Отчёт по лабораторной работе №4

Операционные системы

Козина Дарья Александровна

Содержание

# 1 Цель работы

Получить навыки правильной работы с репозиторием git.

# 2 Задание

1. Выполнить работу для тестового репозитория;
2. Преоюразовать рабочий репозиторий с git-flow и conventional commits.

# 3 Выполнение лабораторной работы

## 3.1 Установка программного обеспечения

Перейдем в режим супер-пользователя (рис. 1).

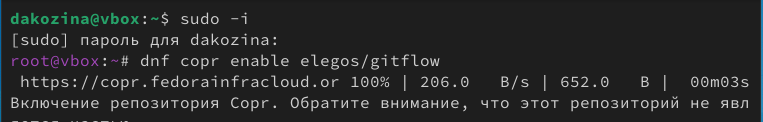


Рис. 1: Режим супер-пользователя

Установим git-flow (рис. 2).

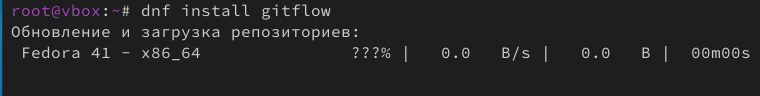


Рис. 2: Установка git-flow

Установим Node.js (рис. 3, рис. 4).

Рис. 3: Установка Node.js

Рис. 3: Установка Node.js

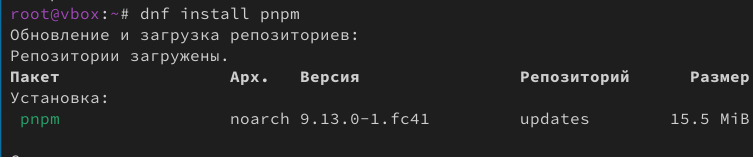


Рис. 4: Установка pnpm

Настроим Node.js. Запустим (рис. 5).

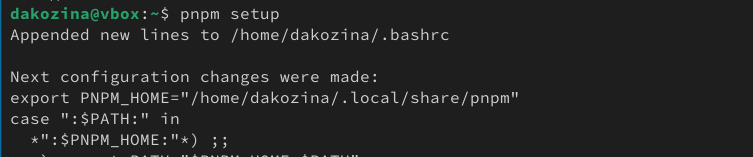


Рис. 5: Запуск

Выполним команду source (рис. 6).

Рис. 6: Команда source

Рис. 6: Команда source

### 3.1.1 Общепринятые коммиты

Добавим commitizen и git-cz (рис. 7).

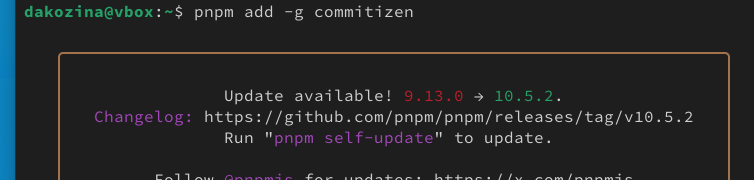


Рис. 7: Comitizen и git-cz

Добавим standard-changelog (рис. 8).

Рис. 8: Standard-changelog

Рис. 8: Standard-changelog

## 3.2 Практический сценарий использования git

Создадим рипозиторий git-extended (рис. 9).

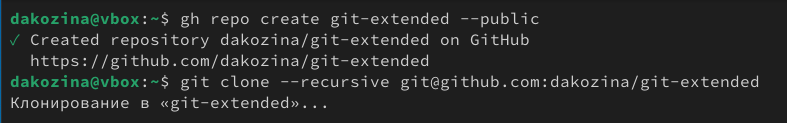


Рис. 9: Создание репозитория

Делаем первый коммит и выложим на GitHub (рис. 10).

