Отчёт по лабораторной работе №4

Операционные системы

Мурашов Иван Вячеславович

Содержание

# 1 Цель работы

Целью данной лабораторной работы является получение навыков правильной работы с репозиториями git.

# 2 Задание

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

# 3 Теоретическое введение

Gitflow Workflow опубликована и популяризована Винсентом Дриссеном.

Gitflow Workflow предполагает выстраивание строгой модели ветвления с учётом выпуска проекта.

Данная модель отлично подходит для организации рабочего процесса на основе релизов.

Работа по модели Gitflow включает создание отдельной ветки для исправлений ошибок в рабочей среде.

Последовательность действий при работе по модели Gitflow:

* Из ветки master создаётся ветка develop.
* Из ветки develop создаётся ветка release.
* Из ветки develop создаются ветки feature.
* Когда работа над веткой feature завершена, она сливается с веткой develop.
* Когда работа над веткой релиза release завершена, она сливается в ветки develop и master.
* Если в master обнаружена проблема, из master создаётся ветка hotfix.
* Когда работа над веткой исправления hotfix завершена, она сливается в ветки develop и master.

# 4 Выполнение лабораторной работы

## 4.1 Установка программного обеспечения

Устанавливаю git-flow (рис. 1).

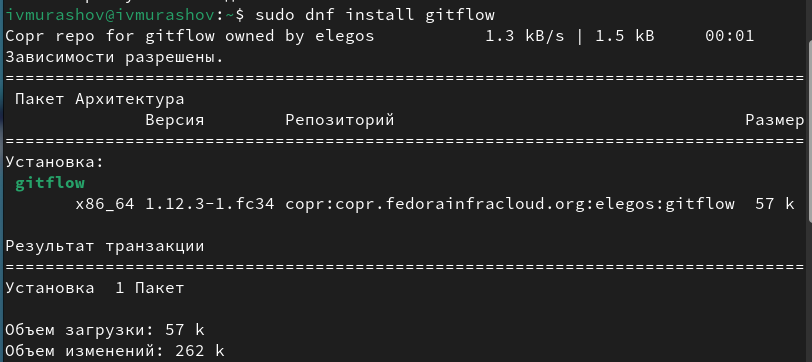


Рис. 1: Установка git-flow

Устанавливаю Node.js (рис. 2).

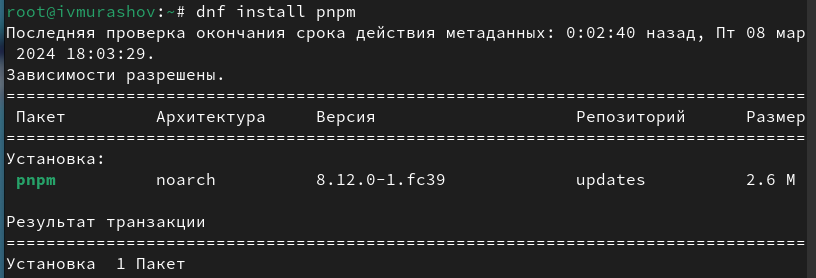


Рис. 2: Установка Node.js

Настраиваю Node.js (рис. 3).

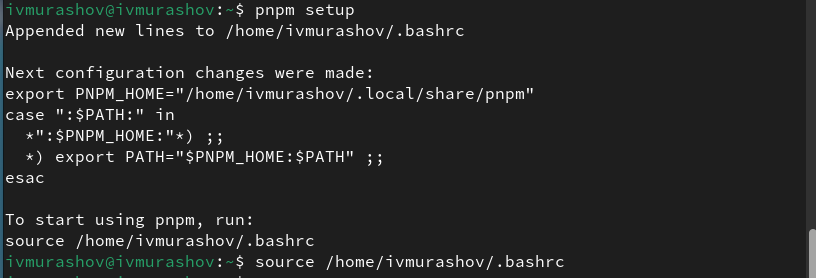


Рис. 3: Настройка Node.js

## 4.2 Общепринятые коммиты

Устанавливаю программу для помощи в форматировании коммитов (рис. 4).

