Отчет по выполнению Лабораторной работы №4

Дисциплина: Архитектура компьютеров и опрерационные системы

Ефремова Полина Александровна

Содержание

Список иллюстраций

Список таблиц

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

Семантическое версионирование описывается в манифесте семантического версионирования. Кратко его можно описать следующим образом: Версия задаётся в виде кортежа МАЖОРНАЯ\_ВЕРСИЯ.МИНОРНАЯ\_ВЕРСИЯ.ПАТЧ. Номер версии следует увеличивать: МАЖОРНУЮ версию, когда сделаны обратно несовместимые изменения API. МИНОРНУЮ версию, когда вы добавляете новую функциональность, не нарушая обратной совместимости. ПАТЧ-версию, когда вы делаете обратно совместимые исправления. Дополнительные обозначения для предрелизных и билд-метаданных возможны как дополнения к МАЖОРНАЯ.МИНОРНАЯ.ПАТЧ формату.

Спецификация Conventional Commits: Соглашение о том, как нужно писать сообщения commit’ов. Совместимо с SemVer. Даже вернее сказать, сильно связано с семантическим версионированием. Регламентирует структуру и основные типы коммитов.

# 4 Выполнение лабораторной работы

## 4.1 Установка ПО. Устнановка git-flow, Node.js. Настройка Node.js. Общепринятые коммиты

1. Устанавливаем nodejs и pnpm (рис. fig. 1).

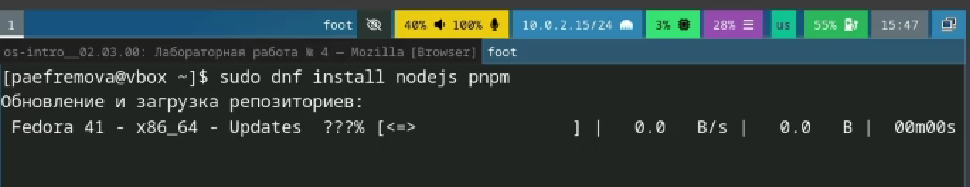


Рис. 1: Установка nodejs и pnpm

1. Устанавливаем git-glow (рис. fig. 2)

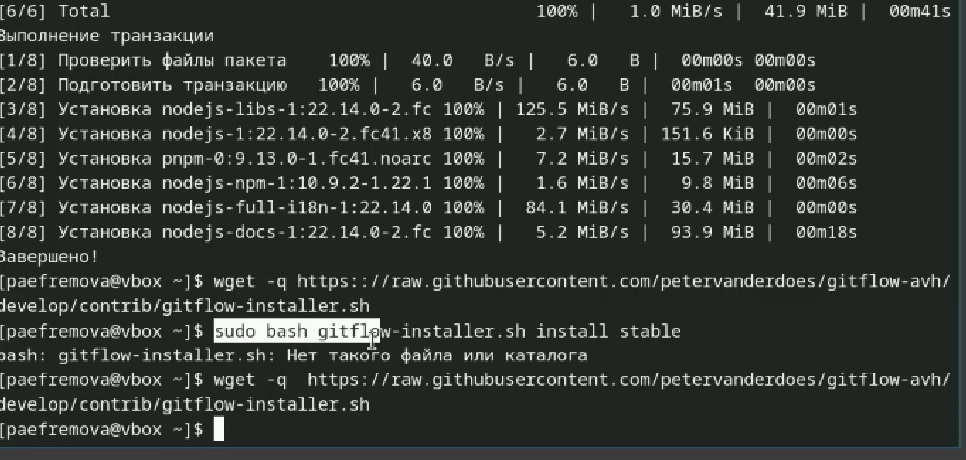


Рис. 2: Установка git-flow

1. Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH. (рис. fig. 3).

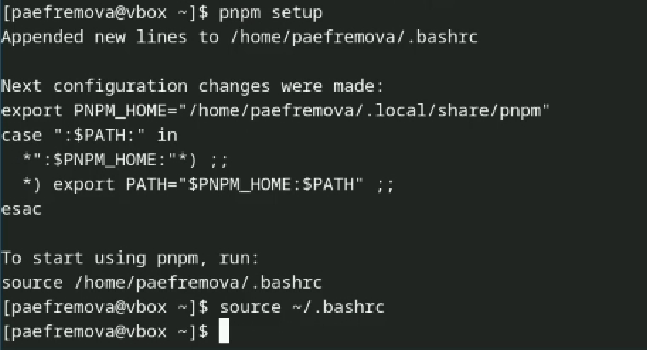


Рис. 3: Настройка Node.js

1. Настраиваем commitizen, standard-changelog. (рис. fig. 4).

