Отчёт по лабораторной работе № 4

Продвинутое использование git.

Саенко Ангелина Андреевна

Содержание

Список иллюстраций

Список таблиц

# Цель работы

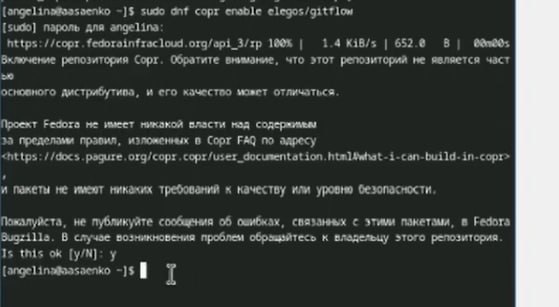
Получение навыков правильной работы с репозиториями git.

# Задание

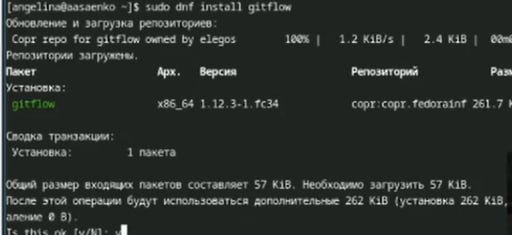
Выполнить работу для тестового репозитория. Преобразовать рабочий репозиторий в репозиторий с git-flow и conventional commits.

# Выполнение лабораторной работы

Для начала установим git-flow из коллекции репозиториев (рис. [-@fig:001]).

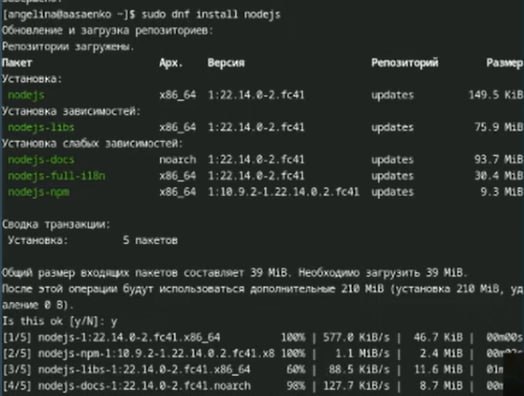


Установка

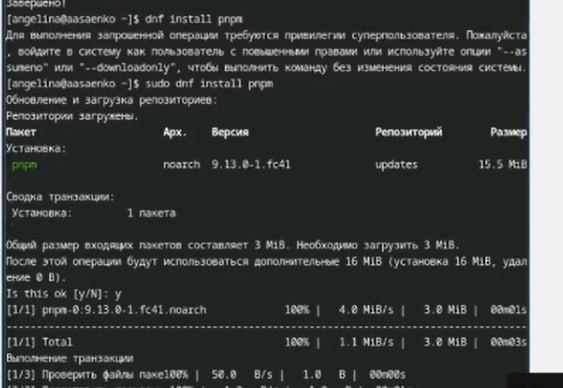


Установка

На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов. Установим ПО

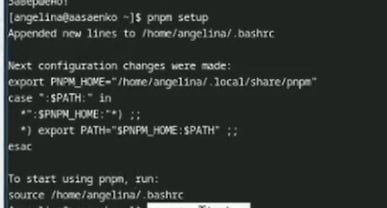


Установка nodejs



Установка pnpm

Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH

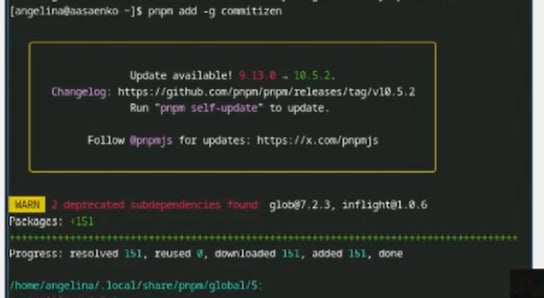


Запуск

Перелогинимся или выполним

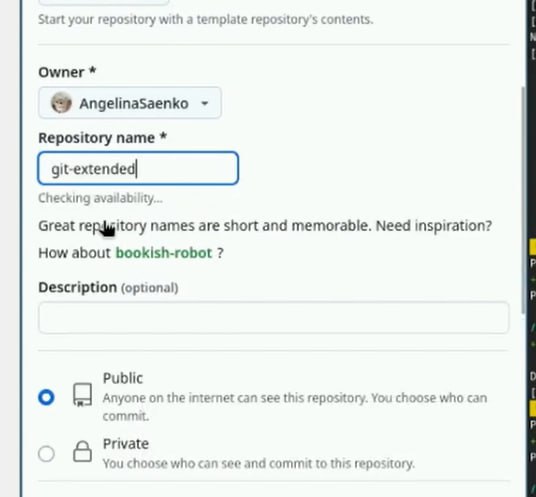
Перелогинимся или выполним

Данная программа commitizen используется для помощи в форматировании коммитов. При этом устанавливается скрипт git-cz, который мы и будем использовать для коммитов.



