Отчёт по лабораторной работе №4

Дисциплина: Операционные системы

Филатов Илья Гурамович

Содержание

# 1 Цель работы

* Получить навыки правильной работы с репозиториями git

# 2 Задание

1. Выполнить работу для тестового репозитория

* Преобразовать рабочий репозиторий в репозиторий с git-flow и conventional commits

# 3 Выполнение лабораторной работы

## 3.1 Установка git-flow

Установим gitflow из коллекции репозиториев Copr (рис. 1).

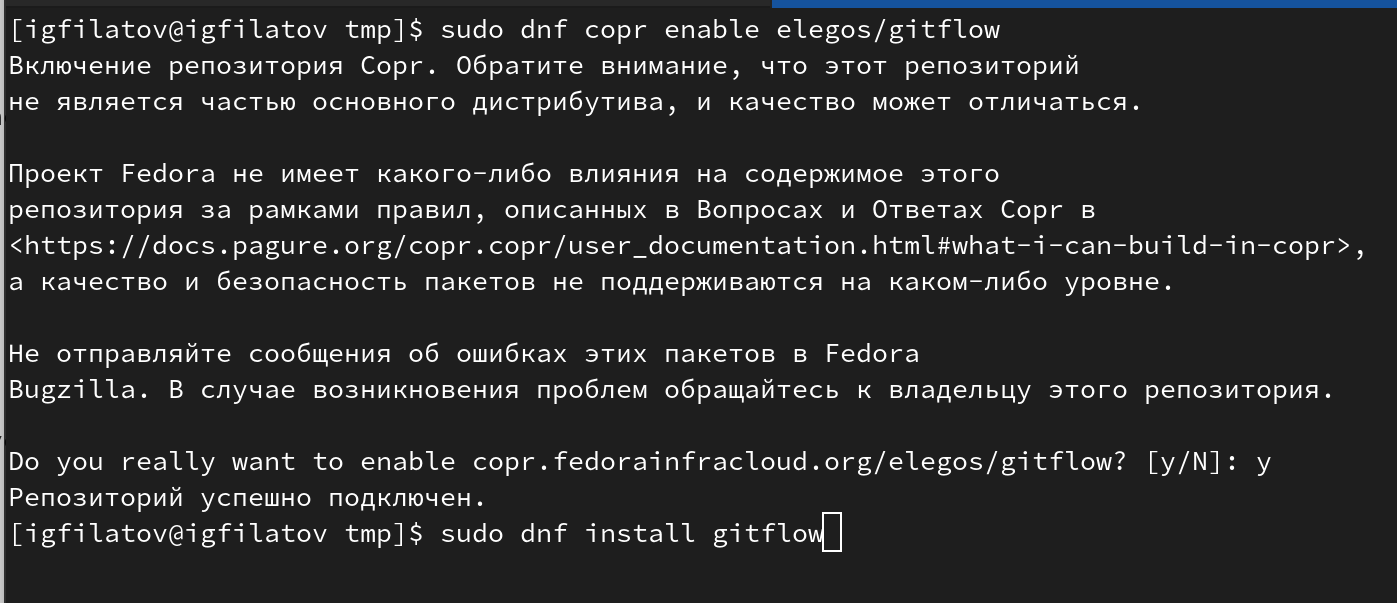


Рис. 1: Установка пакета gitflow

## 3.2 Установка Node.js

Также установим Node.js на котором базируется ПО для семантического версионирования коммитов (рис. 2).

