Лабораторная работа №4

Отчет

Устинова Виктория Вадимовна

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.

## 2.1 Выполнение лабораторной работы

Установка git-flow(рис. 1).

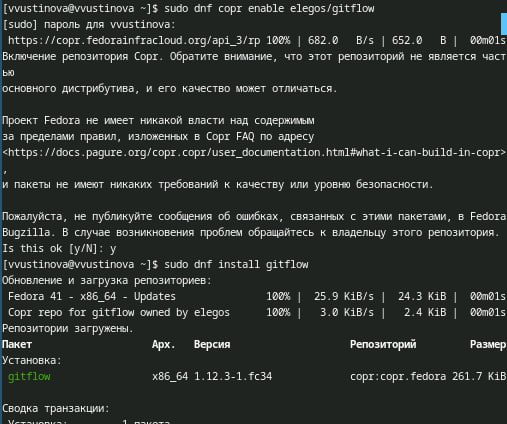


Рис. 1: Вводим команды

Установка Node.js (рис. 2).

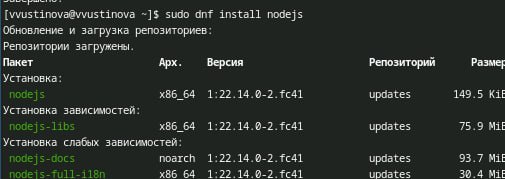


Рис. 2: Вводим команды

Установка Node.js рис. 3).

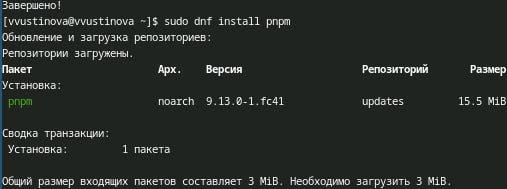


Рис. 3: Вводим команду

Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH.Запустите:pnpm setup и перелогиньтесь(рис. 4).

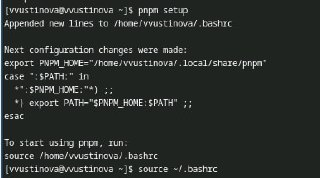


Рис. 4: Запускаем и перелогиниваемся

Данная программа используется для помощи в форматировании коммитов.(рис. 5).

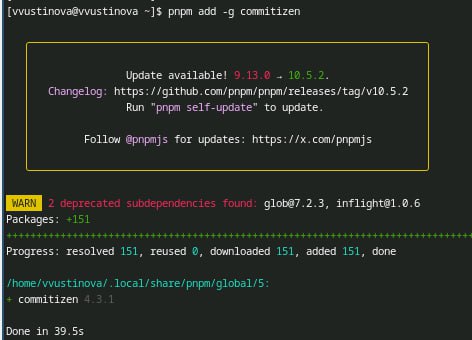


Рис. 5: Вводим команду

Данная программа используется для помощи в создании логов.(рис. 6).

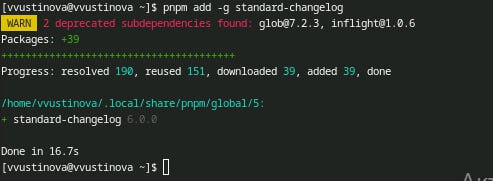


Рис. 6: Вводим команду

Создаем репозиторий в гитхабе git-extended(рис. 7).

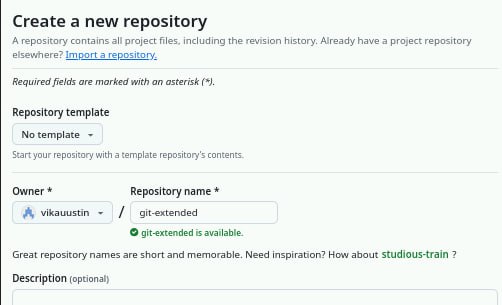


Рис. 7: Создание репозитория

Делаем первый коммит и выкладываем на github (рис. 8).

