Отчёт по лабораторной работе №4

Операционные системы

Ирина Васильевна Панявкина

Содержание

# 1 Цель работы

Цель данной лабораторной 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.

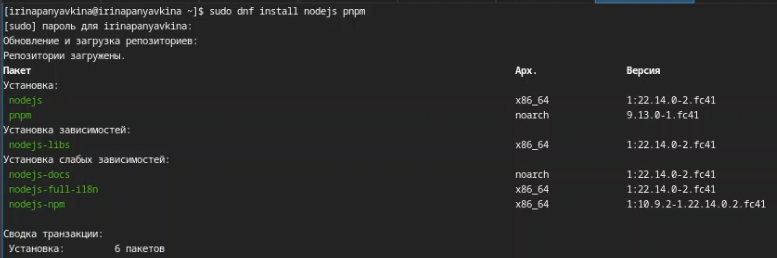
# 3 Теоретическое введение

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

# 4 Выполнение лабораторной работы

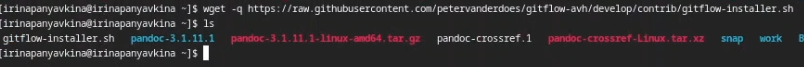
## 4.1 Установка программного обеспечения

На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов, поэтому первым делом устанавливаю nodejs и pnpm (рис. [??]).



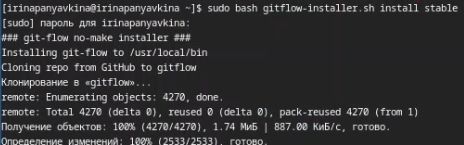
Установка nodejs и pnpm

К сожалению, скачать gitflow по инструкции мне не удалось, поэтому делала я это иным способом. Сначала скачиваю установщик gitflow - gitflow-installer, с помощью команды ls проверяю успешна ли загрузка (рис. [??]).



Скачивание установщика

Теперь устанавливаю gitflow через установщик (рис. [??]).



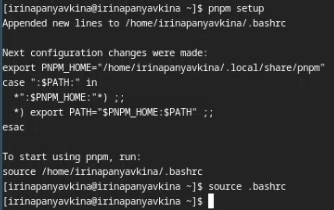
Установка gitflow через установщик

Установщик больше не потребуется, удаляю его с помощью команды rm (рис. [??]).

Установка gitflow

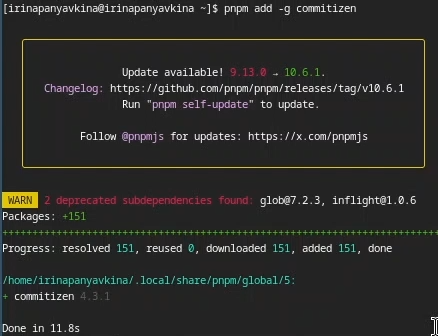
Установка gitflow

Затем настраиваю nodejs. Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH (рис. [??]).



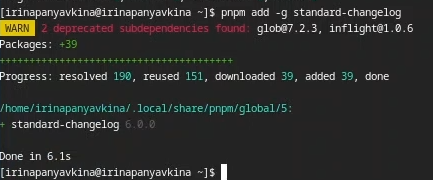
Настройка nodejs

Настраиваю общепринятые коммиты. Добавляю commitizen - программа используется для помощи в форматировании коммитов. (рис. [??]).



Добавление commitizen

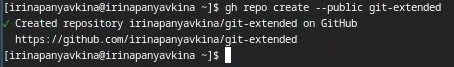
Также добавляю standard-changelog - программа используется для помощи в создании логов (рис. [??]).



Добавление standard-changelog

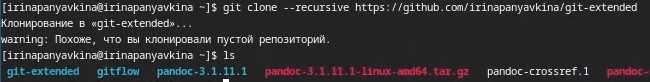
## 4.2 Практический сценарий использования git

Создаю публичный репозиторий на GitHub с помощью команды gh repo create –public и называю его git-extended (рис. [??]).



Создание репозитория git-extended на GitHub

Теперь клонирую его к себе на виртуальную машину (рис. [??]).



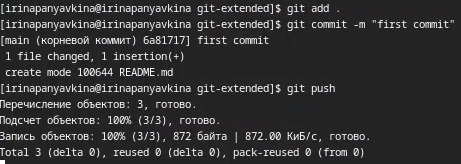
Клонирование репозитория на виртуальную машину

Перемещаюсь в git-extended/ и создаю файл README.md, пишу в нем “text”. Мне это необходимо для создания первого коммита (рис. [??]).

Создание файла README.md

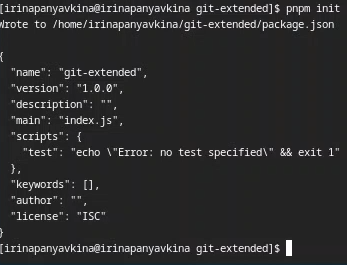
Создание файла README.md

Добавляю изменения через git add ., затем делаю первый коммит с помощью git commit -m “first commit” и выкладываю на github через git push (рис. [??]).



