Лабораторная работа №4

Дисциплина: Операционные системы

Бызова Мария Олеговна

Содержание

# 1 Цель работы

Целью данной лабораторной 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.

Установим git-flow из коллекции репозиториев Copr (https://copr.fedorainfracloud.org/coprs/elegos/gitflow/) (рис. 1 и 2).

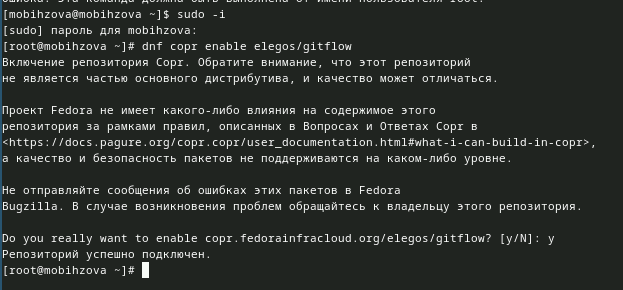


Рис. 1: Установка git-flow

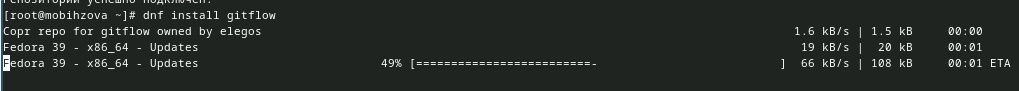


Рис. 2: Установка git-flow

## 3.2 Установка Node.js.

На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов. Установим его (рис. 3 и 4).

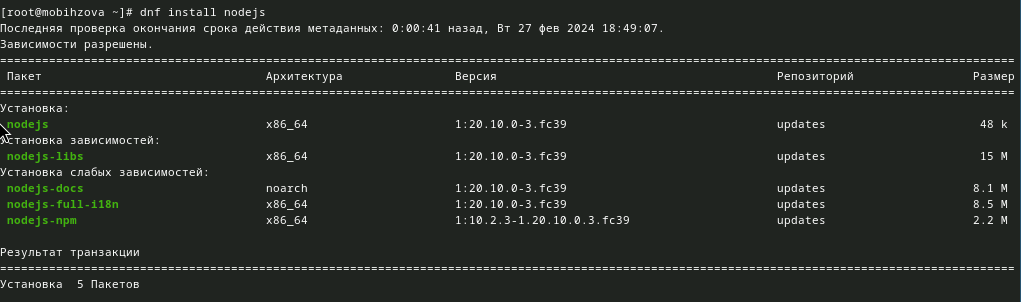


Рис. 3: Установка Node.js

Установка Node.js

Рис. 4: Установка Node.js

## 3.3 Настройка Node.js

Теперь запустим (рис. 1)

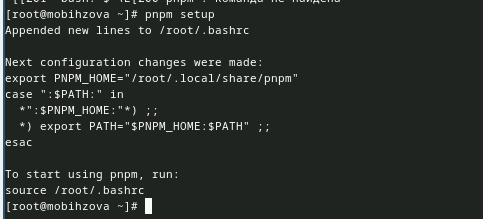


Рис. 5: Запуск

Выполним следующую команду (рис. 6)

Выполнение команды

Рис. 6: Выполнение команды

## 3.4 Общепринятые коммиты.

Введём команду pnpm add -g commitizen. Данная программа используется для помощи в форматировании коммитов. При этом устанавливается скрипт git-cz, который мы и будем использовать для коммитов (рис. 7)

