Отчёт по лабораторной работе №4

Продвинутое использование git.

Жукова Арина Александровна

Содержание

# 1 Цель работы

Получение навыков правильной работы с репозиториями git.

# 2 Задание

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

# 3 Выполнение лабораторной работы

## 3.1 Установка программного обеспечения

1. Установка git-flow.

Необходимо установить git-flow при помощи команд: dnf copr enable elegos/gitflow и dnf install gitflow (рис. 1).

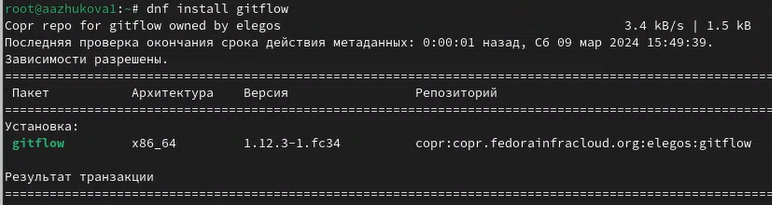


Рис. 1: Установка git-flow

1. Установка Node.js с помощью команд: dnf install nodejs и apt-get install pnpm (рис. 2-3).

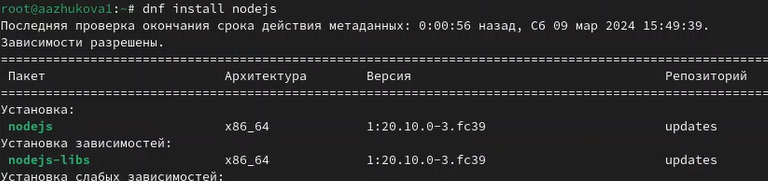


Рис. 2: Установка Node.js

Использование команды apt-get install pnpm

Рис. 3: Использование команды apt-get install pnpm

1. Настройка Node.js

Для дальнейшей работы нам необходимо запустить pnpm при помощи команды pnpm setup (рис. 4).

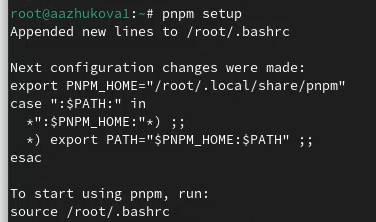


Рис. 4: Настройка Node.js

Выполняем команду: source ~/.bashrc (рис. [-fig. 5).

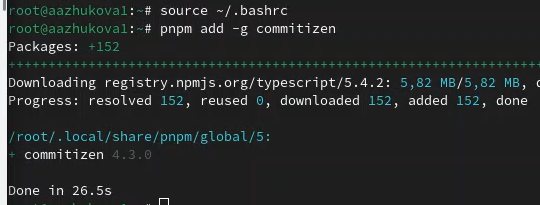


Рис. 5: Использование команды source ~/.bashrc

1. Общепринятые коммиты

* 4.1 commitizen

Для помощи в форматировании коммитов используем pnpm add -g commitizen (рис. [-fig. 6).

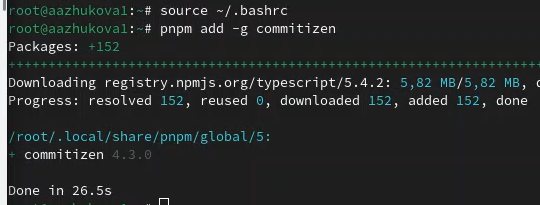


Рис. 6: Формантирование коммитов

4.2 standard-changelog

Для помощи в создании логов используем pnpm add -g standard-changelog (рис. 7).

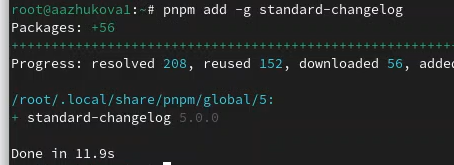


Рис. 7: Создание логов

1. Практический сценарий использования git

* 5.1 Создание репозитория git

Создаём репозиторий на GitHub, называем его git-extended. (рис. 8).

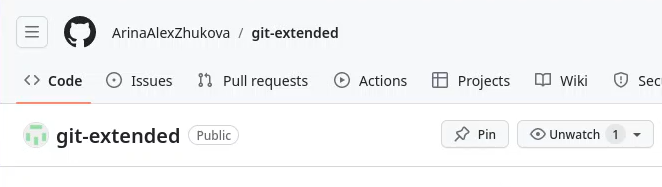


Рис. 8: Создание репозитория

5.2 Делаем первый коммит и выкладываем на github.

Используем команды: git commit -m “first commit”, git remote add origin git@github.com:ArinaAlexZhukova/git-extended.git (рис. 9).

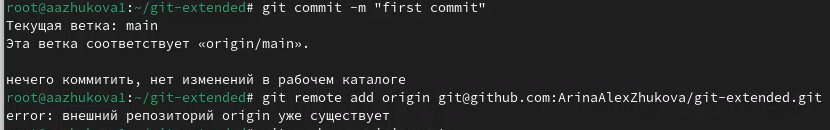


Рис. 9: Создание коммита

git push -u origin master (рис. 10).

