Отчёт по лабораторной работе №4

Операционные системы

Еремина Оксана Андреевна

Содержание

Список таблиц

Список иллюстраций

1 Цель работы

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

2 Задание

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

3 Теоретическое введение

Рабочий процесс Gitflow Workflow. Будем описывать его с использованием пакета git-flow.

- 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. # Выполнение лабораторной работы

1. Установка программного обеспечения

- Установка git-flow

Так как у меня ubuntu, то я устанавливаю ПО с помощью другой команды. (рис.1)

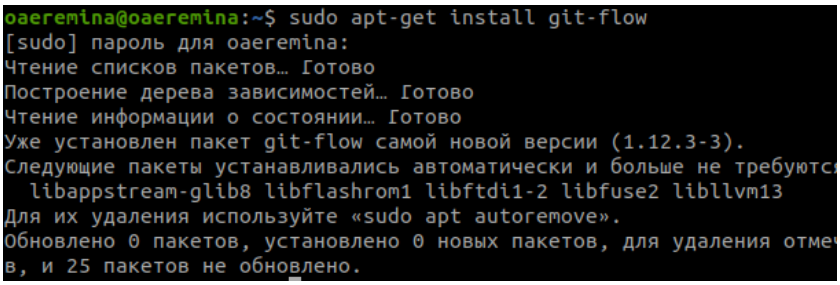

Рис. 3.1: Установка git-flow

- Установка Node.js

То же самое делаю и для node.js (рис.2)

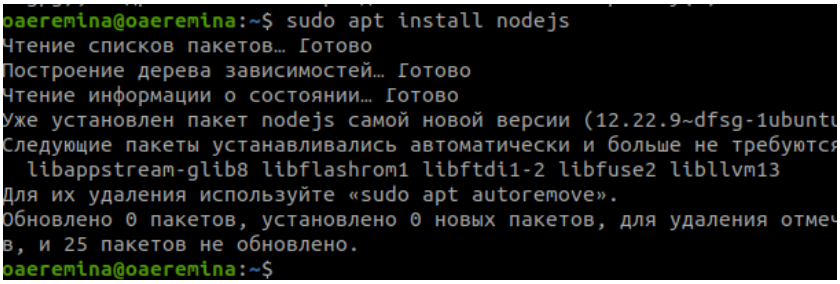

Рис. 3.2: Установка node.js

2. Настройка Node.js

Для работы с Node.js добавляю каталог с исполняемыми файлами, устанавливаемыемыми yarn, в переменную PATH (рис. 3)

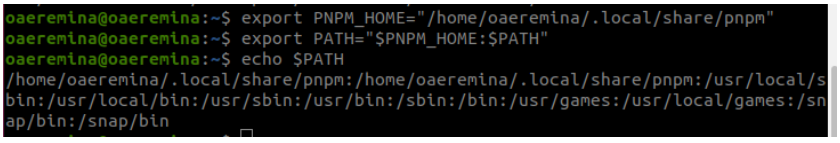

Рис. 3.3: Настройка node.js

3. Общепринятые коммиты

- commitizen

Данная программа используется для помощи в форматировании коммитов. При этом устанавливается скрипт git-cz, который я буду использовать для коммитов. (рис.4)

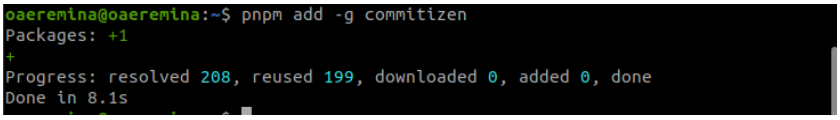

Рис. 3.4: Commitizen

- standard-changelog

Данная программа используется для помощи в создании логов. (рис.5)

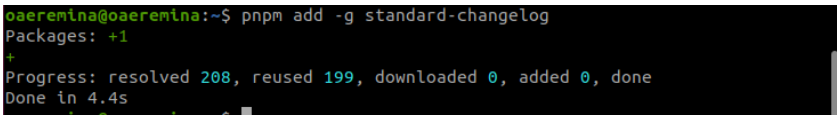

Рис. 3.5: standard-changelog

- Практический сценарий использования git

Создаю репозиторий с именем git-extended. (рис.6)