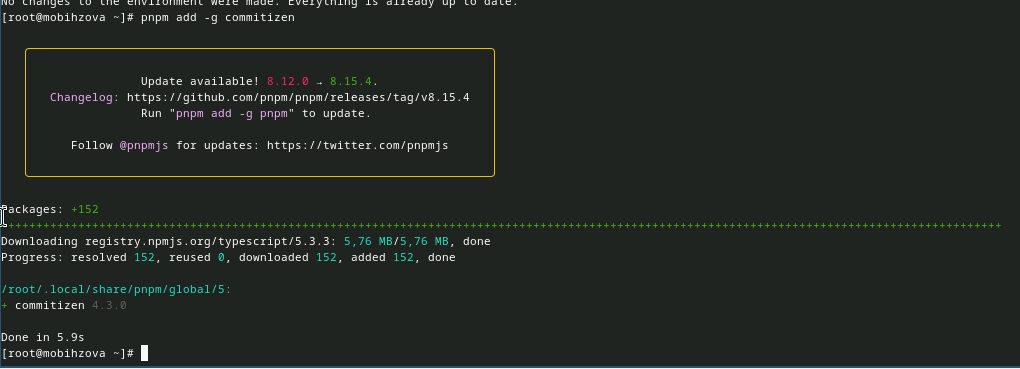


Рис. 7: Commitizen

Введём команду pnpm add -g standard-changelog. Данная программа используется для помощи в создании логов. (рис. 8)

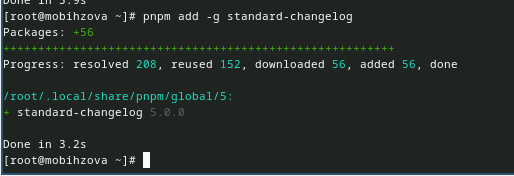


Рис. 8: Standard-changelog

Теперь создадим репозиторий git и подключим репозиторий к github. Предварительно создаём репозиторий на GitHub и клонируем его к себе (рис. 9)

Клонирование репозитория

Рис. 9: Клонирование репозитория

Сделаем первый коммит и выложим на GitHub (рис. 10)

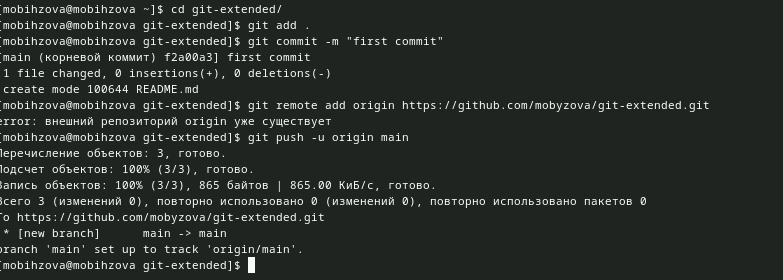


Рис. 10: Первый коммит и загрузка на GitHub

ВЫполним конфигурацию для пакетов Node.js (рис. 11)

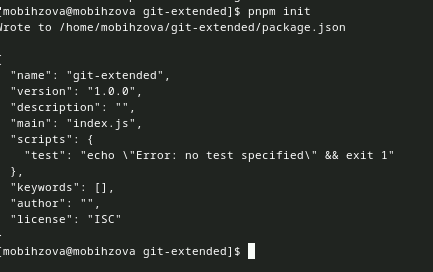


Рис. 11: Конфигурация для пакетов Node.js

Теперь необходимо заполнить несколько параметров пакета (рис. 12)

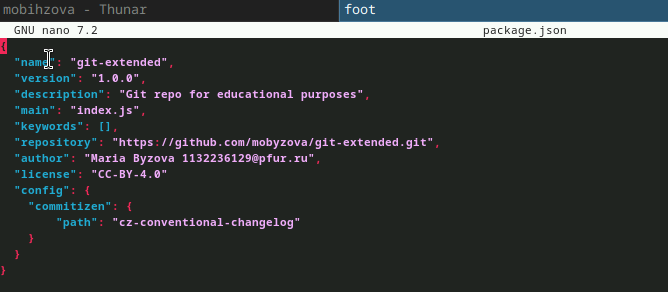


Рис. 12: Заполнение параметров пакета

Добавим новые файлы и выполним коммит (рис. 13)