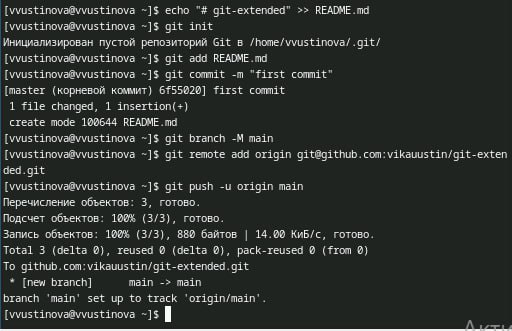


Рис. 8: Вводим комнады из гитхаба

Конфигурация для пакетов Node.js (рис. 9).

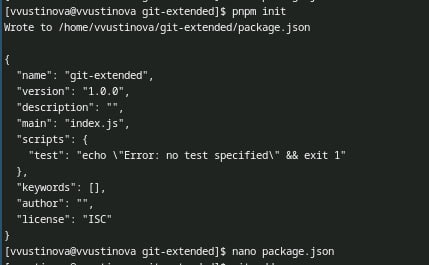


Рис. 9: Вводим команду конфигурации и переходим в редактор

Сконфигурим формат коммитов. Для этого добавим в файл package.json команду для формирования коммитов (рис. 10).

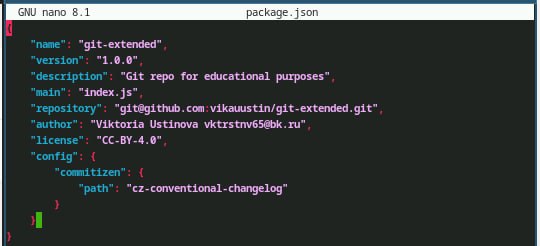


Рис. 10: Редактируем файл

Добавим новые файлы и выполним коммит(рис. 11).

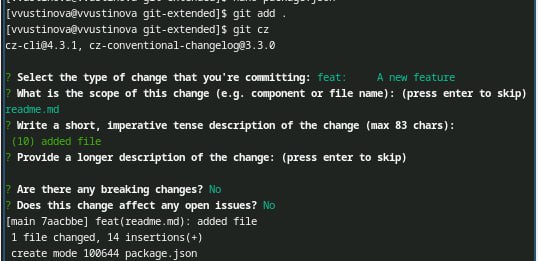


Рис. 11: Вводим команды

Отправляем файлы на гитхаб(рис. 12).

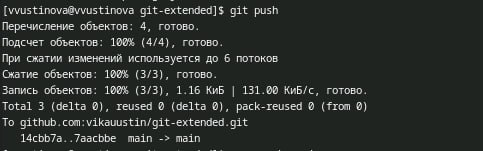


Рис. 12: Используем команду отправки

Инициализируем git-flow и префикс для ярлыков установим в v.(рис. 13).

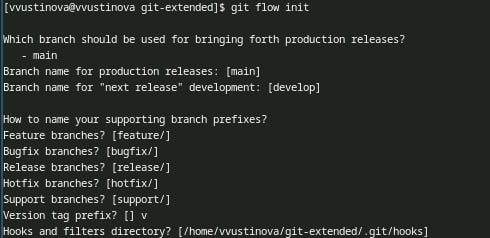


Рис. 13: Устанавливаем префикс

Проверьте, что Вы на ветке develop и загрузите весь репозиторий в хранилище (рис. 14).

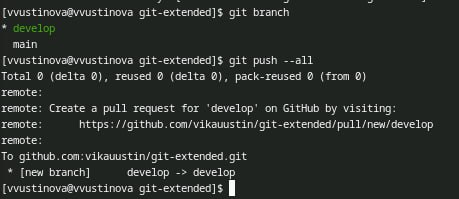


Рис. 14: Мы на той ветке, загружаем

Установите внешнюю ветку как вышестоящую для этой ветки и создадим релиз с версией 1.0.0(рис. 15).

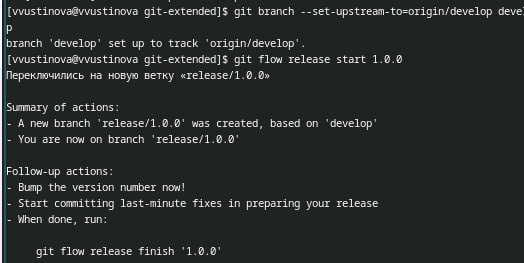


Рис. 15: Вводим команды из туиса

Создадим журнал изменений и добавим журнал изменений в индекс, а также зальём релизную ветку в основную ветку(рис. 16).

