Лабораторная работа №4

Продвинутое использование git

Мухин Тимофей Владимирович

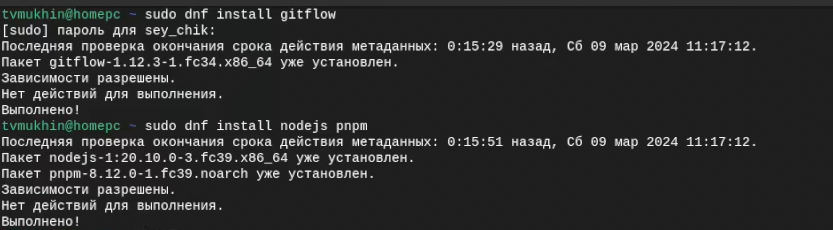
Содержание

# 1 Цель работы

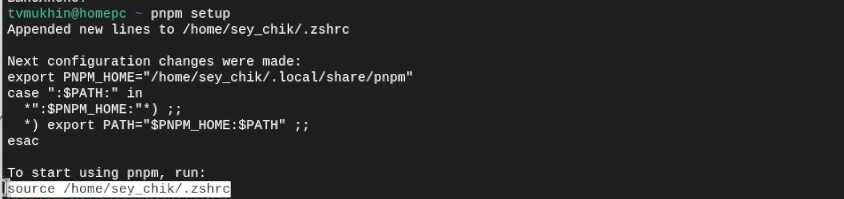
Получение навыков правильной работы с репозиториями git

# 2 Выполнение лабораторной работы

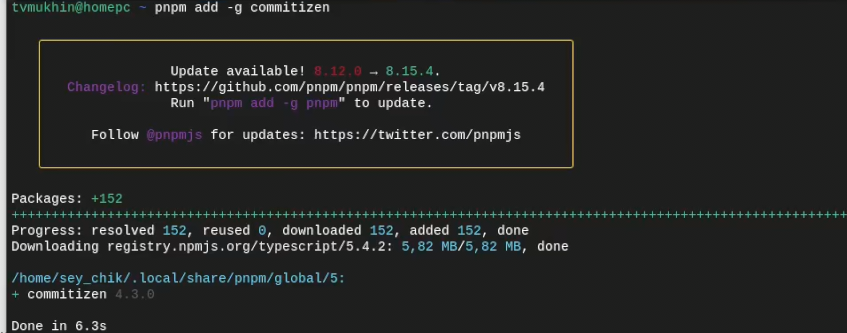
1. Установим gitflow, nodejs и pnpm



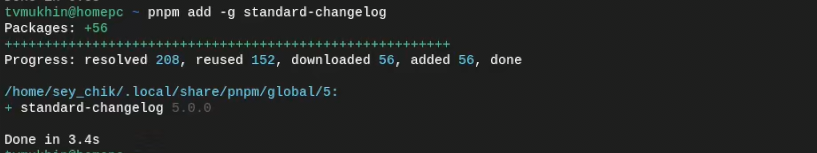
1. Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH.



1. Данная программа(commitizen) используется для помощи в форматировании коммитов



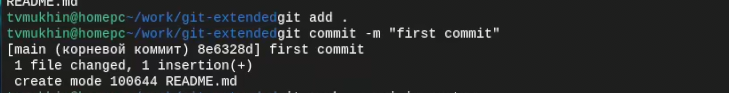
1. Данная программа (standard-changelog) используется для помощи в создании логов



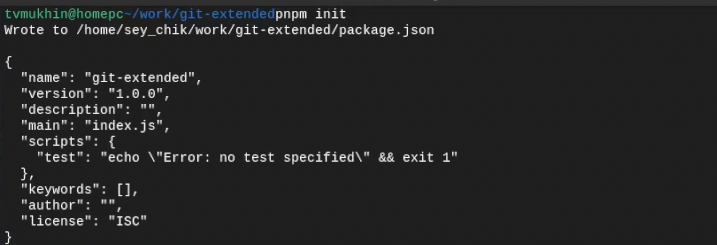
1. Создание репозитория git-extended



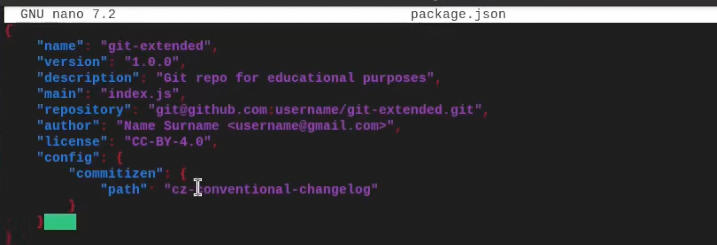
1. Делаем первый коммит и выкладываем на github



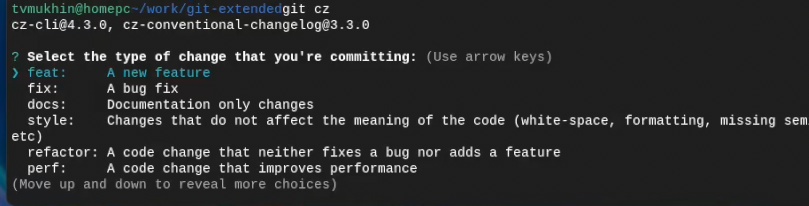
1. Конфигурация общепринятых коммитов



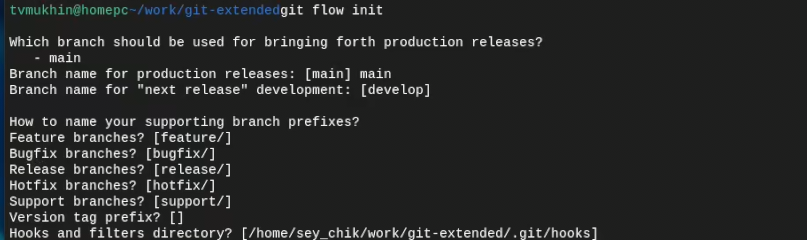
1. Таким образом, файл package.json приобретает вид



