Отчёт по лабораторной работе №4

Операционные системы

Федорова Анжелика Игоревна

Содержание

# 1 Цель работы

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.

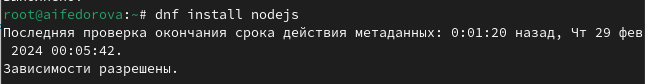
# 3 Теоретическое введение

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

Выполняем установку из коллекции репозиториев Сorp (рис.1 и рис.2).

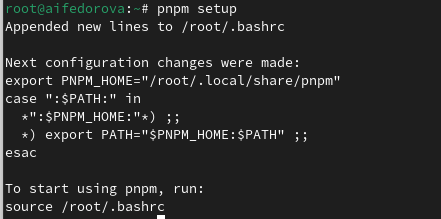




Устанавливаю пакеты Node.js, на которых базируется программное обеспечение для семантического версионирования и общепринятых коммитов (рис.3).



Запускаю нужную команду pnpm setup (рис.4)



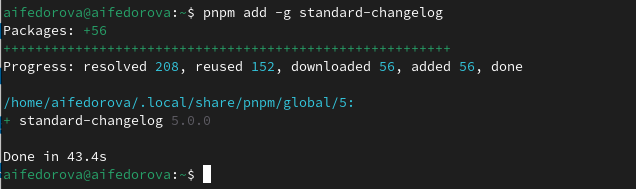
Выполняю команду source ~/.bashrc, для того, чтобы работать с окружением для git-flow. (рис.5)



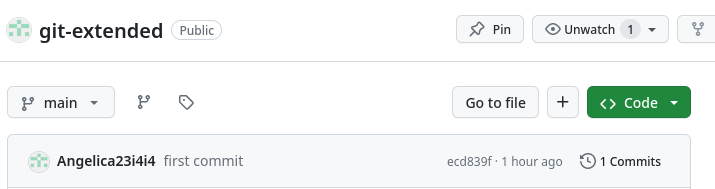
Запускаю команду сommitizen для помощи в форматировании коммитов.(рис.6)



Выполняю команду standard-changelog для помощи в создании логов.(рис.7)



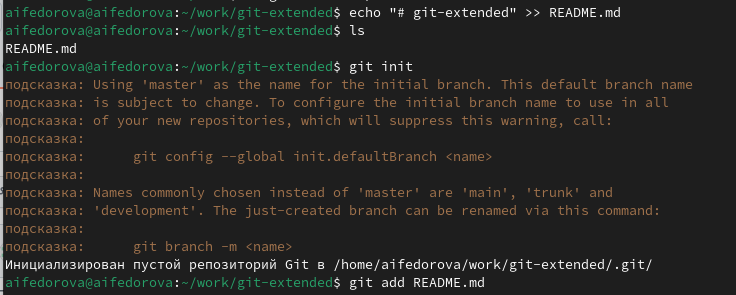
Теперь создаю репозиторий git-extended на Github(рис.8)



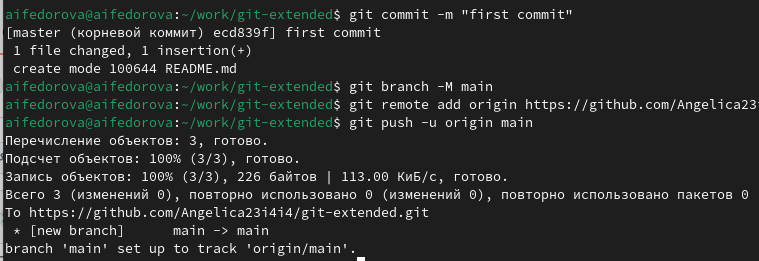
Теперь создаю репозиторий git-extended на консоли и захожу туда (рис.9)



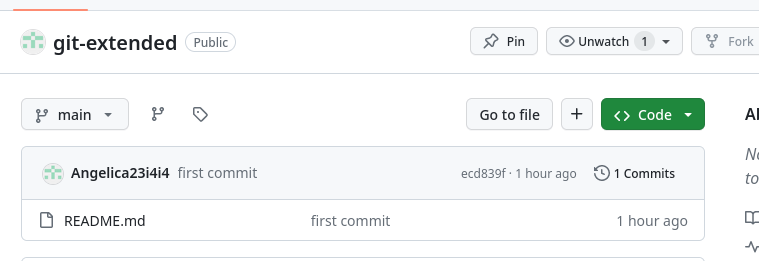
Создаю первый файл README.md и инициализию репозиторий(рис.10)



