Отчёт по лабораторной работе №4

Операционные системы

Чистов Даниил Максимович

Содержание

# 1 Цель работы

Целью данной работы является 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 Выполнение лабораторной работы

## 3.1 Установка git-flow

Для установки gitflow для начала требуется подключить специальный репозиторий (рис. 1).

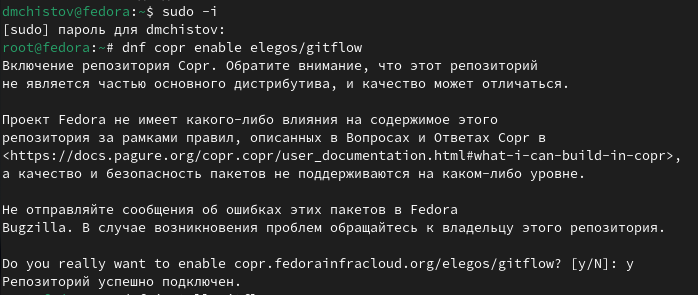


Рис. 1: Подключение репозитория

После успешного подключения, устанавливаю gitflow (рис. 2).

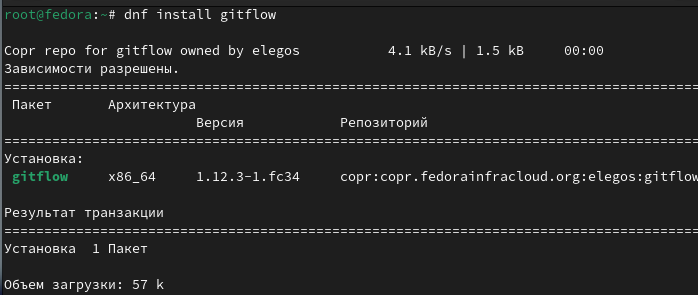


Рис. 2: Установка gitflow

На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов. Устанавливаю nodejs (рис. 3).

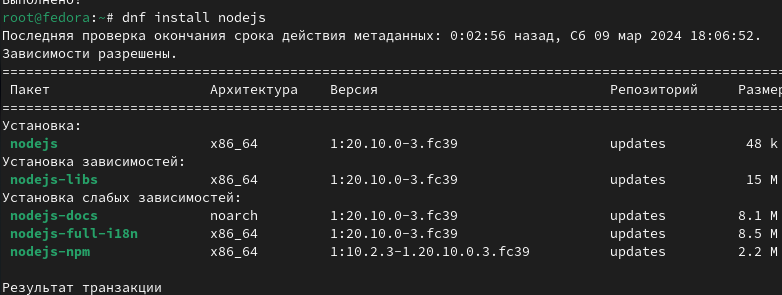


Рис. 3: Установка nodejs

Также устанавливаю pnpm, который требуется для работы nodejs (рис. 4).