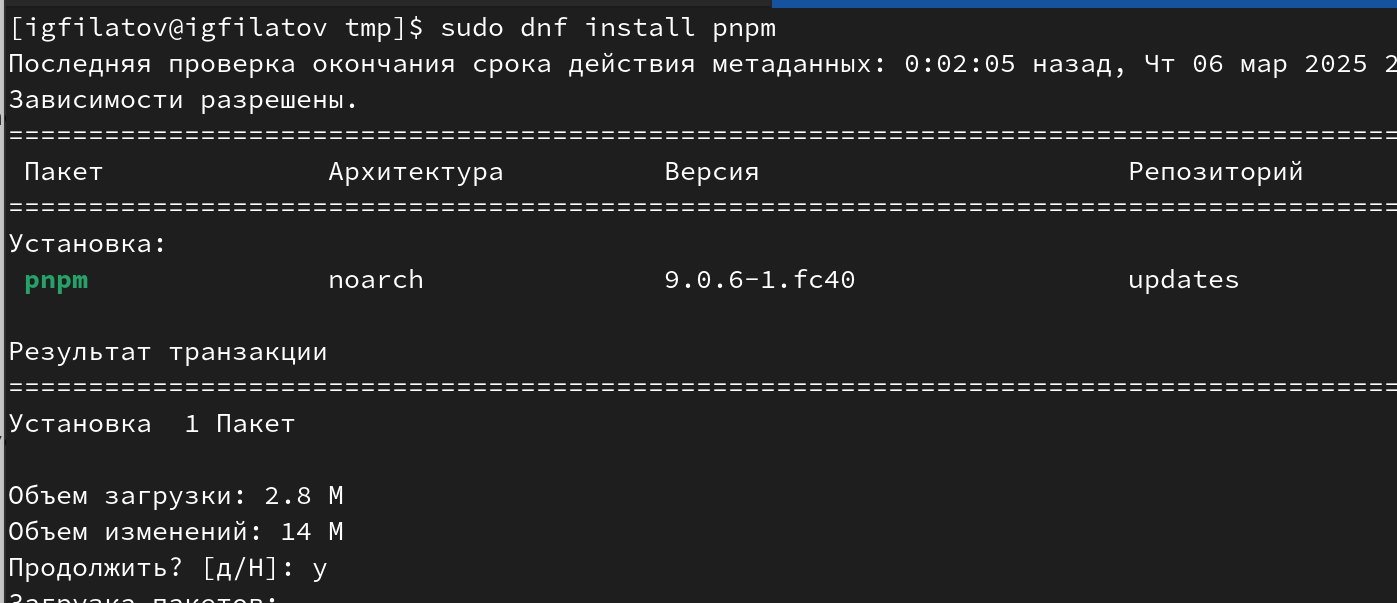


Рис. 2: Установка пакетов 2

## 3.3 Настройки Node.js

Используя комманду pnpm setup и source ~/.bashrc добавляем каталок с исполняемыми файлами от yarn в переменную PATH (рис. 3).

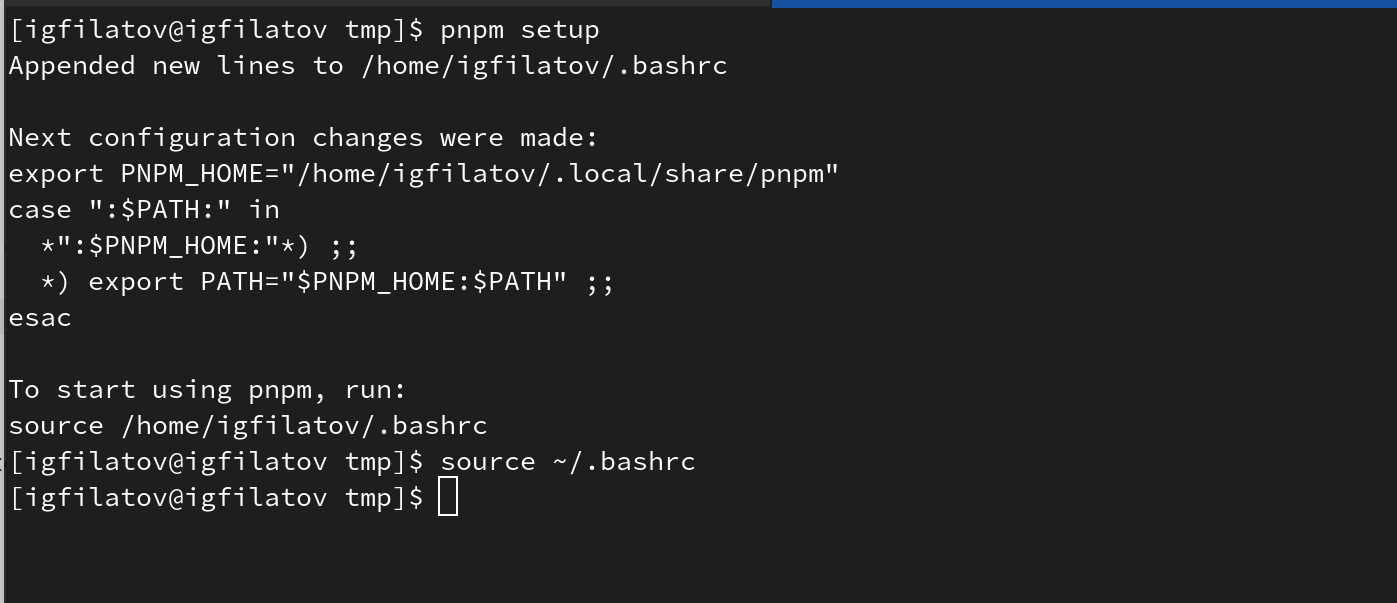


Рис. 3: Настройка Nodejs

Далее установим пакет commitizen для помощи в форматировании коммитов и пакет standard-changelog который используется для помощи в создании логов (рис. 4).