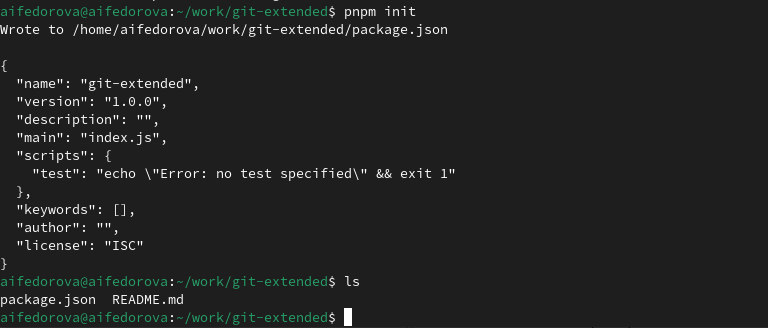
Совершаю первый коммит с помощью команды git commit -m “first commit” и подключаю удаленный репозиторий к консоли с помощью команды git remote add origin (ссылка на репозиторий). Затем отправляю изменения в репозиторий на сайте. (рис.11)



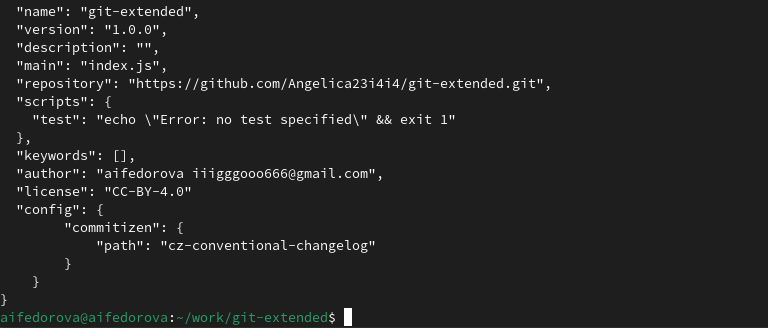
Вижу, что изменнеия ыли успешно отправлены (рис.12)



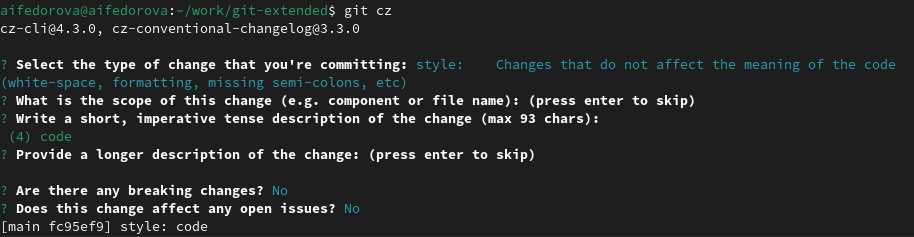
Создаю файл package.json с помощью команды pnpm init (рис.13)



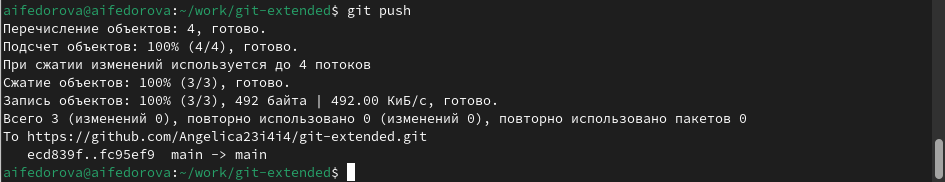
Изменяю лицензию в файле на CC-BY-4.0 и добавляю код с config как было дано в лабораторной работе(рис.14)



Добавлю новые файлы с помощью git add и выполню коммит с помощью git cz(рис.15)



Отправляю изменения в локальный репозиторий(рис.16)



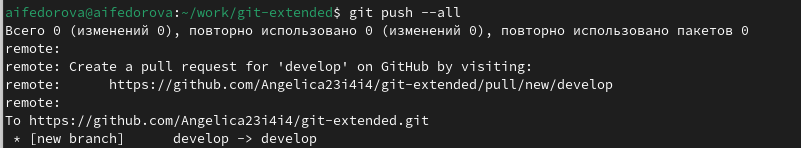
Для работы с конфигурацией git-flow я сначала инициализирую git-flow с git flow init. Префикс для ярлыков установим в v(рис.17)



Проверяю , что я нахожусь на ветке develop (рис.18)



