Отчёт по лабораторной работе

Лабораторная №4

Полина Витальевна Барабаш

Содержание

# 1 Цель работы

Целью данной работы является получение навыков правильной работы с репозиториями git [1].

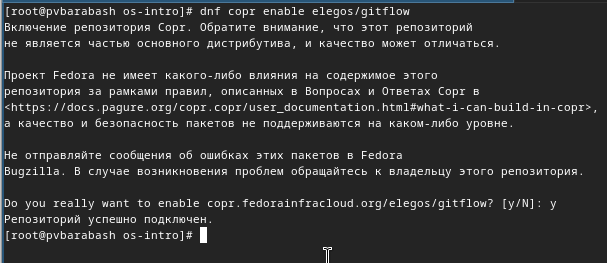
# 2 Задание

Выполнить работу для тестового репозитория.

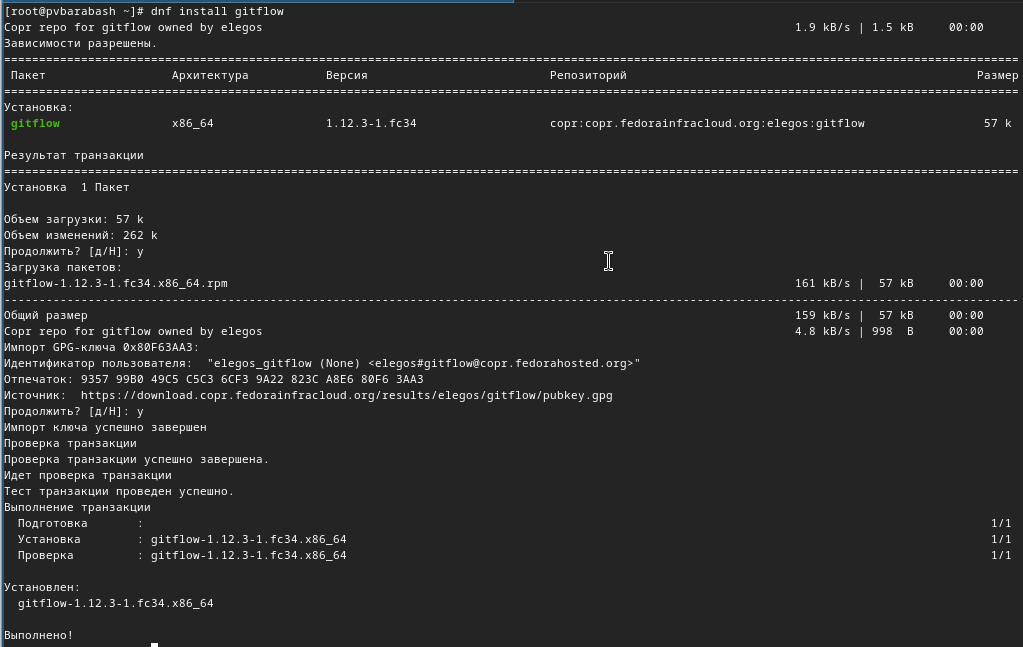
# 3 Выполнение лабораторной работы

**Задание 1.** Установить git-flow.

Из режима суперпользователя я установила git-flow, сначала введя команду dnf copr enable elegos/gitflow (рис. [??]), а затем dnf install gitflow (рис. [??]).