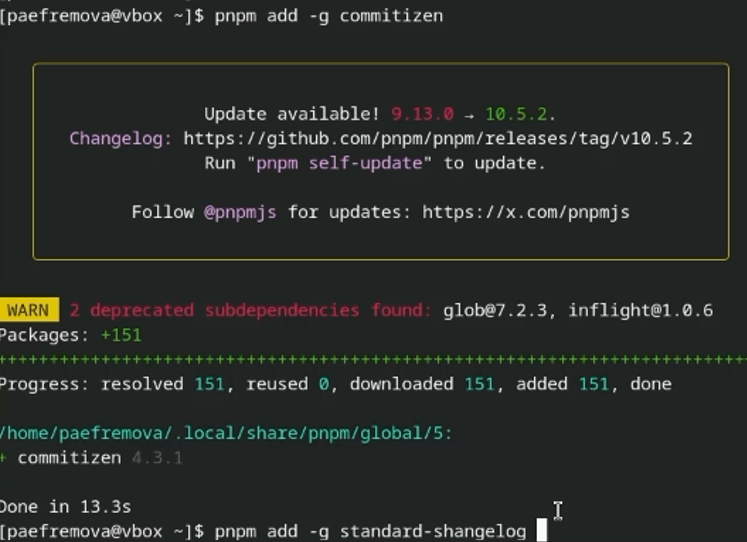


Рис. 4: standard-changelog и commitizen

## 4.2 Практический сценарий использования git. Создание репозитория. Работа с репозиторием.

1. Создаем репозиторий на GitHub. Для примера назовём его git-extended. Делаем первый коммит и выкладываем на github: (рис. fig. 5).

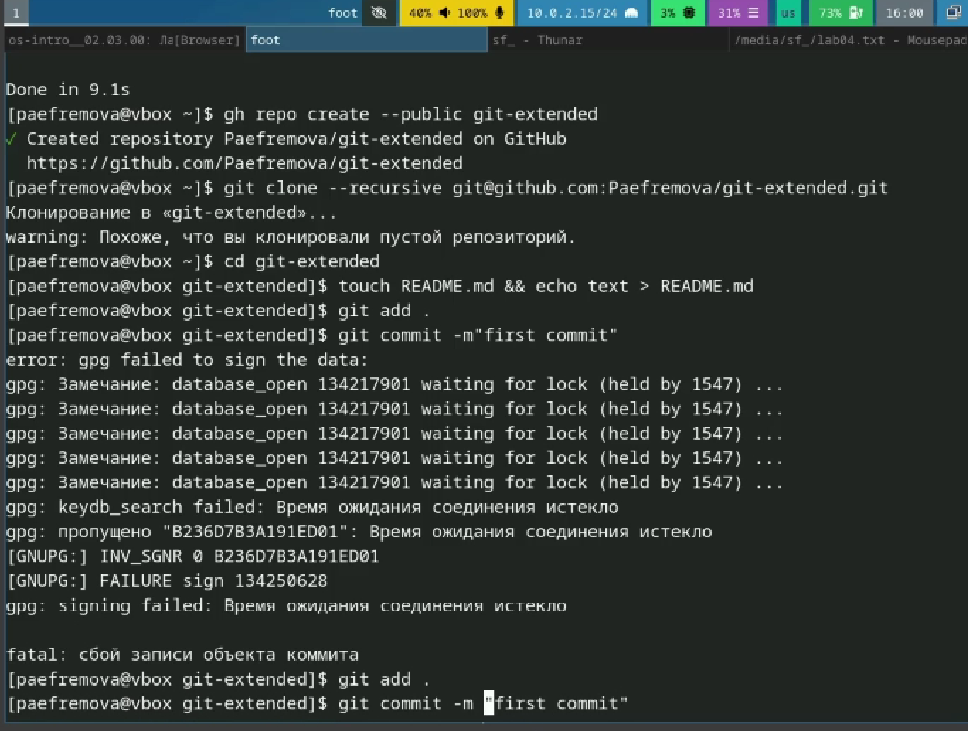


Рис. 5: git-extended

1. Конфигурация общепринятых коммитов. Для этого добавим в файл package.json команду для формирования коммитов: (рис. fig. 6).

