Отчет по лабораторной работе №4

Дисциплина: Операционные системы

Иванов Сергей Владимирович

Содержание

# 1 Цель работы

1. 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 Выполнение лабораторной работы

Установим git-flow из коллекции репозиториев Copr. Активируем репозиторий Corp. (рис. 1).

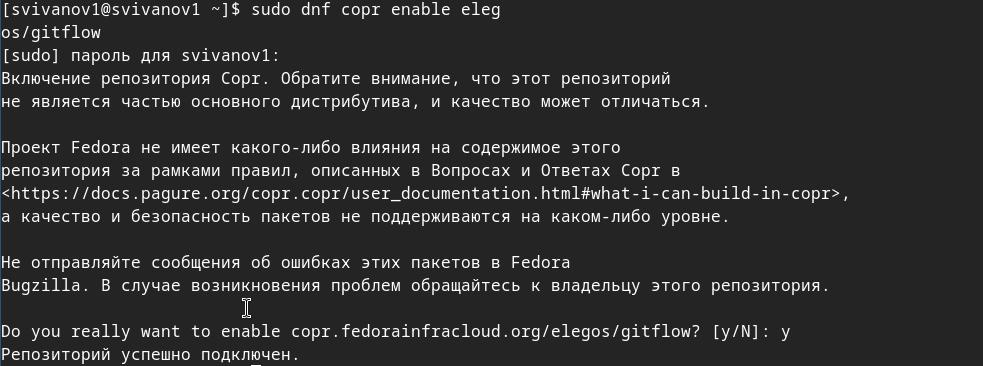


Рис. 1: Активация Corp

Установим git-flow - dnf install gitflow (рис. 2).

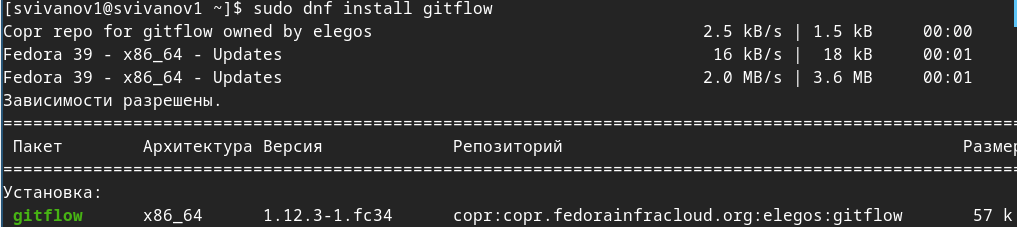
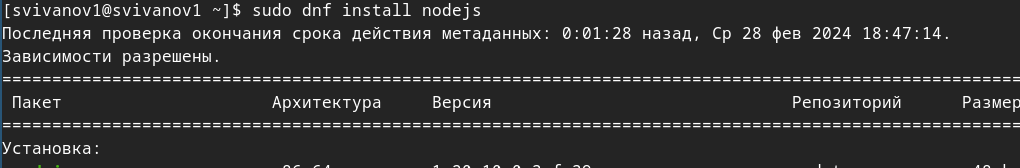


Рис. 2: Установка git-flow

Установим Node.js (рис. 3, 4).



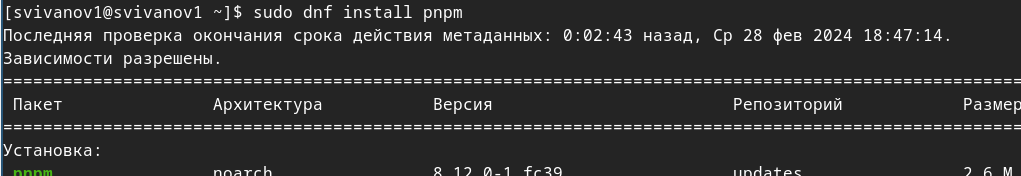


Рис. 3: Установка Node.js

Настроим Node.js. Запустим ‘pnpm setup’ и перелогинимся (рис. 5).