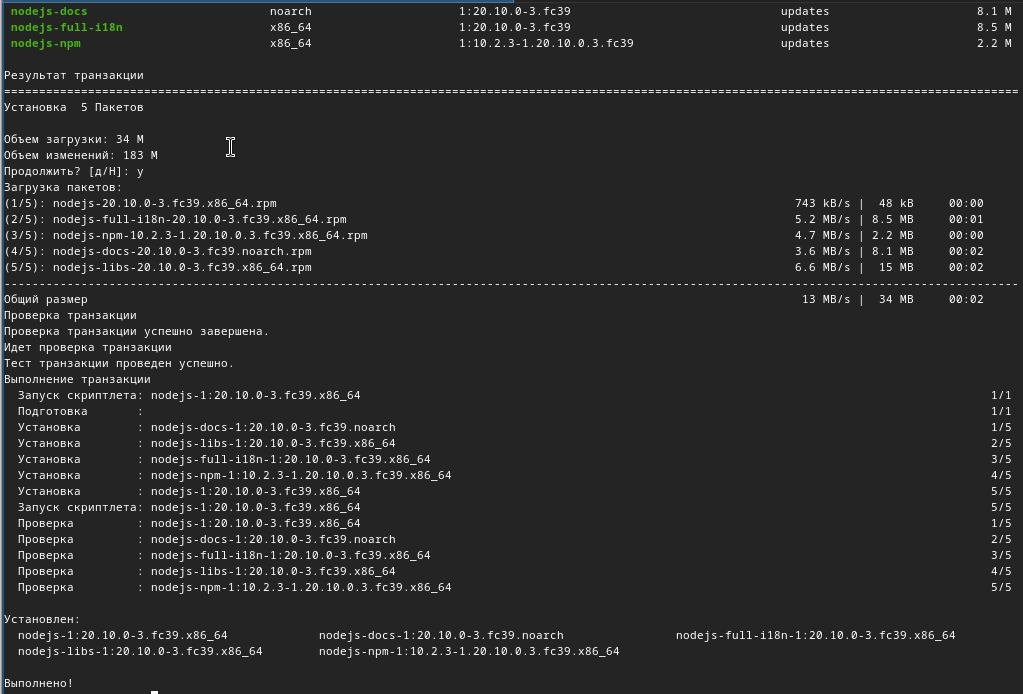
Включение репозитория copr



Установка git-flow

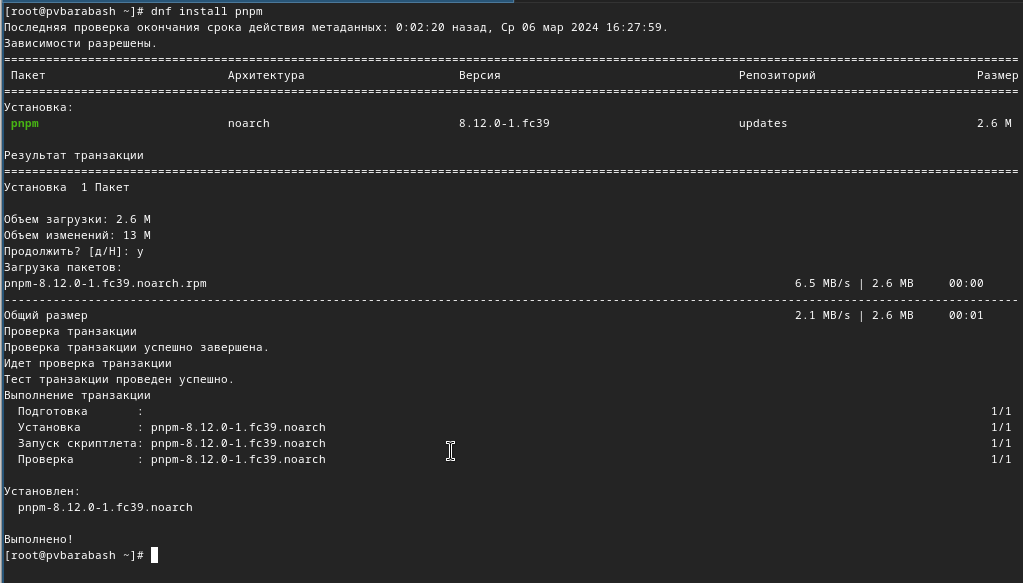
**Задание 2.** Установить Node.js.

Я установила Node.js с помощью команды dnf install nodejs (рис. [??]).



Установка Node.js

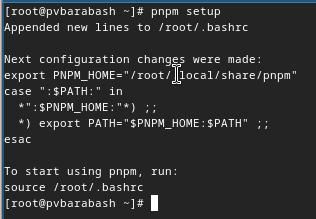
Затем я установила pnpm с помощью команды dnf install pnpm (рис. [??]).



Установка pnpm

**Задание 3.** Настроить Node.js.

Я выполненила команду настройки pnpm setup (рис. [??]).



Настройка pnpm

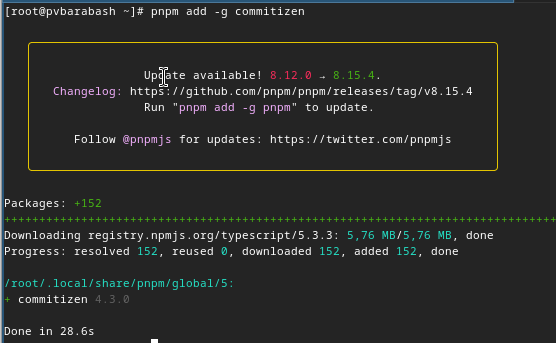
А затем выполнила команду source ~/.bashrc, чтобы использовать pnpm (рис. [??]).

