Лабораторная работа №4

Операционные системы

Машков Илья Евгеньевич

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 Выполнение лабораторной работы

## 3.1 Установка программного обеспечения

### 3.1.1 Установка git-flow

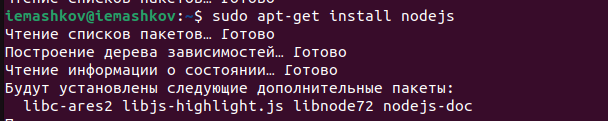
Для начала требуется установить **git-flow** (рис. [??]).

Установка git-flow.

Установка git-flow.

## 3.2 Установка Node.js

Затем устанавливаю **node.js** (рис. [??]).



Установка node.js.

Также понадобится pnpm, с которым возникли проблемы, поэтому я устанавливал его через **wget** (рис. [??]).

Установка pnpm.

Установка pnpm.

## 3.3 Настройка Node.js

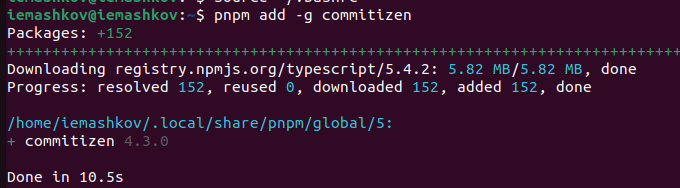
Далее провожу настройку Node.js. Запускаю **pnpm setup**, затем использую команду **source ~/.bashrc**, чтобы перелогиниться (рис. [??]).

Настройка Node.js.

Настройка Node.js.

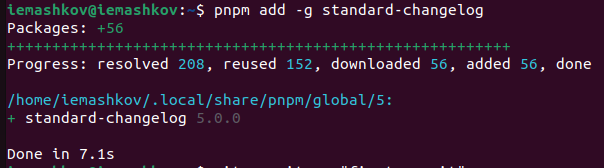
## 3.4 Общепринятые коммиты

1. Добавляю с помощью команды **pnpm add -g** коммит **commitizen**, чтобы эта программа помогала формировать коммиты (рис. [??]).



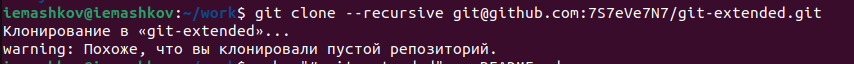
Добавление программы commitizen.

1. Далее, используя ту же команду устанавливаю **standart-changelog**, который будет использоваться для создания логов (рис. [??]).



Добавление программы standart-changelog.

1. Теперь мне нужно создать репозиторий для дальнейшей работы. Для этого мне нужно клонировать этот репозиторий (рис. [??]).



Клонирование репозитория.

Затем создать файл, чтобы наш репозиторий не был пуст (рис. [??]).

Создание файла.

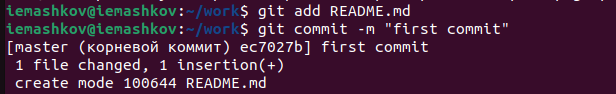
Создание файла.

Инициализирую репозиторий (рис. [??]).

Инициализация репозитория на устройстве.

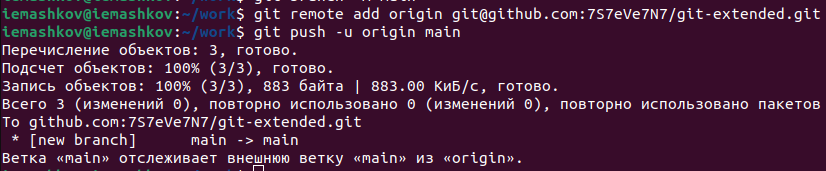
Инициализация репозитория на устройстве.

Теперь добавляю этот файл в список тех, которые я отправлю на сервер (рис. [??]).



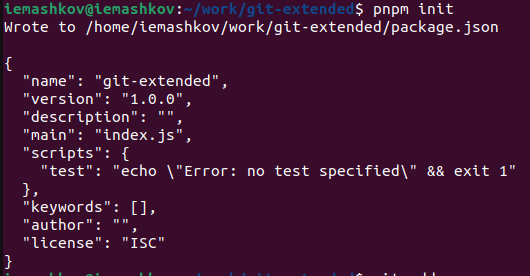
Подготовка к отправлению первого коммита.