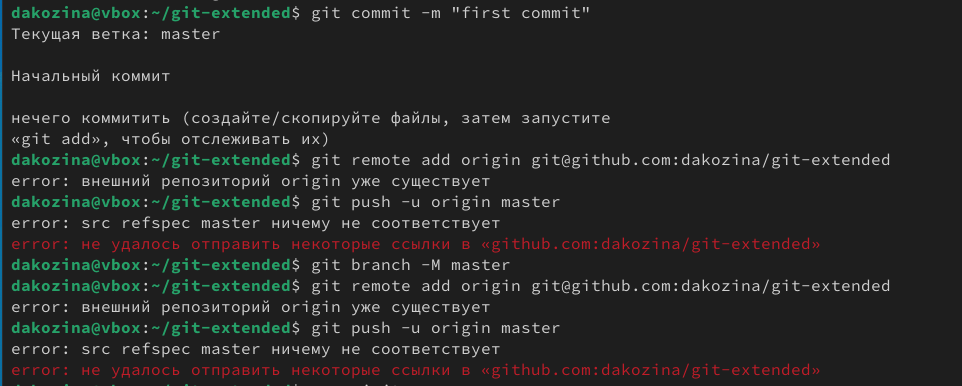


Рис. 10: Первый коммит

Конфигурация для пакетов Node.js (рис. 11).

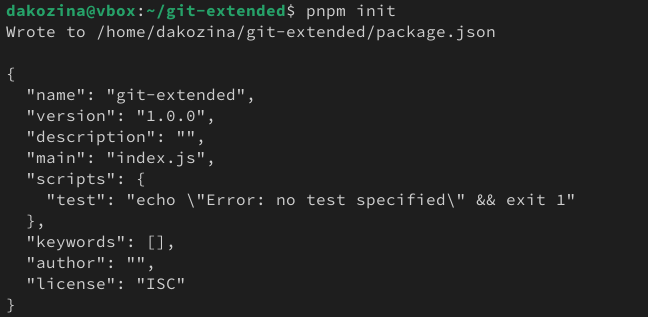


Рис. 11: Конфигурация

Отредактируем файл package.json (рис. 12).

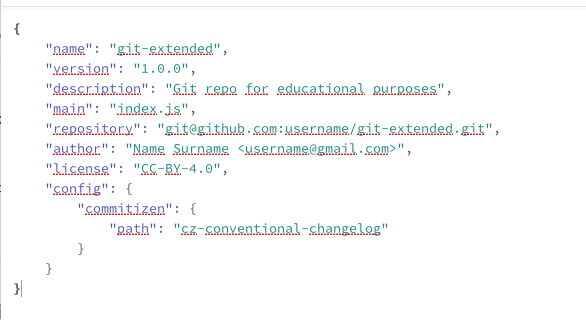
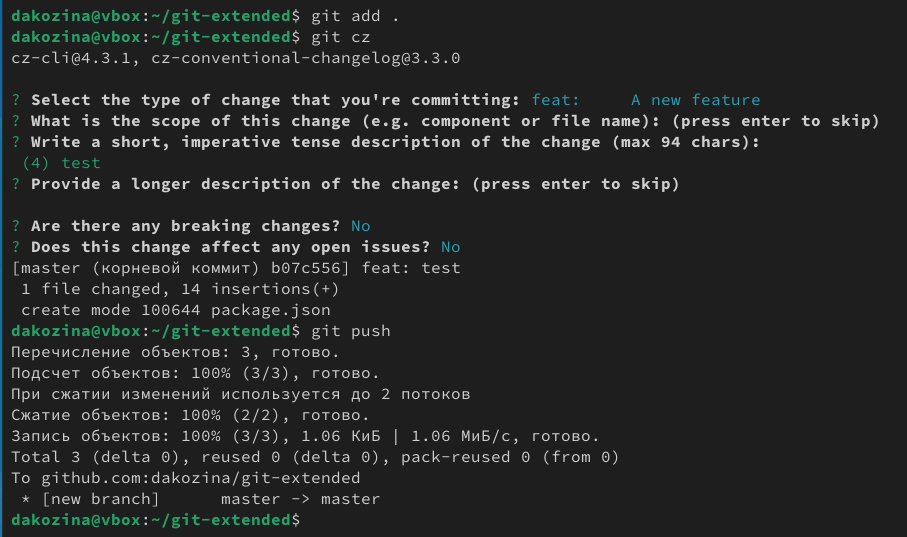


Рис. 12: package.json

Добавим новые файлы, выполним коммит и отправим на github (рис. **¿fig:013?**).



Инициализируем git-flow (рис. 13).

