Отчёт к лабораторной работе №4

Продвинутое использование git

Четвергова Мария Викторовна

Содержание

# 1 Цель работы

Целью 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 convertional commits.

# 3 Выполнение лабораторной работы

## 3.1 Установка программного обеспечения

установим git-flow на ОС Linux. Для этого необходимо установим программное обеспечение из коллекции *Copr*

*dnf install nodejs* *dnf install gitflow*



Рис. 1: Установка пакетов nodejs и gitflow из коллекции Copr

Далее установим Node.js Установим пакет nodejs через командную строки с помощью команд

*dnf install nodejs* *apt-get install pnpm*

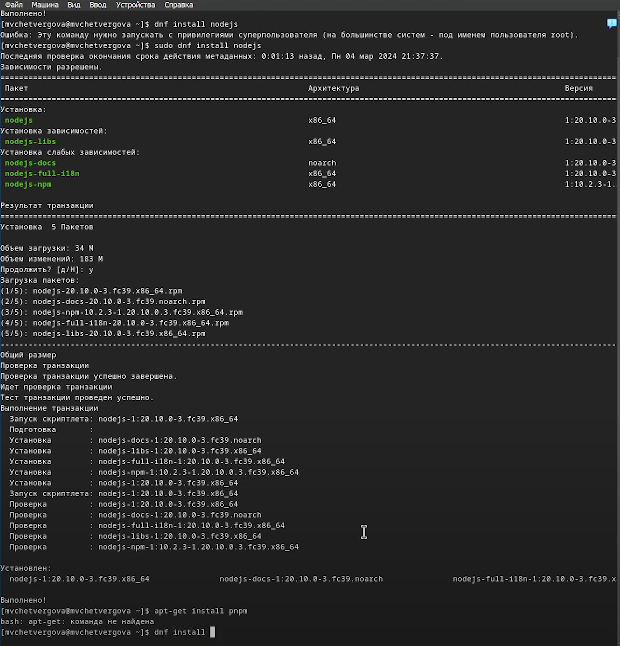


Рис. 2: установим Node.js

Теперь настроим node.js: Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH. Запустим pnpm, перелогинемся и выполним:

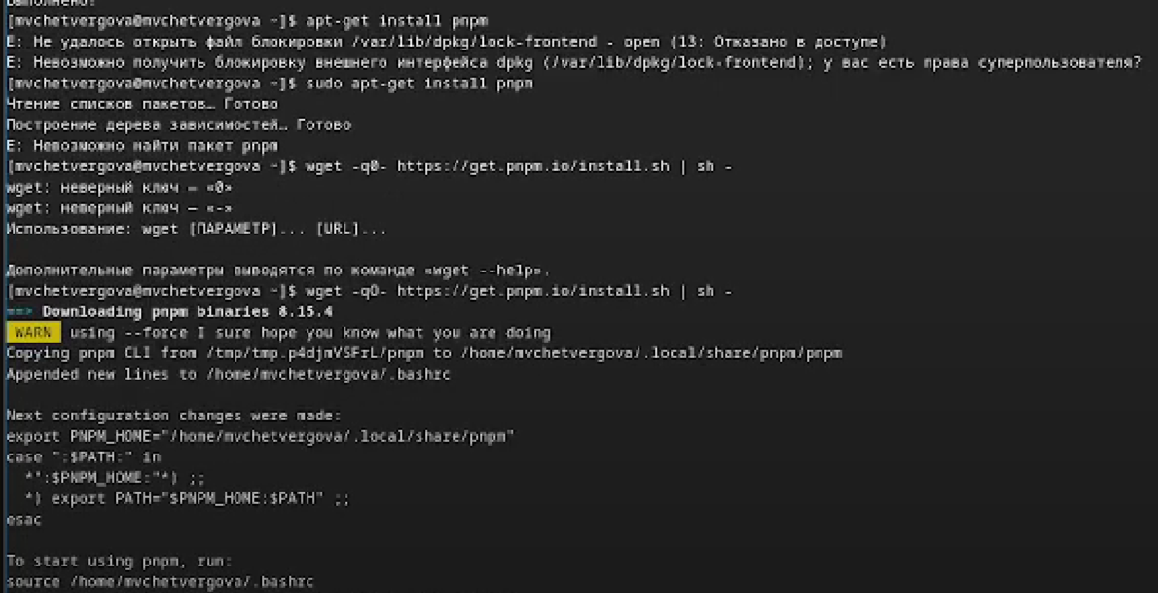


Рис. 3: Добавим каталог с исполняемыми файлами

## 3.2 Общепринятые коммиты.

1. commitizen

* используется для помощи в форматировании коммитов.
* При этом устанавливается скрипт git-cz, который мы и будем использовать для коммитов.

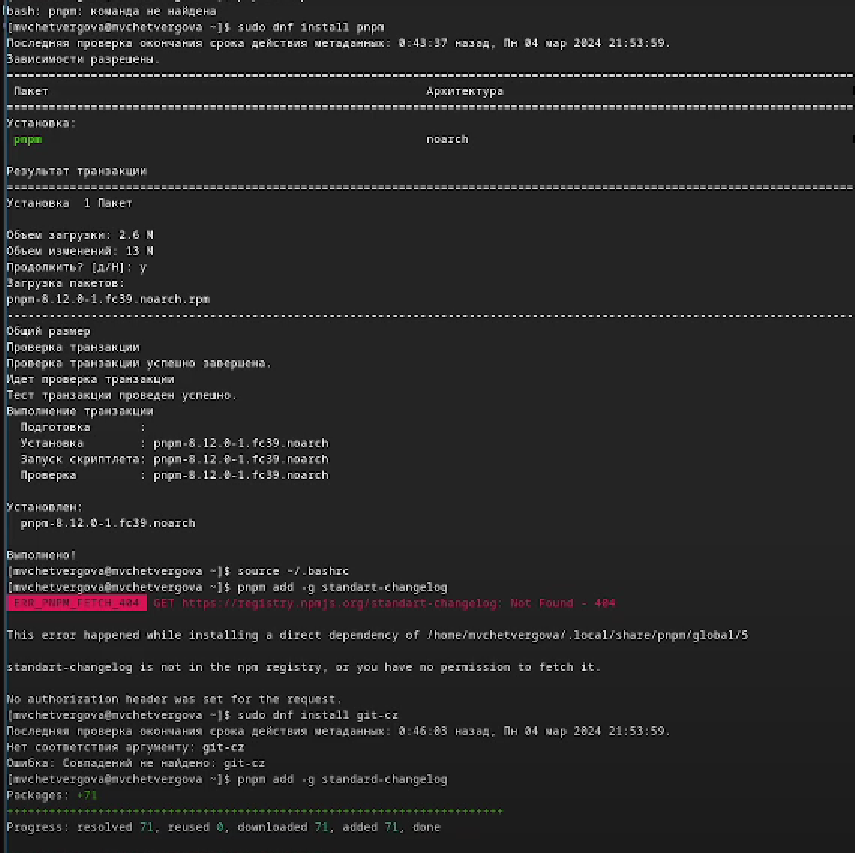


Рис. 4: commitizen

1. standard-changelog

* используется для помощи в создании логов.

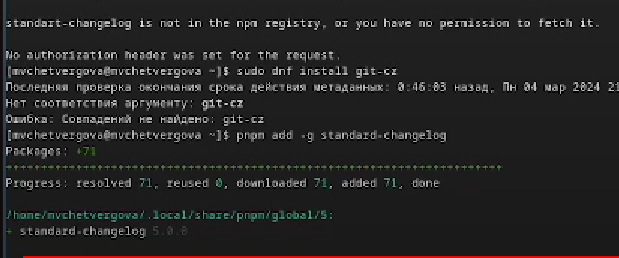


Рис. 5: commitizen

1. Практический сценарий использования git
   1. Создание репозитория git
   2. Подключение репозитория к github -Создать репозиторий на гитхаб
   * Делаем первый коммит