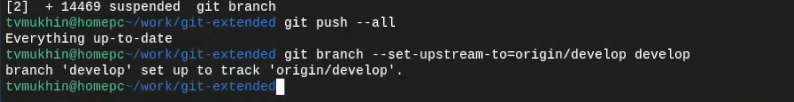
1. Добавим новые файлы и выполним коммит



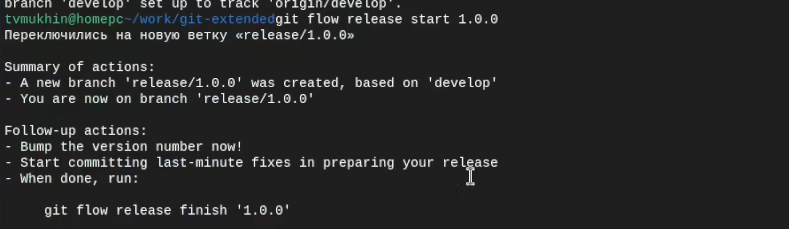
1. Конфигурация git flow. Инициализируем



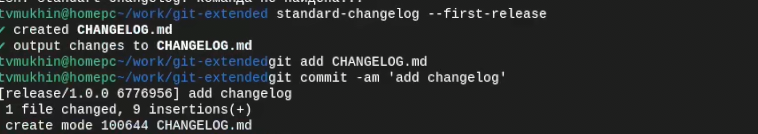
1. Загрузим весь репозиторий в хранилище и установим внешнюю ветку как вышестоящую для этой ветки



1. Создадим релиз с версией 1.0.0



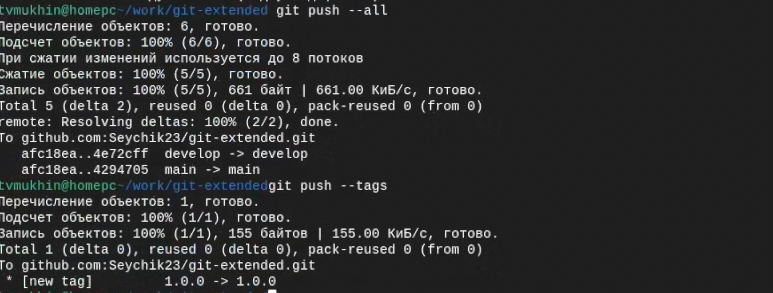
1. Создадим журнал изменений, добавим журнал изменений в индекс



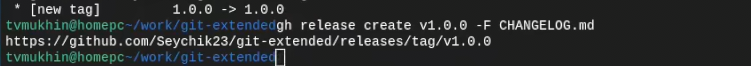
1. Зальём релизную ветку в основную ветку



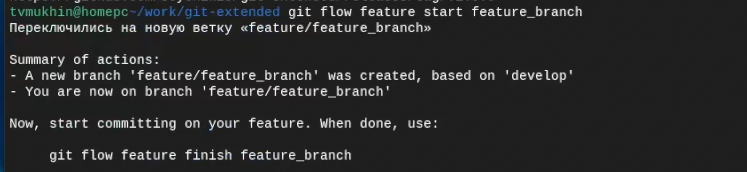
1. Отправим данные на github



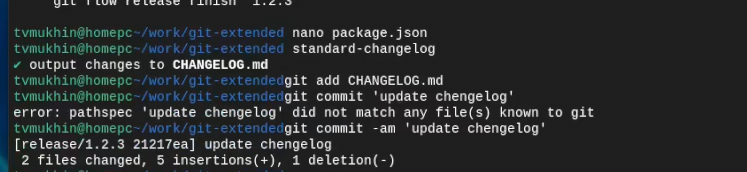
1. Создадим релиз на github. Для этого будем использовать утилиты работы с github



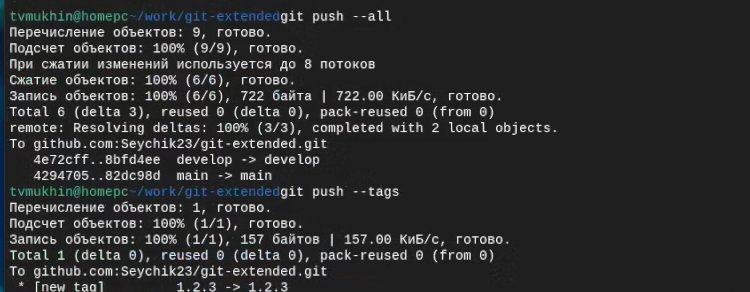
1. Создадим ветку для новой функциональности



1. По окончании разработки новой функциональности следующим шагом следует объединить ветку



1. Обновим номер версии в package.json, создадим журнал изменений, добавим в индекс и отправим данные на сервер



1. Создадим релиз на github с комментарием из журнала изменений



# 3 Выводы

В ходе выполнения работы были изучены новые инструменты git