И, наконец, отправляю мой первый коммит на сервер (рис. [??]).



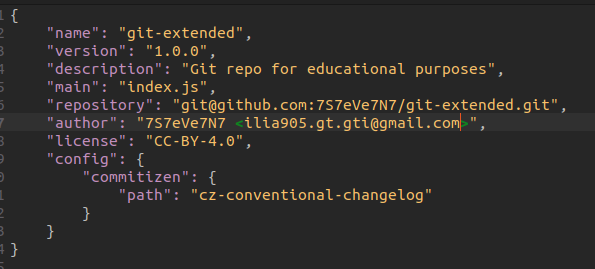
Загрузка первого коммита на сервер.

1. Дальше необходимо конфигурировать общепринятые коммиты. Для этого мне надо прописать команду **pnpm init**, чтобы создать конфигурационный файл (рис. [??]).



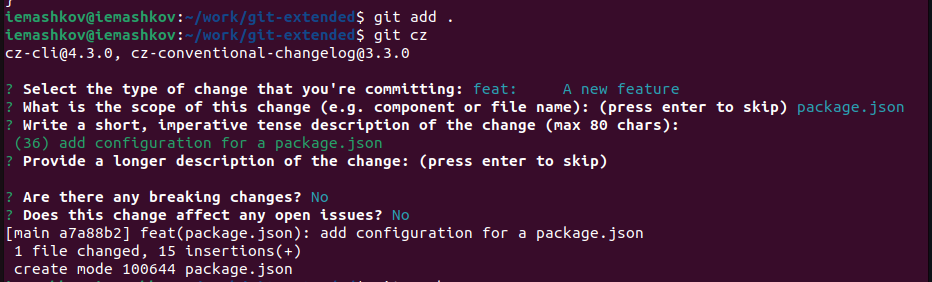
Создание конфигурационного файла.

Конфигурирую формат коммитов в файле (рис. [??]).



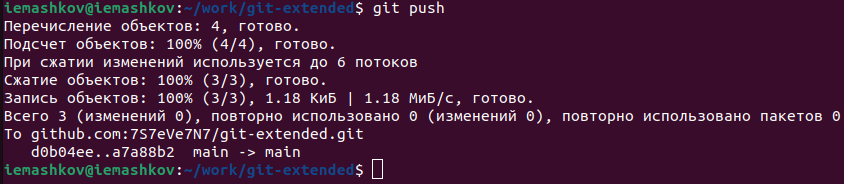
Конфигурация файла.

Добавляю новый файл и выполняю коммит (рис. [??]).



Добавление файла и выполнение коммита.

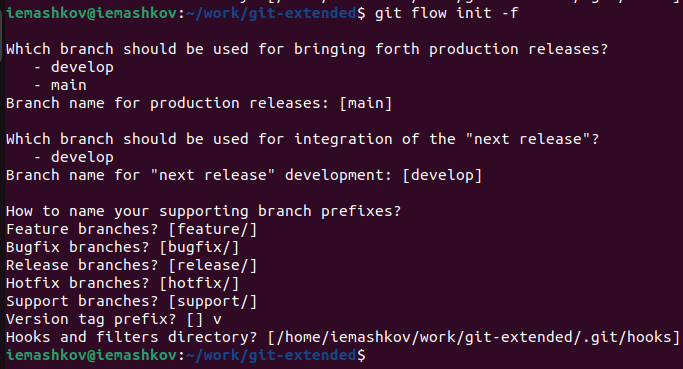
Затем отправляю это на сервер (рис. [??]).



Загрузка файла на сервер.

## 3.5 Конфигурация git flow

Инициализирую **git flow**, указывая префиксом для ярлыков значение **v** (рис. [??]).



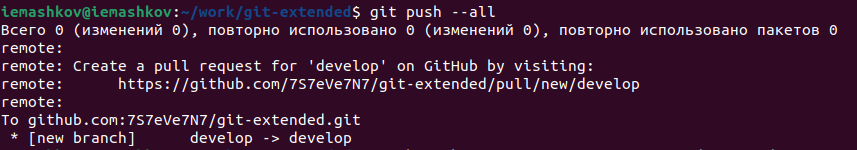
Инициализация git flow.

Проверяю, что нахожусь в ветке **develop** (рис. [??]).