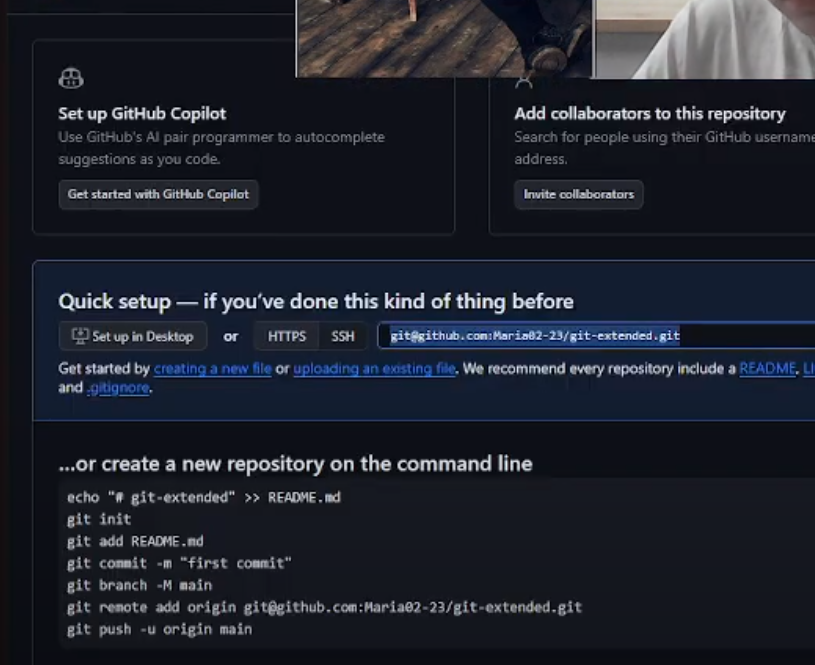


Рис. 6: Подключение репозитория к github

2. Конфигурация обзепринятых коммитов  
- конфигурация для пакетов Node.js  
  
\*pnpm init\*  
  
необходимо заполнить несколько параметров пакета.  
--Название пакета  
--Лицензия пакета  
В конечном счёте файл package.json приобретает следующий вид:

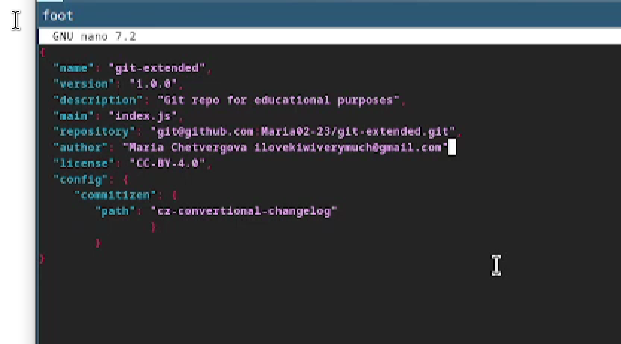
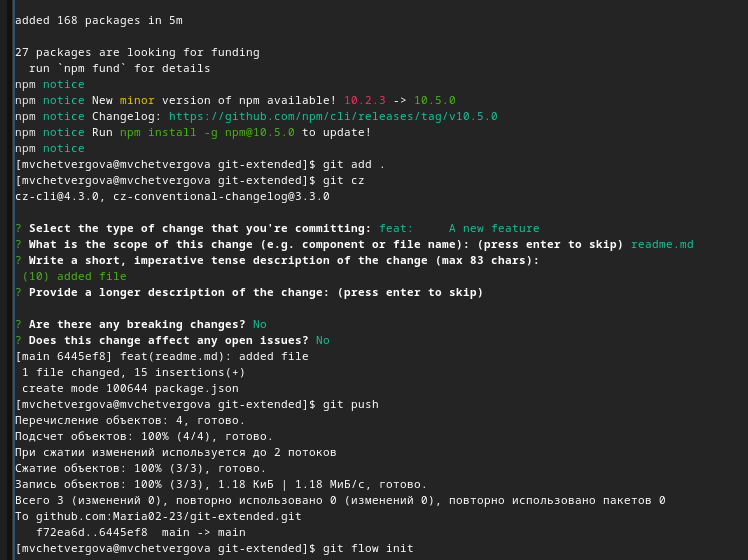