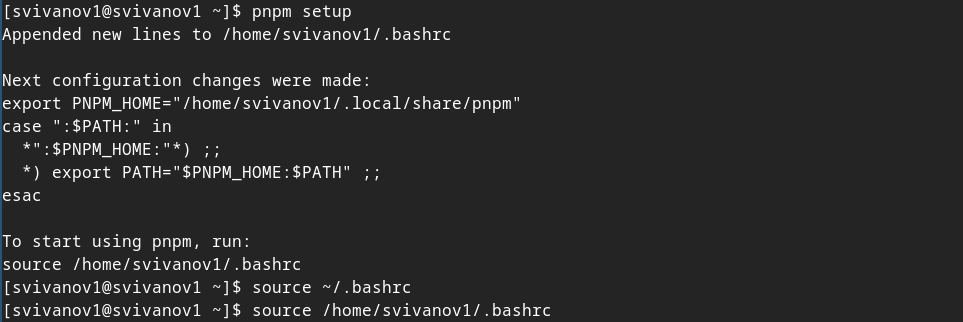


Рис. 4: Настройка Node.js

Установим программу для помощи в форматировании коммитов. (рис. 6).

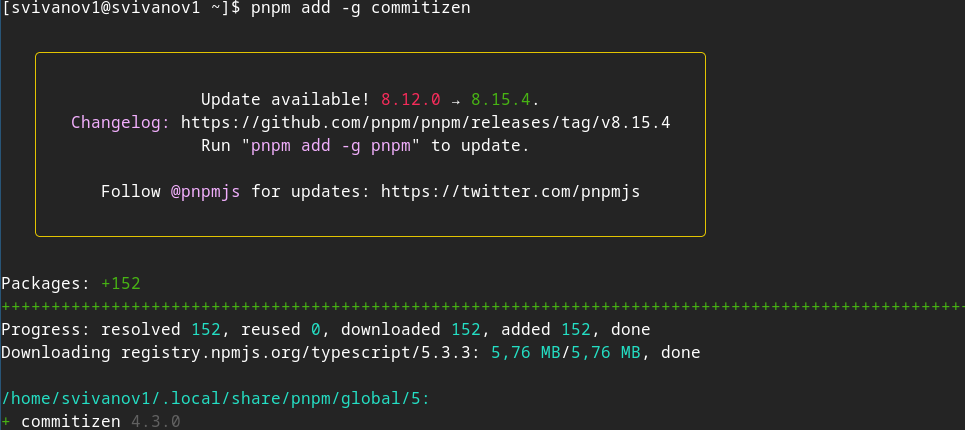


Рис. 5: Commitzen

Установим программу для помощи в создании логов. (рис. 7).

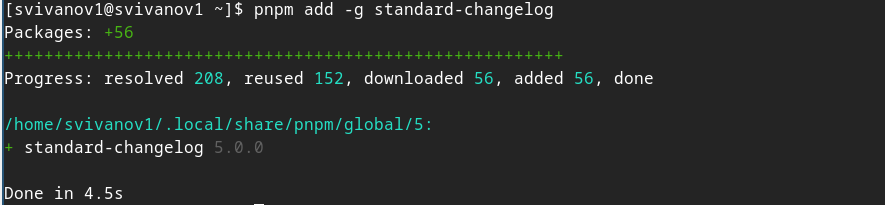


Рис. 6: Standard-changelog

Создадим репозиторий ‘git-extended’ (рис. 8).

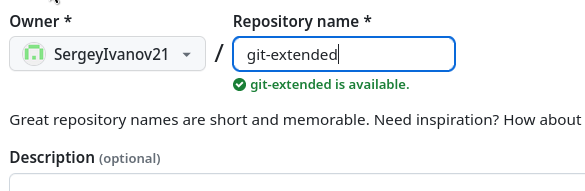


Рис. 7: Создание репозитория

Клонируем репозиторий и создаем файл README.md (рис. 9).