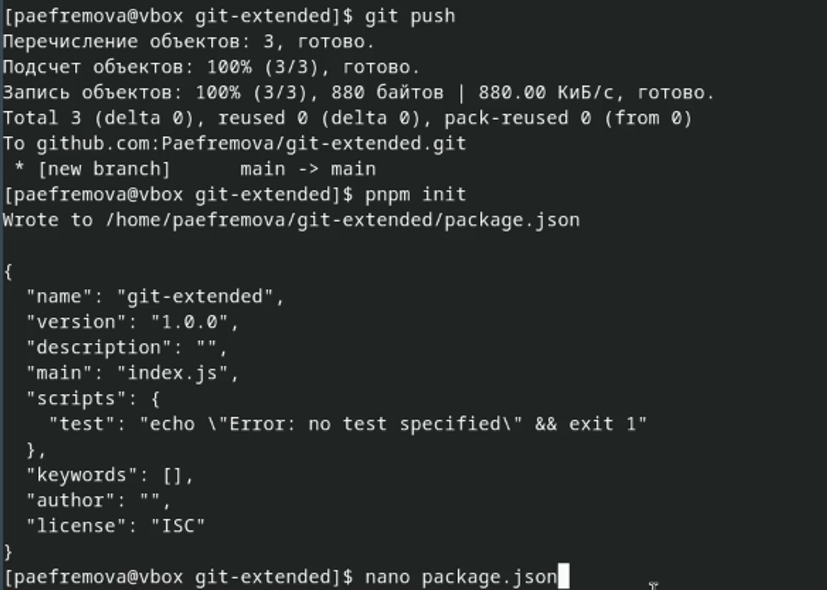


Рис. 6: Конфигурация коммитов

1. Добавляем файлы, выполняем коммит, отправляем на Github. (рис. fig. 7).

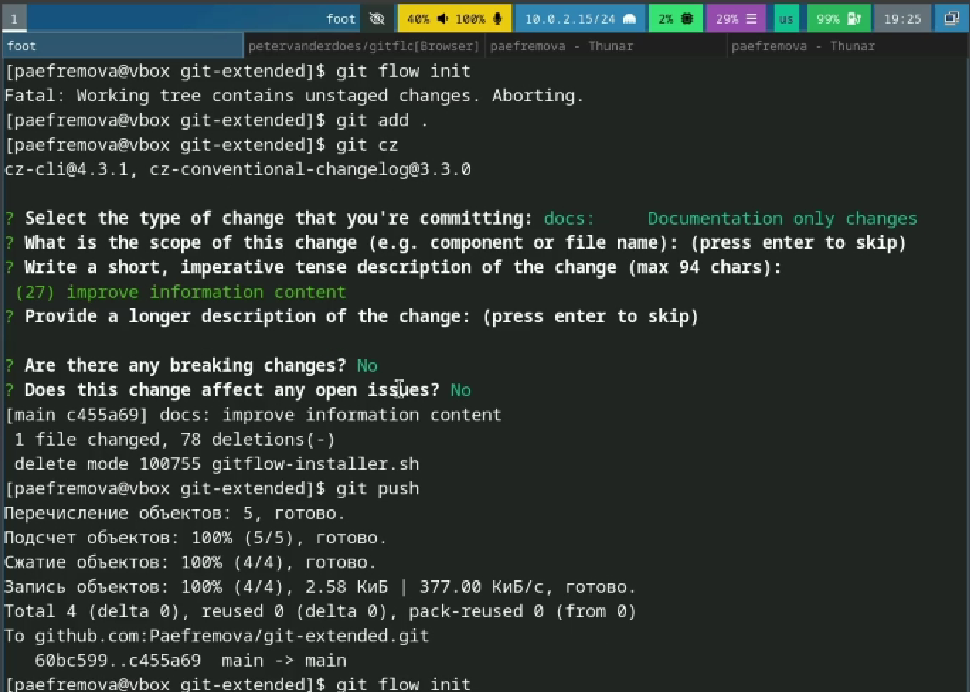


Рис. 7: Отправка изменений на github

1. Инициализируем git-flow Префикс для ярлыков установим в v. Проверьте, что Вы на ветке develop: Загрузите весь репозиторий в хранилище: Установите внешнюю ветку как вышестоящую для этой ветки: (рис. fig. 8).