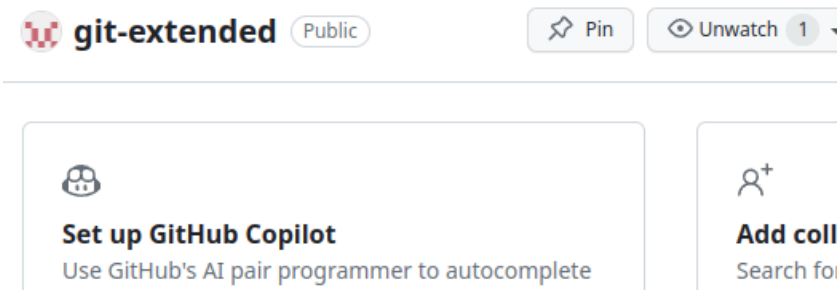

Рис. 3.6: Создание репозитория

Переходу в папку work и клонирую репозиторий в папку (рис.7)

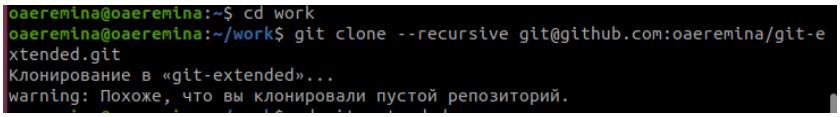

Рис. 3.7: Клонирование репозитория

Делаю первый коммит и выкладываю на github. (рис.8)

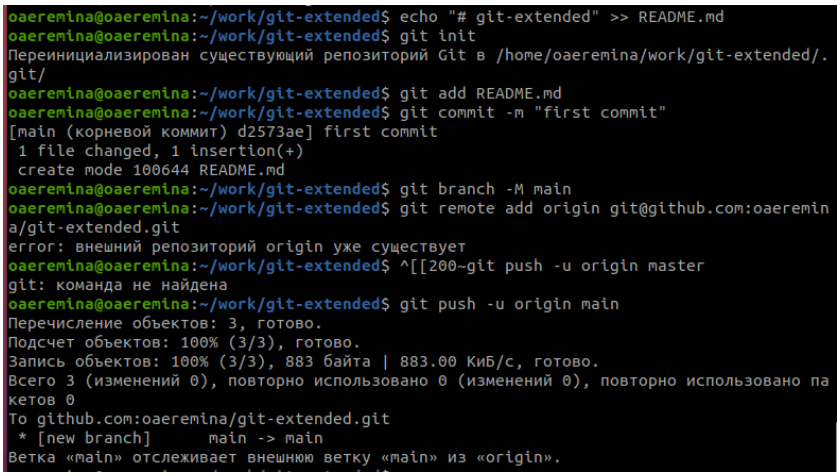

Рис. 3.8: Создание первого коммита

Конфигурация для пакетов Node.js с помощью команды pnpm init (рис.9)

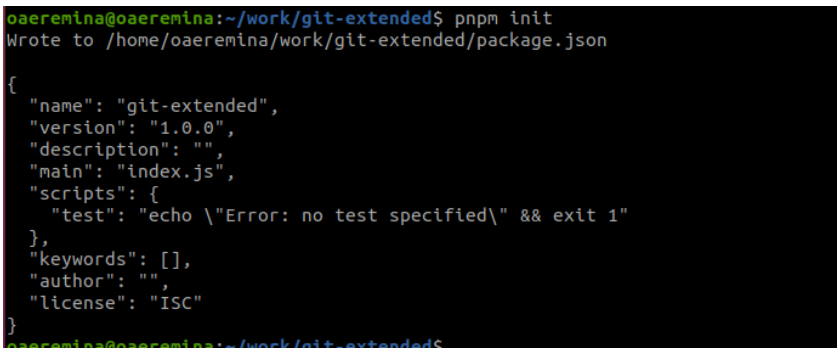

Рис. 3.9: Конфигурация для пакетов Node.js

Заполняю несколько параметров пакеты (рис.10)

