Лабораторная работа No4

Операционные системы

Нелиа Нджову

Содержание

# 1 Цель работы

Цель работы является получение навыков правильной работы с репозиториями git.

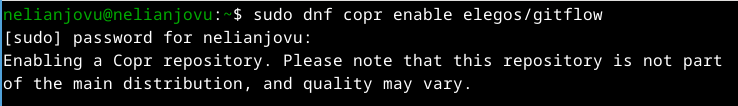
# 2 Задание

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

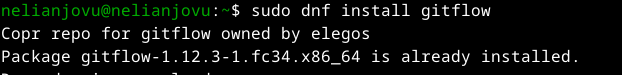
# 3 Выполнение лабораторной работы

**1. Установка git-flow**

Сначала я включаю репозиторий copr(рис.1),после этого устанавливаю gitflow(рис.2)



включение репозиторий copr



установки gitflow

**2. Установка Node.js**

Я устанавливаю nodejs.На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов(рис.3)

установки nodejs

установки nodejs

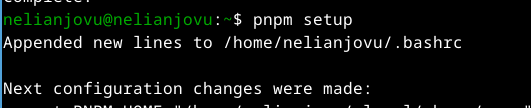
Потом я устанавливаю pnpm(рис.4)

установки pnpm

установки pnpm

**3. Настройка Node.js**

Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH.Я запускаю команду pnpm setup. Эта команда автоматически устанавливает yarn в переменную path(рис 5)



настройка node.js

Затем я выпольняю команду source ~/.bashrc(рис.6)

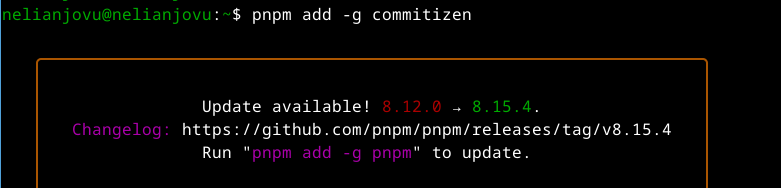
перелогинирование

перелогинирование

**4. Общепринятые коммиты**

*commitizen*

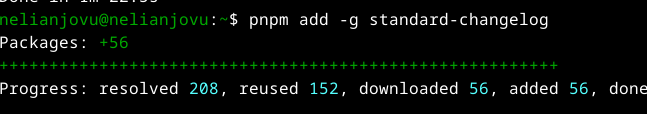
Я запускаю команду, которая используется для помощи в форматировании коммитов.При этом устанавливается скрипт git-cz, который мы и будем использовать для коммитов(рис.7)



commitizen

*standard-changelog*

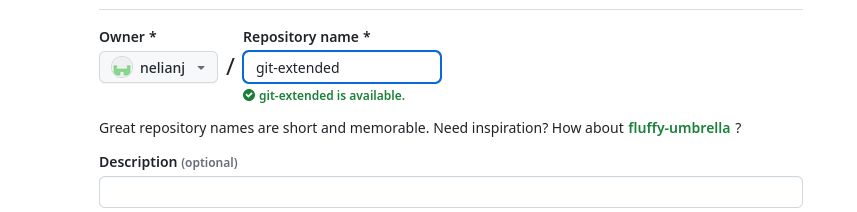
Я запускаю команду, которая испоьзуется для помощи в создании логов(рис.8)



standard-changelog

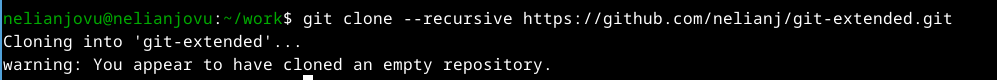
*Практический сценарий использования git*

Я создаю новый репозиторий под названием git-extended(рис.9)