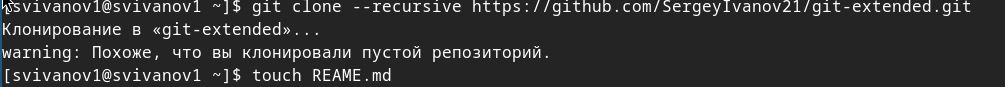


Рис. 8: Клонирование

Делаем коммит и выкладываем на github (рис. 10).

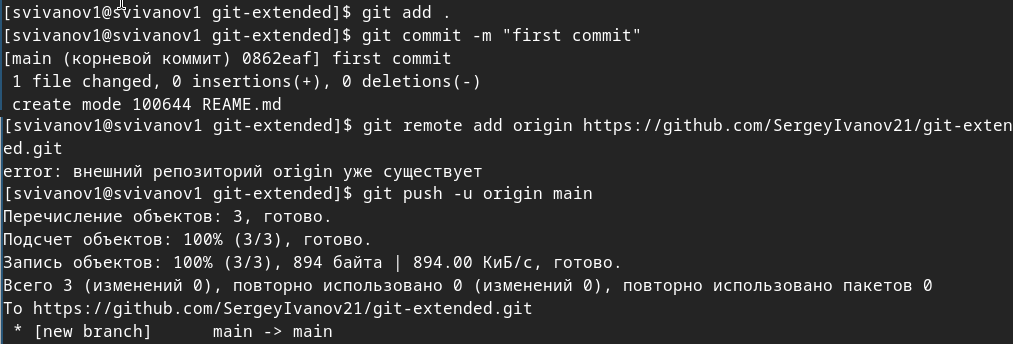


Рис. 9: Отправка на сервер

Сконфигурим формат коммитов. Для этого добавим в файл package.json команду для формирования коммитов (рис. 11).

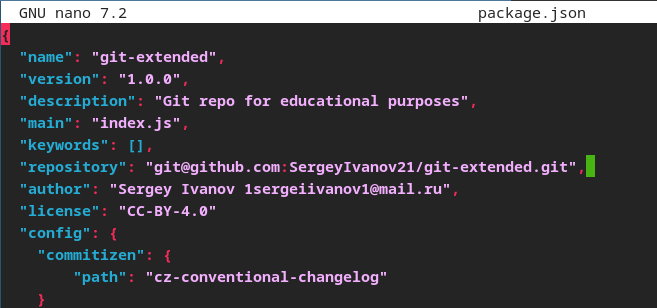


Рис. 10: Конфигурируем коммит

Добавим новые файлы, выполним коммит, отправим на github (рис. 12).

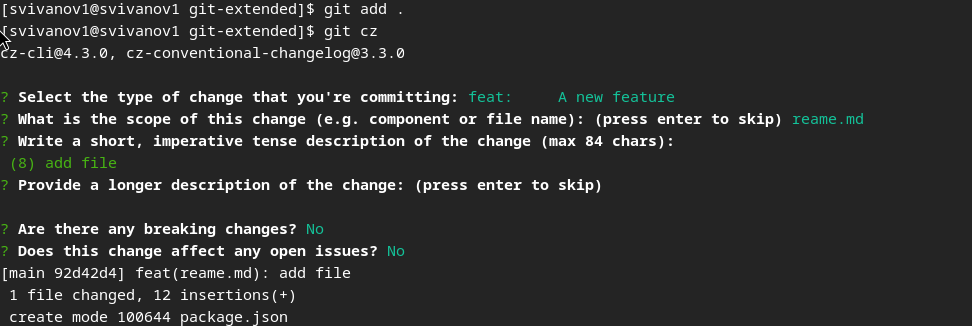


Рис. 11: Отправка коммита

Инициализируем git-flow, префикс для ярлыков установим в v (рис. 13).