Перезагрузка pnpm

Перезагрузка pnpm

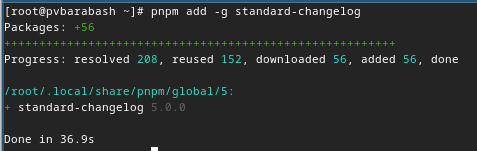
**Задание 4.** Настроить общепринятые коммиты.

Я добавила программу commitizen с помощью команды pnpm add -g commitizen (рис. [??]).



Добавление программы commitizen

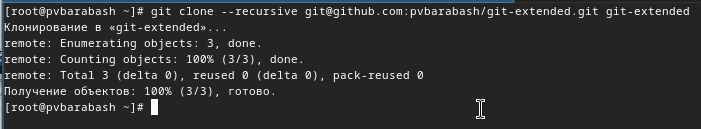
Я добавила программу standard-changelog с помощью команды pnpm add -g standard-changelog (рис. [??]).



Добавление программы standard-changelog

**Задание 5.** Создать репозиторий git.

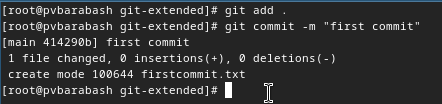
Я создала новый репозиторий на GitHub под названием git-extended и клонировала его в каталог git-extended своей виртуальной машины (рис. [??]).



Создание и копирования репозитория на GitHub

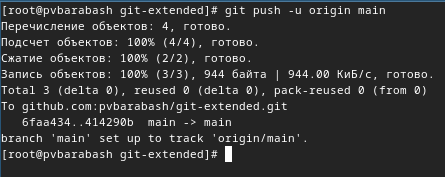
**Задание 6.** Сделать первый коммит и выложить на github.

Я создала файл firstcommit.txt для того, чтобы были какие-то изменения в репозитории. Затем я выполнила команды git add . и git commit -m “first commit” (рис. [??]).



Добавление первого коммита

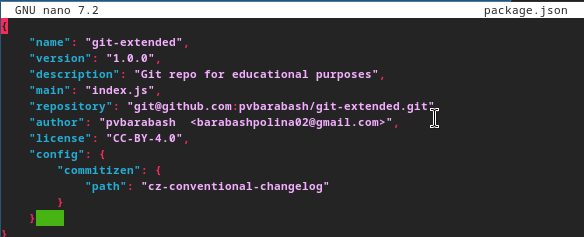
Затем я выложила изменения на github с помощью команды git push -u origin main (рис. [??]).



Выгрузка первого коммита на github

**Задание 7.** Настроить конфигурацию общепринятых коммитов.

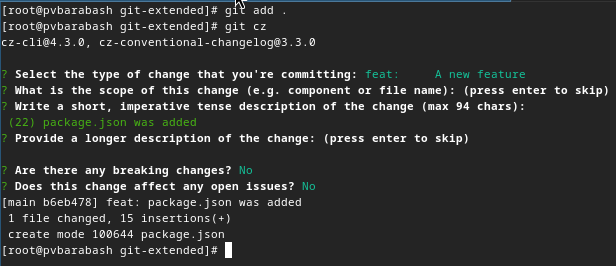
Сначала я инициализировала pnpm с помощью команды pnpm init. Затем открыла файл package.json в nano и изменила текст в соответствии с тем, что выложен на ТУИС (рис. [??]).



Настройка конфигурации общепринятых коммитов

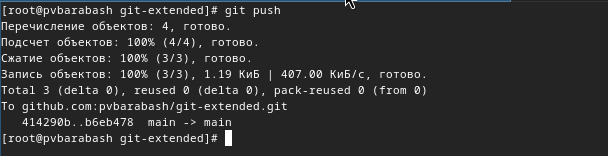
**Задание 8.** Добавить новые файлы, выполнить коммит и отправить на github.

Я добавила новые файлы с помощью команды “git add .”. Выполнила коммит с помощью команды git cz и ответила на выпадающие вопросы (рис. [??]).



Добавление новых файлов и коммита

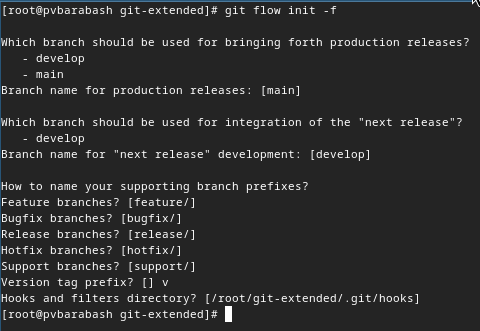
И затем отправила на github с помощью команды git push (рис. [??]).