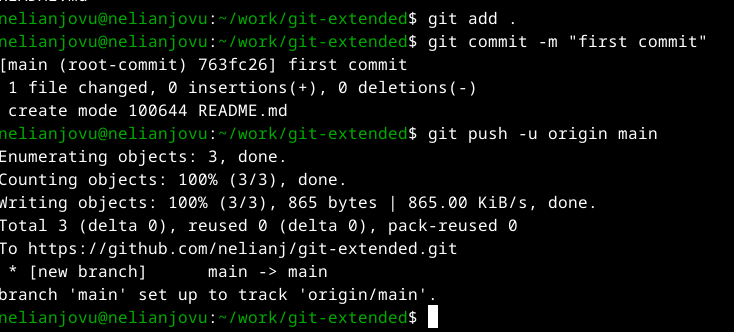
создание новый репозиторий

Я клонирую его в рабочную папку(рис.10)



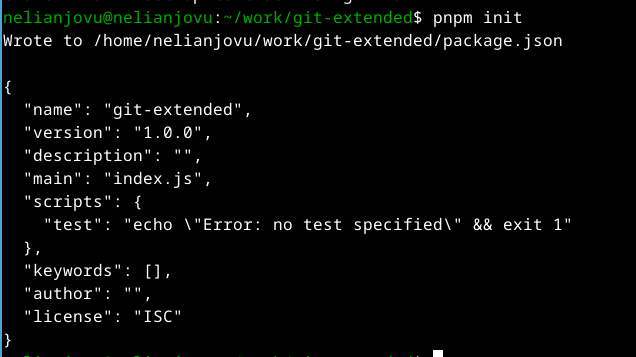
клонирование новый репозиторий

Потом я делаю первый коммитов и выкладываю его на github(рис.11)



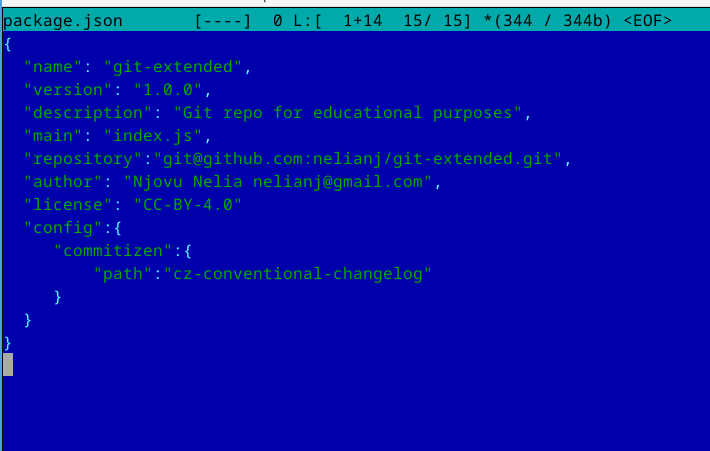
первый коммитов

Я создаю файл package.json с помощью команды pnpm init.Этот файл важен, поскольку он содержит метаданные о проекте(рис.12)



создание файла

Я сконфигурирую формат коммитов. Для этого я дабавлю в файл package.json команду для формирования коммитов(рис.13)