Выкладывание коммита на github

Рис. 10: Выкладывание коммита на github

5.3 Конфигурация общепринятых коммитов

Настраиваем конфигурация для пакетов Node.js при помощи команды: pnpm init (рис. 11).

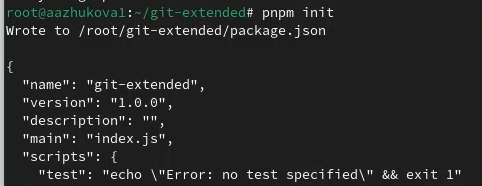


Рис. 11: Настройка конфигурации

Заполняем некоторорые параметры пакета. Сконфигурим формат коммитов, добавляя для в файл package.json команду для формирования коммитов (рис. 12).

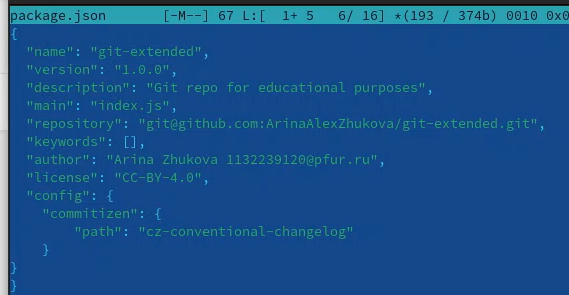


Рис. 12: Файл package.json

При помощи команды git add . добавляем новые файлы. Выполняем коммит при помощи команды git cz, а также выгружаем на github (рис. 13).

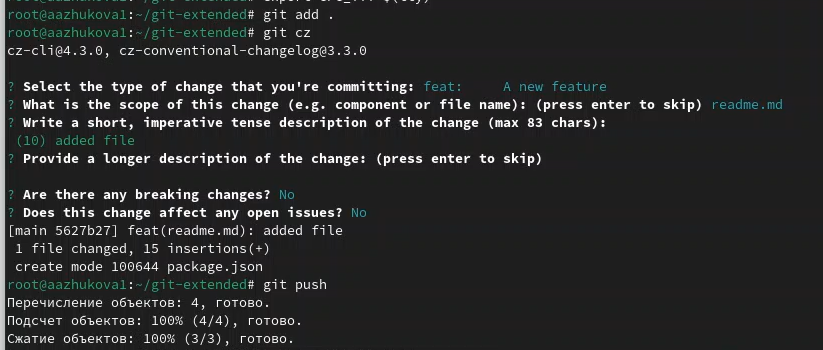


Рис. 13: Выполнение команды git add .; git cz; git push

5.4 Конфигурация git-flow

Инициализируем git-flow при помощи команды git flow init (рис. 14).

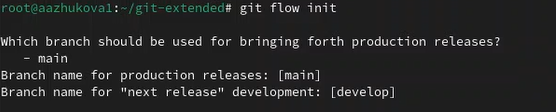


Рис. 14: Иницилизация git-flow

Устанавливаем префикс для ярлыков в v 15).

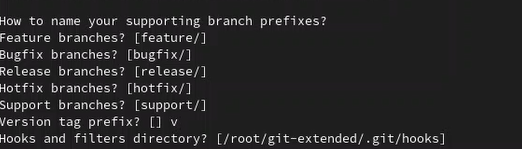


Рис. 15: Префикс для ярлыков

Проверяем, что находимя на ветке branch (рис. 16).

Проверка ветки

Рис. 16: Проверка ветки

Загружаем весь репозиторий в хранилище при помощи git push –all (рис. 17).

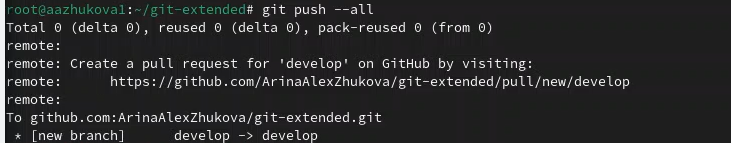


Рис. 17: Выгрузка в хранилище

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

Установка вышестоящей ветки

Рис. 18: Установка вышестоящей ветки

Создание релиза с версией 1.0.0 при помощи git flow release start 1.0.0 (рис. 19).

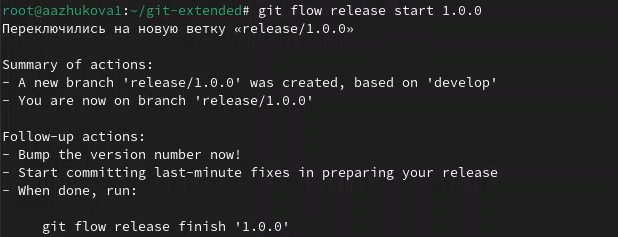


Рис. 19: Создание релиза

Создание журнала изменений с помощью команды standard-changelog –first-release (рис. 20).

Создание журнала изменений

Рис. 20: Создание журнала изменений

Добавление журнала изменений в индекс (рис. 21).

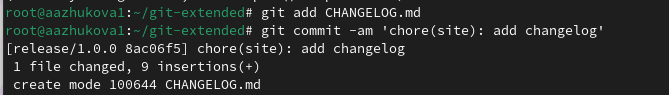


Рис. 21: Добавление журнала изменений

Зальём релизную ветку в основную ветку (рис. 22).