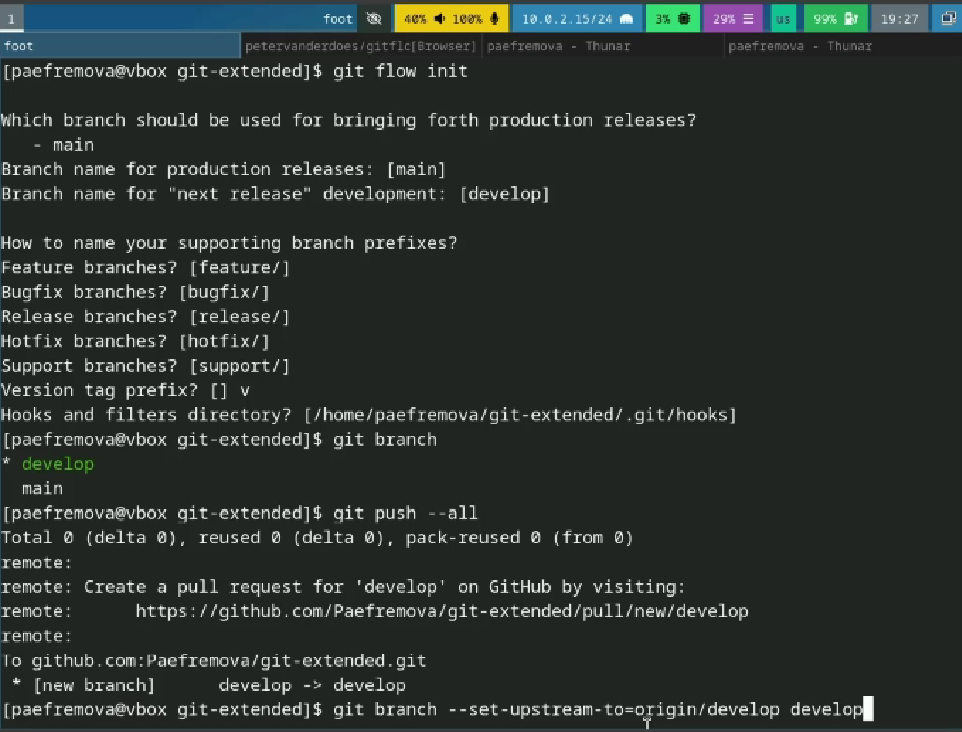


Рис. 8: Часть 1

1. Создадим релиз с версией 1.0.0, Создадим журнал изменений Добавим журнал изменений в индекс Зальём релизную ветку в основную ветку (рис. fig. 9). (рис. fig. 10).