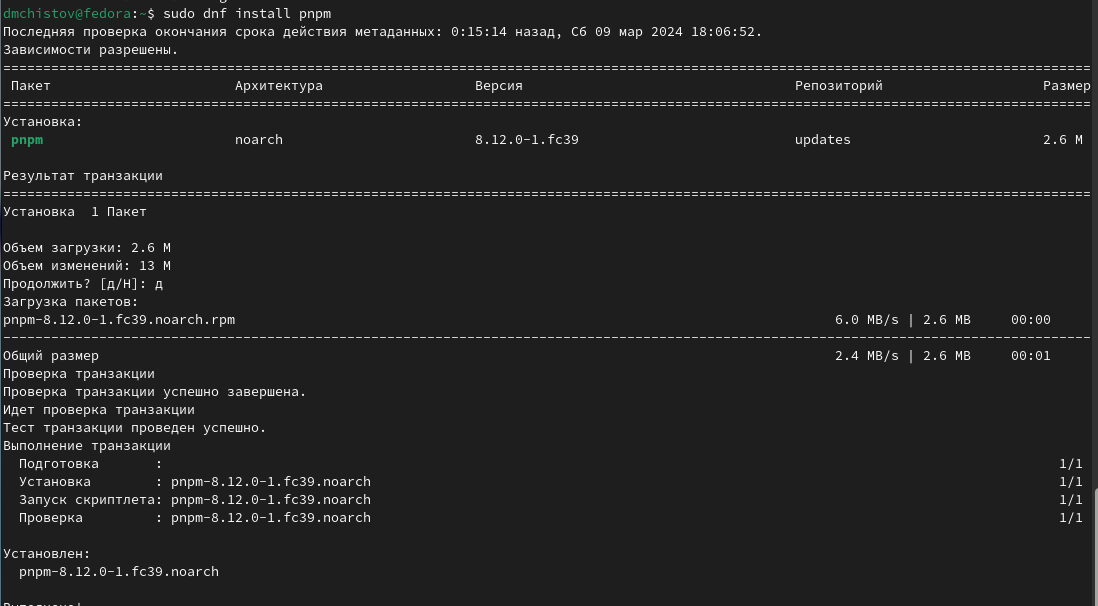


Рис. 4: Установка pnpm

## 3.2 Настройка NodeJS

Запускаю настройку pnpm, после чего перелогиниваюсь (рис. 5).

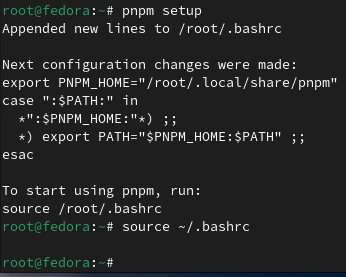


Рис. 5: Настройка pnpm

## 3.3 Общепринятые коммиты

“pnpm add -g commitizen” используется для помощи в форматировании коммитов. При этом устанавливается скрипт git-cz, который и будет использоваться для коммитов (рис. 6).

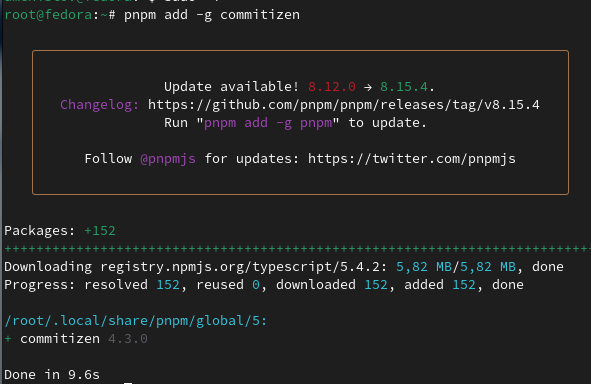


Рис. 6: Один из общепринятых коммитов

“standard-changelog” используется для помощи в создании логов (рис. 7).

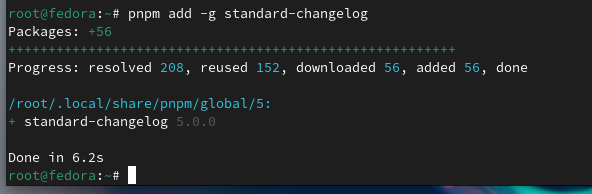


Рис. 7: Ещё один из общепринятых коммитов

## 3.4 Практический сценарий использования git

Требуется создать репозиторий, захожу на сайт GitHub и создаю репозиторий под названием “git-extended” (рис. 8).

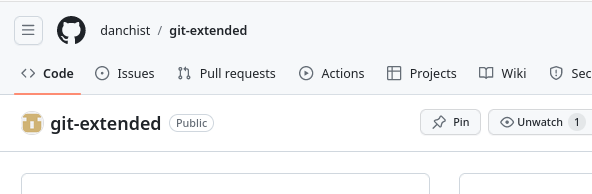


Рис. 8: Новый репозиторий на GitHub

Теперь создаю репозитори локально у себя на виртульной машине, также внутри создаю файл README.md (рис. 9).