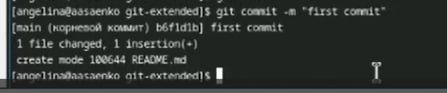
Добавление

Создаём репозиторий на GitHub. Для примера назовём его git-extended

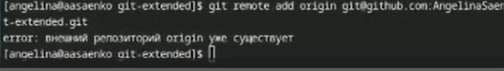


Создание репозитория

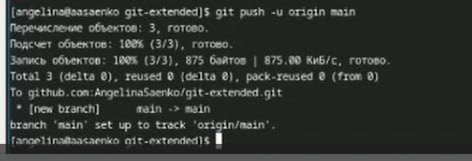
Делаем первый коммит и выкладываем на github



Первый коммит

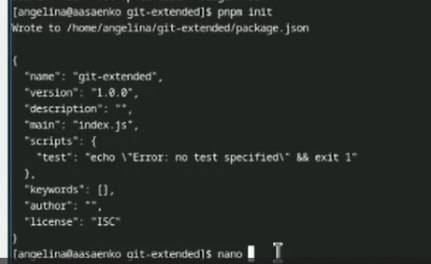


Добавление удалённого репозитория в мой локальный Git-проект



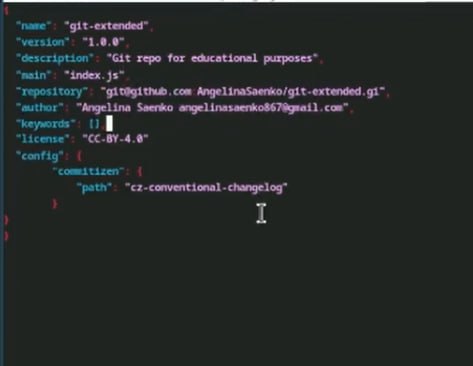
Отправка изменений

Далее сделаем конфигурацию для пакетов Node.js



Конфигурация для пакетов Node.js

Необходимо заполнить несколько параметров пакета. Сконфигурим формат коммитов. Для этого добавим в файл package.json команду для формирования коммитов