Проверка.

Проверка.

Загружаю весь репозиторий в хранилище командой **git push –all** (рис. [??]).



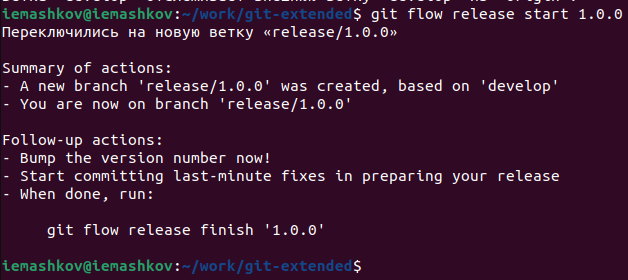
Загрузка всего репозитория в хранилище.

Теперь устанавливаю внешнюю ветку вышестоящей для этой (рис. [??]).

Установка вышестоящей метки.

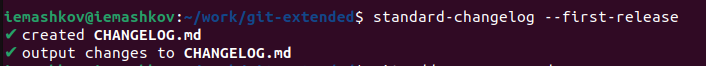
Установка вышестоящей метки.

Создаю релиз с версией **1.0.0** (рис. [??]).



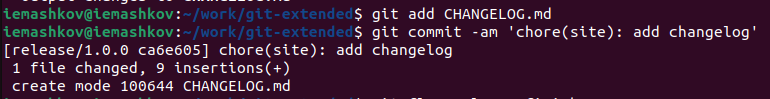
Создание релиза с версией 1.0.0.

Создаю журнал изменений (рис. [??]).



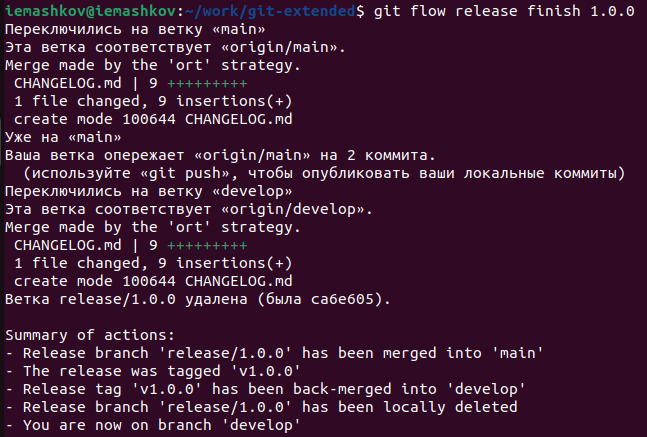
Создание журнала изменений.

Добавляю этот журнал в индекс (рис. [??]).



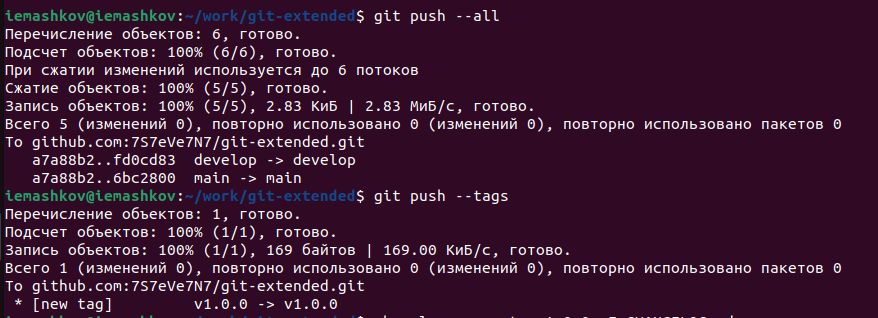
Добавление журнала изменений в индекс.

Заливаю релизную ветку в основную (рис. [??]).



Добавление релизной ветки в основную.

Затем отправляю данные на сервер (рис. [??]).



Загрузка данных на сервер.

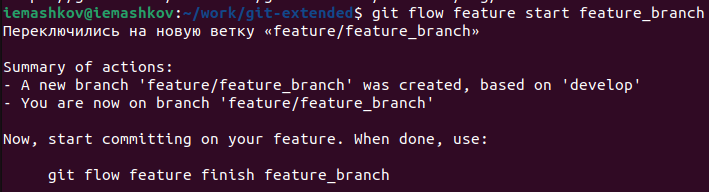
Создаю релиз на github, используя установленные утилиты (рис. [??]).