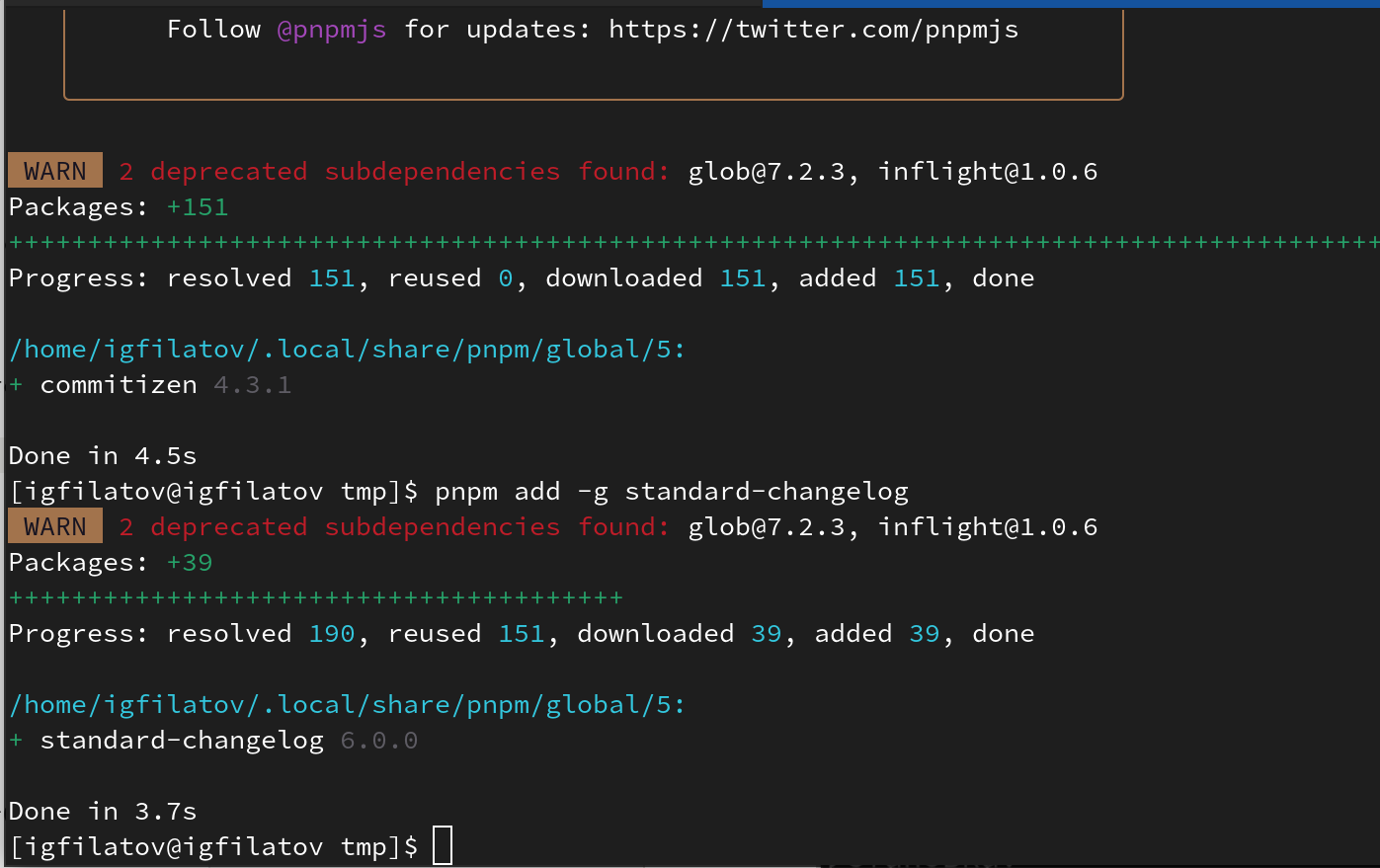


Рис. 4: Установка доп. пакетов

## 3.4 Практический сценарий использования git

### 3.4.1 Создание репозитория github

Создадим учебный репозиторий на github и сделаем первый коммит (рис. 5).