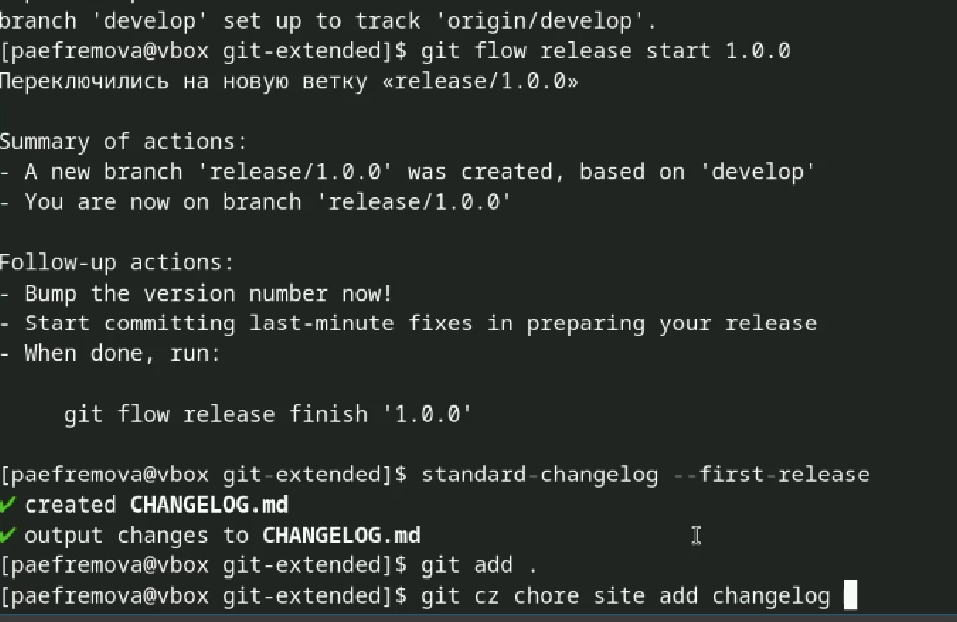


Рис. 9: v 1.0.0

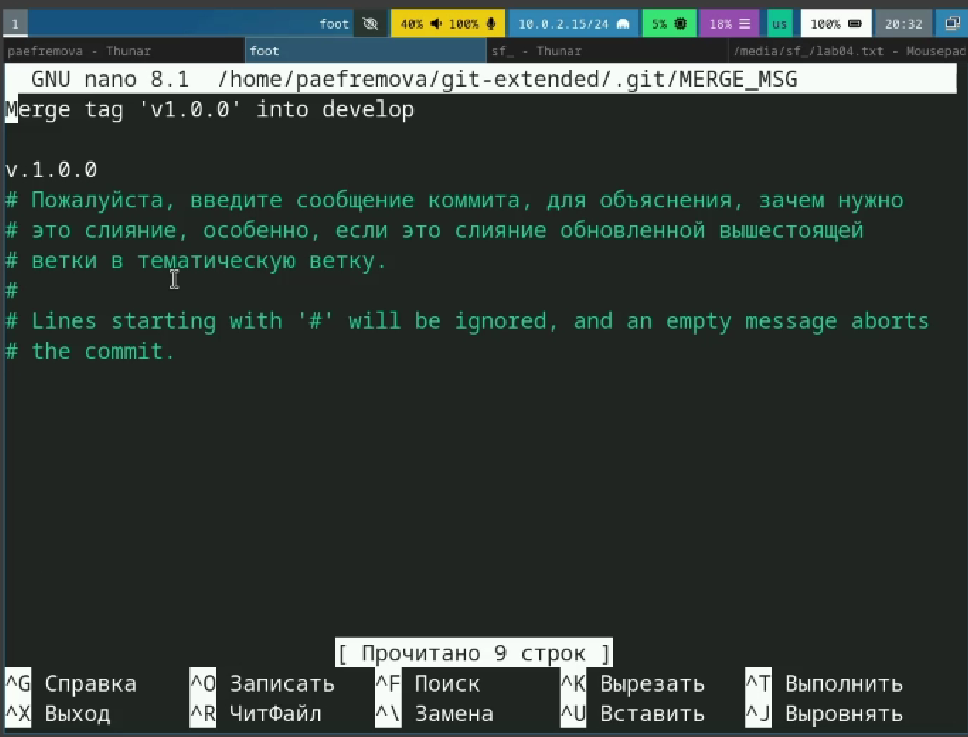


Рис. 10: Залить релизную ветку

1. Отправим данные на github (рис. fig. 11).

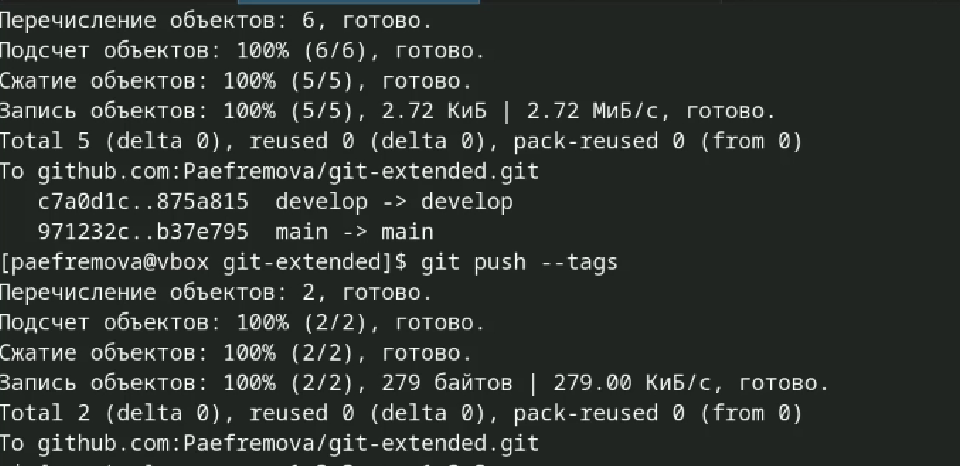


Рис. 11: GitHub

1. Создадим релиз на github, Создадим ветку для новой функциональности. объединить ветку feature\_branch c develop. (рис. fig. 12).