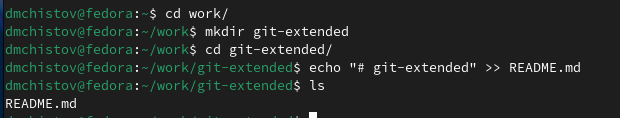


Рис. 9: Новый репозиторий

Инициализирую новый репозиторий и вношу изменения (рис. 10).

Инициализация нового репозитория

Рис. 10: Инициализация нового репозитория

Делаю первый коммит и выкладываю его на GitHub (рис. 11).

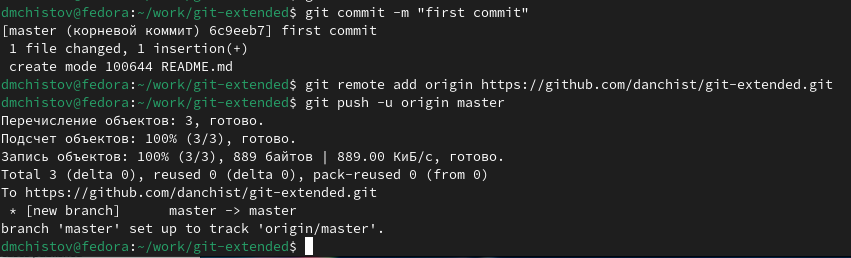


Рис. 11: Первый коммит

Репозиторий успешно прошёл конфигурацию (рис. 12).

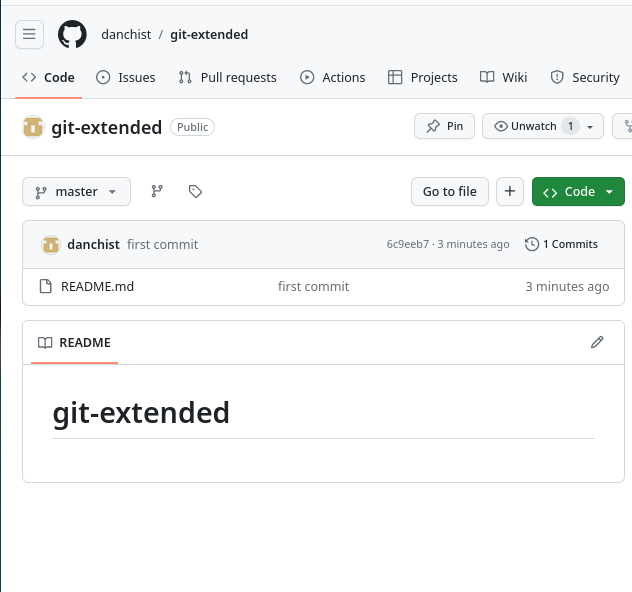


Рис. 12: Новый репозиторий на GitHub

Начинаю конфигурацию пакетов Node.js (рис. 13).

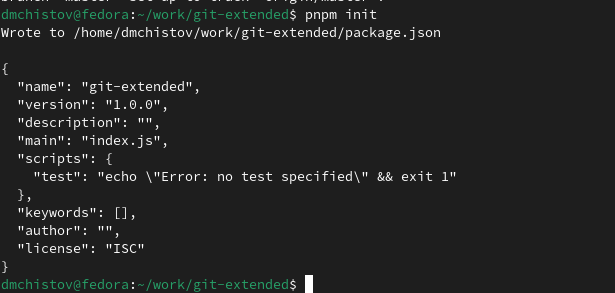


Рис. 13: Конфигурация Node.js

Был создан новый файл package.json. Его нужно отредактировать, как заявлено в задании. Меняю версию на “CC-BY-4.0”, оставляю свои персональные данные - имя, фамилию, почту, ссылку на GitHub (рис. 14).

