Лабораторная работа №4

Продвинутое использование git

Карпова Есения Алексеевна

Содержание

# 1 Цель работы

Получение 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 Теоретическое введение

Рабочий процесс Gitflow:

Рабочий процесс Gitflow Workflow. Будем описывать его с использованием пакета git-flow.

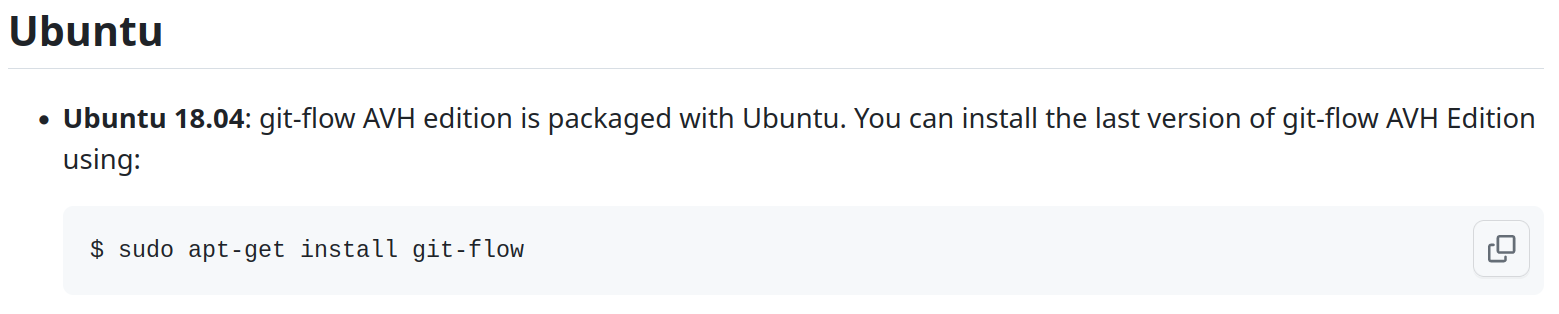
Общая информация:

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. Работа по модели Gitflow включает создание отдельной ветки для исправлений ошибок в рабочей среде. 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

1. Выполнение работы для тестового репозитория

Для начала нужно установить необходимое ПО,в моем случае для ubuntu. Ввожу нужную команду для устновки git-flow (рис. [??]).



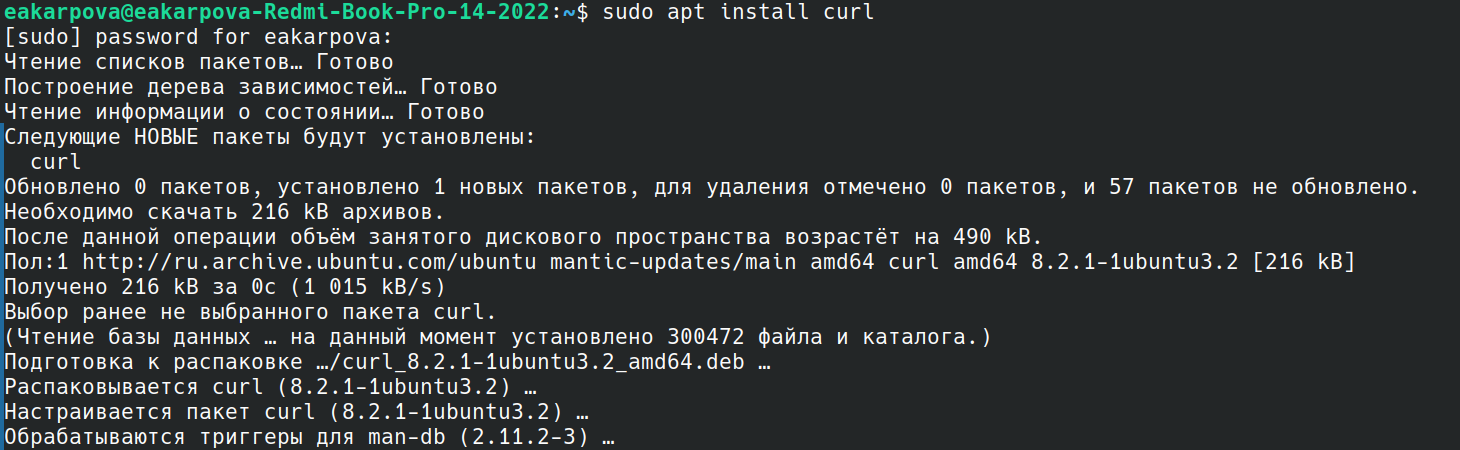
Установка git-flow

Ввожу нужную команду для установки Node.js (рис. [??]).