Изменяем файл

Добавим новые файлы

Добавление новых файлов

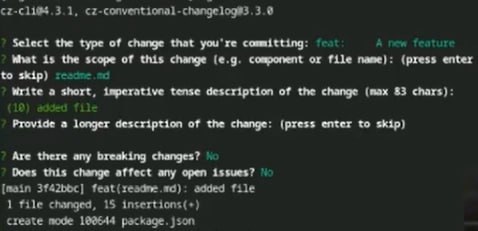
Добавление новых файлов

Выполним коммит

Коммит

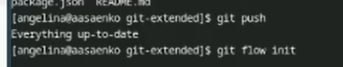
Коммит

Выбираем нужное



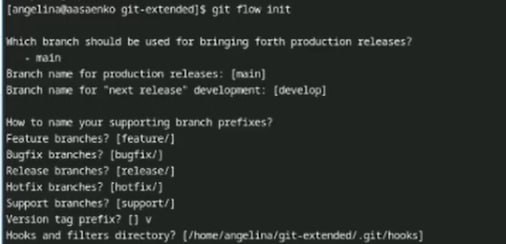
Выбор

Отправим на github



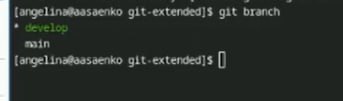
Отправка

Инициализируем git-flow



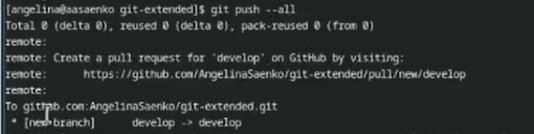
Инициализация git-flow

Проверим , что мы на ветке develop



Проверка

Загрузим весь репозиторий в хранилище



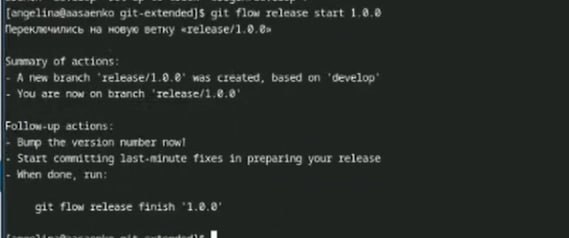
Загрузка репозитория

Установим внешнюю ветку как вышестоящую для этой ветки

Установка внешней ветки

Установка внешней ветки

Создадим релиз с версией 1.0.0



Создание релиза

Создадим журнал изменений

Создание журнала изменений

Создание журнала изменений

Добавим журнал изменений в индекс

Добавление журнала

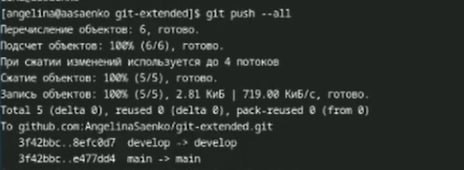
Добавление журнала

Зальём релизную ветку в основную ветку

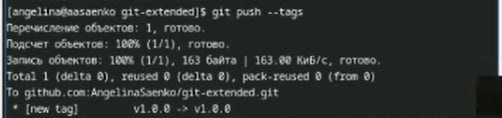
Заливание релизной ветки

Заливание релизной ветки

Отправим данные на github



Отправка данных



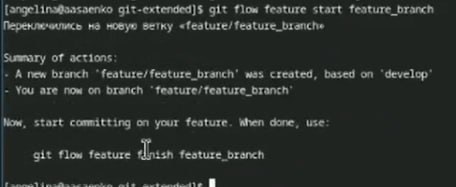
Отправка данных

Создадим релиз на github. Для этого будем использовать утилиты работы с github

Создание релиза

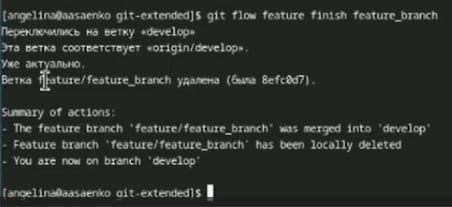
Создание релиза

Создадим ветку для новой функциональности



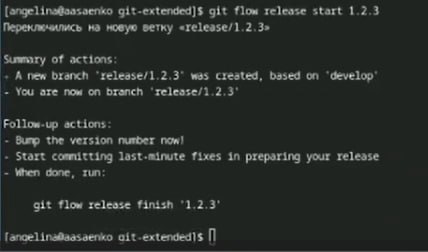
Создание ветки

Далее, продолжаем работу c git как обычно. По окончании разработки новой функциональности следующим шагом следует объединить ветку feature\_branch c develop



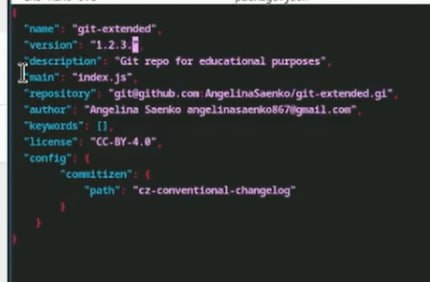
Объединяем ветки

Создадим релиз с версией 1.2.3



Создание релиза

Обновим номер версии в файле package.json. Установим её в 1.2.3



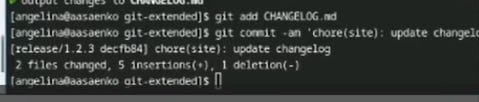
Обновим номер версии в файле package.json. Установим её в 1.2.3

Создадим журнал изменений

Создание журнала изменений

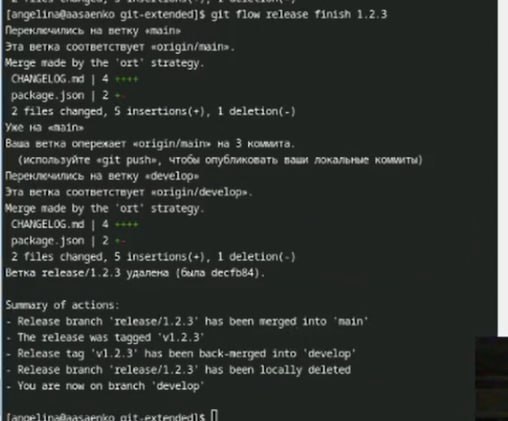
Создание журнала изменений

Добавим журнал изменений в индекс



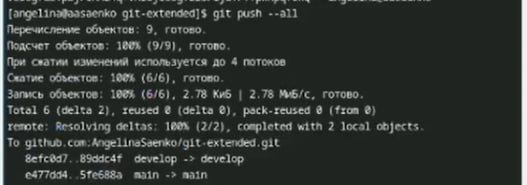
Добавление журнала изменений в индекс

Зальём релизную ветку в основную ветку

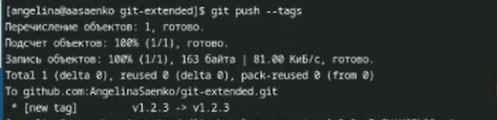


Зальём ветку

Отправим данные на github



Отправка данных



Отправка данных

Создадим релиз на github с комментарием из журнала изменений

Создание релиза

Создание релиза

# Выводы

Освоены инструменты git-flow, commitizen и standard-changelog для управления ветками, семантического версионирования и оформления коммитов. Настроен репозиторий с поддержкой общепринятых коммитов, созданы и завершены релизы с автоматическим обновлением версий и журналов изменений. Получены навыки эффективного управления разработкой и выпуском проектов.

# Список литературы