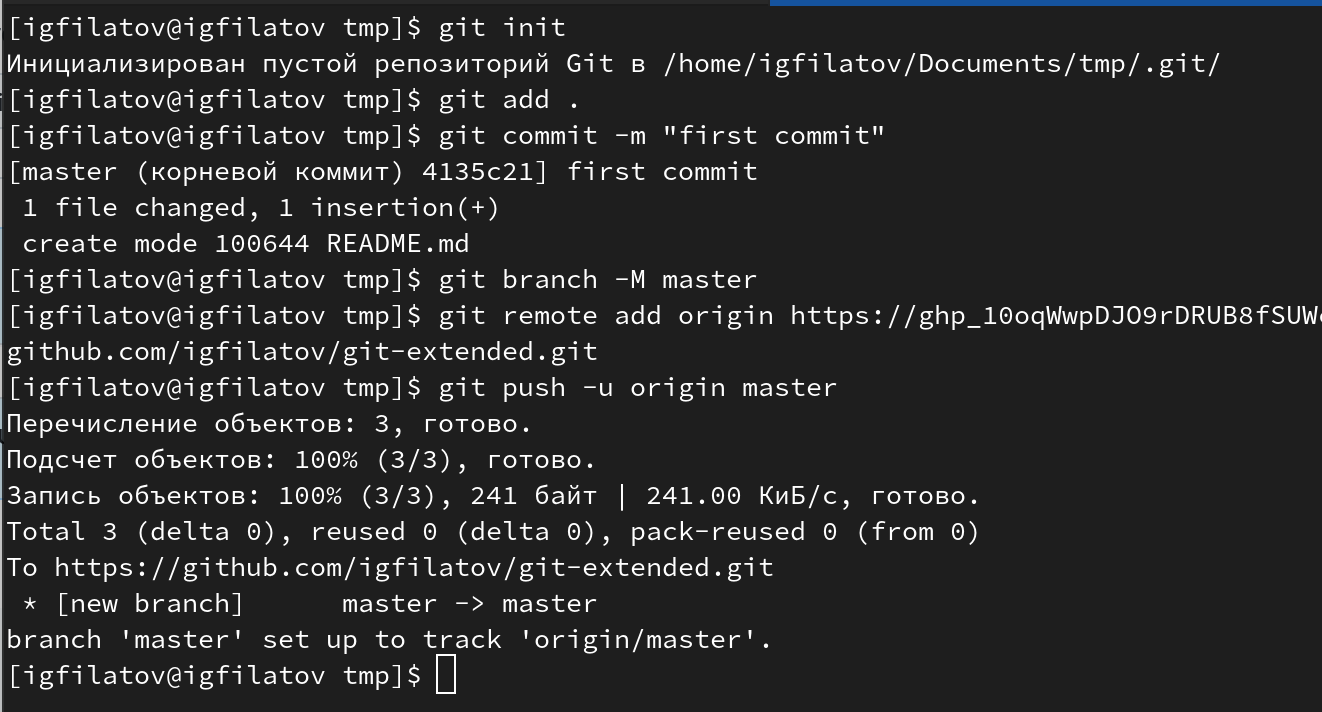


Рис. 5: Инициализация репозитория

Проведём конфигурацию общепринятых коммитов и сделаем повторный коммит (рис. 6).

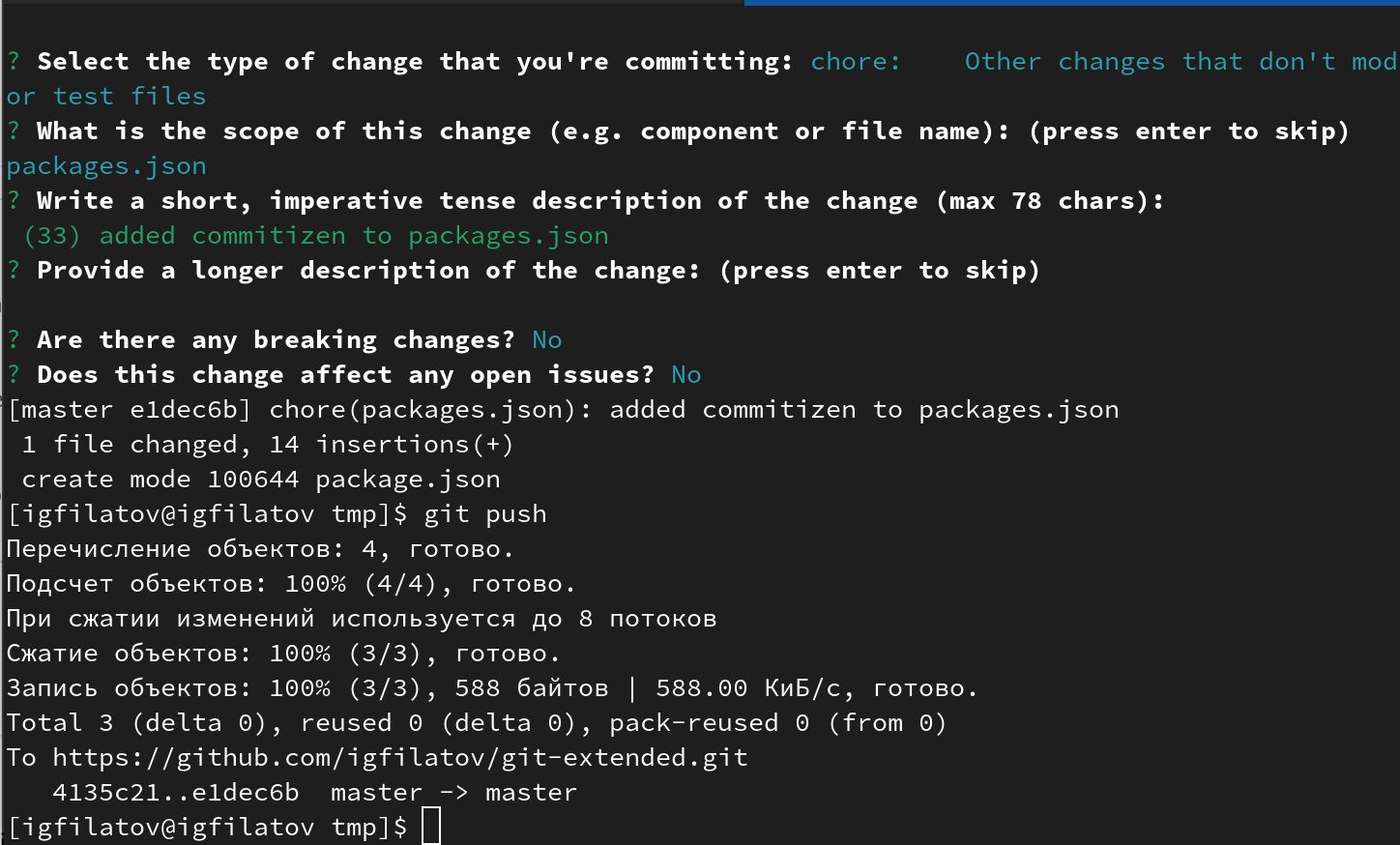


Рис. 6: Конфигурация по стандарту cz

## 3.5 Конфигурация git-flow

Инициализируем git-flow устанавливая префикс для ярлыков в v, проверим что мы на ветке develop и загрузим весь репозиторий в хранилище (рис. 7).