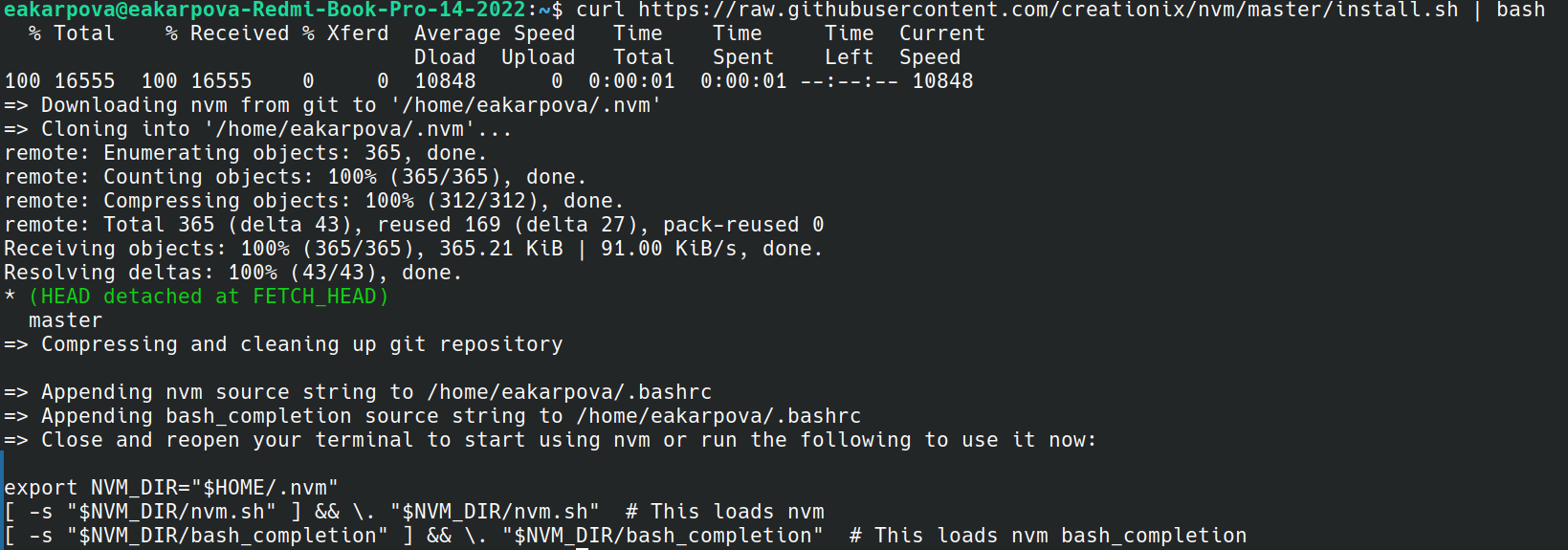
Установка Node.js

Также мне понадобится curl для работы с данными сервера (рис. [??]).



Установка curl

Использую curl без аргументов для выполнения HTTP-запроса (рис. [??]).



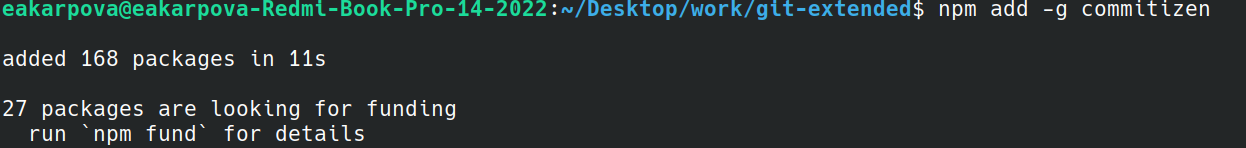
HTTP-запрос

Ввожу команду для перезагрузки (рис. [??]).

Ввод команды

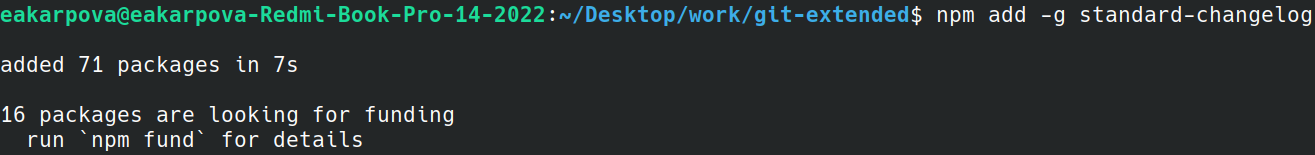
Ввод команды

Для формотирования коммитов и скрипта git-cz использую приведенную ниже программу (рис. [??]).



