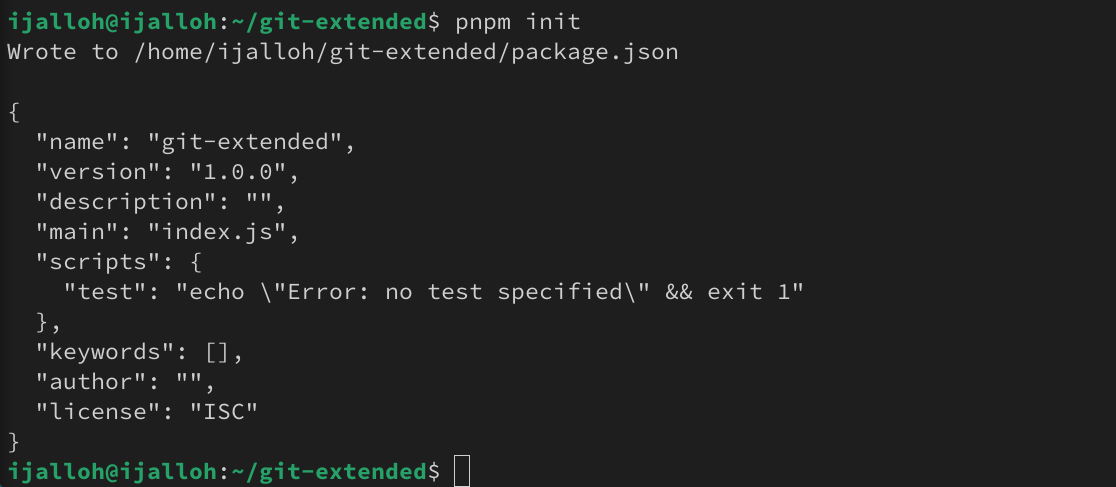


Рис. 13: Инициализирование pnpm

Заполняю несколько параметров пакета (файл package.json):

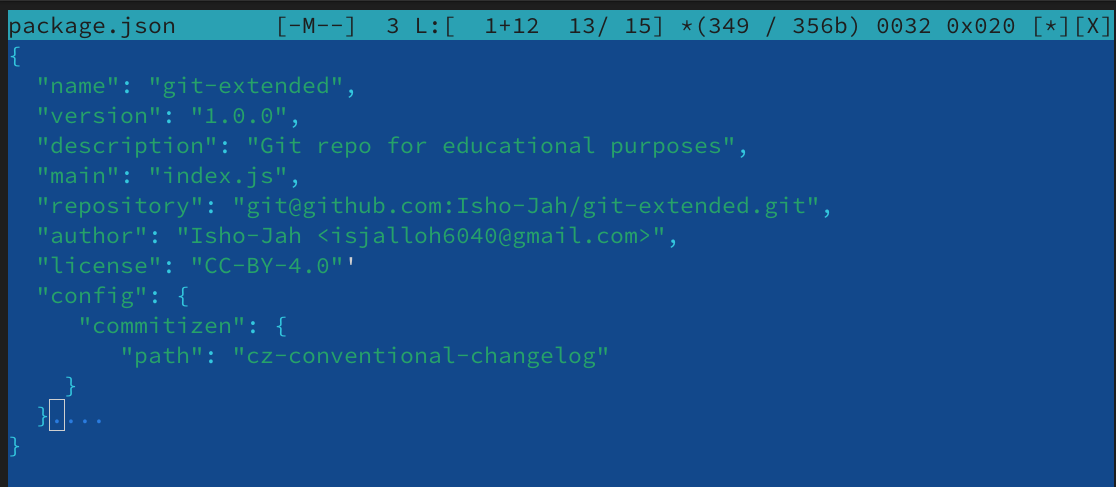


Рис. 14: Заполнение пакетов

Добавляю новые файлы и выполняю коммит (указиваю тип коммит (feat)) и отправляю на гит:

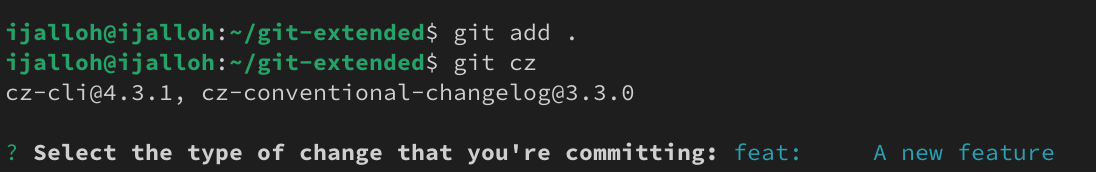


Рис. 15: Добавление новых файлов

Инициализирую git-flow и указываю ветки:

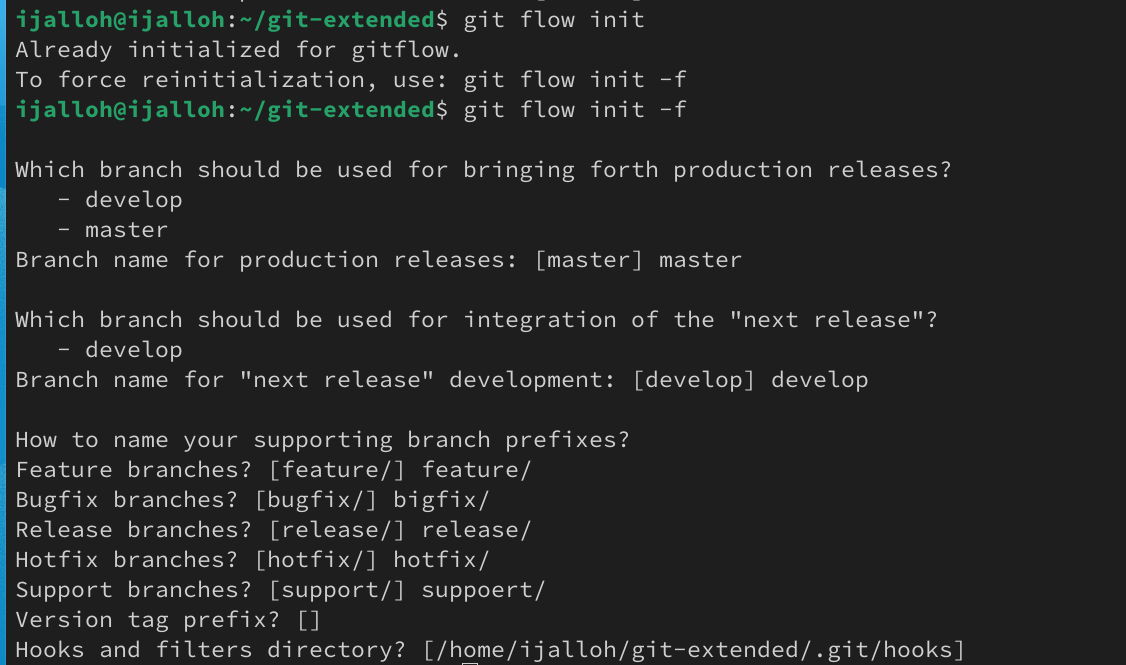


Рис. 16: Инициализирование git-flow

Преверяю что я на ветке develop с промощью git branch:

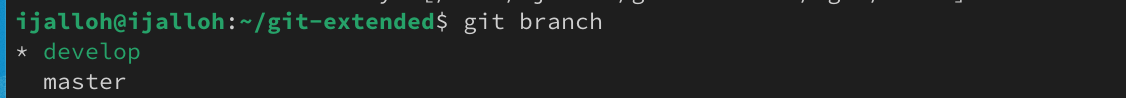


Рис. 17: Проверка ветки

Загружаю весь репозиторий в хранилище с помощью git push –all:

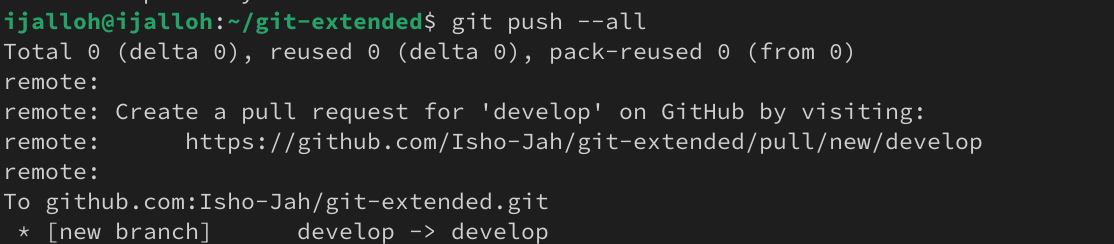


Рис. 18: Загрузка репозиторий

Установливаю внешнюю ветку как вышестоящую для этой ветки (develop):

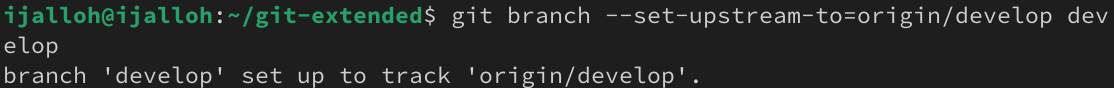


Рис. 19: установка внешней ветки

Создаю релиз с версией 1.0.0 и журнал изменений (standard-changelog)::

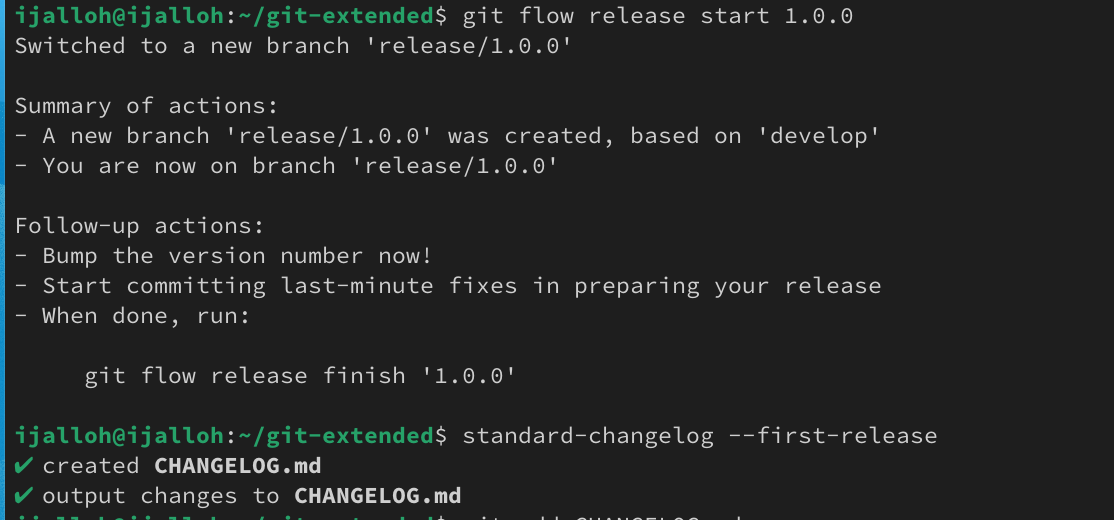


Рис. 20: Создание релиза

Добавляю журнал изменений в индекс:

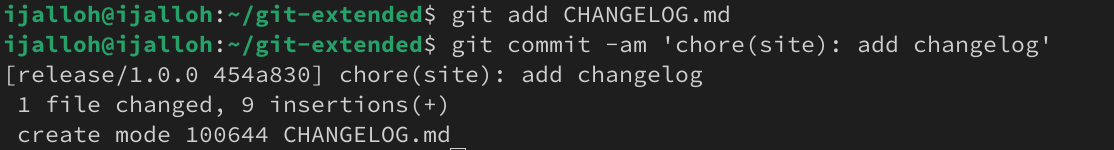


Рис. 21: Добавление журнала изменений

Залью релизную ветку в основную ветку:

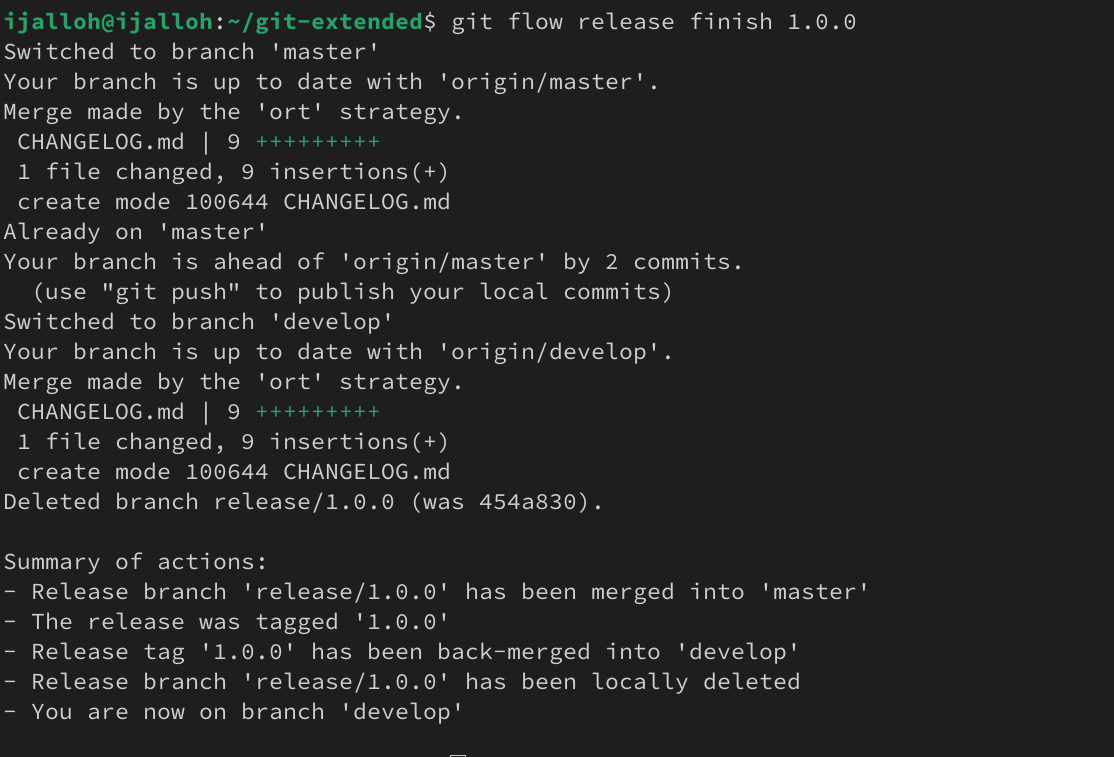


Рис. 22: Замена ветки

Отправляю данные на github:

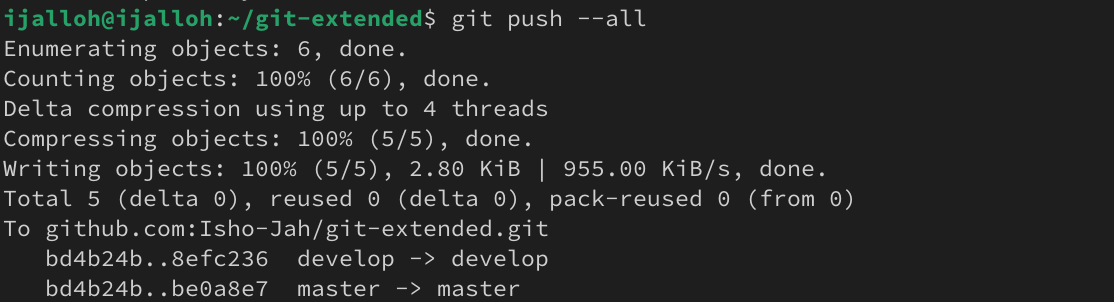


Рис. 23: Отправка на гит

Создаю релиз на github. Для этого использую утилиты работы с github gh (gh release create):

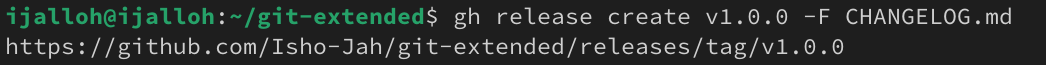


Рис. 24: Создание релиза

Создаю ветку для новой функциональности:

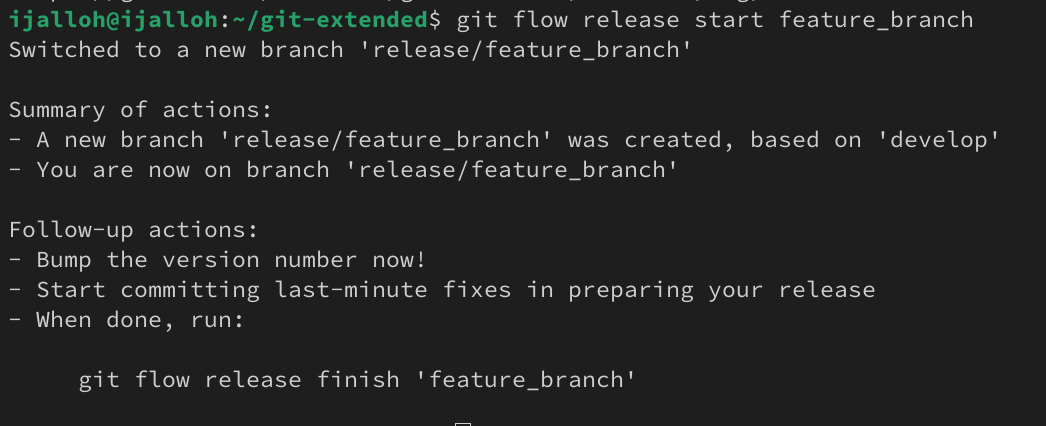


Рис. 25: Создание ветки feature\_branch

Далее, продолжаю работу c git как обычно. Создаю релиз с версией 1.2.3:

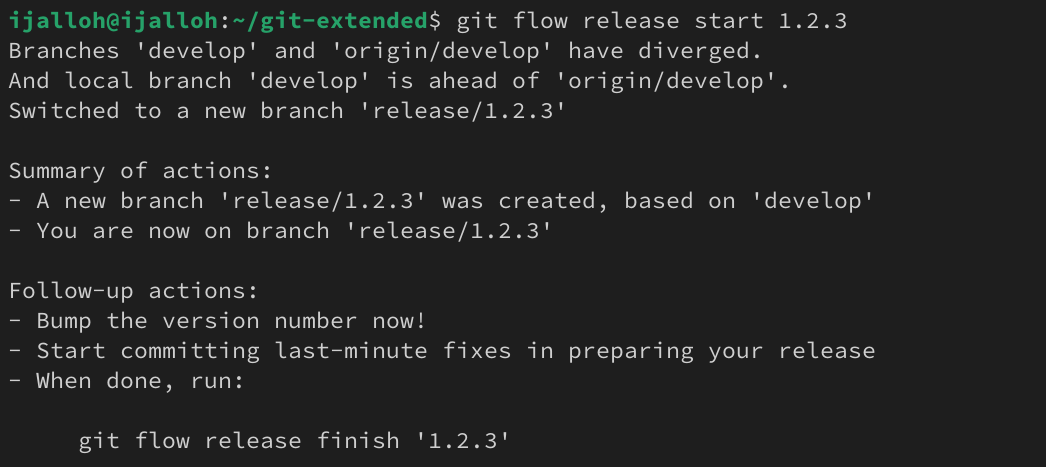


Рис. 26: Создание релиза с версией 1.2.3

Обновляю номер версии в файле package.json в 1.2.3:

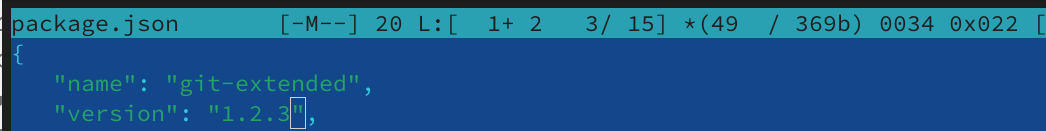


Рис. 27: Обновление номера версии

Создаю журнал изменений (standard-changelog):

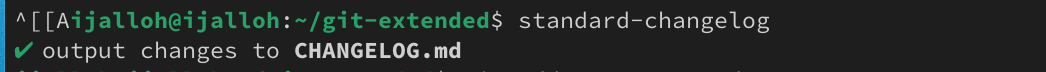


Рис. 28: Создание нового журнала изменений

Добавляю журнал изменений в индекс:

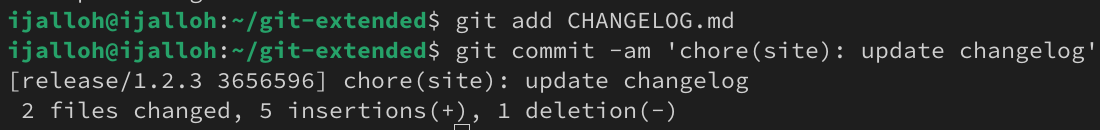


Рис. 29: Добавление журнала

Залью релизную ветку в основную ветку:

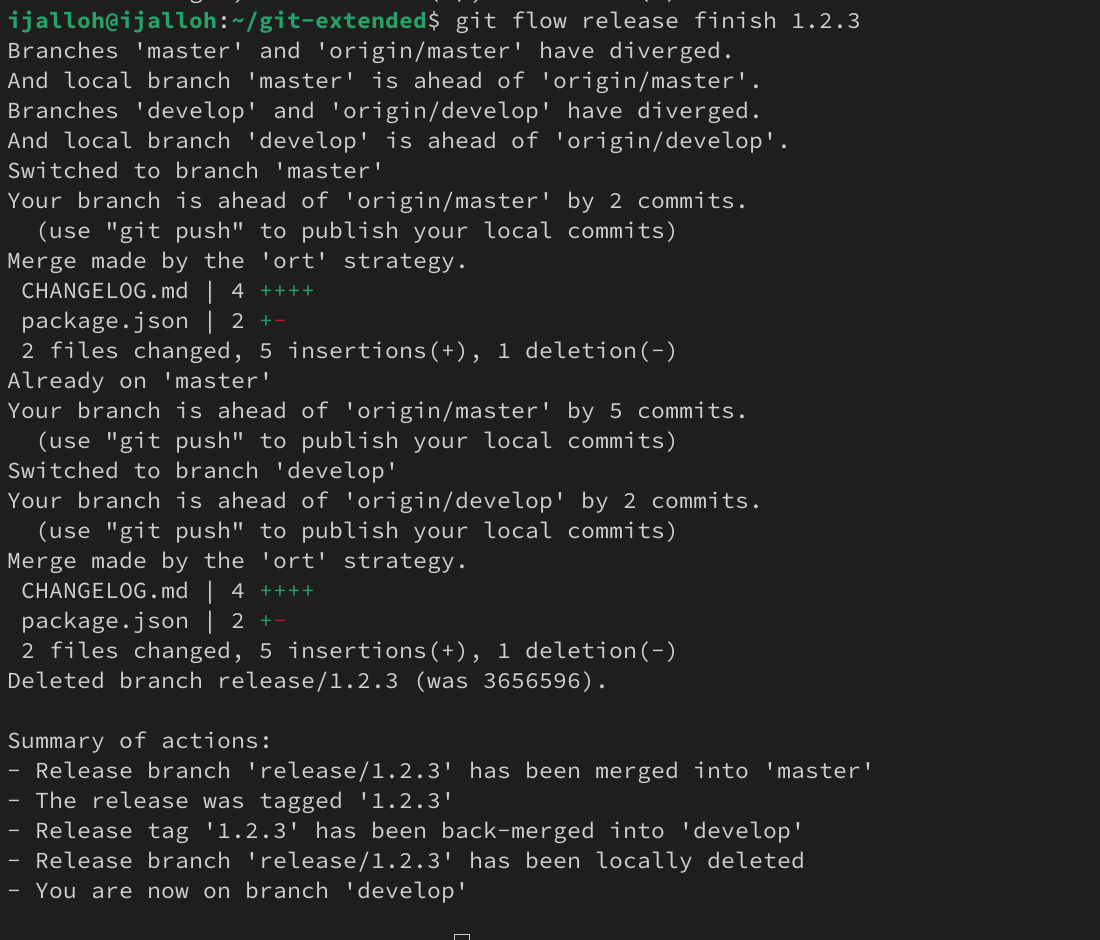


Рис. 30: Замена ветки

Отправляю данные на github:

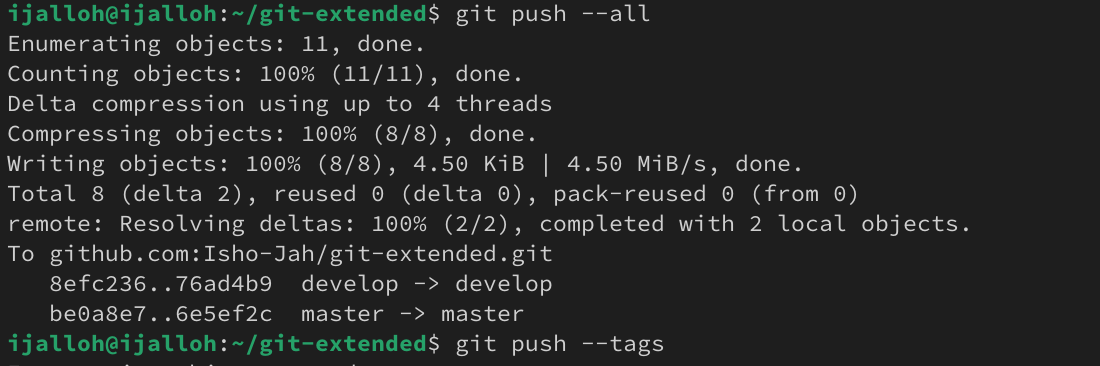


Рис. 31: Отправка данных

Создаю релиз на github с комментарием из журнала изменений:

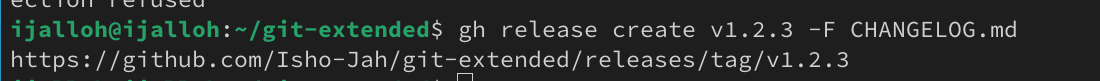


Рис. 32: Cоздание релиза

# 4 Выводы

При выполнение работы я получила навыки правильной работы с репозиториями git.

# Список литературы

[Архитектура ЭВМ](https://esystem.rudn.ru/mod/page/view.php?id=1098794&forceview=1)