Отчёт по лабораторной работе №4

Операционные системы

Андреева Софья Владимировна

Содержание

# 1 Цель работы

Получение навыков правильной работы с репозиториями git.

# 2 Выполнение работы

Установим git-flow из коллекции репозиториев Cop (рис. fig. 1).

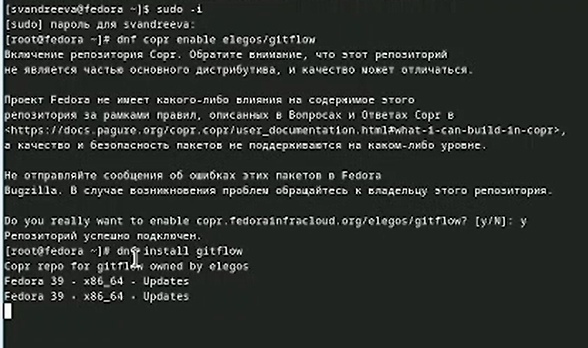


Рис. 1: Установим git-flow

Установка Node.js.На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов (рис. fig. 2).

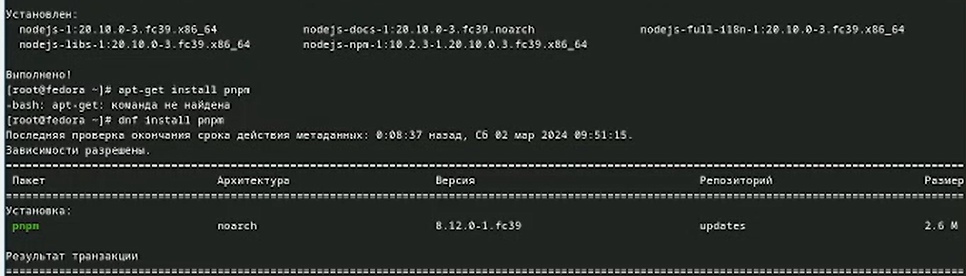


Рис. 2: Установка Node

Запустим pnpm setup, выполним source ~/.bashrc.Выполним pnpm add -g commitizen.При этом устанавливается скрипт git-cz, который мы и будем использовать для коммитов (рис. fig. 3).

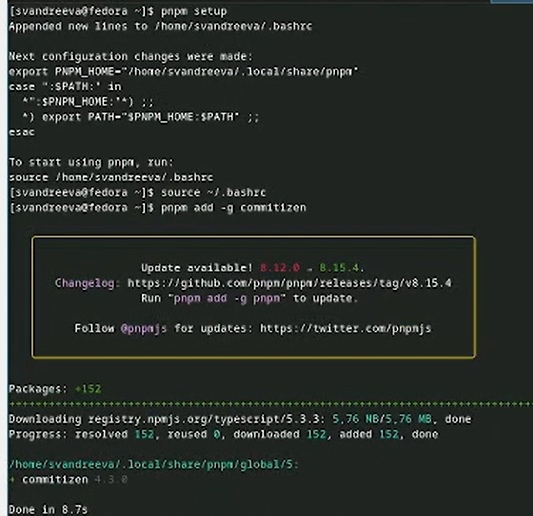


Рис. 3: Выполним pnpm add -g commitizen

Добавим standard-changelog.Данная программа используется для помощи в создании логов (рис. fig. 4).



Рис. 4: Добавим standard-changelog

Создадим репозиторий на GitHub. Назовём его git-extended.Клонируем его.(рис. fig. 5).

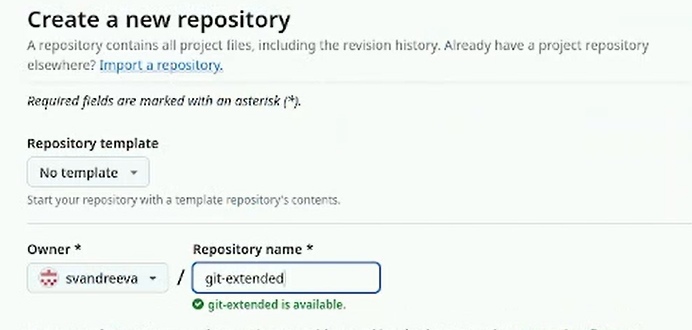


Рис. 5: Создадим репозиторий на GitHub

Делаем первый коммит и выкладываем на github (рис. fig. 6).

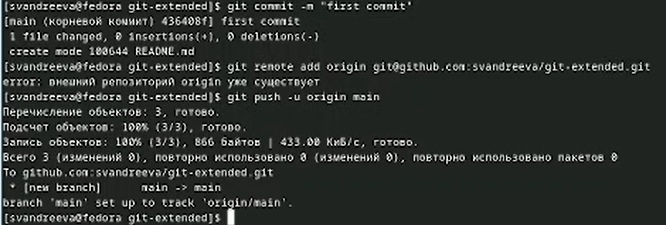


Рис. 6: Делаем первый коммит

Создаем конфигурацию для пакетов Node.js.Необходимо заполнить несколько параметров пакета. Выбираем лицензию CC-BY-4.0.Добавим в файл package.json команду для формирования коммитов (рис. fig. 7).