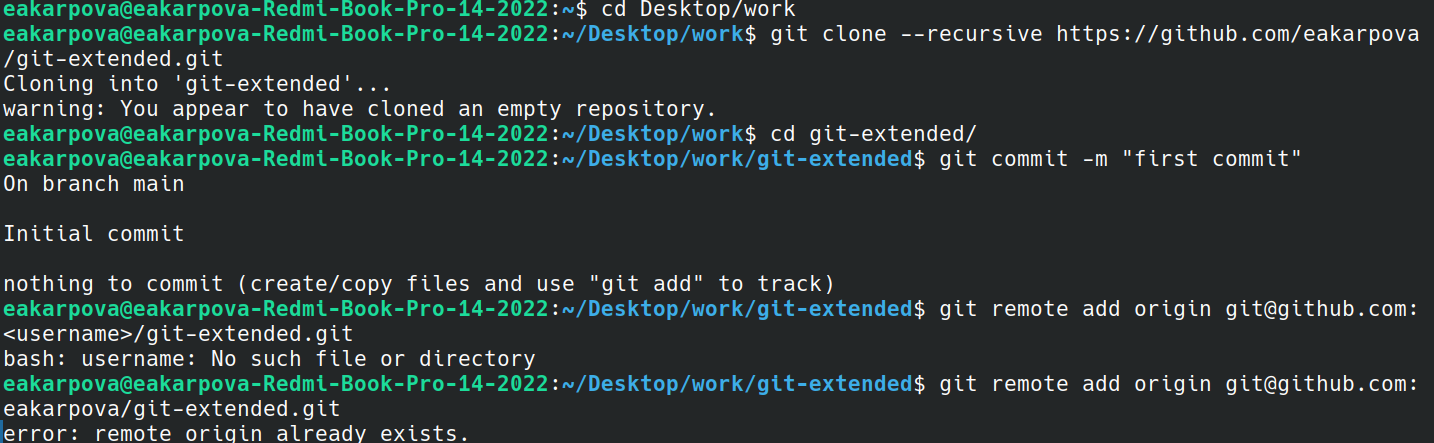
Commitizen

Для создания логов использую приведенную ниже программу (рис. [??]).



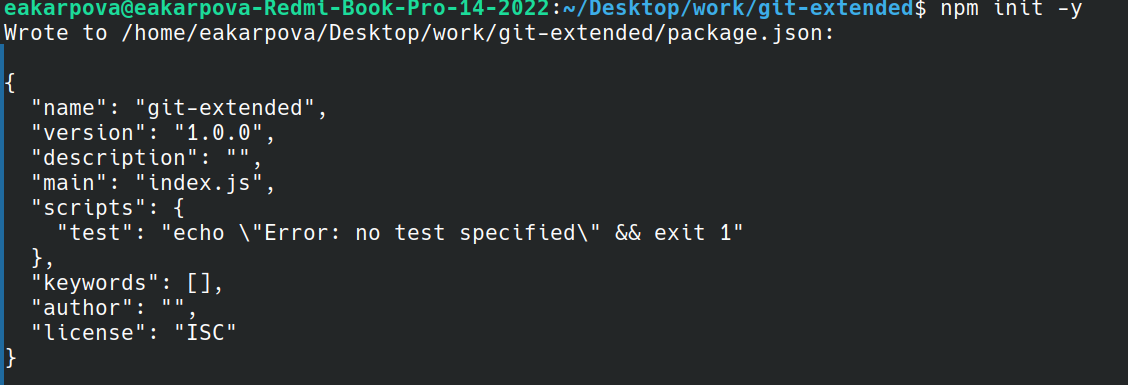
Standard-changelog

Создаю репозиторий git-extended, делаю первый коммит и выкладываю на github(рис. [??]).