Создание релиза на github.

Создание релиза на github.

## 3.6 Работа с репозиторием git

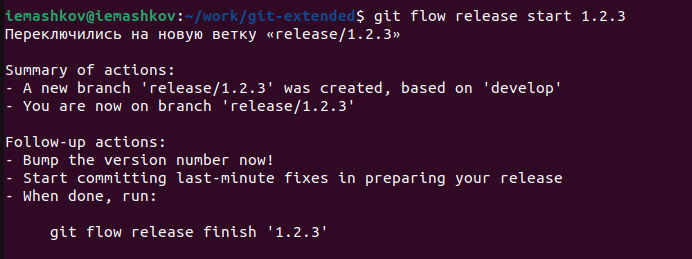
1. Создаю ветку для новой функциональности (рис. [??]).



Создание ветки для новой функциональности.

Далее объединяю эту ветку с ветками **feature\_branch** и **develop** с помощью команды **git flow feature finish feature\_branch**

1. Создаю релиз с версией **1.2.3** (рис. [??]).



Создание релиза с новой версией.

Обновляю номер версии в файле **package.json** (рис. [??]).

Обновление данных в файле package.json.

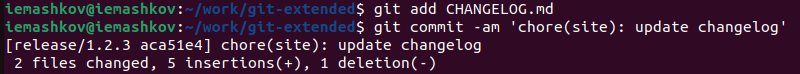
Обновление данных в файле package.json.

Создаю журнал изменений (рис. [??]).

Создание журнала изменений.

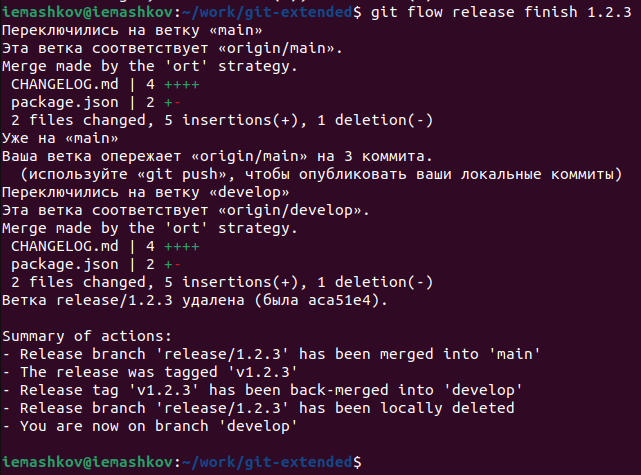
Создание журнала изменений.

И добавляю его в индекс (рис. [??]).



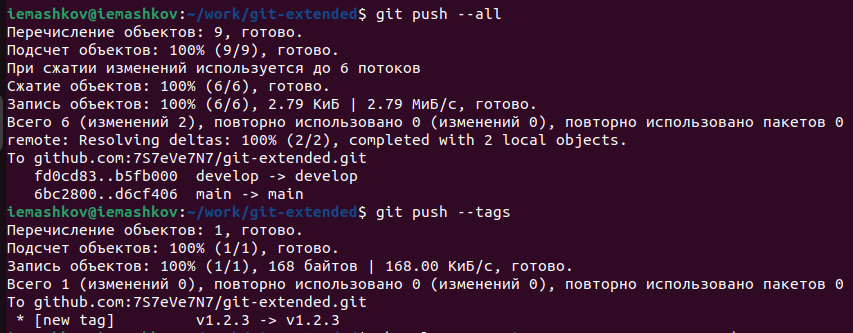
Добавление журнала изменений в индекс.

Заливаю релизную ветку в основную (рис. [??]).



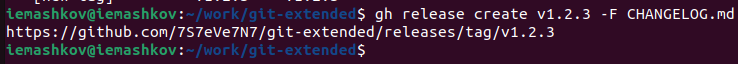
Добавление релизной ветки в основную.

Отправляю данные на github (рис. [??]).



Отправка данных на сервер.

Создаю релиз на github с комментарием из журнала изменений (рис. [??]).



Создание релиза.

# 4 Выводы

При выполнении лабораторной работы я получил навыки для правильной работы с репозиториями git.

# Список литературы

[Операционные системы](https://esystem.rudn.ru/mod/page/view.php?id=1098794&forceview=1)