Отчет по лабораторной работе № 4

Продвинутое использование git

Алиса Алексеевна Скобеева

Содержание

# 1 Цель работы

Получение навыков правильной работы с репозиториями git.

# 2 Задание

Создать репозиторий и выполнить с ним команды, которые указаны на платформе.

# 3 Выполнение лабораторной работы

Установка git-flow Устанавливаем git-flow из коллекции репозиториев Copr:

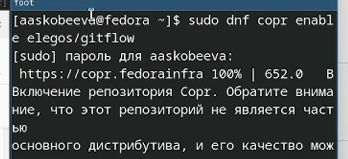


Рис. 1: Выполняем команду

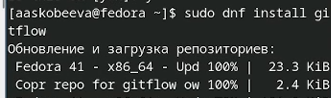


Рис. 2: Выполняем следующую команду

Установка Node.js Последовательно выполняем команды:

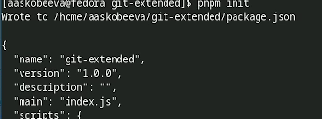


Рис. 3: Ввод команд

Запускаем:

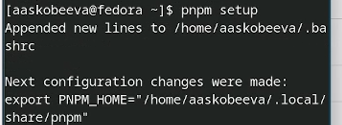


Рис. 4: Выполняем pnpm setup

Общепринятые коммиты Добавляем коммиты commitizen и standart-changelog

Установка коммита commitizen

Рис. 5: Установка коммита commitizen

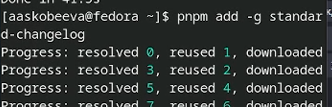


Рис. 6: Установка коммита standart-changelog

Создание репозитория git

Создаем репозиторий git. Называем его git-extended.

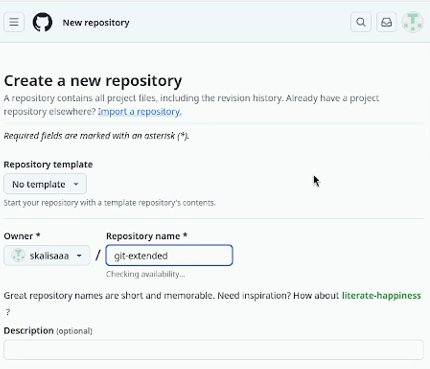


Рис. 7: Создаем репозиторий на GitHub

Добавляем какой-нибудь файл, чтобы репозиторий корректно работал. Назовем его TOUCHME. Откроем его, и напишем произвольный текст. После, можем создавать первый коммит.

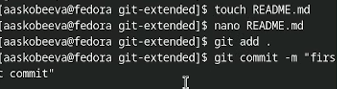


Рис. 8: Последовательно выполняем все действия

Добавляем коммит:

Добавление коммита 1

Рис. 9: Добавление коммита 1

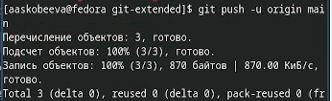


Рис. 10: Добавление коммита 2

Конфигурация общепринятых коммитов Выполняем команду pnpm init:

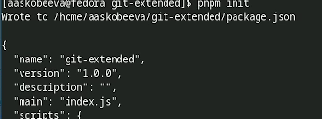


Рис. 11: Выполняем команду

Последовательно заполняем файл так, как от нас требуется:

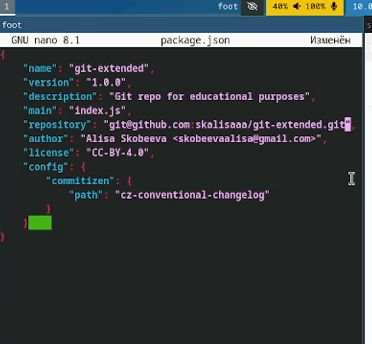


Рис. 12: Редактируем файл

Добавляем новые файлы с помощью git add . Далее выполняем коммит git cz и отправляем все на GitHub:

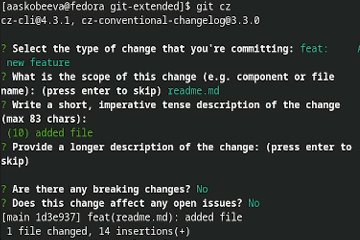


Рис. 13: Выполняем команды

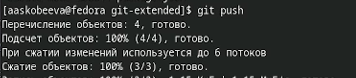


Рис. 14: Выполняем git push для отправки на GitHub

Конфигурация git-flow Инициализируем git-flow с помощью git flow init и выбираем нужные нам параметры:

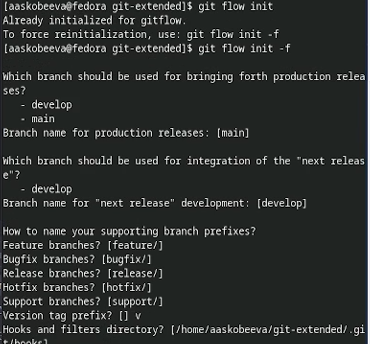


Рис. 15: Выбор нужных параметров

Проверяем, находимся ли мы на ветке develop и далее загружаем весь репозиторий в хранилище:

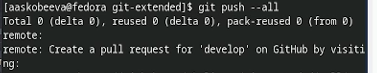


Рис. 16: Выполняем команду

Устанавливаем внешнюю ветку как вышестоящую для этой ветки:

Выполняем команду

Рис. 17: Выполняем команду

Далее, создаем релиз с версией 1.0.0, создаем журнал изменений и добавляем его в индекс:

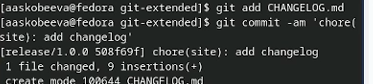


Рис. 18: Выполняем команды

Заливаем релизную ветку в основную фетку и отправляем данные на github. Далее создаем релиз на github. Для этого используем утилиты работы с github:

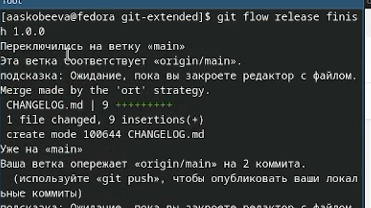


Рис. 19: Выполняем команду

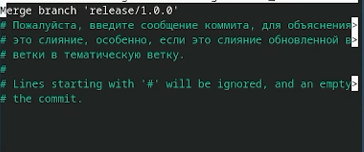


Рис. 20: Выполняем команду

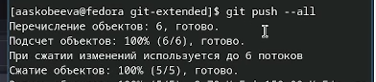


Рис. 21: Выполняем команду

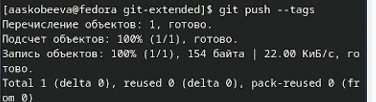


Рис. 22: Выполняем команду

Выполняем команду

Рис. 23: Выполняем команду

Работа с репозиторием git. Разработка новой функциональности Создаем ветку для новой функциональности, далее, продолжаем работу с git как обычно, и по окончании объединяем ветку feature\_branch с develop:

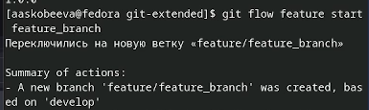


Рис. 24: Выполняем команду

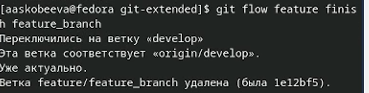


Рис. 25: Выполняем команду

Создание релиза git-flow Создаем релиз с версией 1.2.3. Обновляем номер версии в файле package.json. Далее, создаем журнал изменений и добавляем его в индекс. Далее, заливаем релизную ветку в основную ветку и отправляем данные на GitHub. Создаем релиз на github c комментарием из журнала изменений:

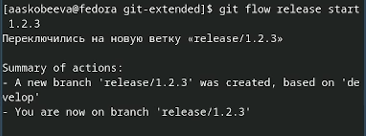


Рис. 26: Последовательно выполняем все команды 1

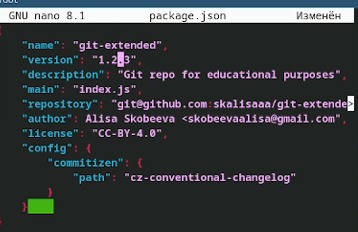


Рис. 27: Последовательно выполняем все команды 2

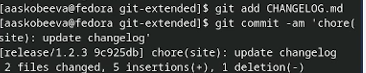


Рис. 28: Последовательно выполняем все команды 3

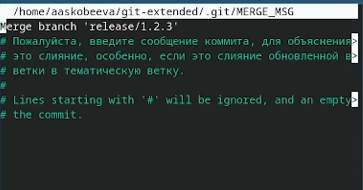


Рис. 29: Последовательно выполняем все команды 4

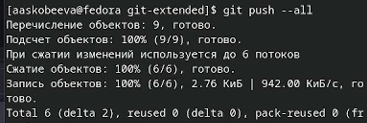


Рис. 30: Последовательно выполняем все команды 5

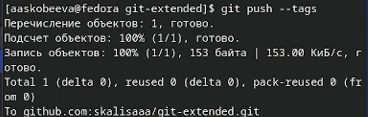


Рис. 31: Последовательно выполняем все команды 6

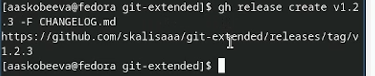


Рис. 32: Последовательно выполняем все команды 7

# 4 Выводы

В ходе выполнения лабораторной работы мы приобрели навыки продвинутого использования git.