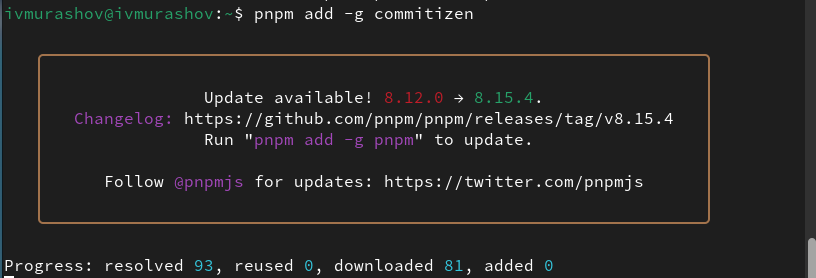


Рис. 4: Установка ПО

Устанавливаю программу для помощи в создании логов (рис. 5).

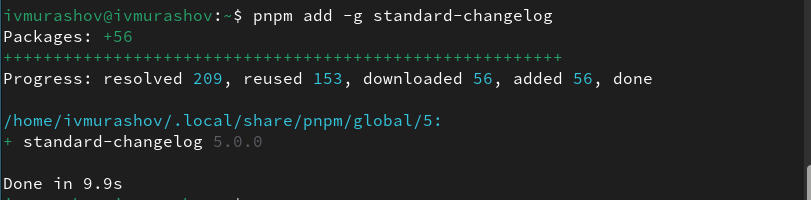


Рис. 5: Установка ПО

Создаю и клонирую репозиторий (рис. 6).

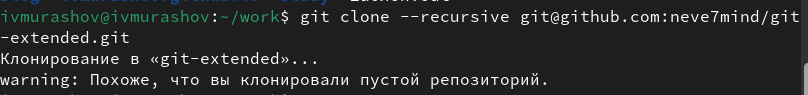


Рис. 6: Клонирование репозитория

Настраиваю репозиторий, делаю 1ый коммит и выкладываю его на github (рис. 7).

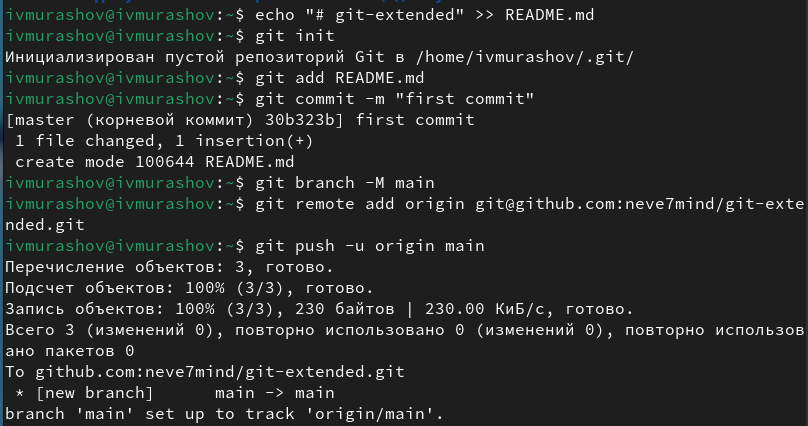


Рис. 7: Настройка репозитория

Настраиваю конфигурацию для для пакетов Node.js (рис. 8).

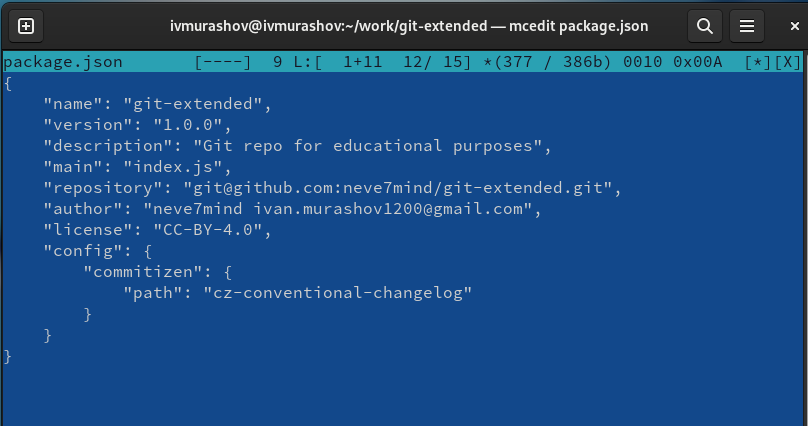


Рис. 8: Настройка конфигураций

Добавляю новые файлы, выполняю коммит и отправляю файлы на сервер (рис. 9).