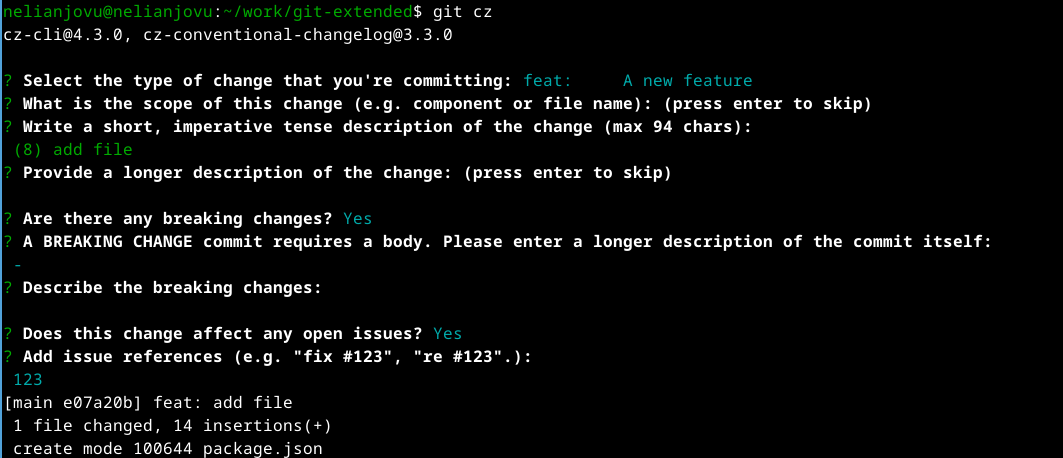
Конфигурация для пакетов Node.js

Потом добавлю новый файл на github(рис.14)

добавление файла на git

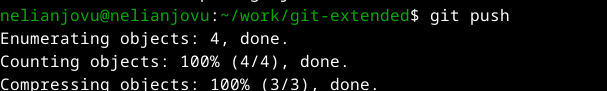
добавление файла на git

Я выпоняю коммитов с помощью git cz(рис.15)



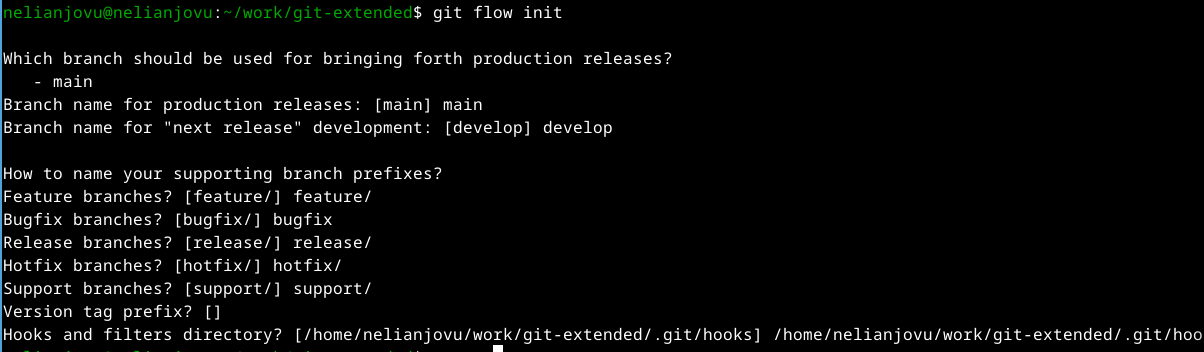
выполнение коммитов

Затем я отправляю на github(рис.16)



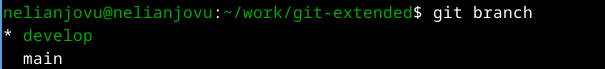
отправление файла

Я инициализирую git flow с помощью команду git flow init(рис.17)



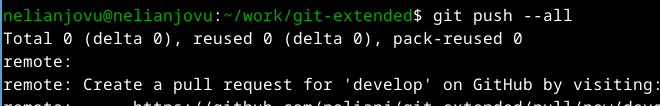
инициализирование git flow

Я проверяю что я на ветке develop(рис.18)



проверка если я на ветке develop

Я загружаю весь репозиторий в хранилище(рис.19)



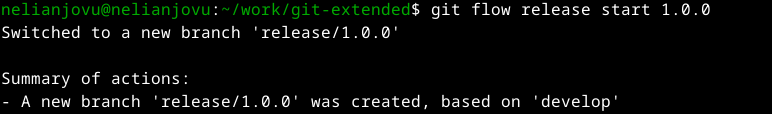
загрузка весь репозиторий

Я устанавливаю внешнюю ветку как вышестоящую для этой ветки(рис.20)