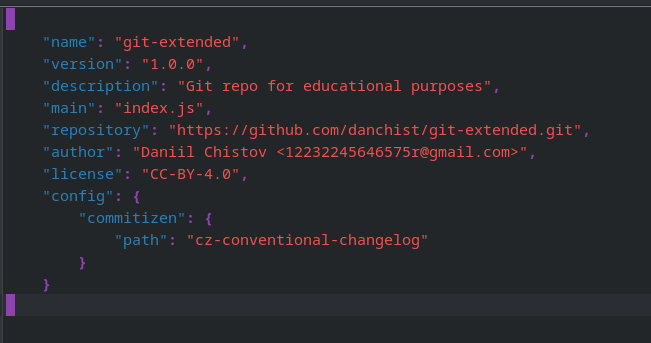


Рис. 14: Изменение файла

Добавляю новые файлы и выполняю коммит, пишу, что изменения не масштабные и указываю новый файл. После всего этого, выполняю git push (рис. 15).

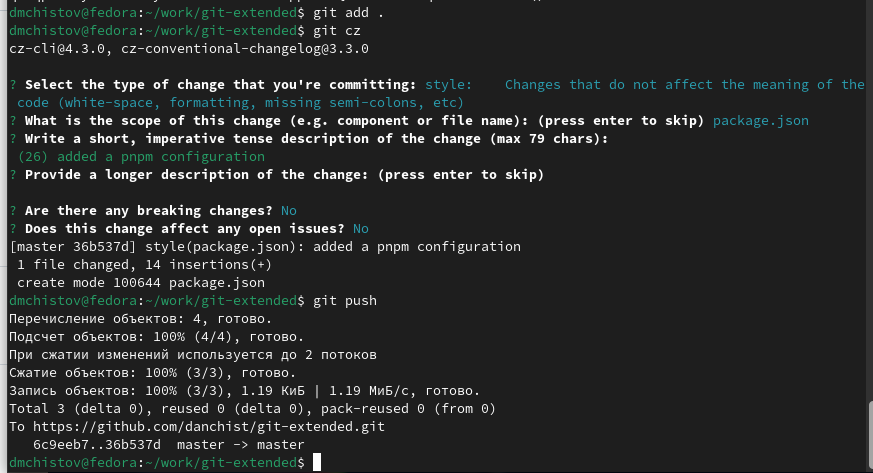


Рис. 15: Выполняю коммит

## 3.5 Конфигурация gitflow

Инициализирую gitflow, указываю префикс “v” для ярлыков (рис. 16).

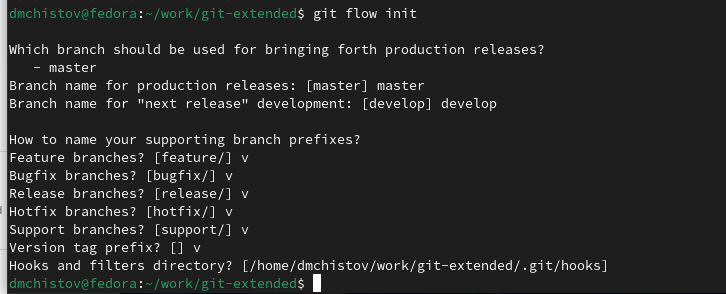


Рис. 16: Конфигурация gitflow

Удостоверяюсь, что я нахожу в ветке develop. Всё точно (рис. 17).

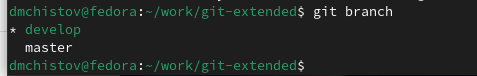


Рис. 17: Текущая ветка

Загружаю весь репозиторий в хранилище (рис. 18).

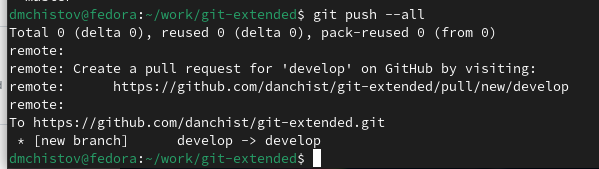


Рис. 18: Загрузка репозитория

Установливаю внешнюю ветку как вышестоящую для этой ветки (рис. 19).