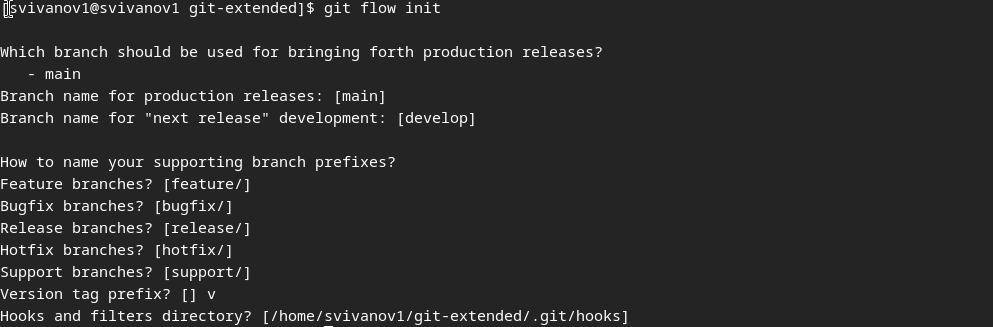


Рис. 12: Конфигурация git-flow

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

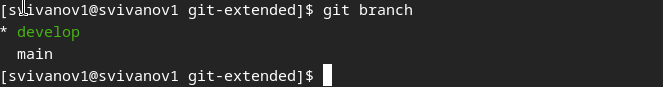


Рис. 13: develop

Загрузим весь репозиторий в хранилище (рис. 15).

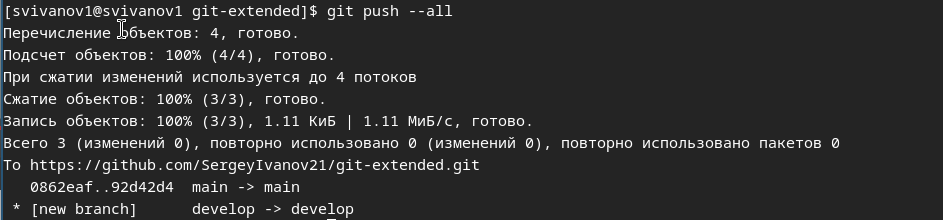


Рис. 14: Загрузка репозитория

Установим внешнюю ветку как вышестоящую для этой ветки и создадим релиз с версией 1.0.0 (рис. 16).

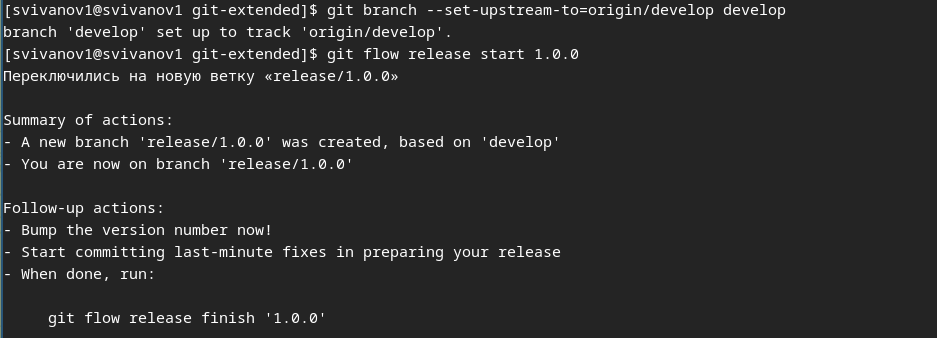


Рис. 15: Создаем релиз

Создадим журнал изменений и добавим журнал изменений в индекс (рис. 17).

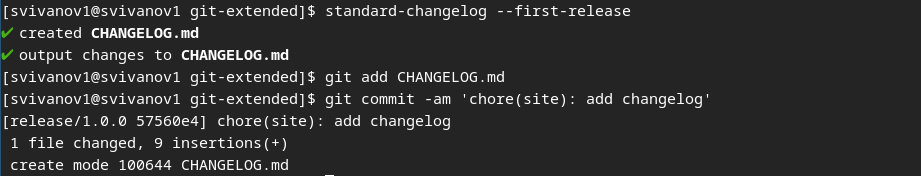


Рис. 16: Журнал изменений

Зальём релизную ветку в основную ветку (рис. 18).