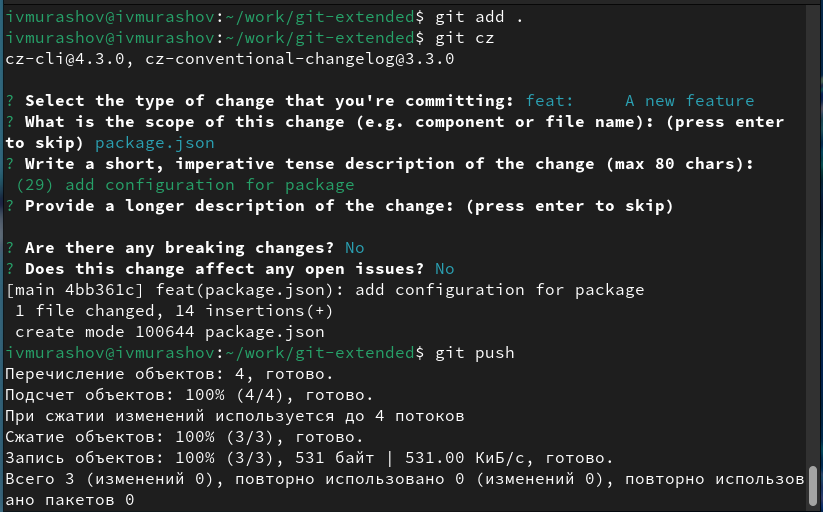


Рис. 9: Отправка файлов на сервер

Инициализирую git-flow (рис. 10).

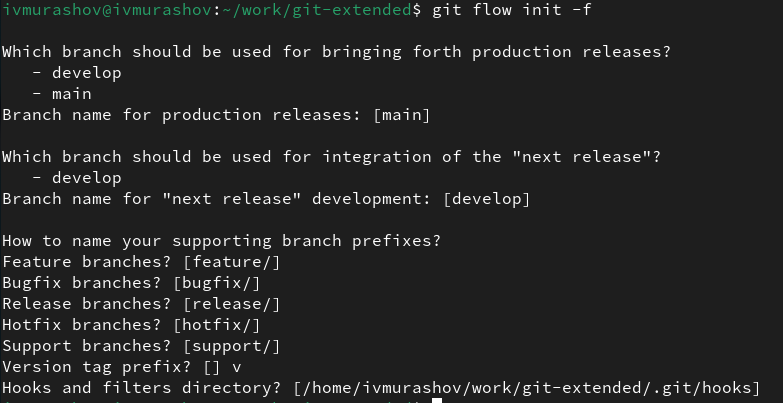


Рис. 10: Инициализация git-flow

Проверяю, что я на ветке develop, загружаю весь репозиторий в хранилище (рис. 11).

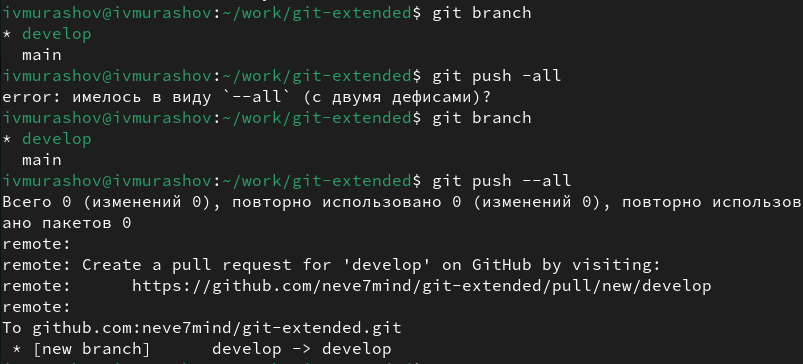


Рис. 11: Отправка файлов на сервер

Устанавливаю внешнюю ветку как вышестоящую для этой ветки (рис. 12).