Работа с ветками

Рис. 19: Работа с ветками

Создадаю релиз с версией 1.0.0 (рис. 20).

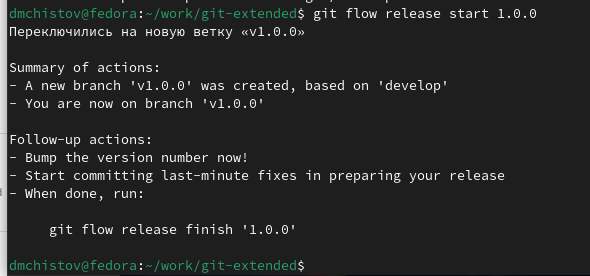


Рис. 20: Новый релиз - 1.0.0.

Создаю журнал изменений (рис. 21).

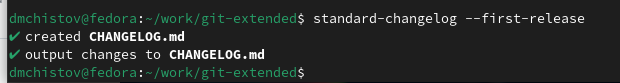


Рис. 21: Журнал изменений - 1.0.0.

Добавляю журнал изменений в индекс (рис. 22).

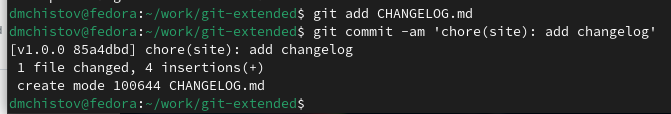


Рис. 22: Добавление журнала изменений

Загружаю релизную ветку в основную ветку, от меня требуют описать изменения (рис. 23).

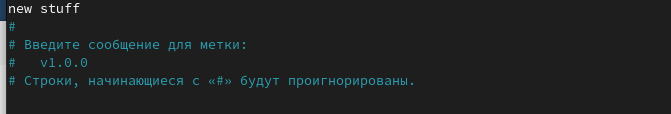


Рис. 23: Описание изменений

Релизная ветка успешно загружена (рис. 24).

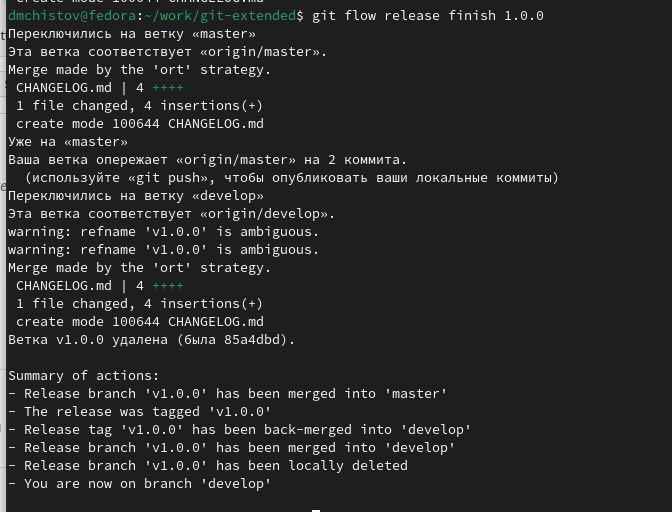


Рис. 24: Успех

Отправляю данные на GitHub - пишу “git push –all” (рис. 25), а затем “git push –tags” (рис. 26).

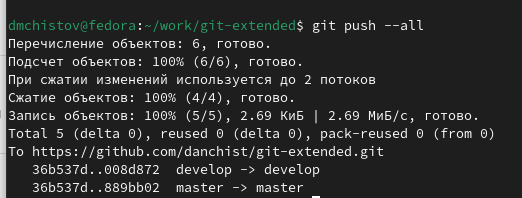


Рис. 25: git push –all

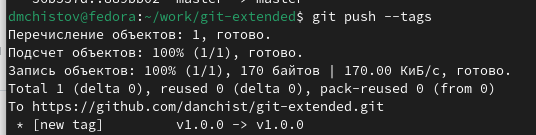


Рис. 26: git push –tags

Создаю релиз на GitHub (рис. 27).

