Отчёт по лабораторной работе №4

Дисциплина: операционные системы

Мориссала Донзо , НКАБд-01-24

Содержание

# Цель работы

Цель данной работы является получением навыков правильной работы с репозиториями git.

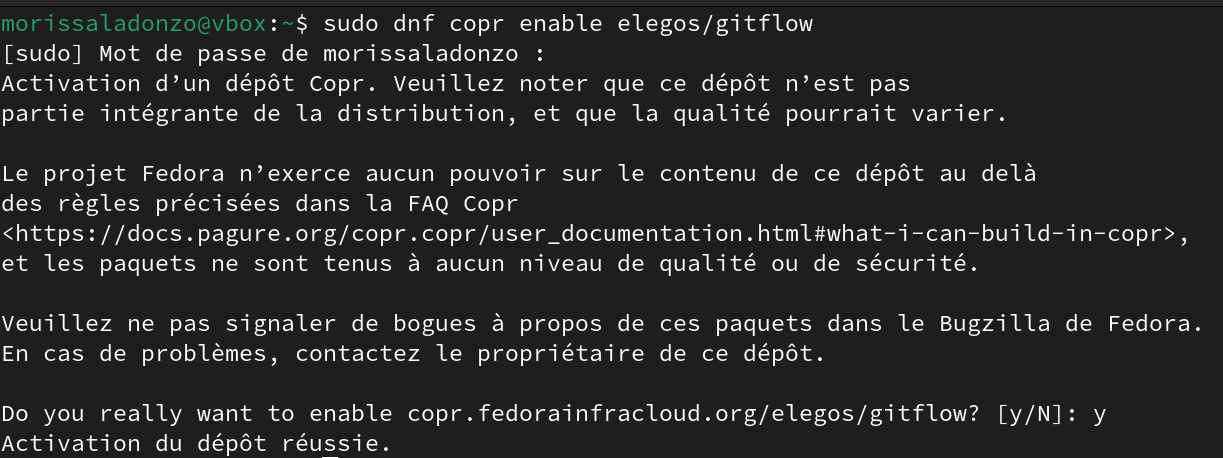
# Задание

1. Выполнить работу для тестового репозитория.
2. Преобразовать рабочий репозиторий в репозиторий с git-flow и conventional commits.