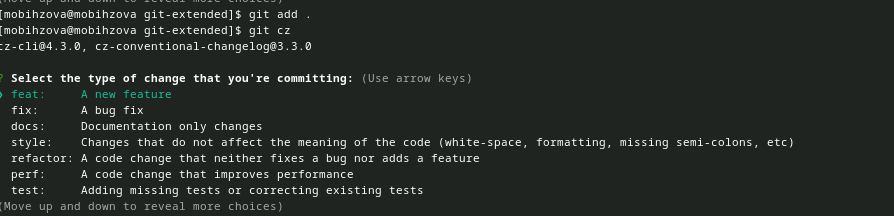


Рис. 13: Добавление новых файлов и выполнение коммита

Отправим на github (рис. 14)

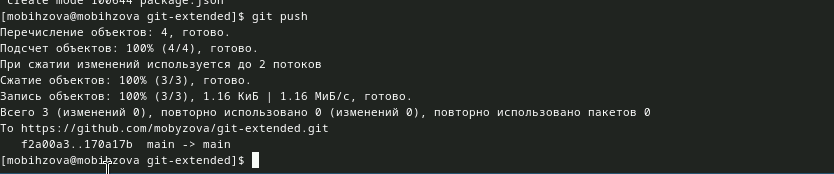


Рис. 14: ОТправка на github

Инициализируем git-flow и префикс для ярлыков установим в v (рис. 15)

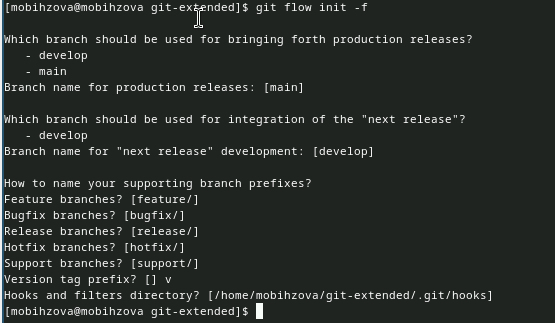


Рис. 15: Инициализация git-flow

Проверим, что мы на ветке develop (рис. 16)

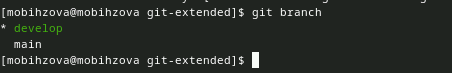


Рис. 16: Проверка ветки

Загрузим весь репозиторий в хранилище (рис. 17)

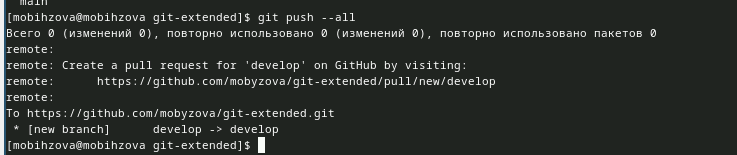


Рис. 17: Загрузка репозитория в хранилище

Установим внешнюю ветку как вышестоящую для этой ветки (рис. 18)

Установка ветки

Рис. 18: Установка ветки

Создадим релиз с версией 1.0.0 (рис. 19)

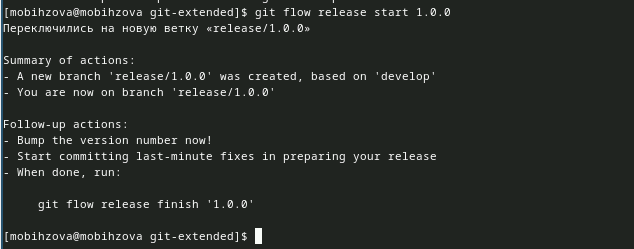


Рис. 19: Создание релиза

Создадим журнал изменений (рис. 20)

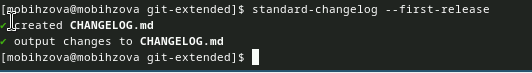


Рис. 20: Создание журнала изменений

Добавим журнал изменений в индекс (рис. 21)

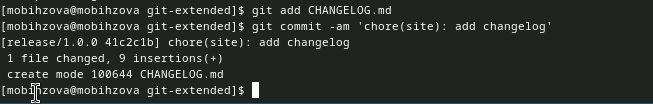


Рис. 21: Добавление журнала изменений в индекс

Зальём релизную ветку в основную ветку (рис. 22)