Релиз 1.0.0 на GitHub

Рис. 27: Релиз 1.0.0 на GitHub

## 3.6 Пример работы с репозиторием git

Создаю ветку для новой функциональности (рис. 28).

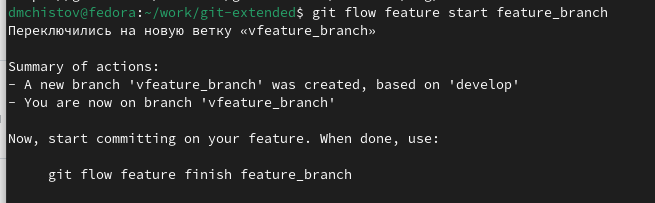


Рис. 28: Ветка новой функциональности

Допустим, я сделал какую-нибудь новую функцию. После окончанию своей работы, завершаю ветку новой функциональности - объединяю её с веткой develop (рис. 29).

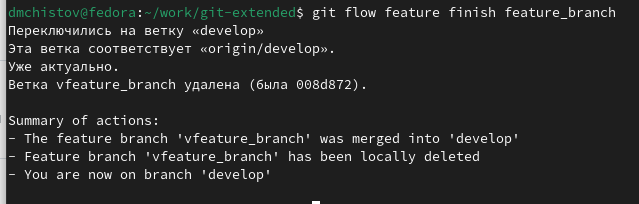


Рис. 29: Завершение ветки новой функциональности

Создаю новый релиз на Gitflow - 1.2.3 (рис. 30).

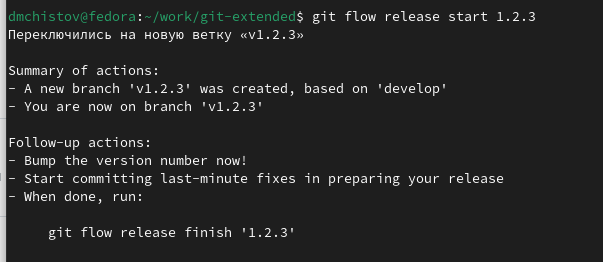


Рис. 30: Релиз на Gitflow - 1.2.3

Обновляю номер версии в файле package.json на 1.2.3 (рис. 31).

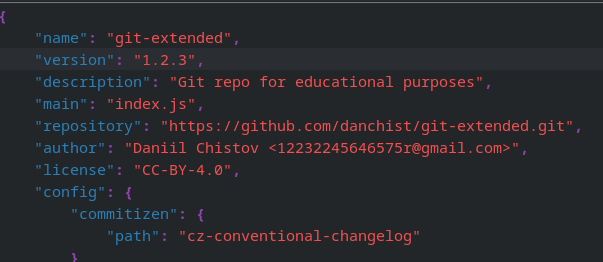


Рис. 31: Новый файл package.json

Создаю журнал изменений, добавляю его в индекс, делаю новый коммит (рис. 32).

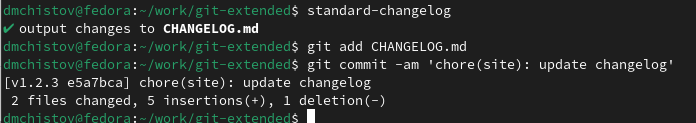


Рис. 32: Новый коммит с журналом изменений

Загружаю релизную ветку в основную. Описываю изменения - “новый номер” (рис. 33).

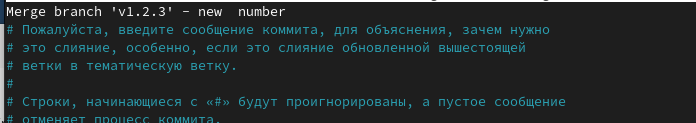


Рис. 33: “Новый номер”

Добавляю подпись к изменениям (рис. 34).