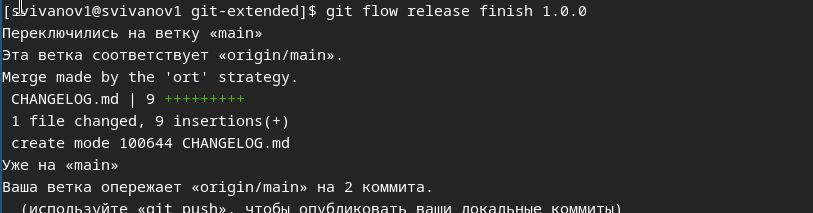


Рис. 17: Заливаем релизную ветку

Отправим данные на github (рис. 19).

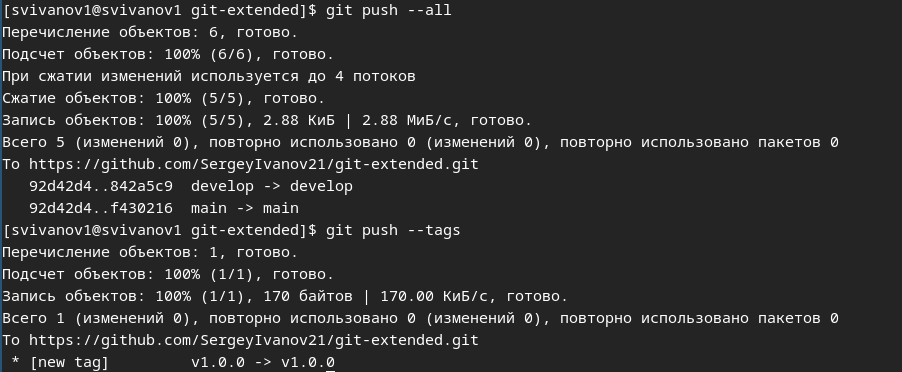


Рис. 18: Отправка

Создадим релиз на github. (рис. 20).

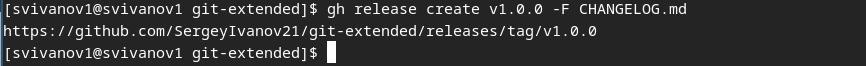


Рис. 19: Создаем релиз

Создадим ветку для новой функциональности, объединим ветку feature\_branch c develop. (рис. 21).

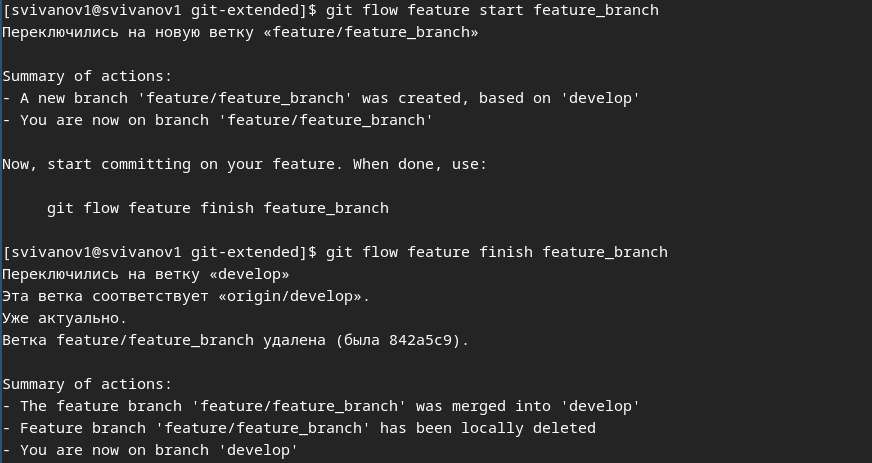


Рис. 20: Новая ветка

Создадим релиз с версией 1.2.3. (рис. 22).