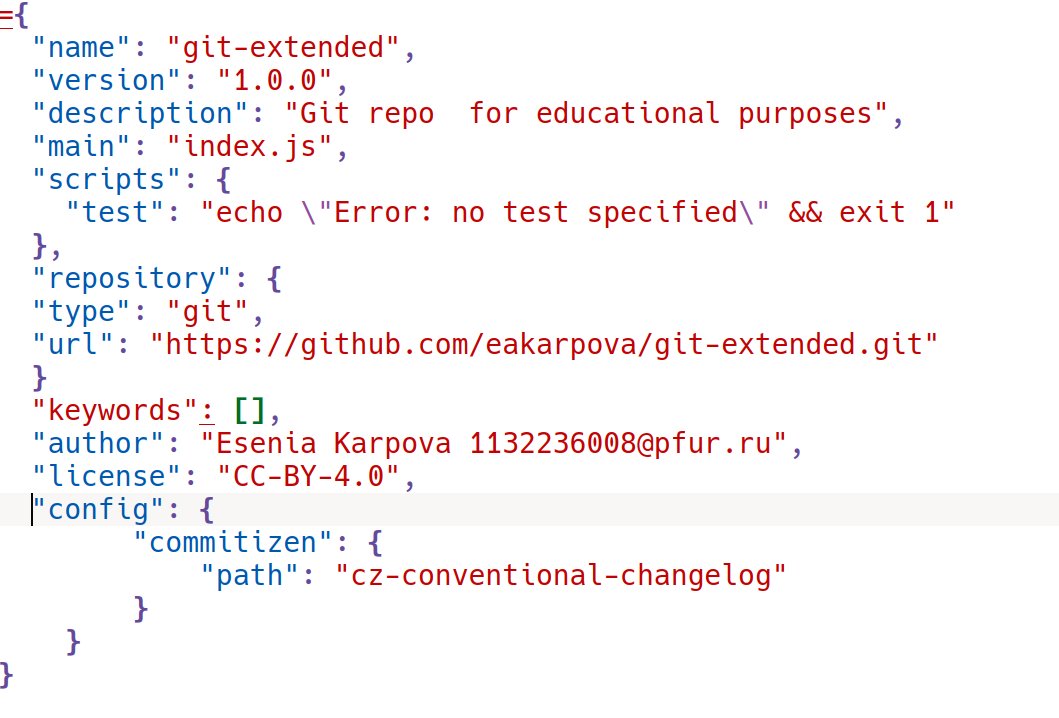
Создание репозитория git

Для конфигурации общепринятых коммитов ввожу команду (рис. [??]).

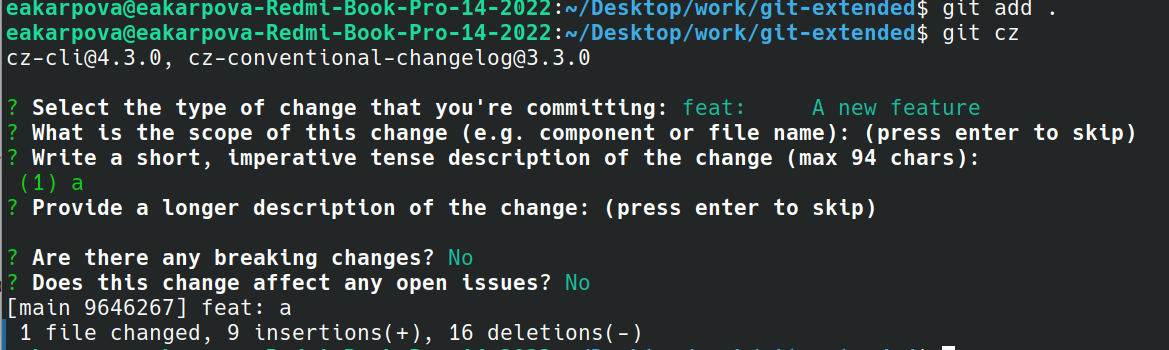


Конфигурация для пакетов Node.js

Для того чтобы сконфигурировать формат коммитов добавим в исходный файл команду для формирования коммитов (рис. [??]).