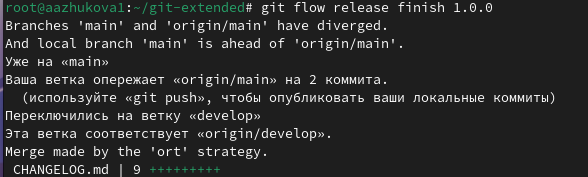


Рис. 22: Соединение релизной ветки и основной

Отправка данных на github (рис. 23-24).

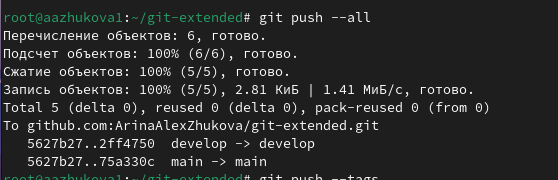


Рис. 23: Отправка данных на github

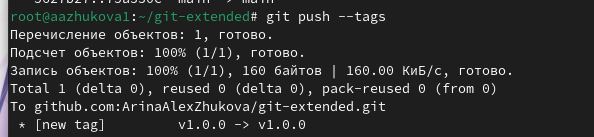


Рис. 24: Отправка данных на github

Создание релиза на github (рис. 25).

Создание релиза

Рис. 25: Создание релиза

1. Работа с репозиторием git

* 6.1 Разработка новой функциональности

Создание ветки для новой функциональности (рис. 26).

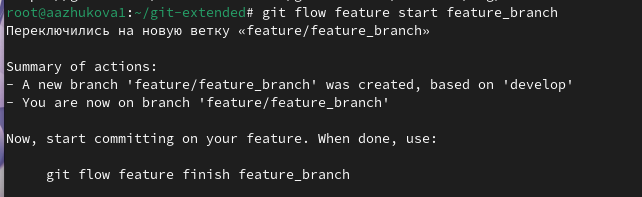


Рис. 26: Создание ветки

Объединение ветки feature\_branch c develop (рис. 27).



Рис. 27: Объединение веток

6.2 Создание релиза git-flow

Создаём релиз с версией 1.2.3 (рис. 28).

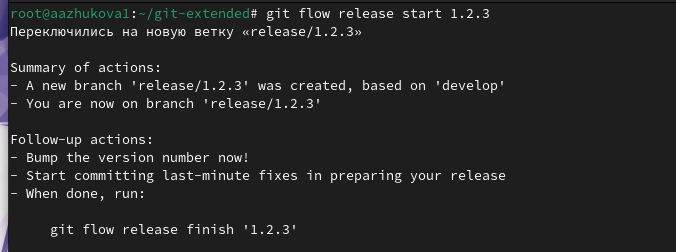


Рис. 28: Создание релиза

Обновляем номер версии в файле package.json. Создаём журнал изменений (рис. 29).

Создание журнала изменений

Рис. 29: Создание журнала изменений

Добавим журнал изменений в индекс (рис. 30).

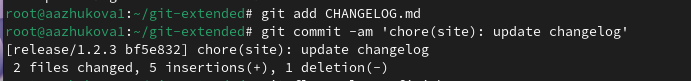


Рис. 30: Добавление журнала изменений

Зальём релизную ветку в основную ветку (рис. 31).

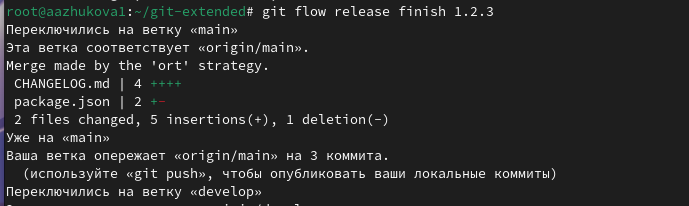


Рис. 31: Соединение релизной ветки и основной

Отправка данных на github (рис. 32-33).

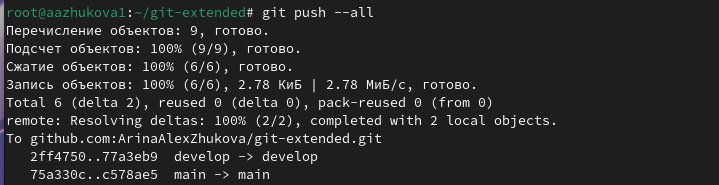


Рис. 32: Отправка данных на github

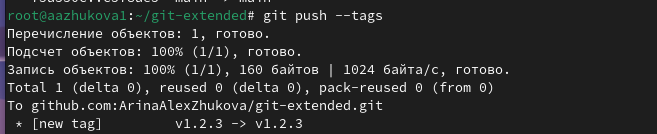


Рис. 33: Отправка данных на github

Создание релиза на github с комментарием из журнала изменений (рис. 34).

Создание релиза

Рис. 34: Создание релиза

# 4 Выводы

Нами были изучены и опробованы на практике многие возможности git-flow.

# Список литературы