вышестоящая ветка

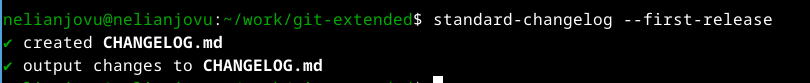
вышестоящая ветка

Я создаю релиз с версиеи 1.0.0(рис.22)



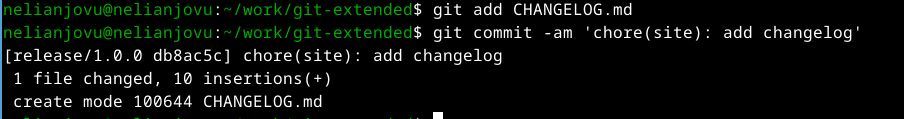
создание релиз с версиеи 1.0.0

Потом я создаю журнал изменений(рис.23)



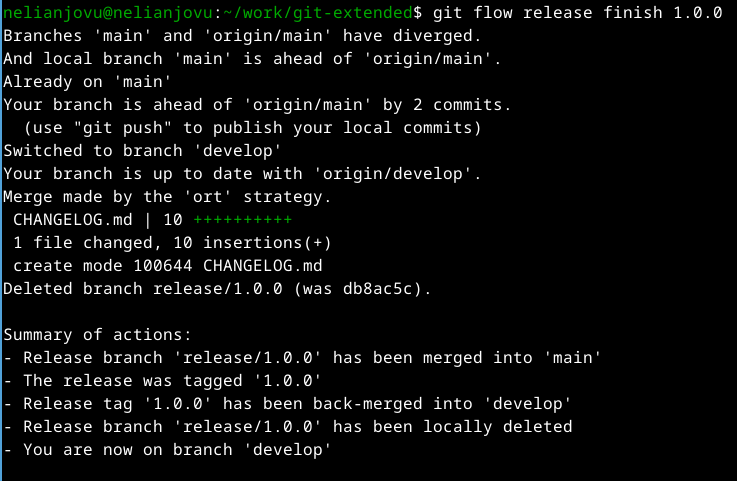
создание журнала изменений

После этого я дабовлю журнал изменений в индекс(рис.24)



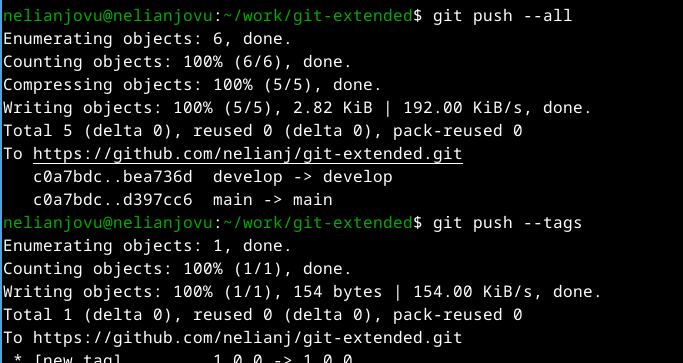
дабовление журнала изменнений в индекс

Я заполняю релизную ветку в основную ветку(рис.25)



заполнения релизной ветки

Я отправляю данные на github(рис.26)



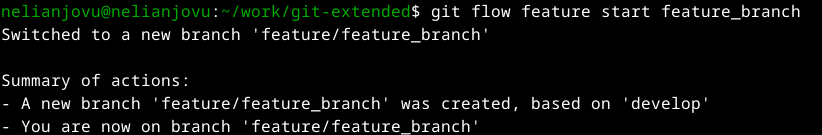
отправление на github

Я создаю релиз на github.Для этого буду использовать утилиты работы с github “gh release create v1.0.0 -F CHANGELOG.md”(рис.27)

создание релиз на github

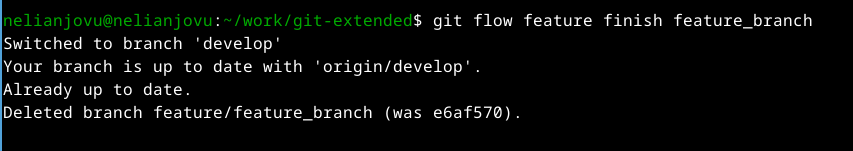
создание релиз на github

Я создаю ветку для новой функциональности(рис.28)