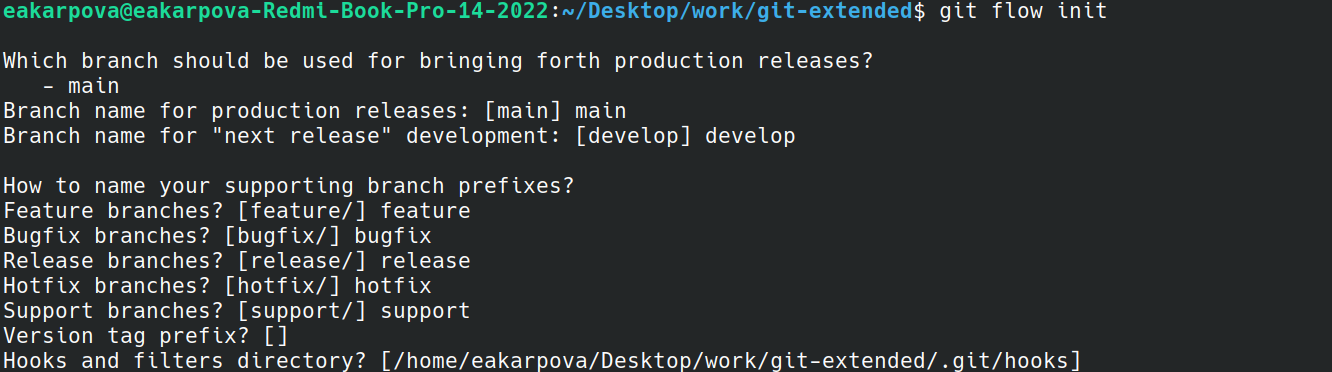
 width=100%}

Добвыим новые файлы и выполним коммит, после чего отправим на github(рис. [??]).



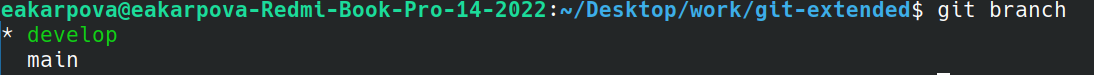
Добавление новых файлов

Инициализируем git-flow (рис. [??]).



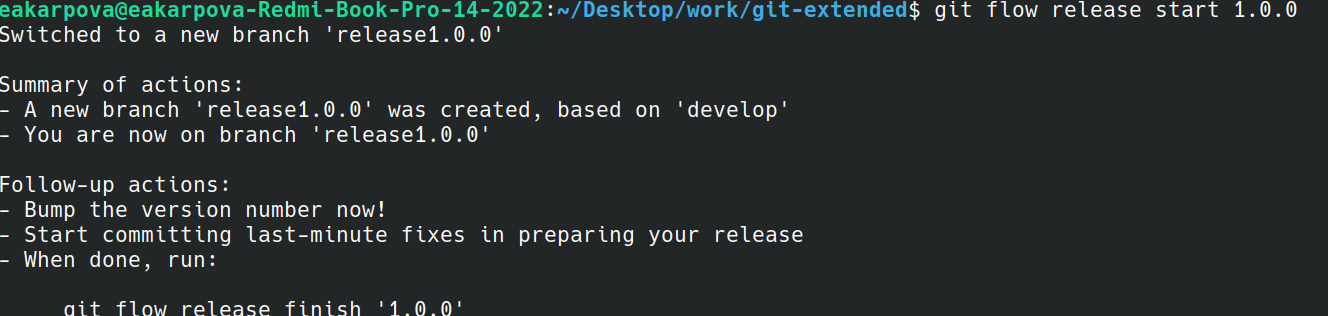
Конфигурация git-flow

Проверим, что находимся на ветке develop (рис. [??]).



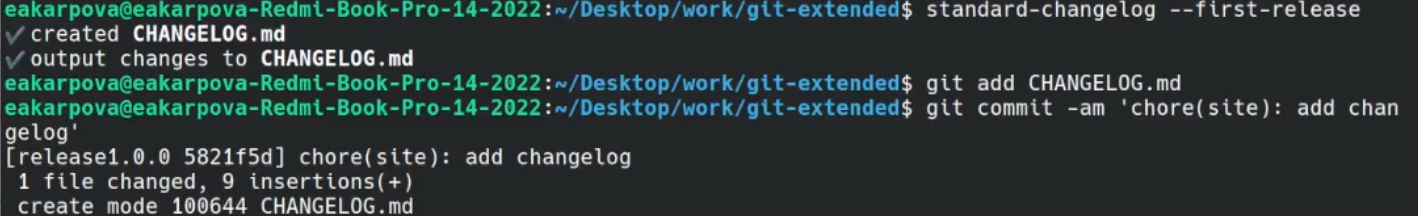
Проверка нахождения на ветке develop

Загрузим весь репозиторий в хранилище и установим внешнюю ветку как вышестоящую, после чего создадим релиз с версией 1.0.0. (рис. [??]).