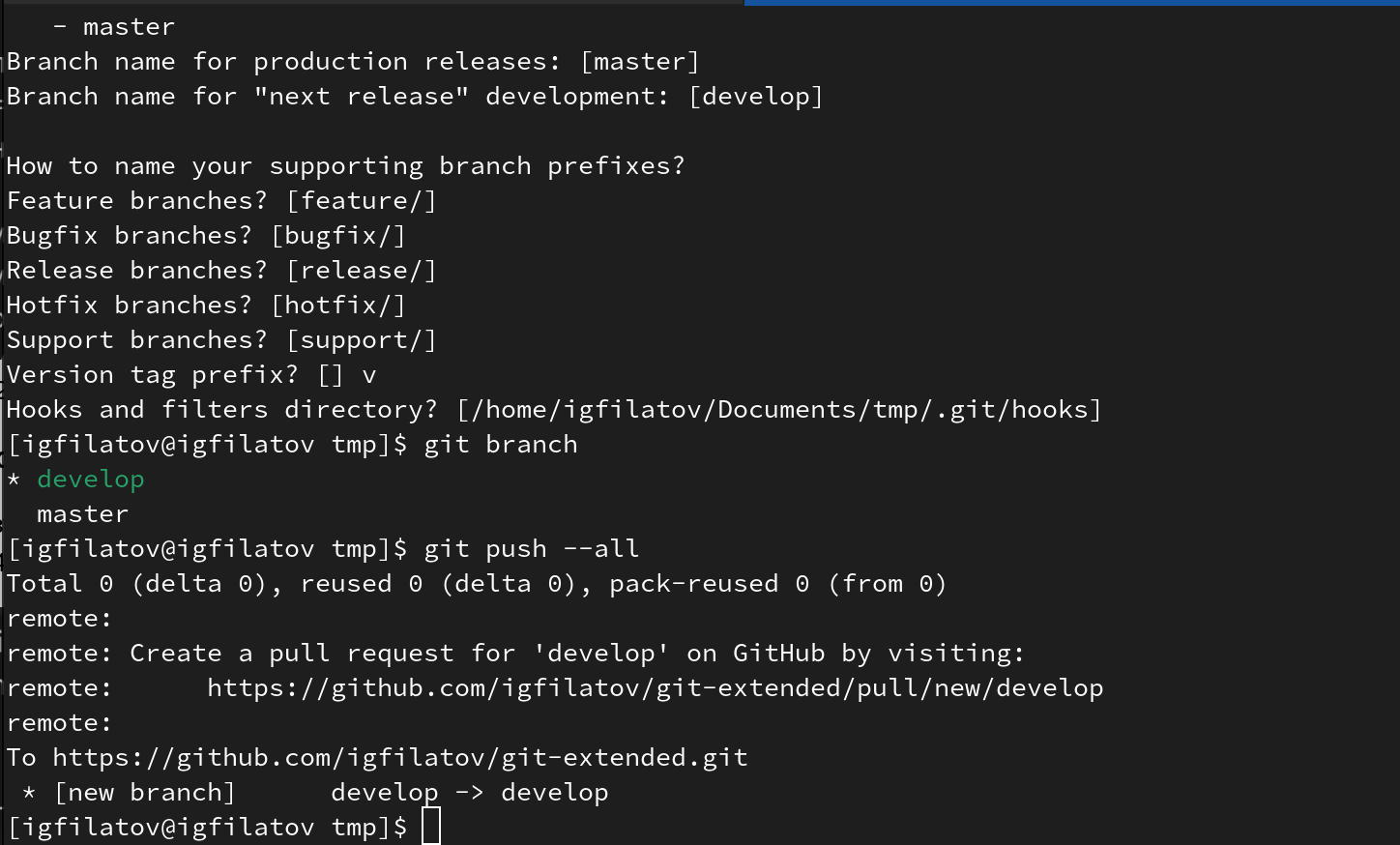


Рис. 7: Инициализация git flow

Установим внешнюю ветку как вышестоящую для этой ветки и создадим релиз с версией 1.0.0 и журнал изменений (рис. 8).