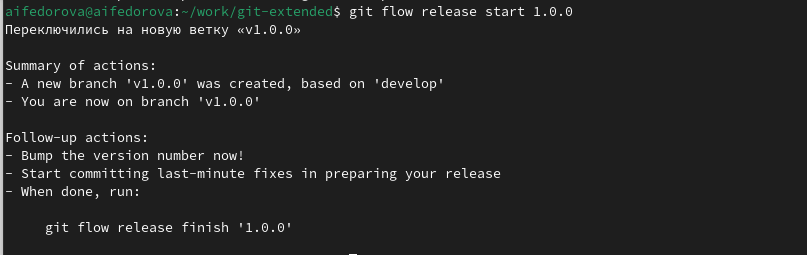
Загружаю весь репозиторий в хранилище с помощью git push –all (рис.19)



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



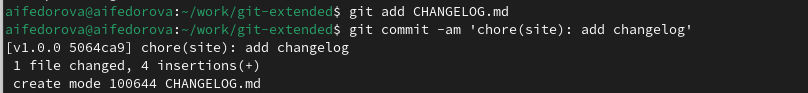
Создадаю релиз с версией 1.0.0 (рис.21)



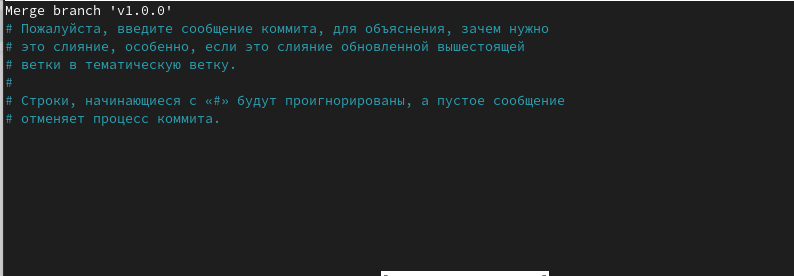
Создам журнал изменений (рис.22)



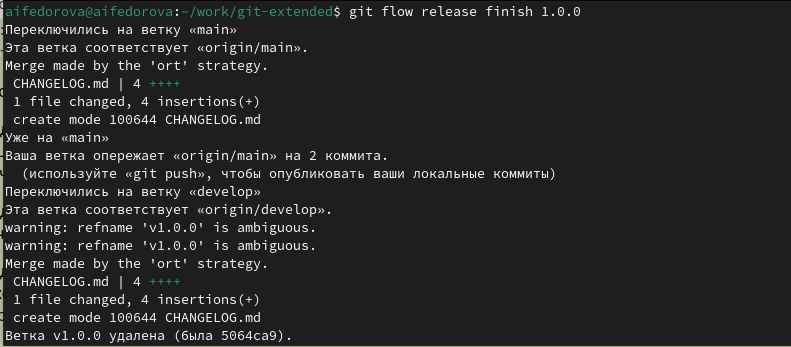
Добавлю журнал изменений в индекс (рис.23)



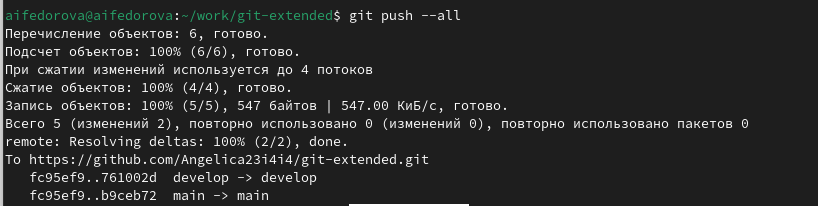
Теперь я должна положить релизную ветку в основную ветку. Далее я ввожу необходимые сообщения для обозначения цели изменений (рис.24)



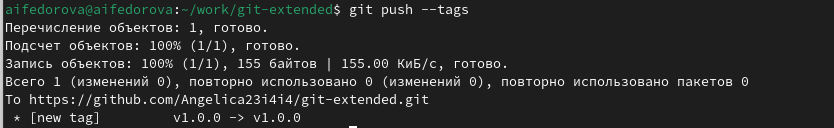
Вижу, что данный процесс был успешно завершен (рис.25)



Далее я отправляю данные на github (рис.26)



Затем отправляю в репозиторий все проставленные теги (рис.27)



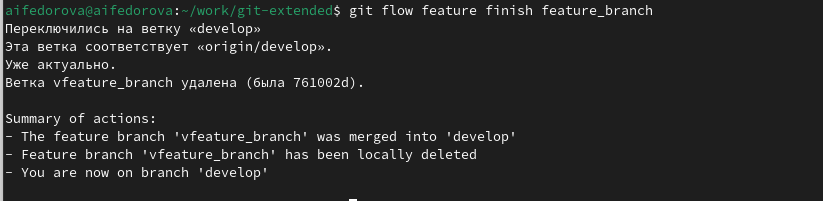
Создам релиз на github. Для этого буду использовать утилиты работы с github (рис.28)