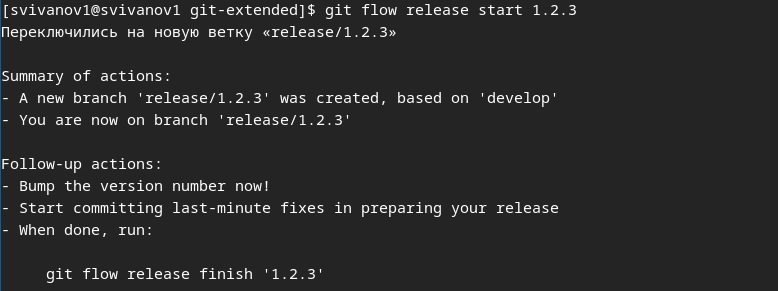


Рис. 21: Создаем релиз

Обновим номер версии в файле package.json. Установим её в 1.2.3. (рис. 23).

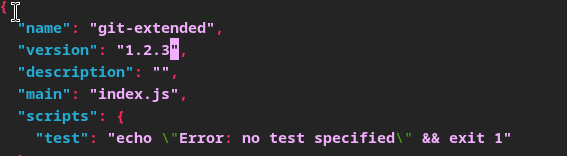


Рис. 22: Обновляем номер версии

Создадим журнал изменений (рис. 24).

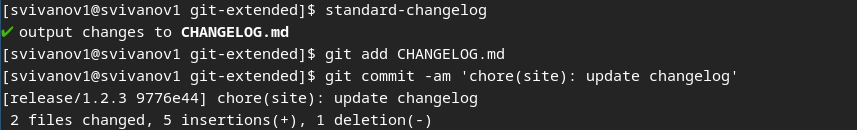


Рис. 23: Создаем релиз

Добавим журнал изменений в индекс и зальём релизную ветку в основную ветку. (рис. 25).

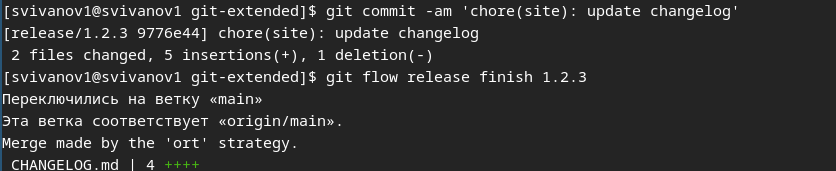


Рис. 24: Заливаем ветку

Отправим данные на github. (рис. 26).

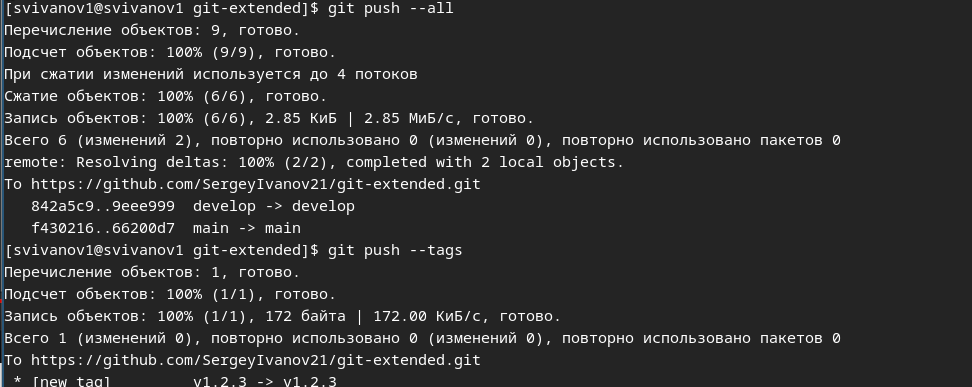


Рис. 25: Отправляем данные

Создадим релиз на github с комментарием из журнала изменений (рис. 27)

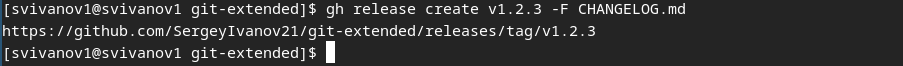


Рис. 26: Создаем релиз с комментарием

# 4 Выводы

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