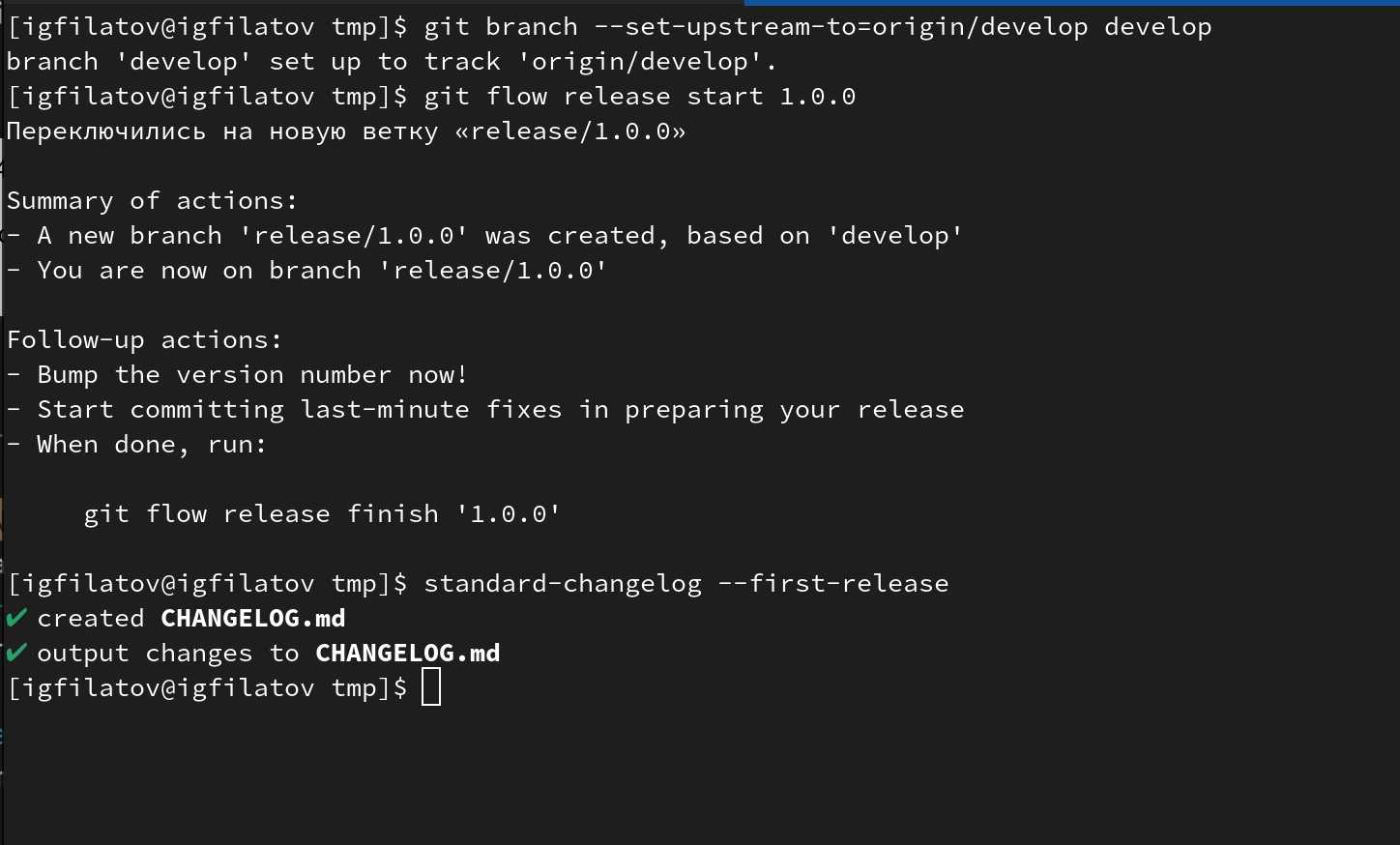


Рис. 8: Настройка ветки develop

Добавим изменения в индекс, зельём релизную ветвь в основную, отправим данные и создадим релиз на github (рис. 9).