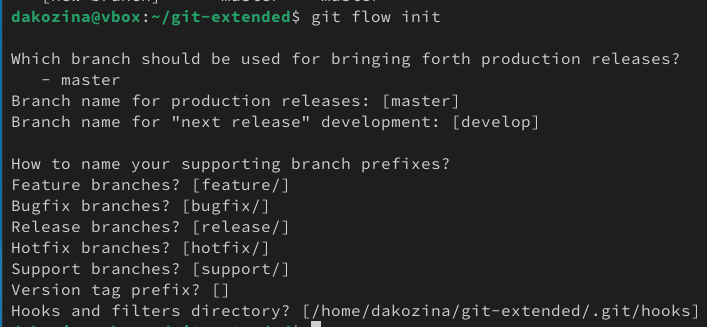


Рис. 13: Инициализация

Проверим, что мы на ветке develop (рис. 14).

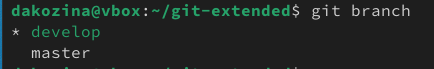


Рис. 14: Проверка

Загрузим весь репозиторий в хранилище (рис. 15).

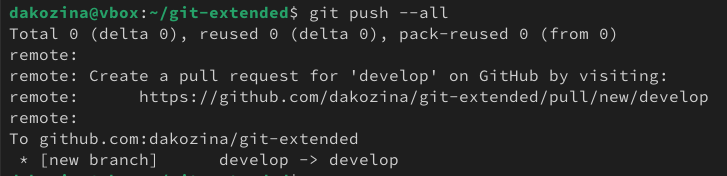


Рис. 15: Загрузка репозитория

Установим внешнюю ветку как вышестоящую для этой ветки (рис. 16).

Рис. 16: Установка внешней ветки

Рис. 16: Установка внешней ветки

Создадим релиз с версией 1.0.0 (рис. 17).

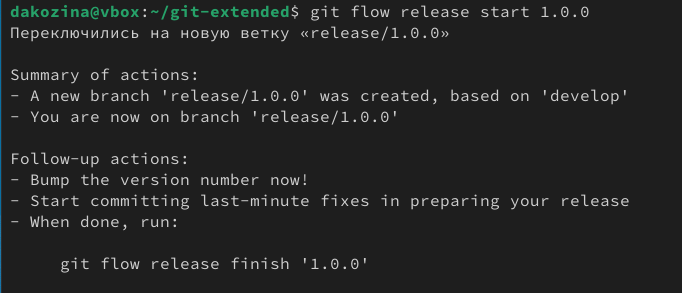


Рис. 17: Создание релиза

Создадим журнал изменений (рис. 18).

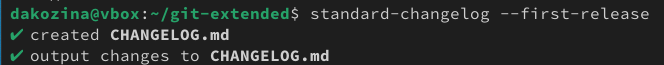


Рис. 18: Создание журнала

Создадим журнал изменений в индексе (рис. 19).

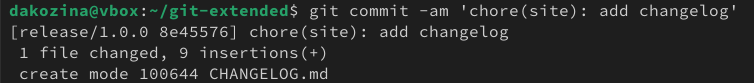
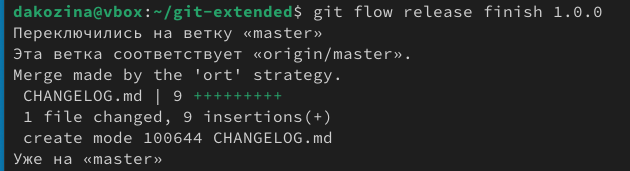


Рис. 19: Создание журнала

Зальем релизную ветку в основную ветку (рис. **¿fig:021?**).



Отправим данные на github (рис. 20).

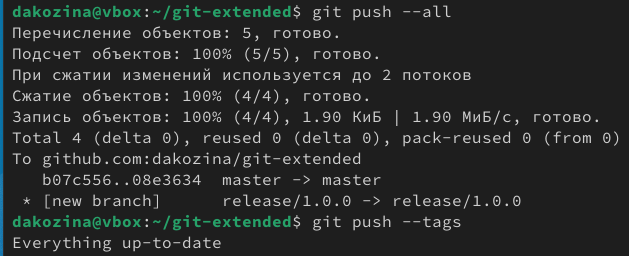


Рис. 20: Отправка данных

Создадим релиз на github. Для этошо будем использовать утилиты работы с github (рис. 21).

Рис. 21: Создание релиза

Рис. 21: Создание релиза

Создадим ветку для новой функциональности (рис. 22).