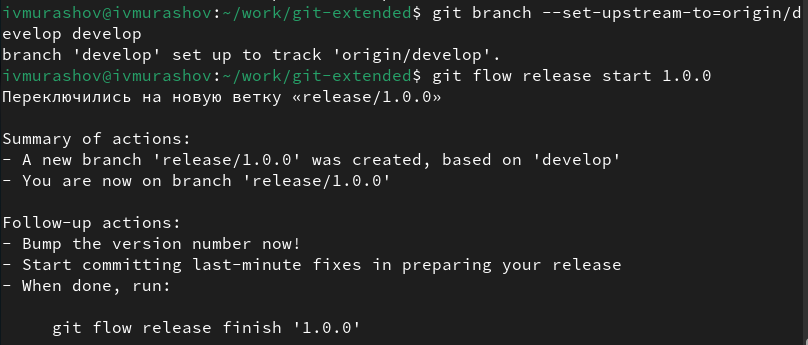


Рис. 12: Работа с ветками

Создадаю релиз с версией 1.0.0, журнал изменений, добавляю журнал изменений в индекс и заливаю релизную ветку в основную (рис. 13).

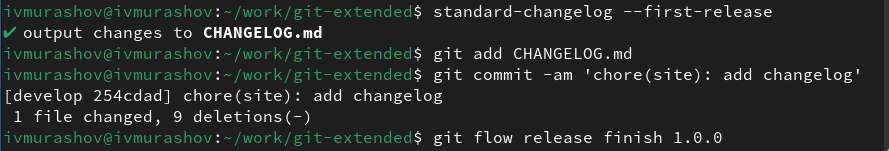


Рис. 13: Работа с ветками

Отправляю данные на github (рис. 14).

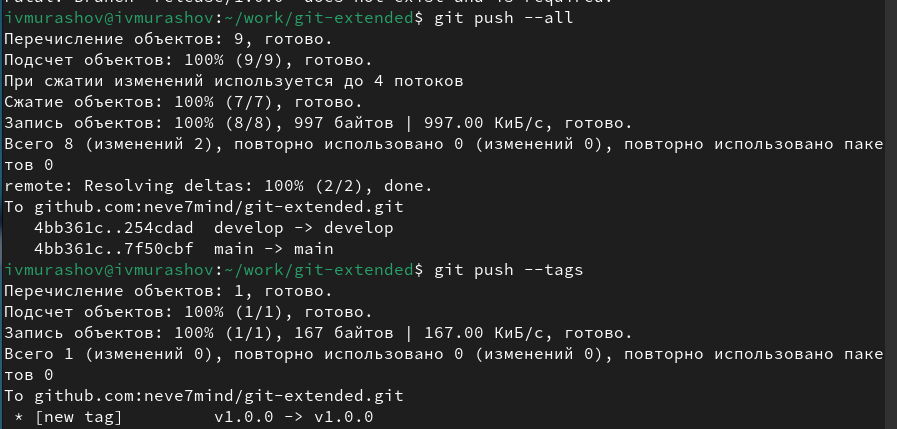


Рис. 14: Отправка файлов на сервер

Создадаю релиз на github (рис. 15).

Создание релиза

Рис. 15: Создание релиза

Создадаю ветку для новой функциональности и по окончании разработки новой функциональности объединяю ветку feature\_branch c develop (рис. 16).

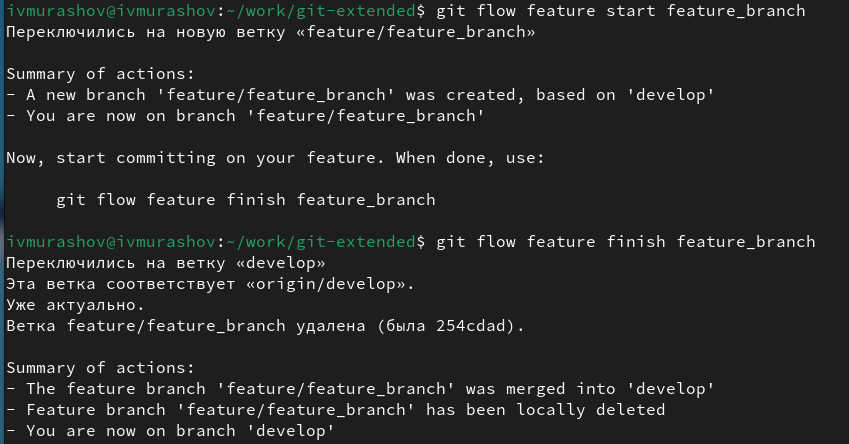


Рис. 16: Создание и объединение веток

Создадаю релиз с версией 1.2.3 (рис. 17).