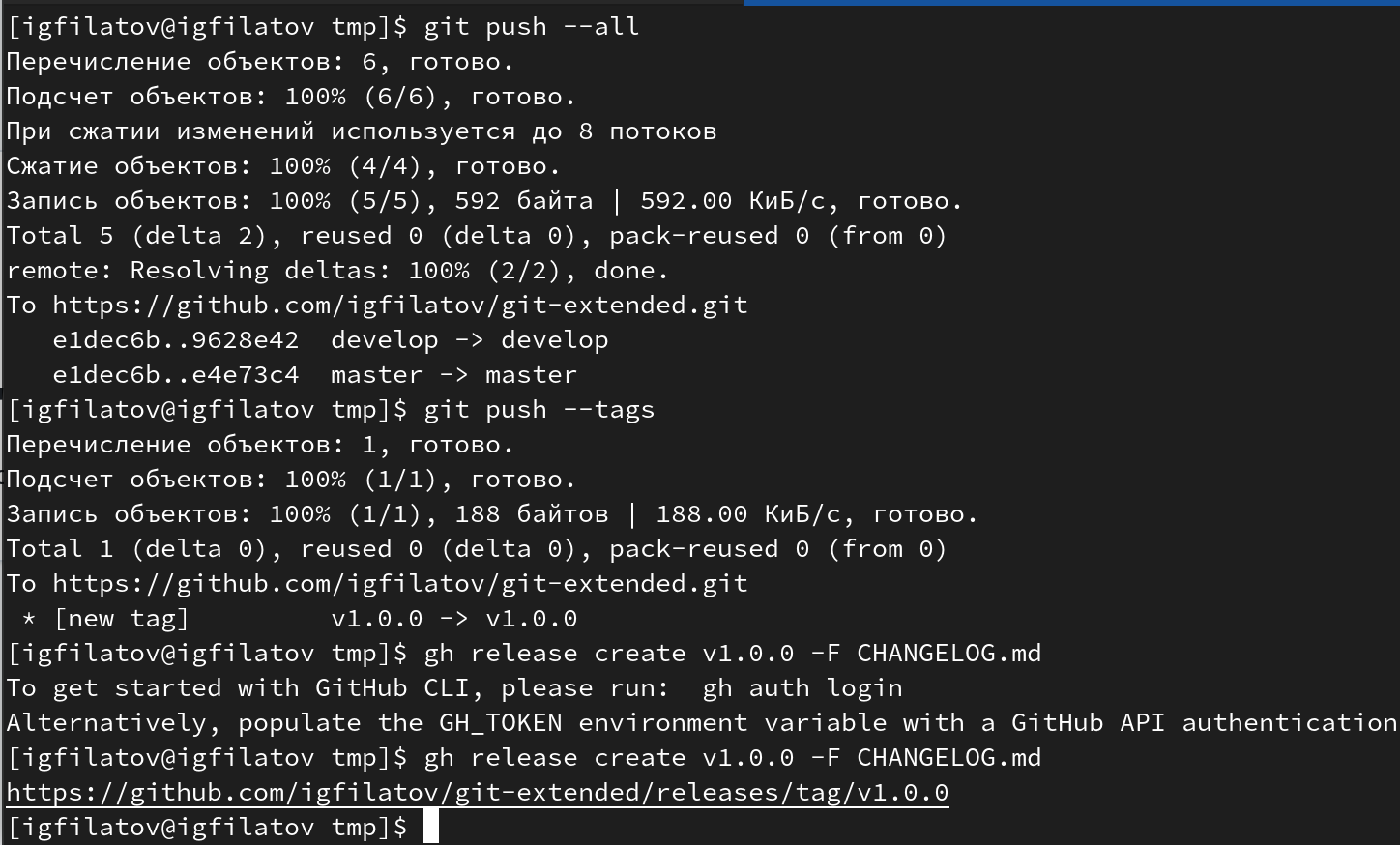


Рис. 9: Создание релиза

## 3.6 Работа с репозиторием git-flow

Для демонстрации создадим ветку новой функциональности и добавим релиз с версией 1.2.3 (рис. 10).

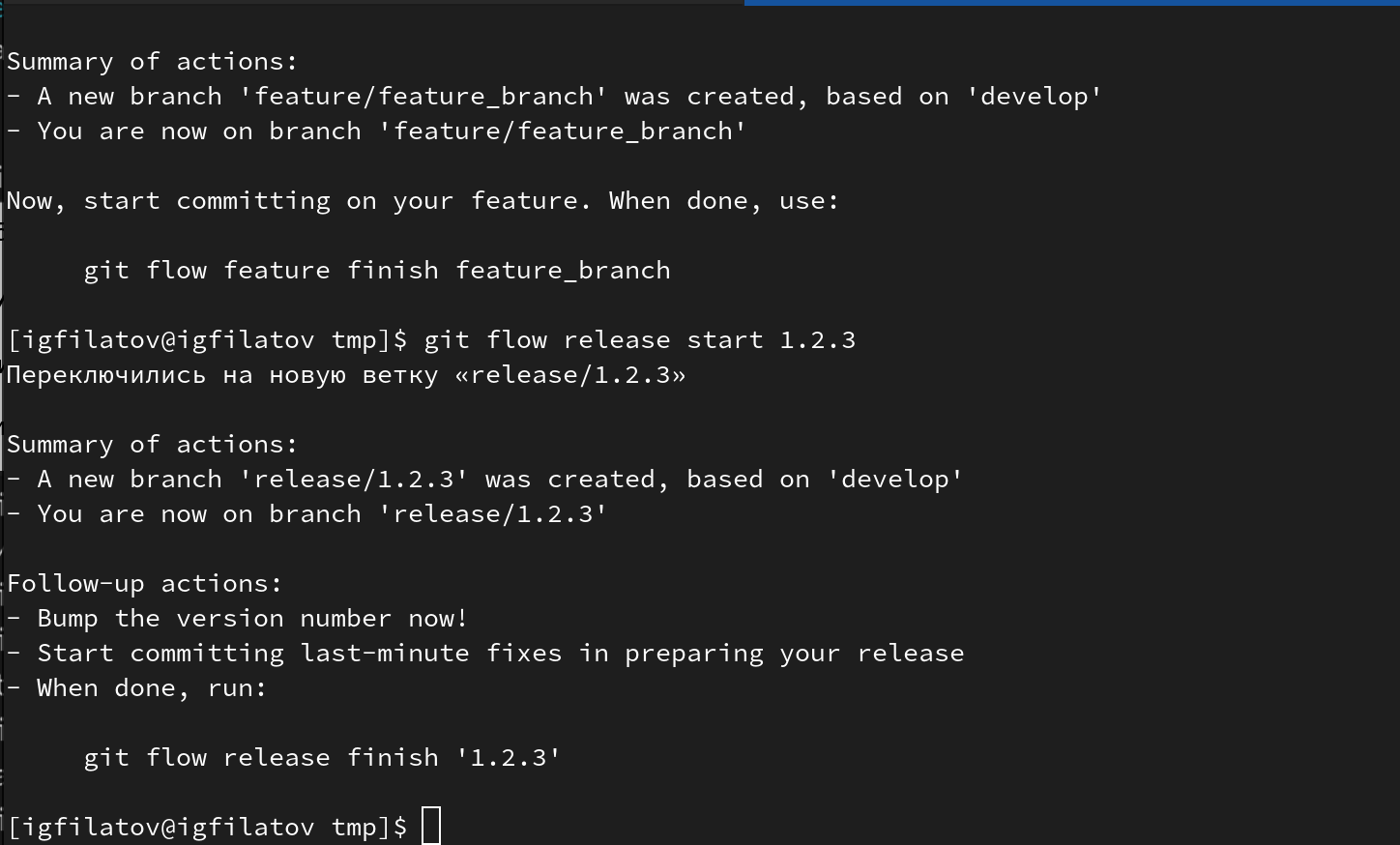


Рис. 10: Создание ветки новой функциональности

Обновим номер версии в файле package.json, создадим журнал изменений и добавим в индекс (рис. 11).