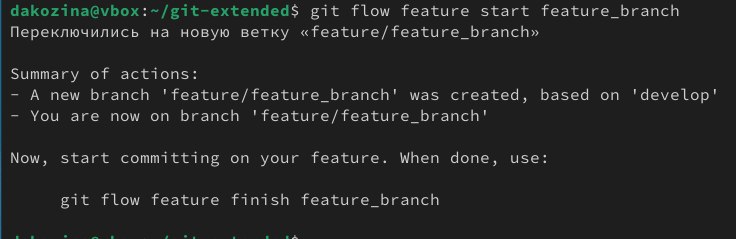
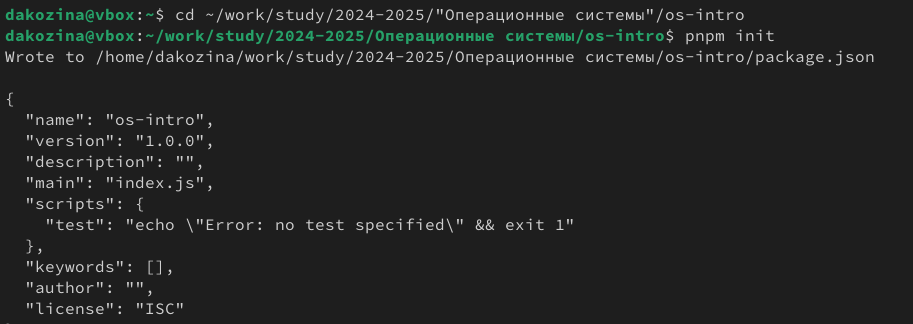


Рис. 22: Создание ветки

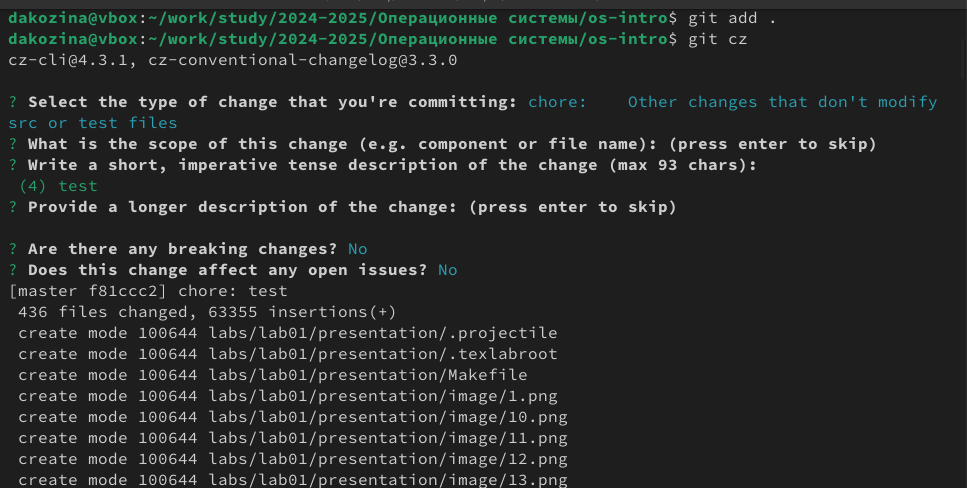
Продолжим работу с git как обычно. В конце разработки новой функциональности следующим шагом объединим ветку feature\_branch с develop (рис. **¿fig:025?**).

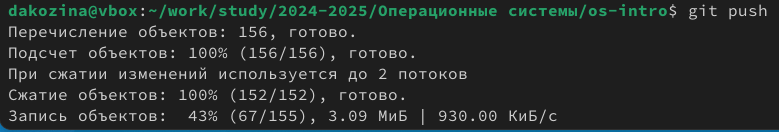


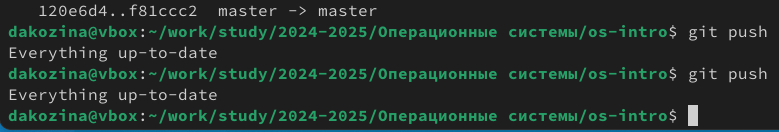
Проделаем все то же самое для нашего основного репозитория (рис. **¿fig:026?**, рис. **¿fig:027?**, рис. **¿fig:028?**, рис. **¿fig:029?**, рис. **¿fig:030?**).











# 4 Выводы

В ходе лабораторной работы мы получили навыки правильной работы с репозиторием git.