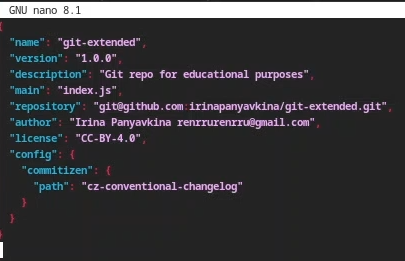
Создание первого коммита

Приступим к конфигурации общепринятых коммитов. Осуществляю конфигурацию для пакетов Node.js (рис. [??]).



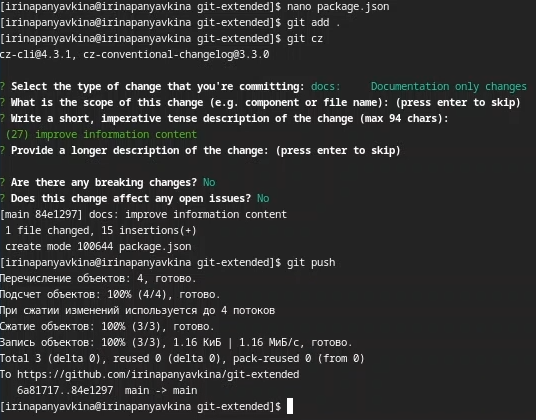
Конфигурация для пакетов Node.js

Редактирую файл package.json. Заполняю несколько параметров пакета (рис. [??]).



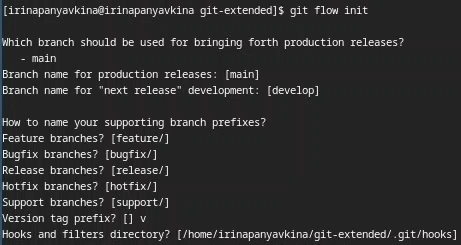
Редактирование файла package.json

Теперь добавляю новые файлы через git add ., выполняю коммит по новой команде git cz, отправляю на github через git push (рис. [??]).



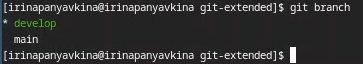
Выполнение коммита по команде git cz

Осуществим конфигурацию git-flow. Инициализирую git-flow с помощью команды git flow init. Префикс для ярлыков установливаю в v. (рис. [??]).



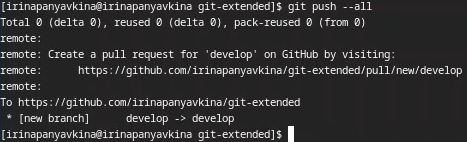
Конфигурация git-flow

Проверяю, что я нахожусь на ветке develop: git branch (рис. [??]).



Проверка ветки

Загружаю весь репозиторий в хранилище через команду push –all (рис. [??]).



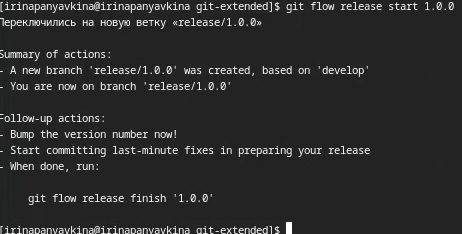
Загрузка всего репозитория в хранилище

Устанавливаю внешнюю ветку как вышестоящую для этой ветки командой git branch –set-upstream-to=origin/develop develop (рис. [??]).

Установка внешней ветки как вышестоящей

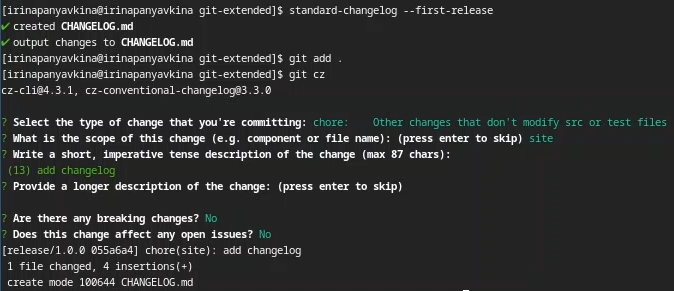
Установка внешней ветки как вышестоящей

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



Создание релиза с версией 1.0.0

Создаю журнал изменений и добавляю журнал изменений в индекс, в вопросах программы указываю ‘chore(site): add changelog’ (рис. [??]).



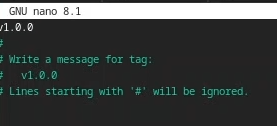
Создание журнала изменений и добавление его в индекс

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

Добавление релизной ветки в основную ветку

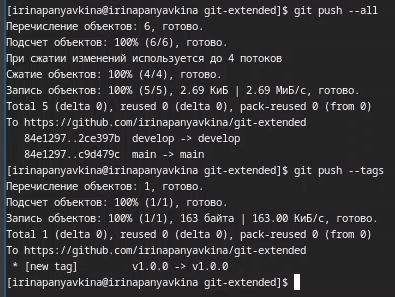
Добавление релизной ветки в основную ветку

Указываю версию релиза 1.0.0 в открывшемся файле (рис. [??]).