Рис. 7: Конфигурация обзепринятых коммитов

* Добавим новые файлы, выполним коммит и отправим на гитхаб:



1. Конфигурация git-flow

* Инициализируем git-flow через команду

*git flow init*

Префикс для ярлыков установим в v.

* Проверим, что мы на ветке develop
* Загрузим весь репозиторий в хранилище через номанду *git push –all*
* Установим внещнюю ветку как вышестоящую для этой ветки
* Создадим релиз с версией 1.0.0 и создадим журнал изменений
* Зальём релизную ветку в основную ветку, отправим данные на гитхаб и создадим релиз на гитхаб

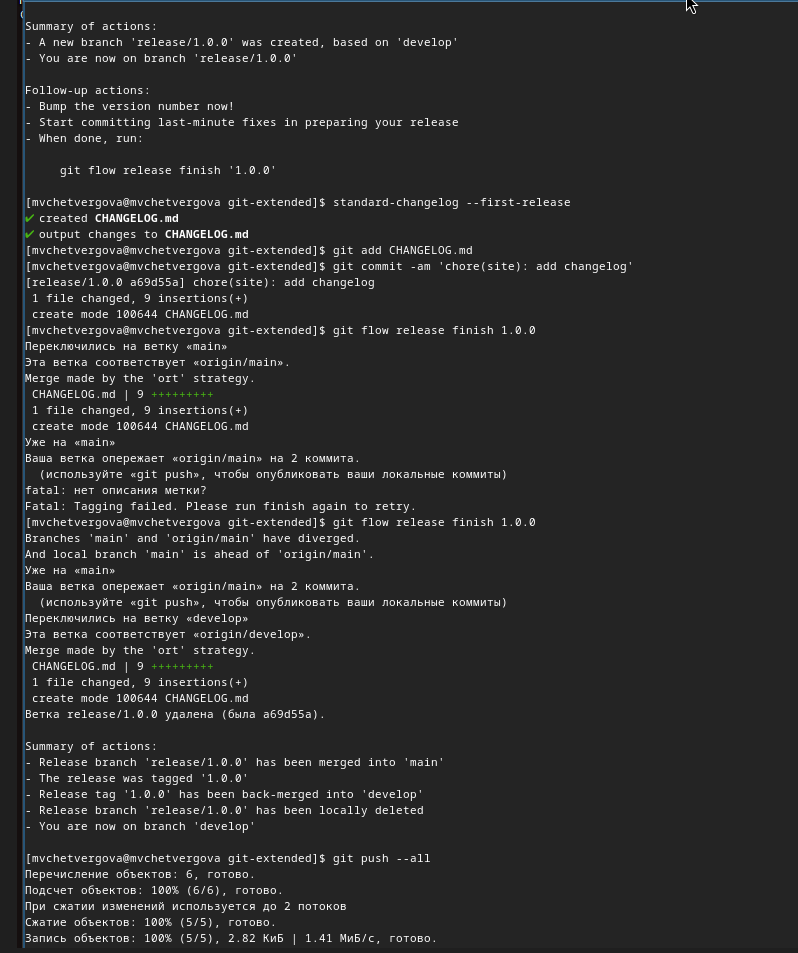


Рис. 8: выполнение вышеприведённых программ

1. Работа с репозиторием git

* создадим релиз с версией 1.2.3 и укажем эту новую версию в package.json