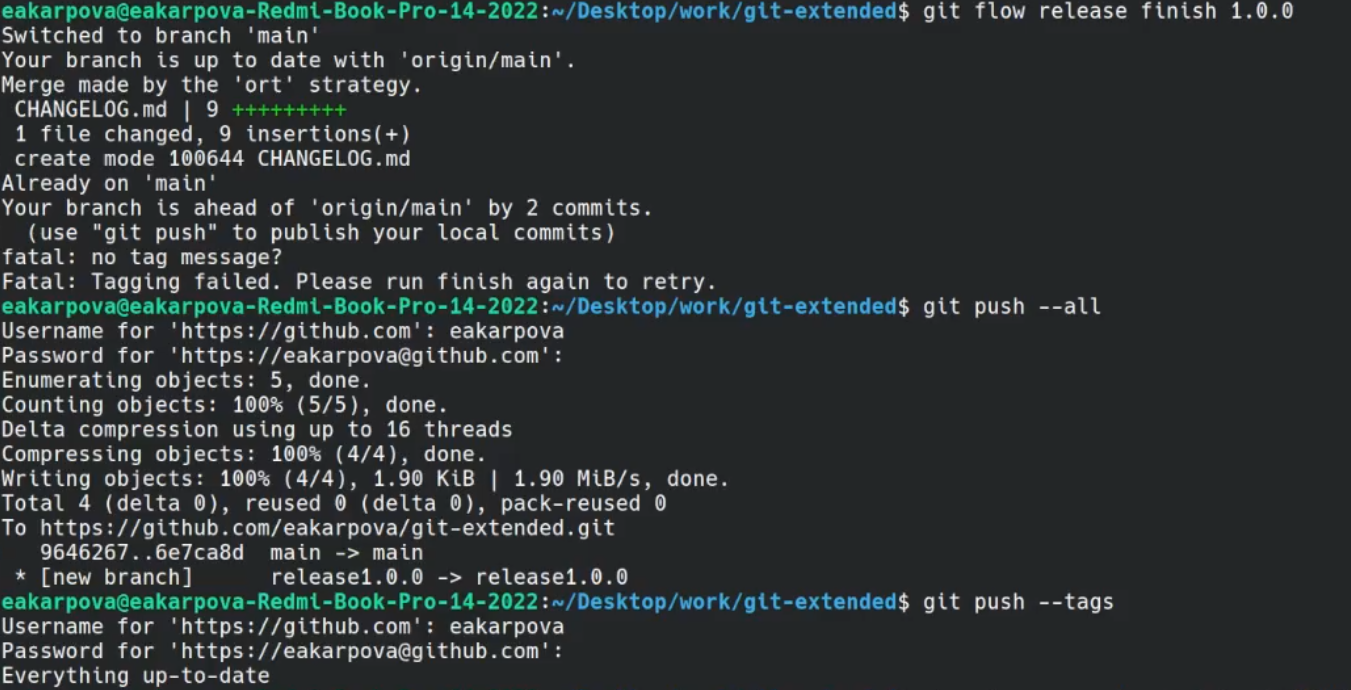
Создание релиза

Создаем журнал изменений и добавляем его в индекс (рис. [??]).



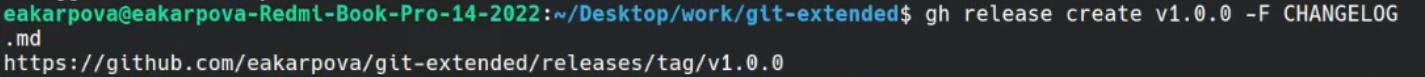
Создание журнала изменений

Зальем релизную ветку в основную и отправим данные на github (рис. [??]).



Залив релиза

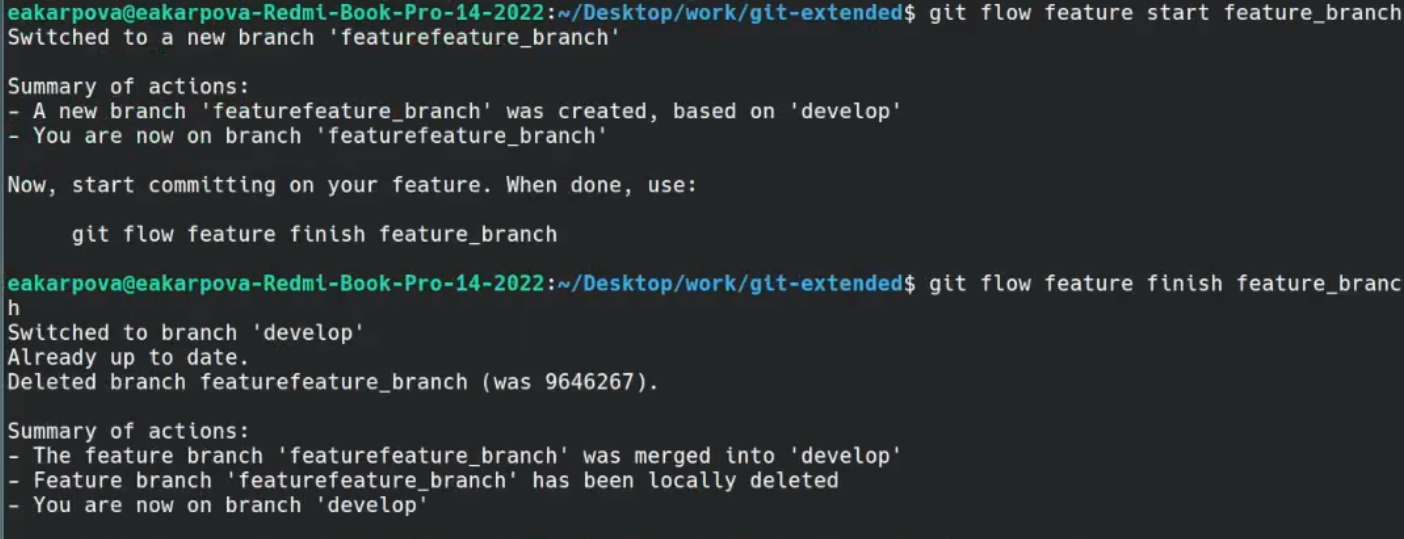
Создадим релиз на github с помощью утилиты (рис. [??]).