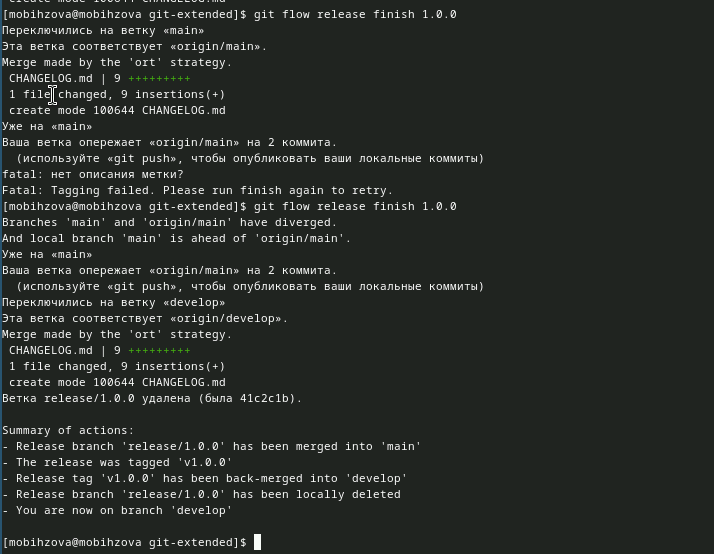


Рис. 22: Заливание релизной ветки в основную ветку

Отправим данные на github (рис. 23)

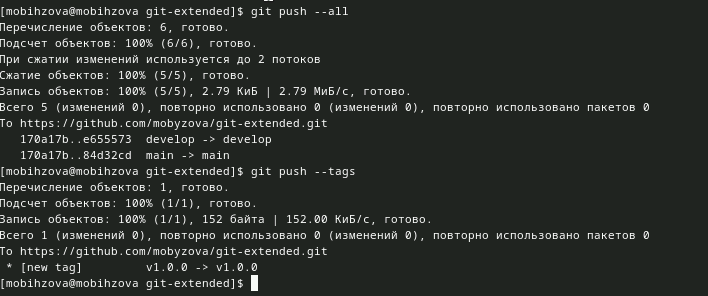


Рис. 23: Отправка данных на github

Создадим релиз на github. Для этого будем использовать утилиты работы с github (рис. 24)

Создание релиза на github

Рис. 24: Создание релиза на github

Создадим ветку для новой функциональности (рис. 25)

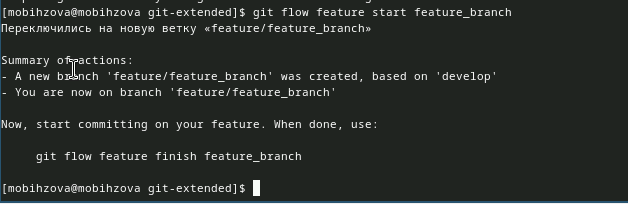


Рис. 25: Создание ветки для новой функциональности

По окончании разработки новой функциональности следующим шагом следует объединить ветку feature\_branch c develop (рис. 26)

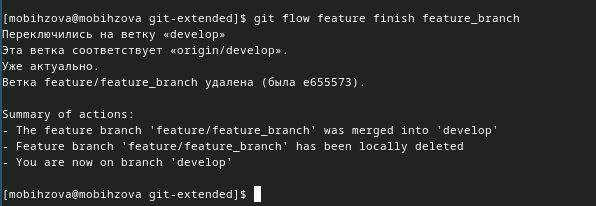


Рис. 26: Объединение веток

Создадим релиз с версией 1.2.3 (рис. 27)

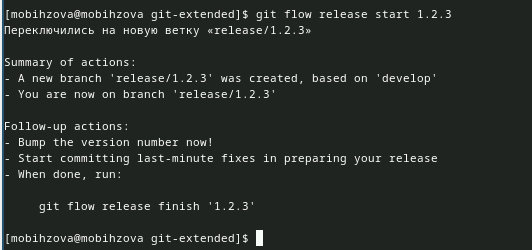


Рис. 27: Создание релиза

Обновим номер версии в файле package.json (рис. 28)