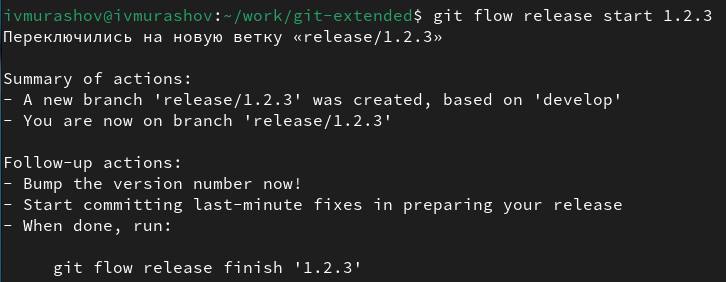


Рис. 17: Создание релиза

Обновляю номер версии в файле package.json (рис. 18).

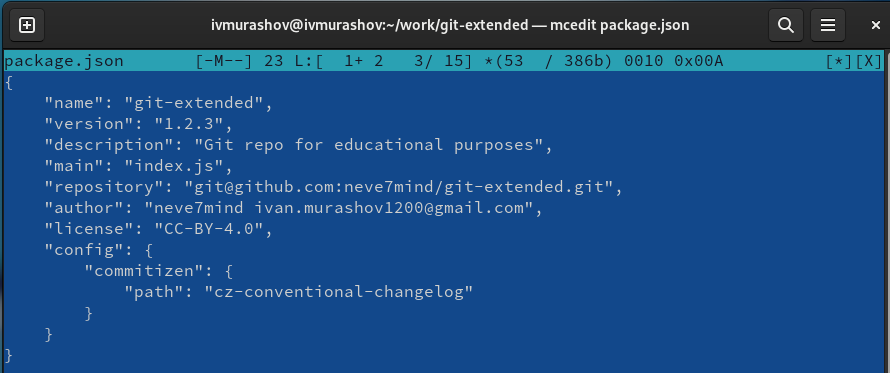


Рис. 18: Настройка конфигураций

Заливаю релизную ветку в основную, после чего отправляю данные на github и создаю релиз на github с комментарием из журнала изменений (рис. 19).

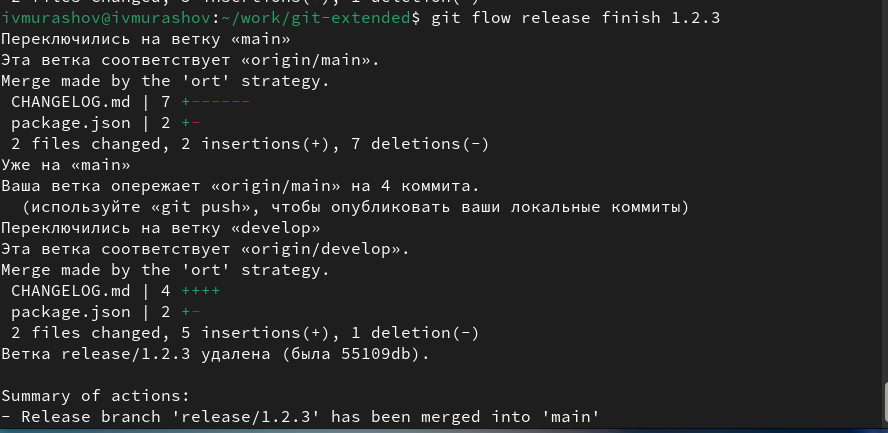


Рис. 19: Работа с ветками

# 5 Выводы

В ходе выполнения данной лабораторной работы я получил навыки правильной работы с репозиториями git.