Создание релиза на github

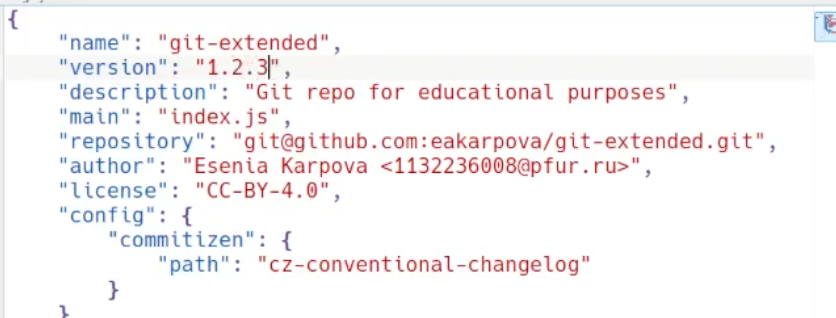
1. Преобразовывание рабочего репозитория в репозиторий с git-flow и conventional commits.

Создадим ветку для новой функциональности (рис. [??]).



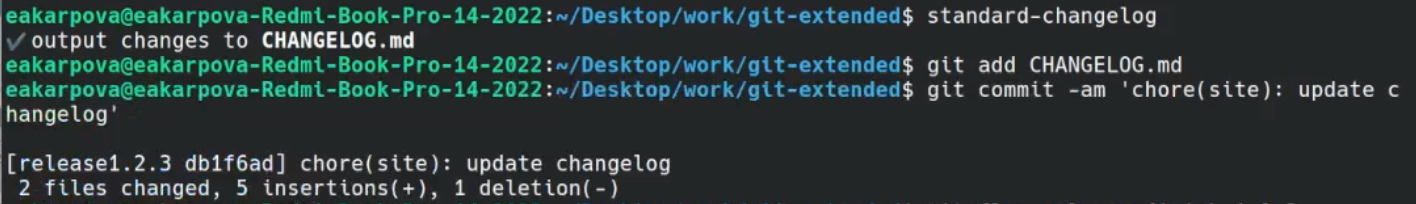
Создание ветки

Создадим релиз с версией 1.2.3 и обновим номер версии в файле package.json (рис. [??]).