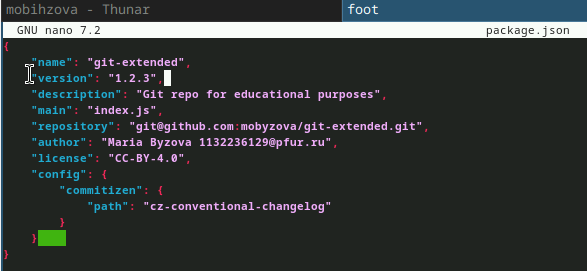


Рис. 28: Обновление версии

Создадим журнал изменений (рис. 29)

Создание журнала изменений

Рис. 29: Создание журнала изменений

Добавим журнал изменений в индекс (рис. 30)

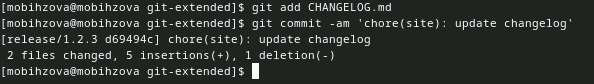


Рис. 30: Добавление журнала изменений в индекс

Зальём релизную ветку в основную ветку (рис. 31)

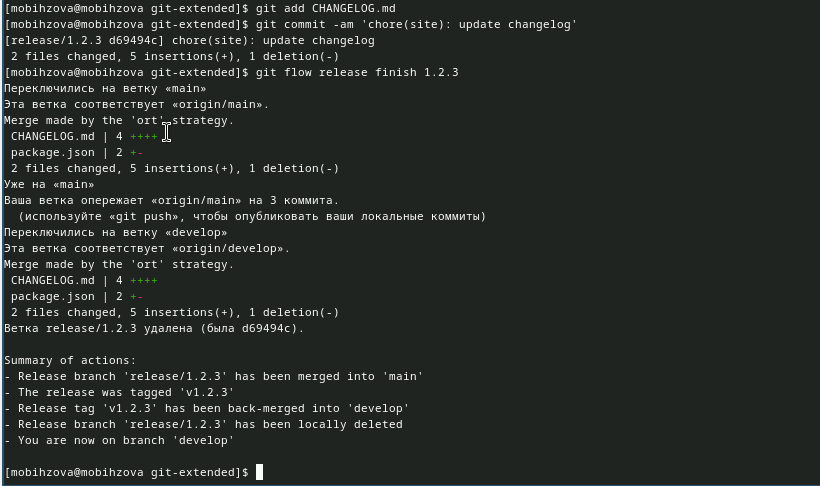


Рис. 31: Заливание релизной ветки в основную ветку

Отправим данные на github (рис. 32)

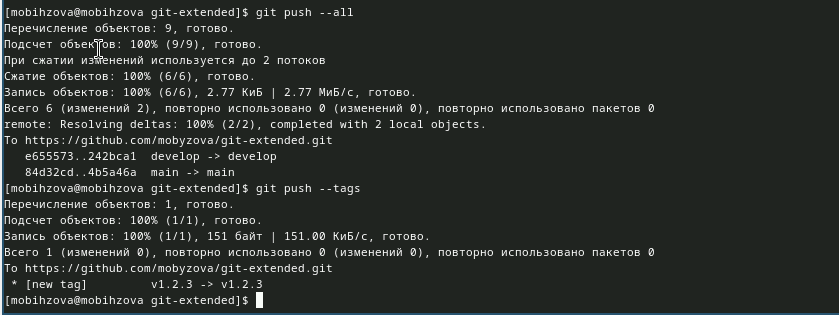


Рис. 32: Отправка данных на github

Создадим релиз на github с комментарием из журнала изменений (рис. 33)

Создание релиза на github с комментарием журнала изменений

Рис. 33: Создание релиза на github с комментарием журнала изменений

# 4 Выводы

В ходе выполнения данной лабораторной работы я получила навыки правильной работы с репозиториями git.

# Список литературы

1. Лабораторная работа №4 [Электронный ресурс] URL: https://esystem.rudn.ru/mod/page/view.php?id=1098937#org5411099