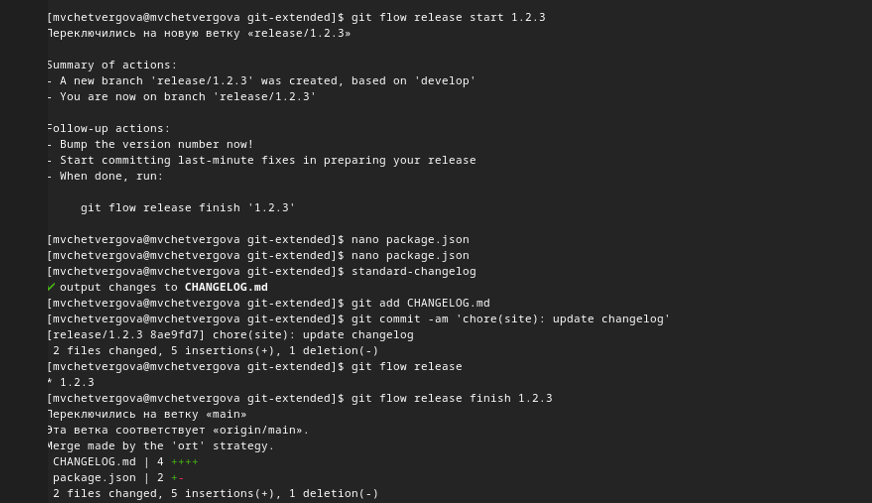


Рис. 9: создание релиза новой версии

* создадим журнал изменений и добавим его в индекс

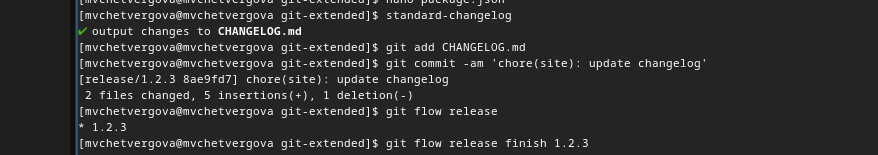


Рис. 10: создание журнала изменений

* Зальём релизную ветку в основную ветку и отправим данные на гитхаб

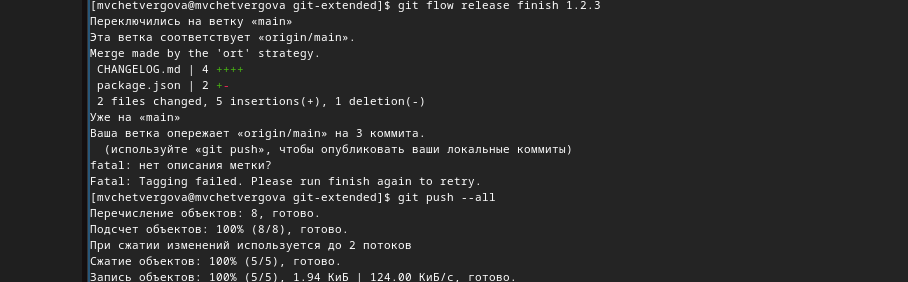


Рис. 11: заливаем релизную ветку в основную ветку

* В конце создадим релиз на github с комментариями из журнала изменений

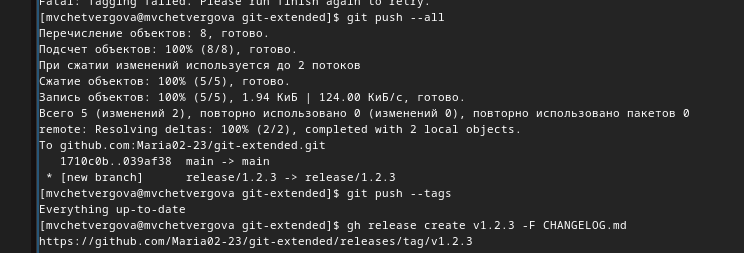


Рис. 12: Создадим релиз на гитхаб с комментариями из журнала

# 4 Выводы

В ходе выполнения лабораторной работы №4 Мы приобрали ценные знания и навыки по продвинутому использованию системы git.

# Список литературы