Отправка файлов на github

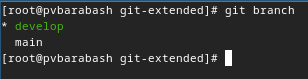
**Задание 9.** Задать конфигурацию git-flow.

Я инициализировала git-flow с помощью команды git flow init, установив префикс для ярлыков в v (рис. [??]).



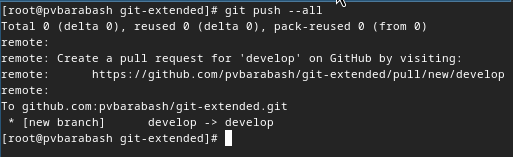
Инициализация git-flow

Я проверила, что нахожусь на ветке develop с помощью команды git branch (рис. [??]).



Проверка местонахождения

Я загрузила весь репозиторий в хранилище с помощью команды git push –all (рис. [??]).



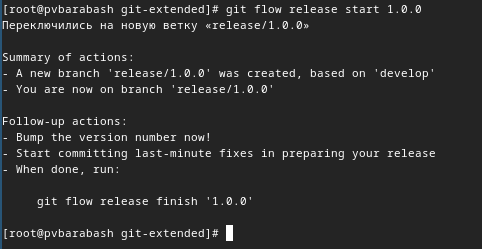
Проверка местонахождения

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

Проверка местонахождения

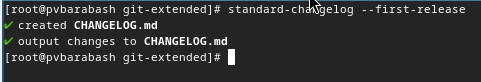
Проверка местонахождения

Я создала релиз с версией 1.0.0 с помощью git flow release start 1.0.0 (рис. [??]).



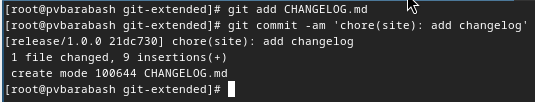
Отправка файлов на сервер

Я создала журнал изменений с помощью standard-changelog –first-release (рис. [??]).



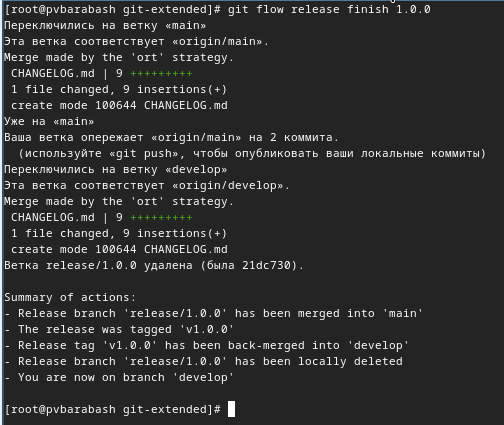
Создание журнала изменений

Я добавила журнал изменений в индекс с помощью команд git add CHANGELOG.md и git commit -am ‘chore(site): add changelog’ (рис. [??]).



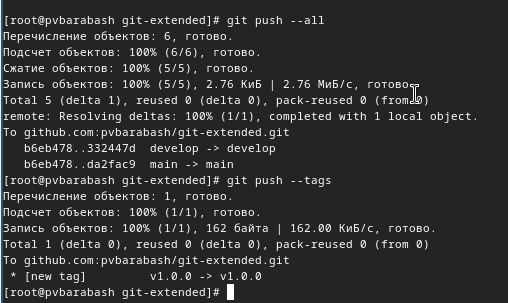
Добавление журнала изменений в индекс

Я залила релизную ветку в основную ветку с помощью git flow release finish 1.0.0 (рис. [??]).



Слитие релизной ветки в основную

Я отправила данные на github с помощью команд git push –all и git push –tags (рис. [??]).



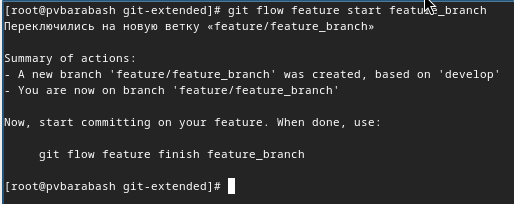
Отправка данных на github

Я создала релиз на github. Для этого использовала утилиты работы с github: gh release create v1.0.0 -F CHANGELOG.md (рис. [??]).