Рис. 11: Изменение версии

Сделаем коммит, снова зальём ветку в основную и после отправки данных на github создаём релиз (рис. 12).

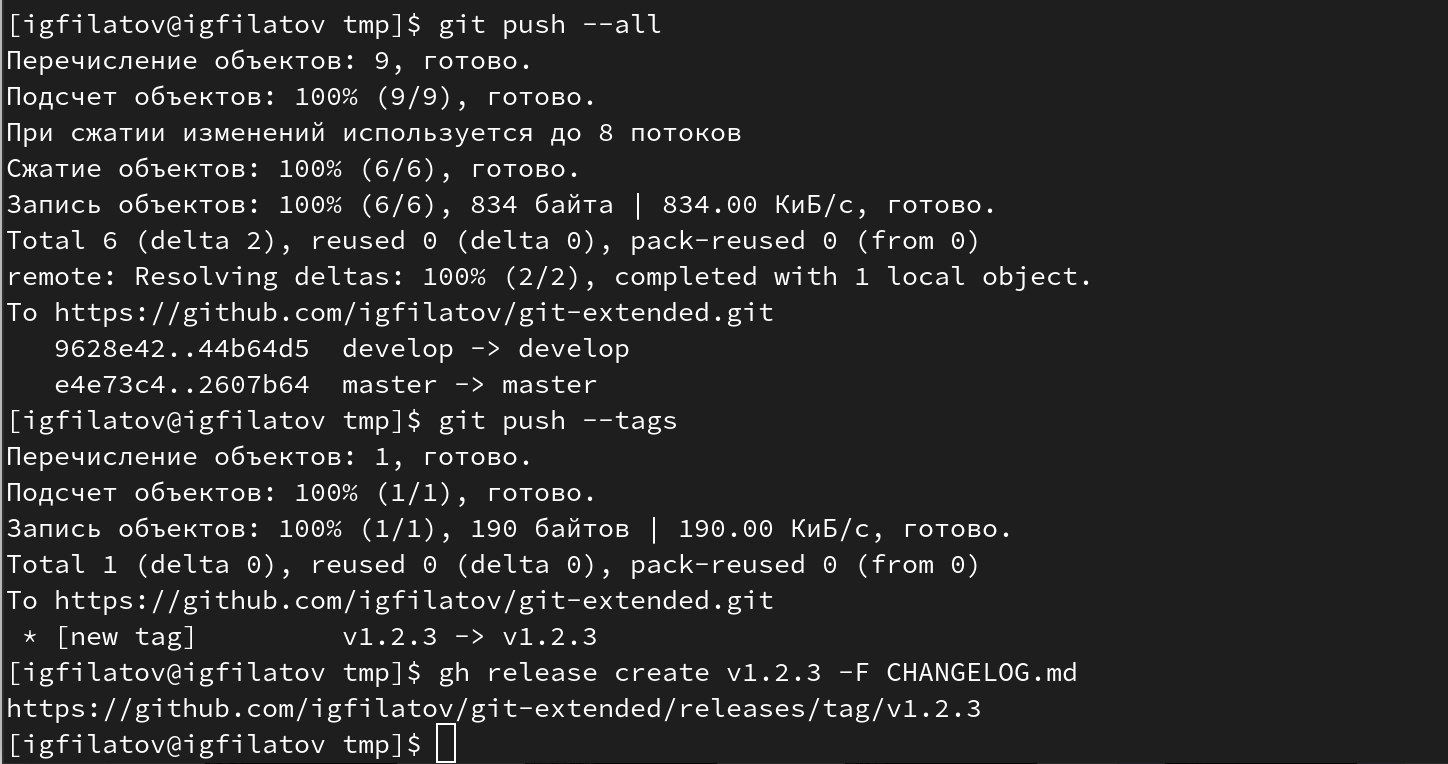


Рис. 12: Создание тестового релиза 1.2.3

В конце концов преобразуем наш рабочий репозиторий в репозиторий с git-flow и conventional commits

# 4 Выводы

Я приобрёл навыки правильной работы с репозиториями git, git-flow и conventional commits

# 5 Список литературы

1. [Архитектура ЭВМ](https://esystem.rudn.ru/pluginfile.php/2089548/mod_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%968.%20%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D1%86%D0%B8%D0%BA%D0%BB%D0%B0.%20%D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0%20%D0%B0%D1%80%D0%B3%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%B2%20%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D0%BE%D0%B9%20%D1%81%D1%82%D1%80%D0%BE%D0%BA%D0%B8..pdf)