Рис. 3.10: Параметры пакета

Добавляю новые файлы, выполняю коммит и отправляю на github (рис.11)

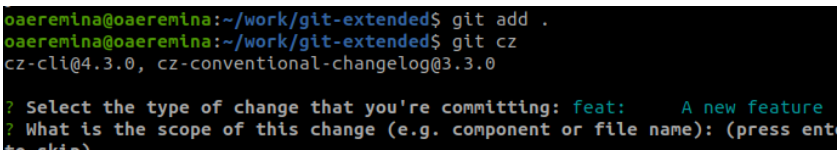

Рис. 3.11: Работа с git

Инициализирую git-flow (рис.12)

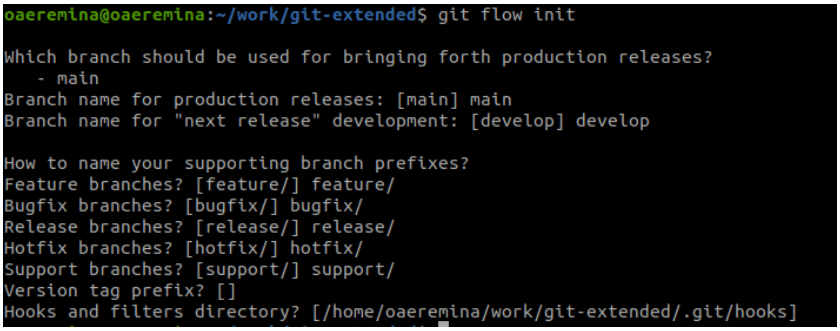

Рис. 3.12: Инициализация git-flow

Проверяю, что нахожусь на ветке develop и загружаю весь репозиторий в хранилище (рис.13)

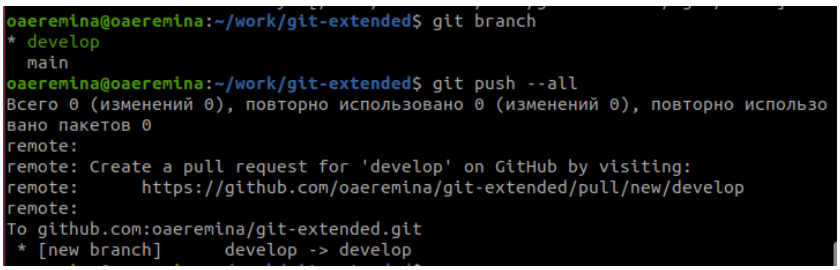

Рис. 3.13: Загружка в хранилище

Установливаю внешнюю ветку как вышестоящую для этой ветки (рис.14)

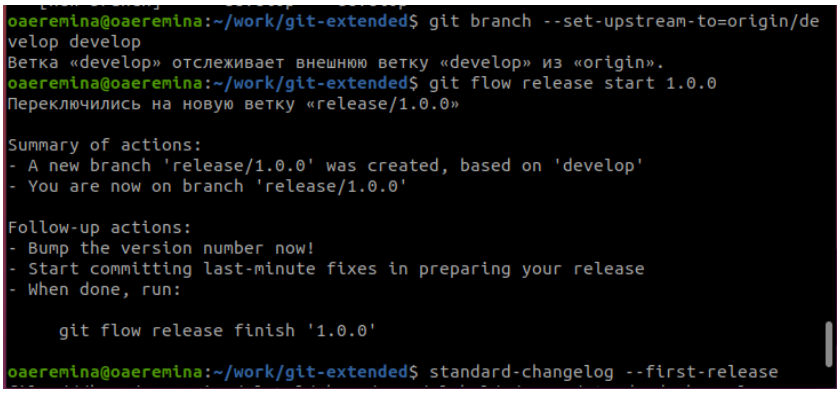

Рис. 3.14: Установка внешней ветки

4 Выводы

При выполнении данной лабораторной работы я получила практические навыки правильной работы с репозиториями github

Список литературы

https://esystem.rudn.ru/mod/page/view.php?id=1098794#org363eddd