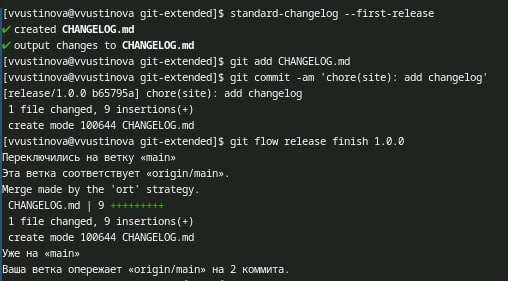


Рис. 16: делаем все вышесказанное

Отправим данные на github и создадим релиз на github. Для этого будем использовать утилиты работы с github (рис. 17).

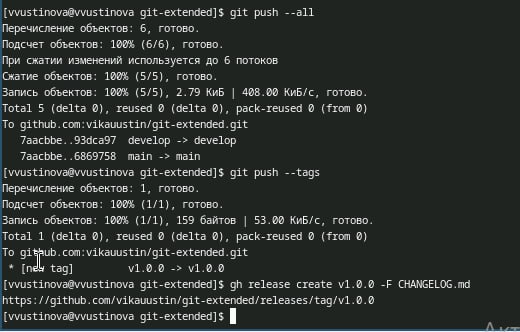


Рис. 17: Отправляем данные и создаем релиз

Создадим ветку для новой функциональности и объединяем ветку feature\_branch c develop(рис. 18).

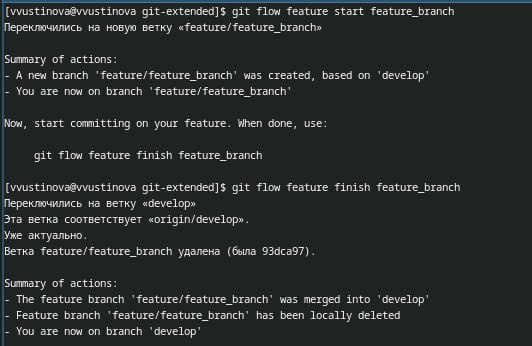


Рис. 18: Вводим команды

оздадим релиз с версией 1.2.3 и откроем редактор(рис. 19).

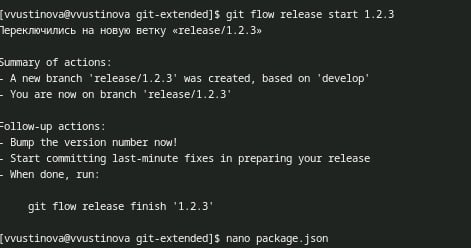


Рис. 19: Меняем версию в редакторе

Создадим журнал изменений, добавим журнал изменений в индекс и зальём релизную ветку в основную ветку(рис. 20).

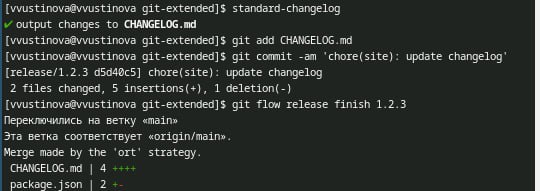


Рис. 20: Вводим команды

Отправим данные на github (рис. 21).

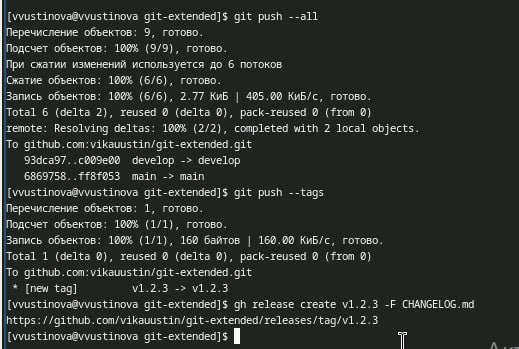


Рис. 21: Все отправилось успешно

# 3 Выводы

У нас получилось получить навыкы правильной работы с репозиториями git.