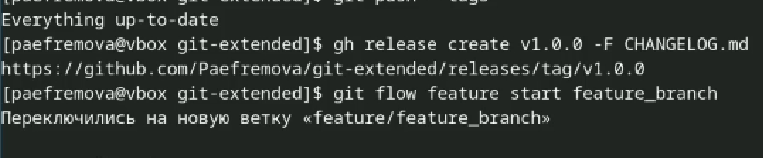


Рис. 12: Создадим релиз на github.

1. Создадим релиз с версией 1.2.3. Обновим номер версии в файле package.json. Установим её в 1.2.3. (рис. fig. 13). (рис. fig. 14).

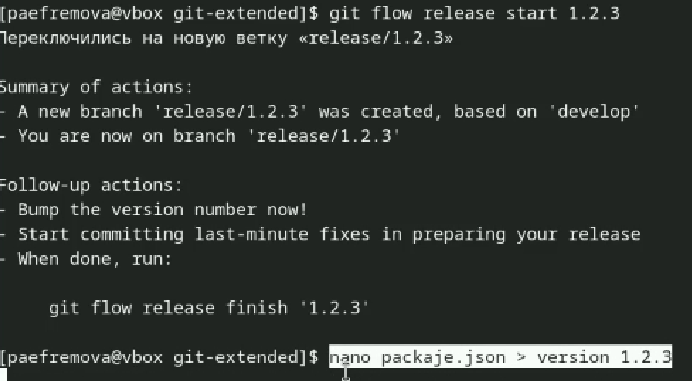


Рис. 13: Релиз v1.2.3.

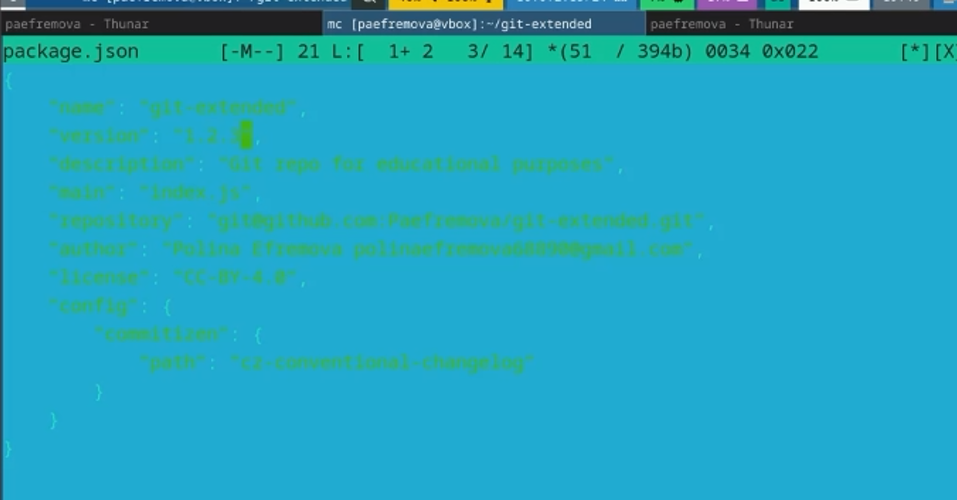


Рис. 14: Изменение package.json

1. Создадим журнал изменений Добавим журнал изменений в индекс, зальем релизную ветку в основную ветку (рис. fig. 15).