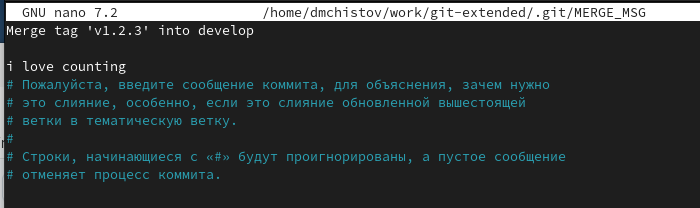


Рис. 34: “Новый номер”

Ветка успешно загружена (рис. 35).

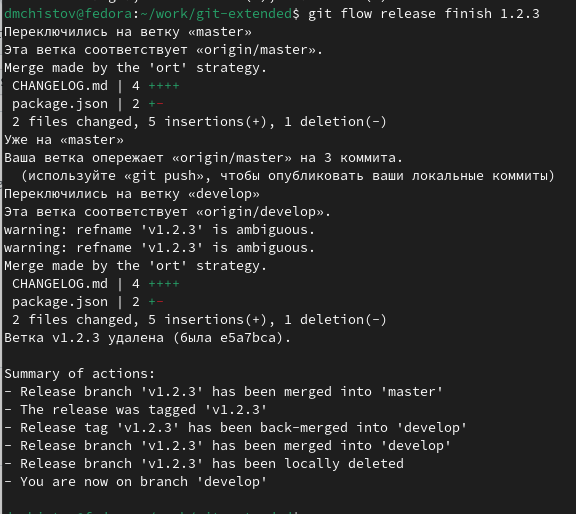


Рис. 35: Успешное слияние веток

Отправляю изменения на GitHub (рис. 36).

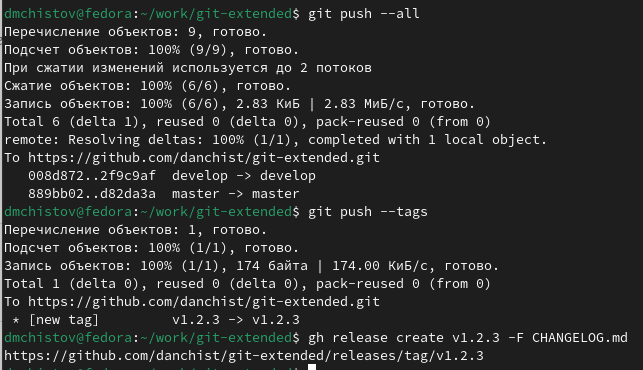


Рис. 36: Новый релиз на GitHub’е - 1.2.3

Задание выполнено успешно! (рис. 37).

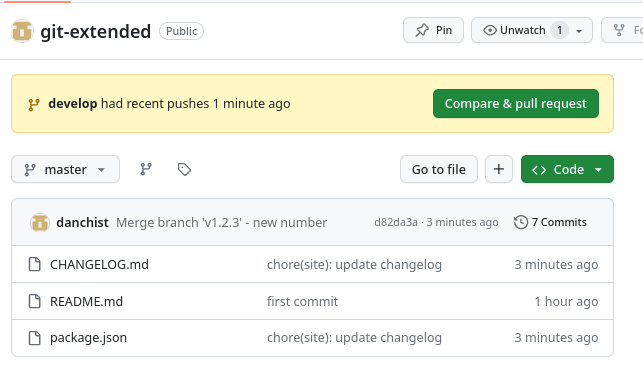


Рис. 37: Релиз успешно загружен

# 4 Выводы

Выполняя данную работу я получил навыки правильной работы с репозиториями git.

# Список литературы

[Лабораторная работы №2](https://esystem.rudn.ru/mod/page/view.php?id=1098794)