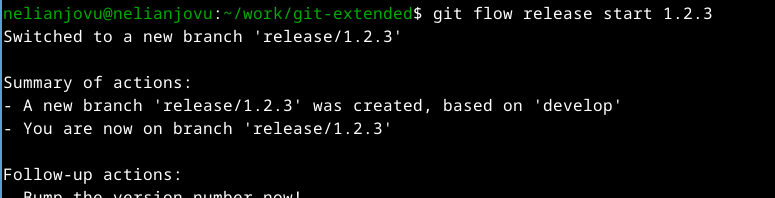
создание ветки

По окончании разработки новой функциональности следующим шагом следует объединить ветку feature\_branch c develop(рис.28)



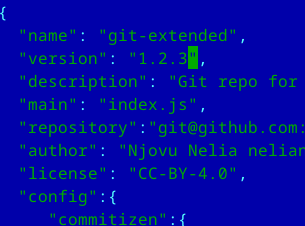
объединение ветки

Я создаю релиз с версиеи 1.2.3(рис.29)



создание релиз с версиеи 1.2.3

Я обновлю номер версии в файле package.json. Установливаю её в 1.2.3(рис.30)



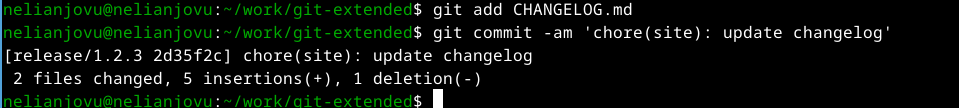
обновление номер версии

Я создаю журнал изменений(рис.31)

создание журнала изменений

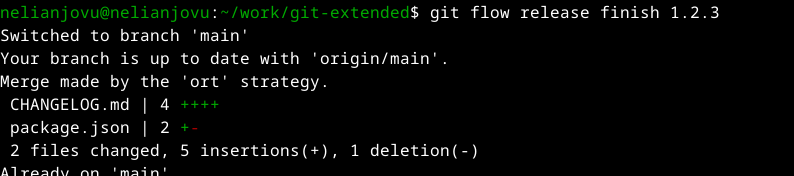
создание журнала изменений

Я добавляю журнал изменений в индекс(рис.32)



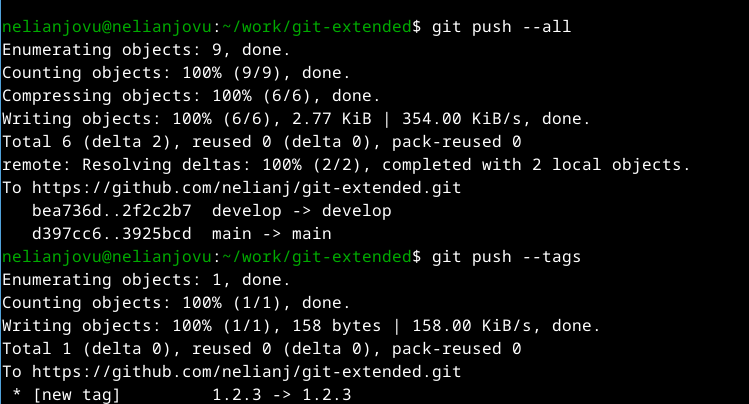
дабовление журнала изменнений в индекс

Заполняю релизную ветку в основную ветку(рис.33)



заполнение релизной ветки

Я отправяю данные на github(рис.34)



отправление на github

Я создаю релиз на github с комментарием из журнала изменений(рис.35)

создание релиз на github

создание релиз на github

# 4 Выводы

Выполняя эту лабораторную работу, я получаю навыки правильной работы с репозиториями git.

# Список литературы

Лабораторная работа № 4