Созданиие релиза на github

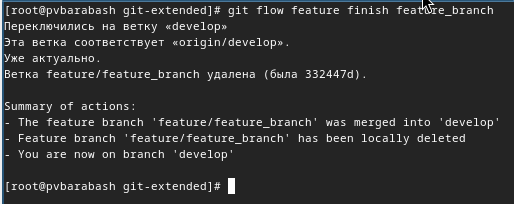
Созданиие релиза на github

Я создала ветку для новой функциональности с помощью команды git flow feature start feature\_branch (рис. [??]).



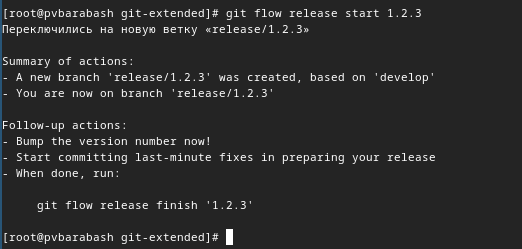
Создание ветки для новой функциональности

Следующим шагом я объединила ветку feature\_branch c develop с помощью git flow feature finish feature\_branch (рис. [??]).



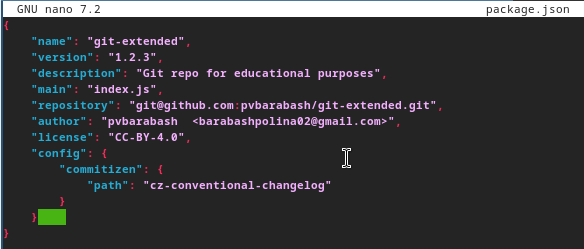
Объединение веток

Я создала релиз с версией 1.2.3 с помощью команды git flow release start 1.2.3 (рис. [??]).



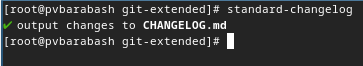
Создание релиза с версией 1.2.3

Я обновила номер версии в файле package.json: установила её в 1.2.3 (рис. [??]).



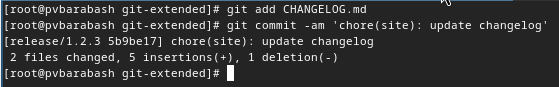
Обновление номера версии

Я создала журнал изменений с помощью команды standard-changelog (рис. [??]).



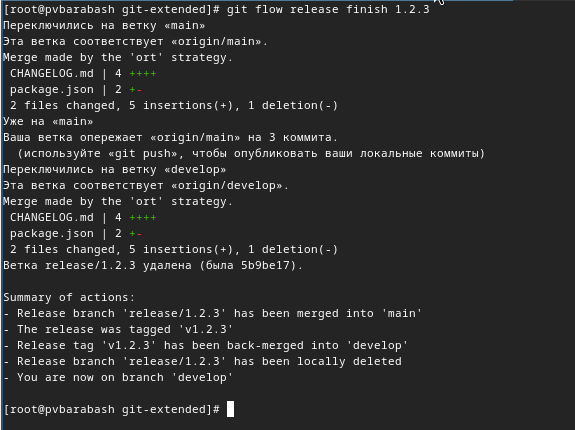
Создание журнала изменений

Я добавила журнал изменений в индекс с помощью команд git add CHANGELOG.md и git commit -am ‘chore(site): update changelog’ (рис. [??]).



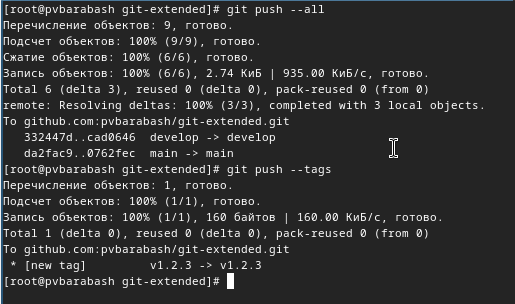
Добавление журнала изменений в индекс

Я залила релизную ветку в основную ветку с помощью команды git flow release finish 1.2.3 (рис. [??]).



Объединение веток

Я отправила данные на github с помощью команд git push –all и git push –tags (рис. [??]).



Отправление данных на github

Я создала релиз на github с комментарием из журнала изменений с помощью команды gh release create v1.2.3 -F CHANGELOG.md (рис. [??]).

Создание релиза на github

Создание релиза на github

# 4 Выводы

При выполнении данной лабораторной работы я получила навыки правильной работы с репозиториями git, настроила конфигурацию общепринятых коммитов.

# Список литературы

1. ТУИС [Электронный ресурс]. РУДН. URL: <https://esystem.rudn.ru/mod/page/view.php?id=1098937>.