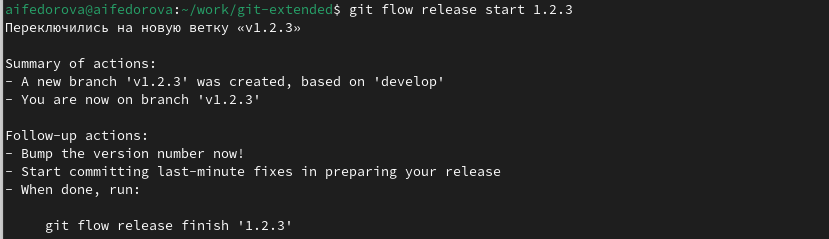
Для дополнительного примеры работы с релизами я создам ветку для новой функциональности (рис.29)



Объединяю ветку feature\_branch c develop (рис.30)



Теперь я создаю релиз с версией 1.2.3 (рис.31)



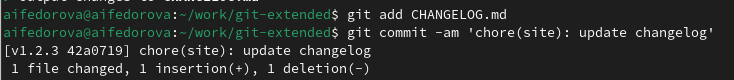
Я снова редактирую файл package.json, меняя текущую версию на 1.2.3. (рис.32)



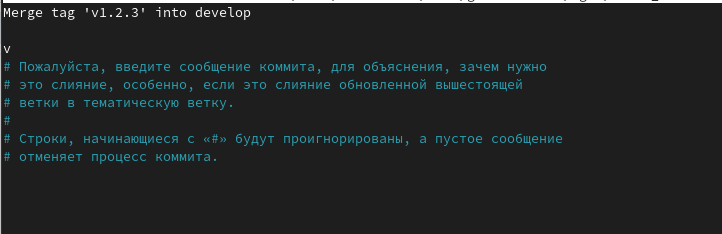
Снова создам журнал изменений (рис.33)



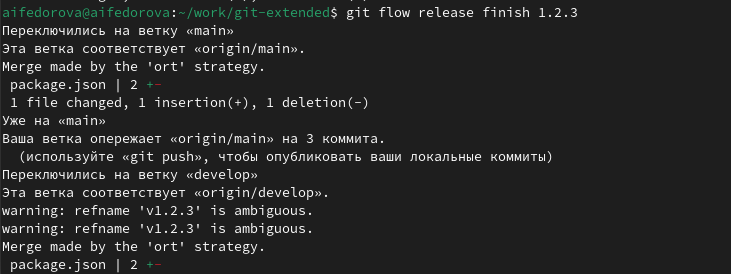
Добавлю журнал изменений в индекс (рис.34)



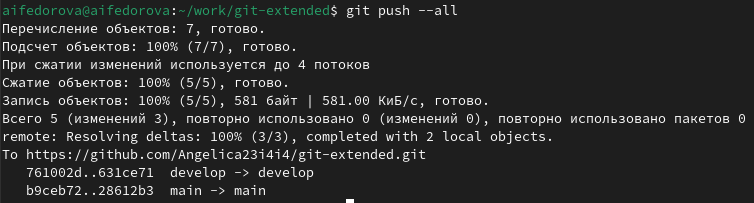
Снова перенаправляю релизную ветку в основную ветку и отвечаю на необходимые вопросы для этого(ярлык v оставлю прежним) (рис.35)



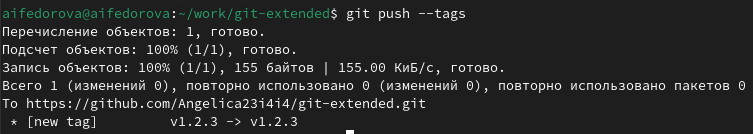
Вижу, что процесс завершился без ошибок(рис.36)



Отправляю данные на github (рис.37)



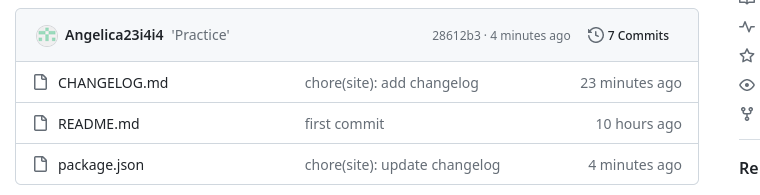
Далее отправляю все теги на github(рис.38)



Снова создам релиз на github с комментарием из журнала изменений (рис.39)



Захожу на в свой репозиторий на сайте и вижу, что продленные изменения и созданные файлы успешно лежат в созданном репозитории.(рис.40)



# 5 Выводы

Я получила навыки правильной работы с репозиториями git.

# 6 Список литературы

[Лабораторная работа №4](https://esystem.rudn.ru/mod/page/view.php?id=1098794)