Рис. 7: Создаем конфигурацию для пакетов Node.js

Добавим новые файлы, выполним коммит и отправим на github (рис. fig. 8).

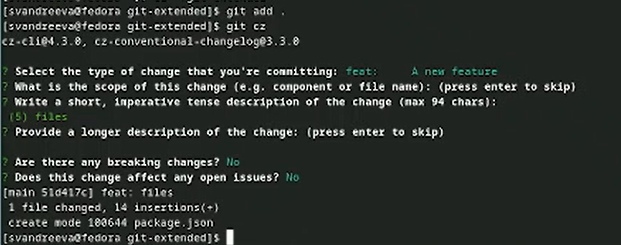


Рис. 8: Добавим новые файлы

Инициализируем git-flow.Префикс для ярлыков установим в v.Проверим, что мы на ветке develop.Загрузим весь репозиторий в хранилище (рис.@fig:009).

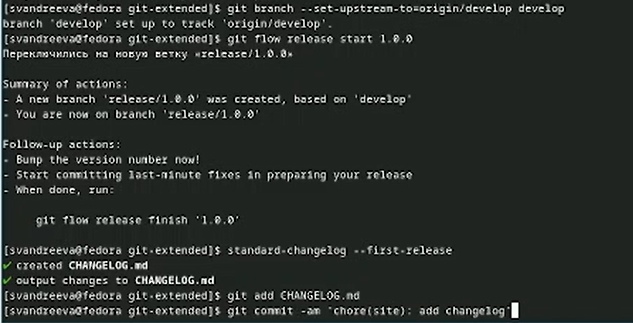


Рис. 9: Инициализируем git-flow

Установим внешнюю ветку как вышестоящую. Создадим релиз с версией 1.0.0. Создадим журнал изменений. Добавим журнал изменений в индекс. Зальём релизную ветку в основную ветку и отправим данные на github (рис. fig. 10).



Рис. 10: Добавим журнал изменений в индекс

Создадим релиз на github. Для этого будем использовать утилиты работы с github.Создадим ветку для новой функциональности: git flow feature start feature\_branch. По окончании разработки новой функциональности следующим шагом следует объединяем ветку feature\_branch c develop. Создадим релиз git-flow с версией 1.2.3. Обновим номер версии в файле package.json. (рис. fig. 11).

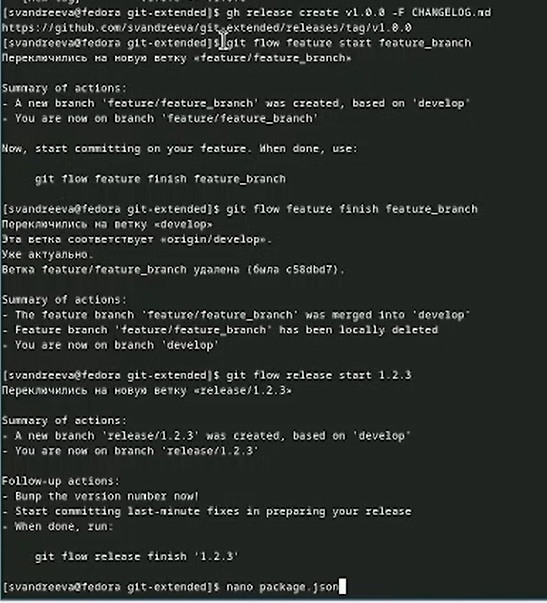


Рис. 11: Создадим релиз на github

Создадим журнал изменений. Добавим журнал изменений в индекс. Зальём релизную ветку в основную ветку (рис. fig. 12).

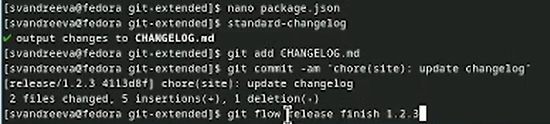


Рис. 12: Добавим журнал изменений в индекс

Отправим данные на github и создадим релиз на github с комментарием из журнала изменений (рис. fig. 13).

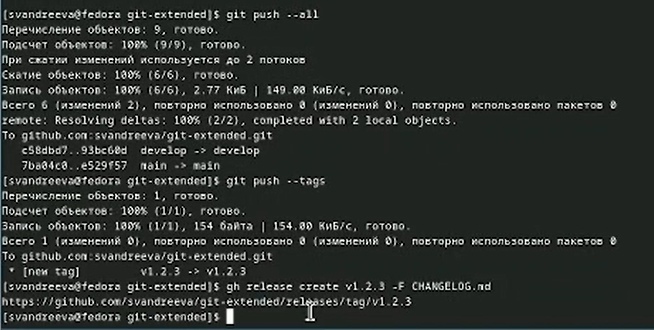


Рис. 13: Отправим данные на github и создадим релиз

# 3 Выводы

Я получила навыки правильной работы с репозиториями git.