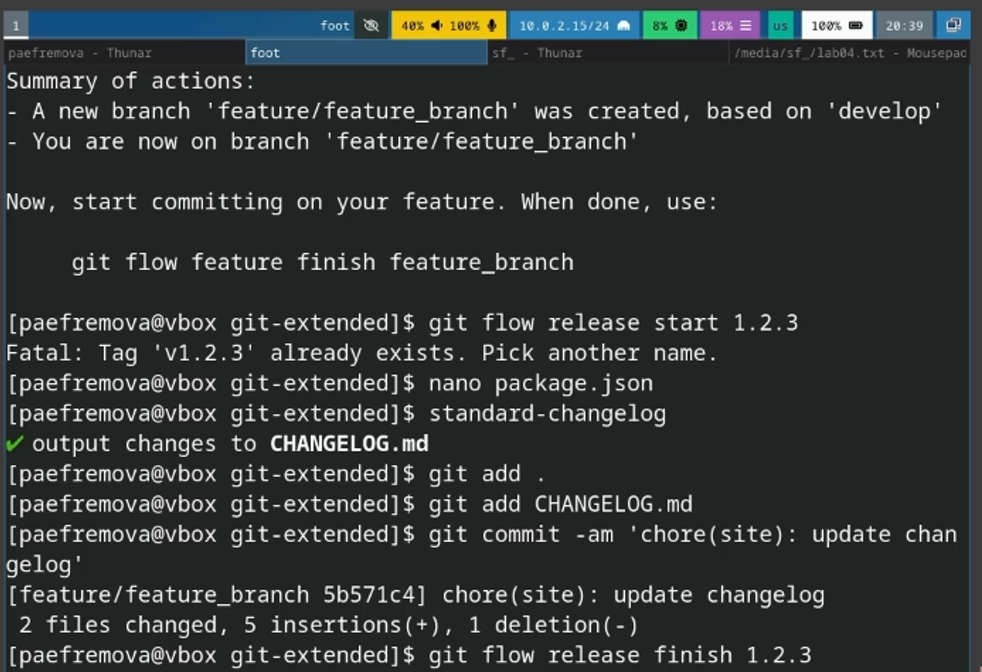


Рис. 15: журнал изменений

14.Отправим данные на github Создадим релиз на github с комментарием из журнала изменений: (рис. fig. 16).

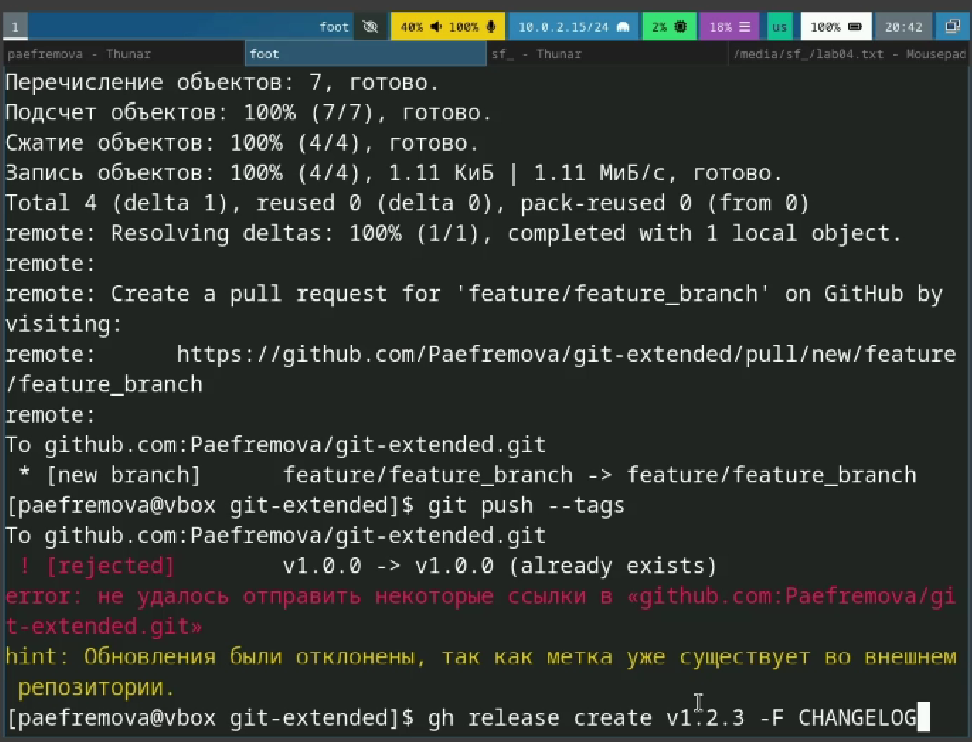


Рис. 16: v 1.2.3 stop

# 5 Выводы

Мы получили навыки правильной работы с репозиториями git, в особенности с git-flow. Теперь я могу загружать данные через release.

# Список литературы

[Рабочий процесс с Gitflow(электронный ресурс)](https://yamadharma.github.io/ru/post/2021/04/18/gitflow-workflow/)