Указание версии релиза в файле

Отправляю данные на github, используя две команды git push –all и git push –tags (рис. [??]).



Отправка данных на github

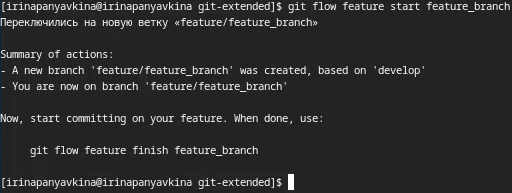
Создаю релиз на github. Для этого использую утилиты работы с github. (рис. [??]).

Создание релиза на github

Создание релиза на github

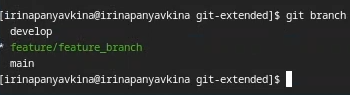
## 4.3 Работа с репозиторием git

Перехожу к разработке новой функциональности. Создаю ветку для новой функциональности (рис. [??]).



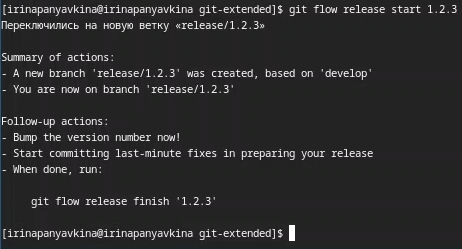
Создание ветки для новой функциональности

Проверяю, что я нахожусь на ветке feture/feature\_branch (рис. [??]).



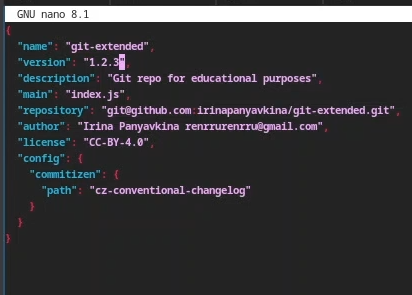
Проверка ветки

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



Создание релиза с версией 1.2.3

Редактирую файл package.json. Обновляю версию релиза на 1.2.3. (рис. [??]).



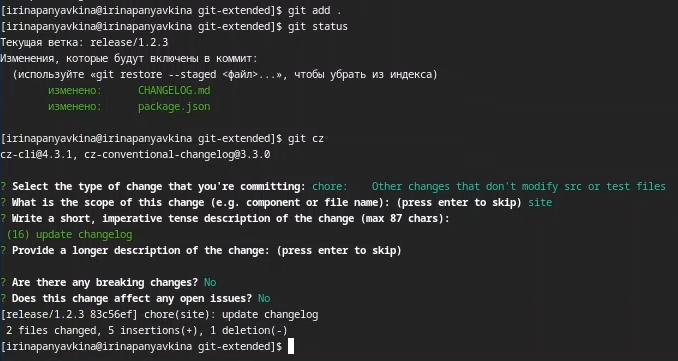
Редактирование файла package.json

Создаю журнал изменений (рис. [??]).

Создание журнала изменений

Создание журнала изменений

Теперь добавляю новые файлы через git add ., с помощью git status проверяю изменения, выполняю коммит команде git cz, добавляю журнал изменений в индекс, в вопросах программы указываю ‘chore(site): update changelog’ (рис. [??]).



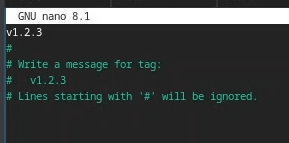
Выполнение коммита и добавление журнала изменений в индекс

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

Добавление релизной ветки в основную ветку

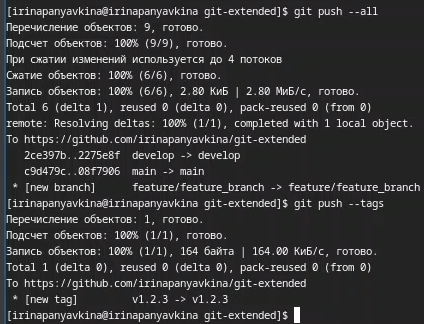
Добавление релизной ветки в основную ветку

Указываю версию релиза 1.2.3 в открывшемся файле (рис. [??]).



Указание версии релиза в файле

Отправляю данные на github, используя две команды git push –all и git push –tags (рис. [??]).



Отправка данных на github

Создаю релиз на github. Для этого использую утилиты работы с github. (рис. [??]).

Создание релиза на github

Создание релиза на github

# 5 Выводы

Во время выполнения лабораторной работы, я получила навыки правильной работы с репозиториями git.

# Список литературы

1. Лабораторная работа №4 [Электронный ресурс] URL: https://esystem.rudn.ru/mod/page/view.php?id=1224375