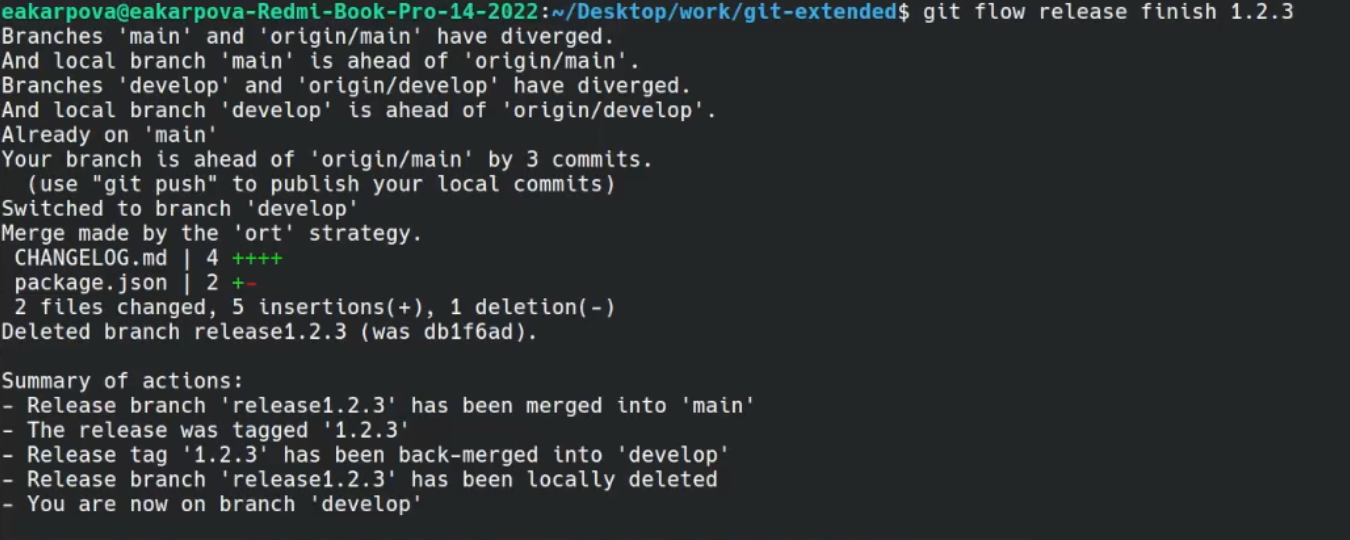
Обновление версии в файле

Создадим журнал изменений и добавим его в индекс (рис. [??]).



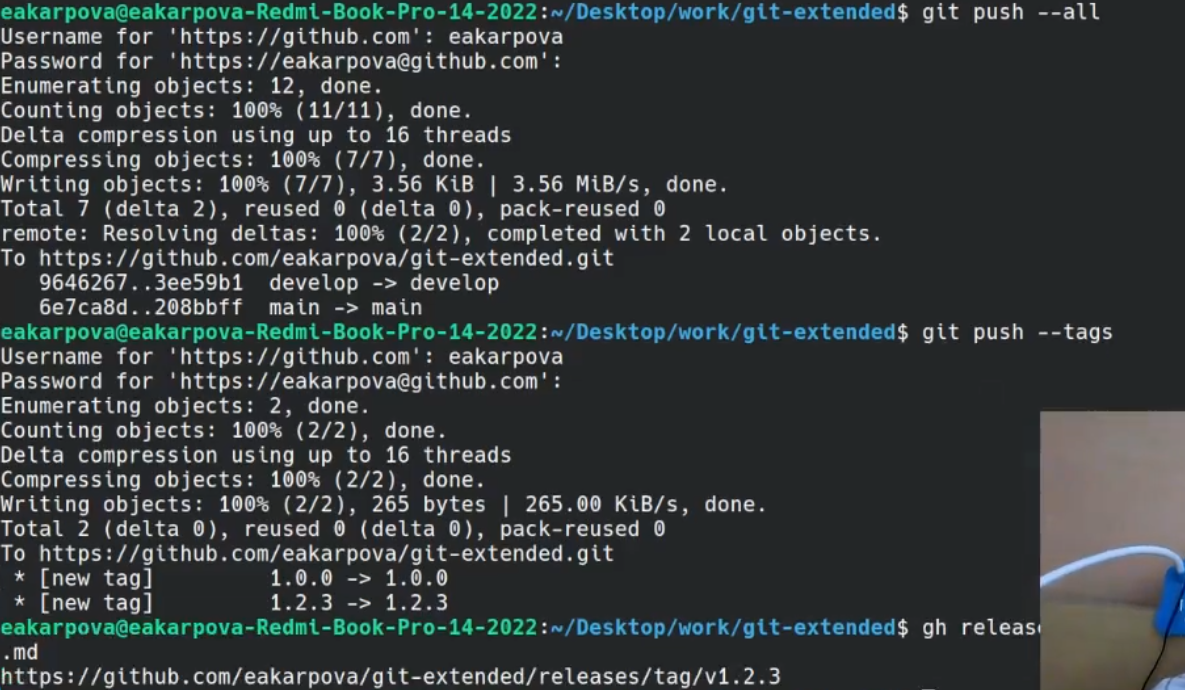
Создание журнала и добавление его в индекс

Зальем релизную ветку в основную (рис. [??]).



Залив релизной ветки

Отправим данные на github (рис. [??]).



Отправка данных на github

(рис. [??]).

# 5 Выводы

Я получила навыки правильной работы с репозиториями git