# Выполнение лабораторной работы

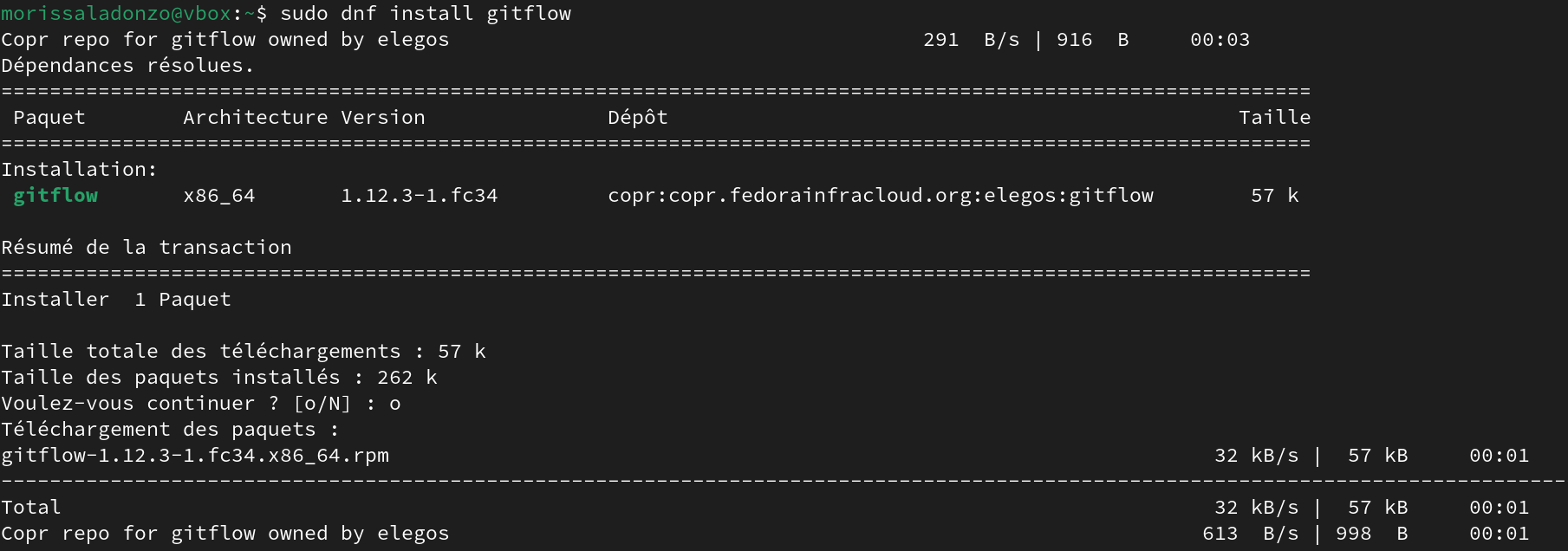
## Установка git-flow

Gitflow Workflow предполагает выстраивание строгой модели ветвления с учётом выпуска проекта. Сначала я включаю репозиторий copr:



включение copr

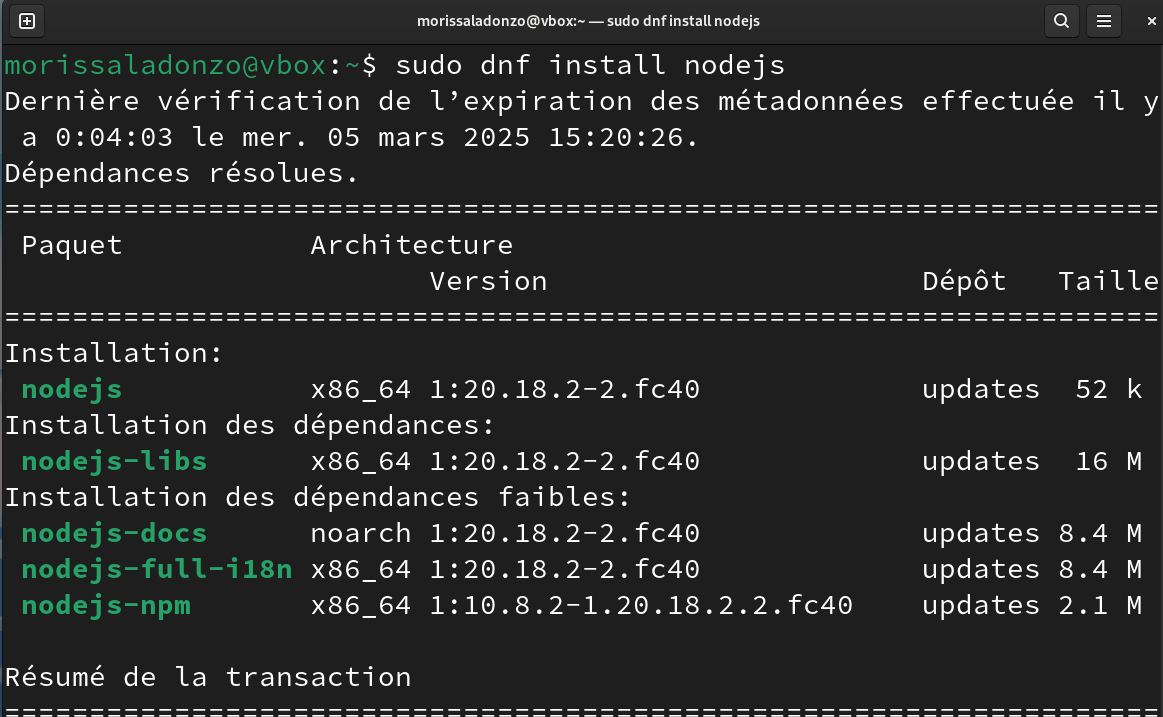
Используя dnf install скачаю gitflow:



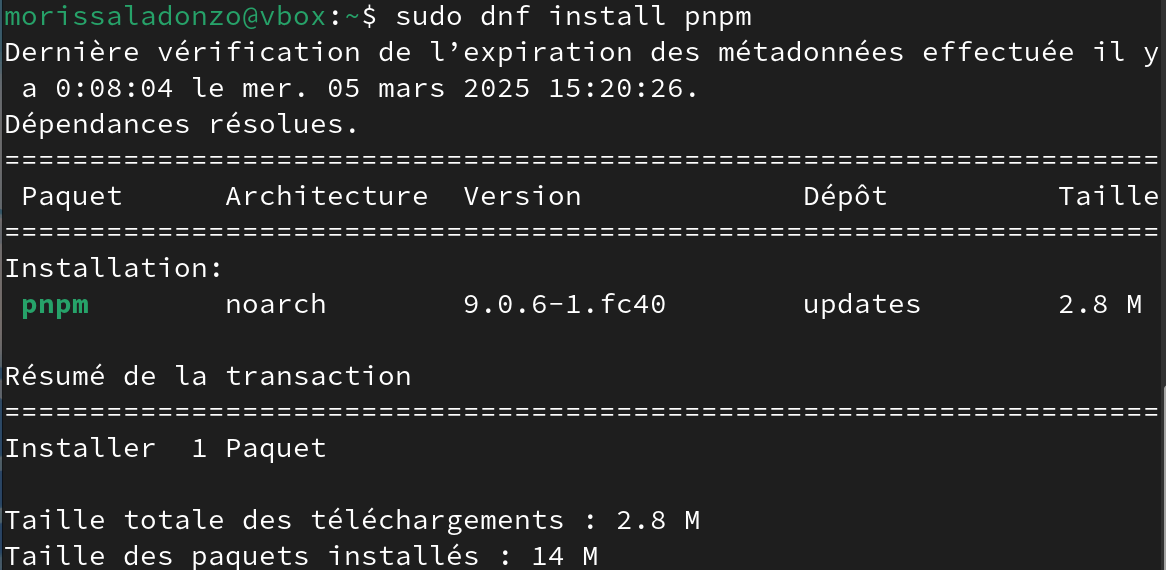
dnf install gitflow

## Установка Node.js

Для семантического версионирования и общепринятых коммитов я устанавливаю Nodejs и pnpm:



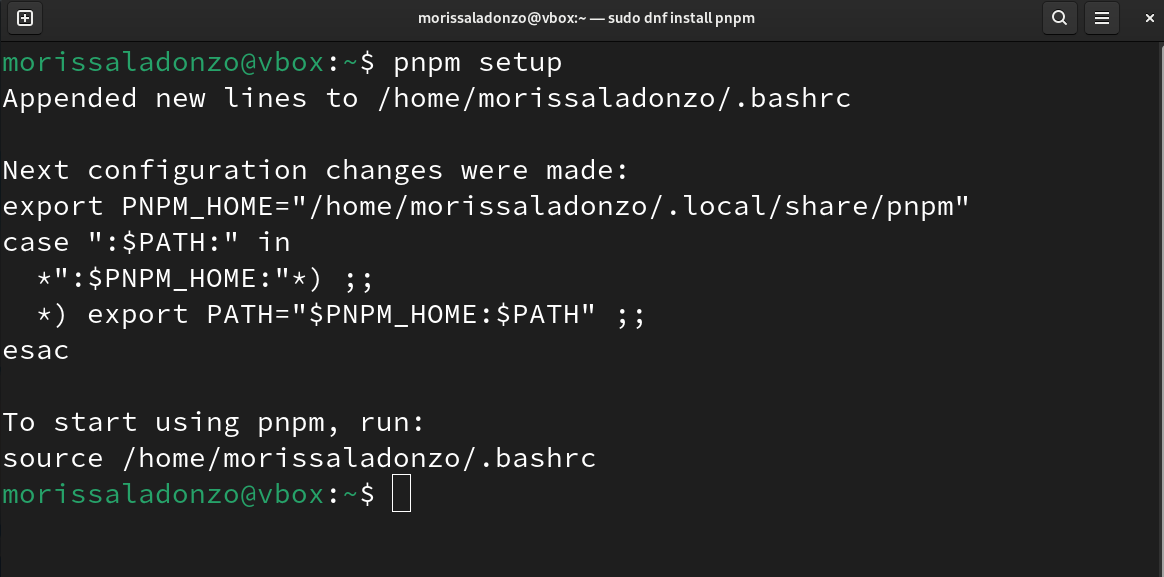
Установка Nodejs



Установка pnpm

## Настройка Node.js

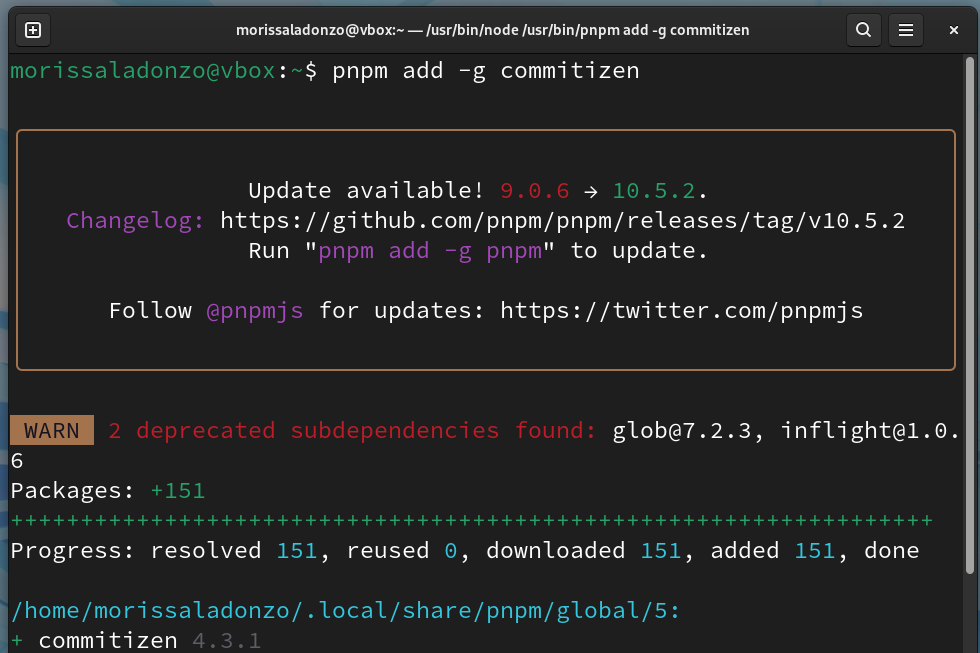
Запуская pnpm setup я добавляю каталог с исполняемыми файлами, устанавливаемыми yarn для работы с Node.js в переменную PATH:



Запуск pnpm

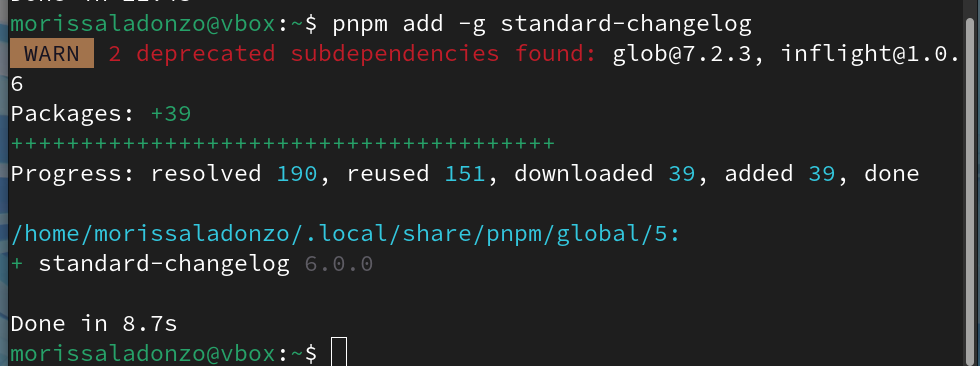
## Общепринятые коммиты

Для помощи в форматировании коммитов добавляю программу commitizen:



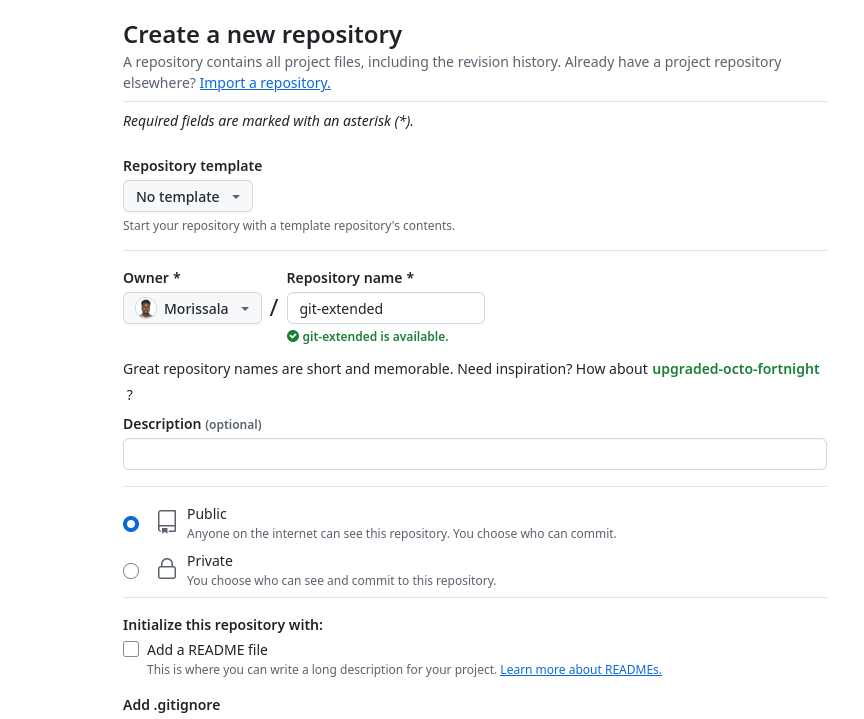
добавление commitizen

Добавляю standard-changelog для помощи в создании логов:



добавление standard-changelog

Создаю репозиторий на GitHub назову его git-extended:



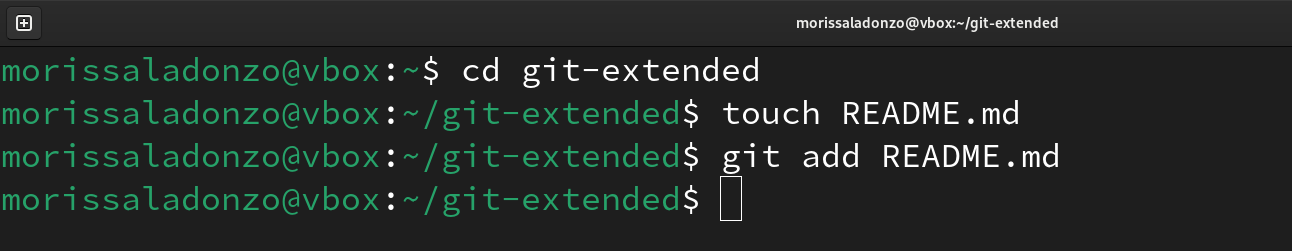
Создание git-extended

Я клонирую его в каталог git-extended:



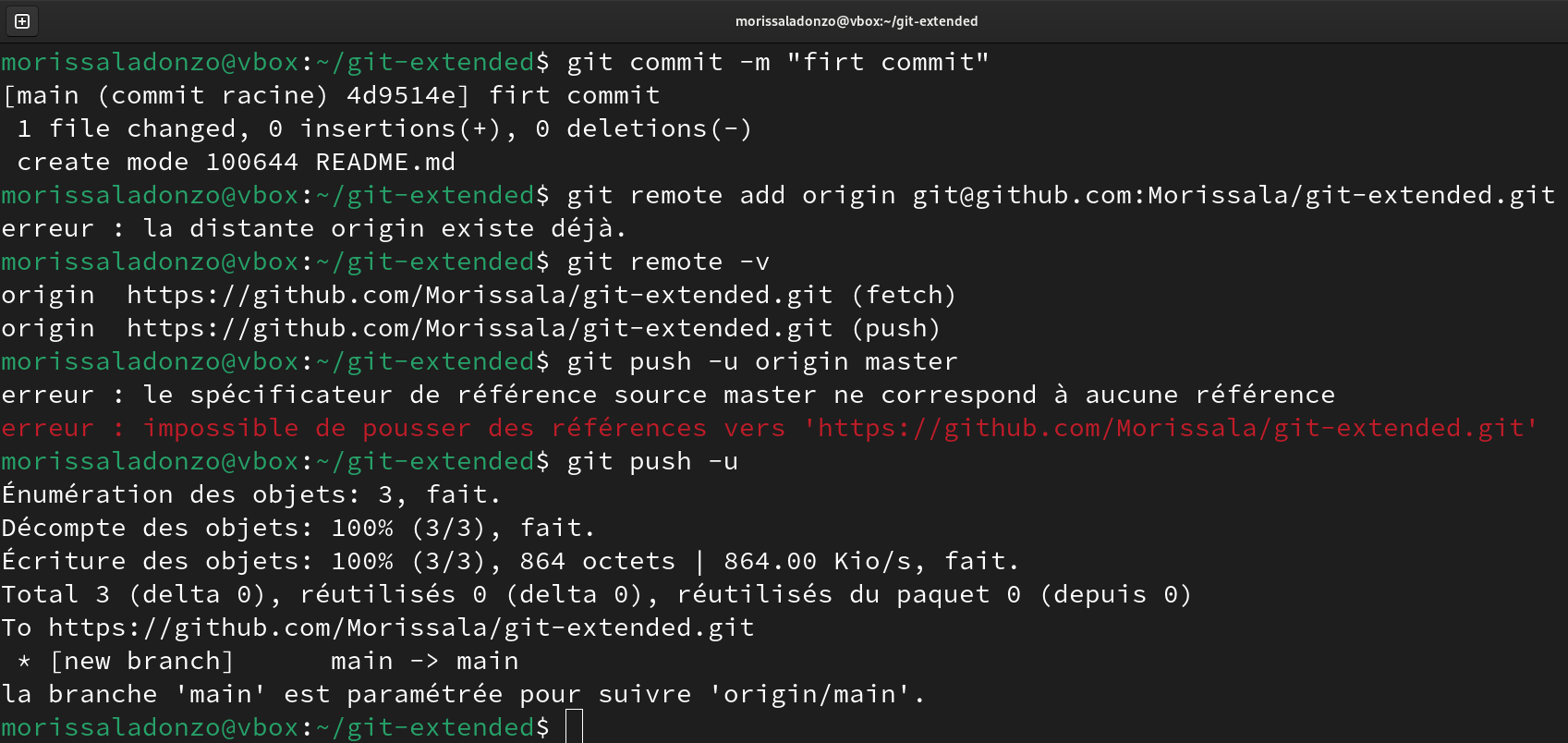
клонирование git-extended

Создаю и добавляю файл README.md:



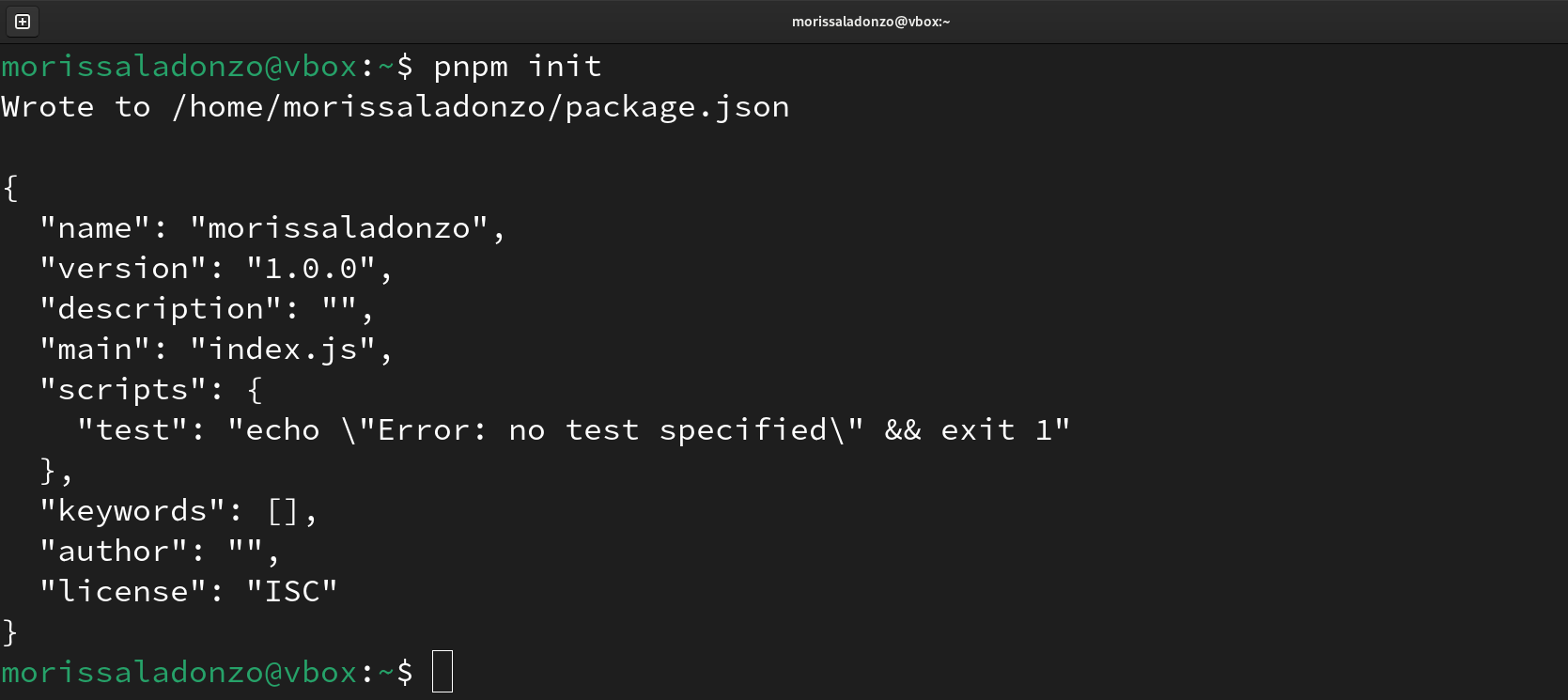
Добавление readme.md

Делаю первый коммит и выкладываю на github:



Первый коммит

Я инициализирую pnpm:



Инициализирование pnpm

Заполняю несколько параметров пакета (файл package.json):



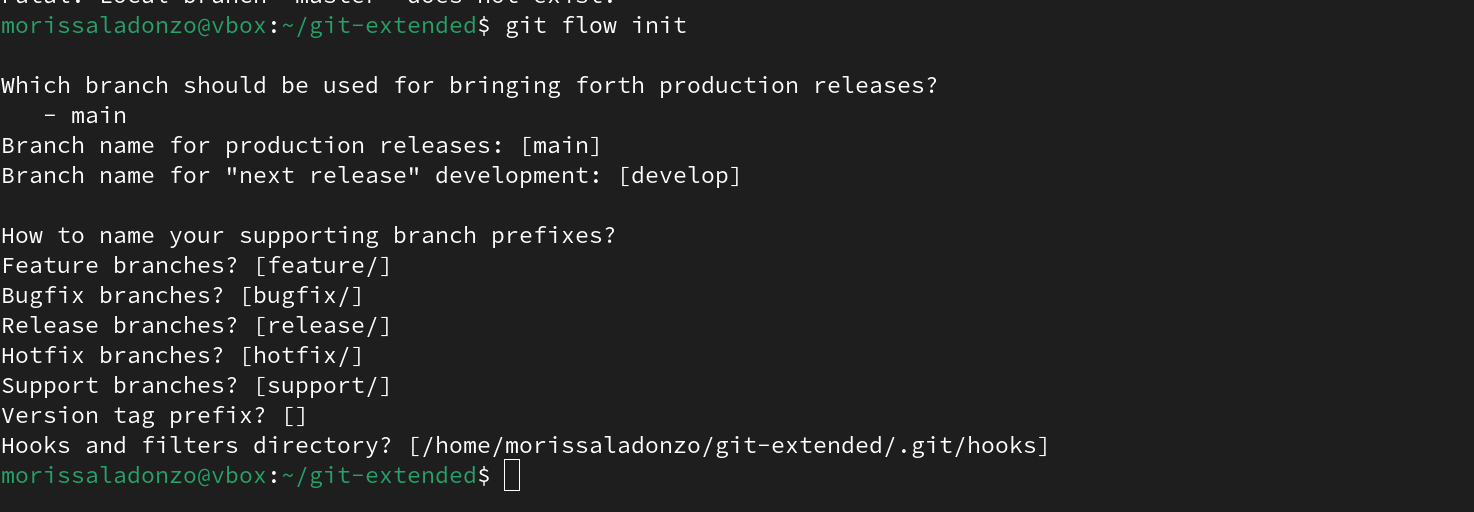
Заполнение пакетов

Добавляю новые файлы и выполняю коммит (указиваю тип коммит (feat)) и отправляю на гит:



Добавление новых файлов

Инициализирую git-flow и указываю ветки:



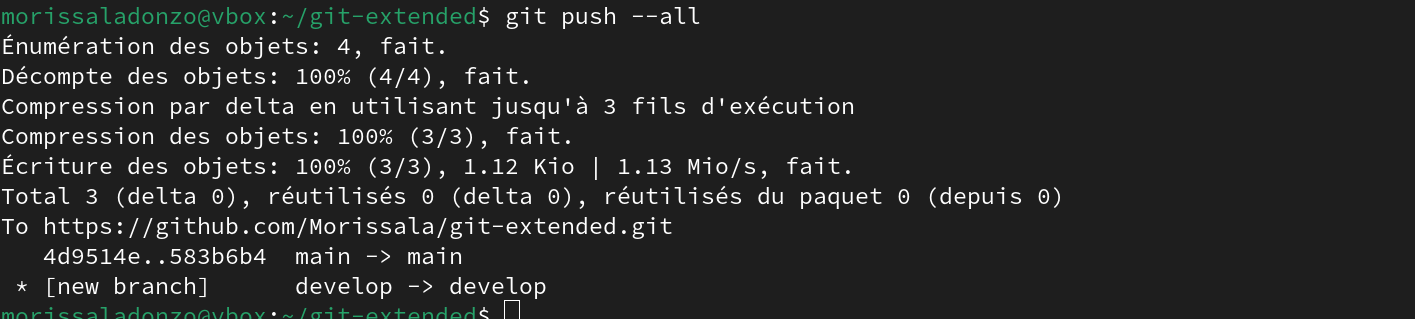
Инициализирование git-flow

Преверяю что я на ветке develop с промощью git branch:



Проверка ветки

Загружаю весь репозиторий в хранилище с помощью git push –all:



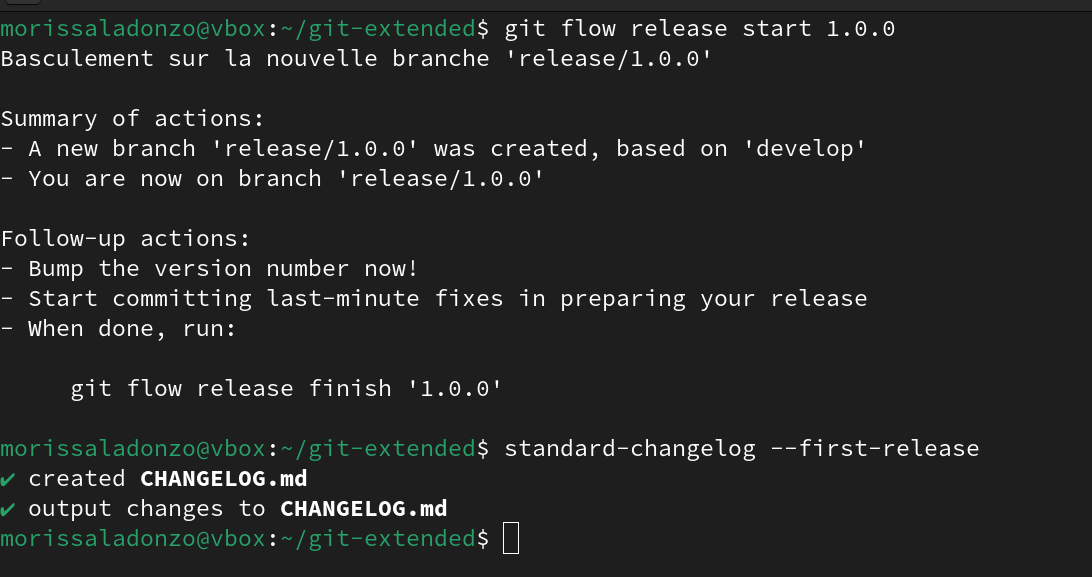
Загрузка репозиторий

Установливаю внешнюю ветку как вышестоящую для этой ветки (develop):

установка внешней ветки

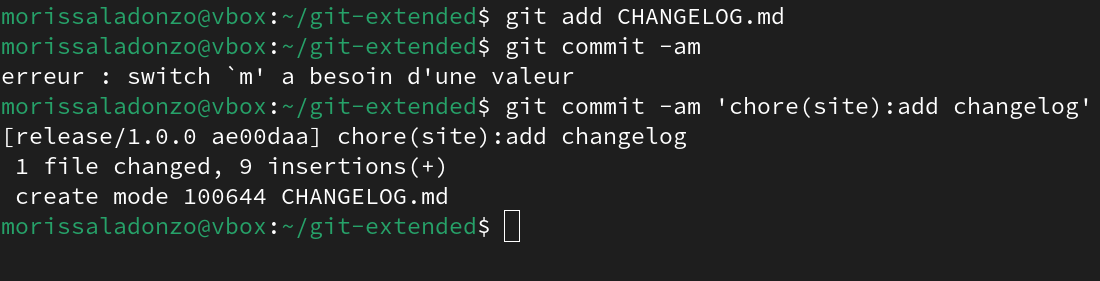
установка внешней ветки

Создаю релиз с версией 1.0.0 и журнал изменений (standard-changelog)::



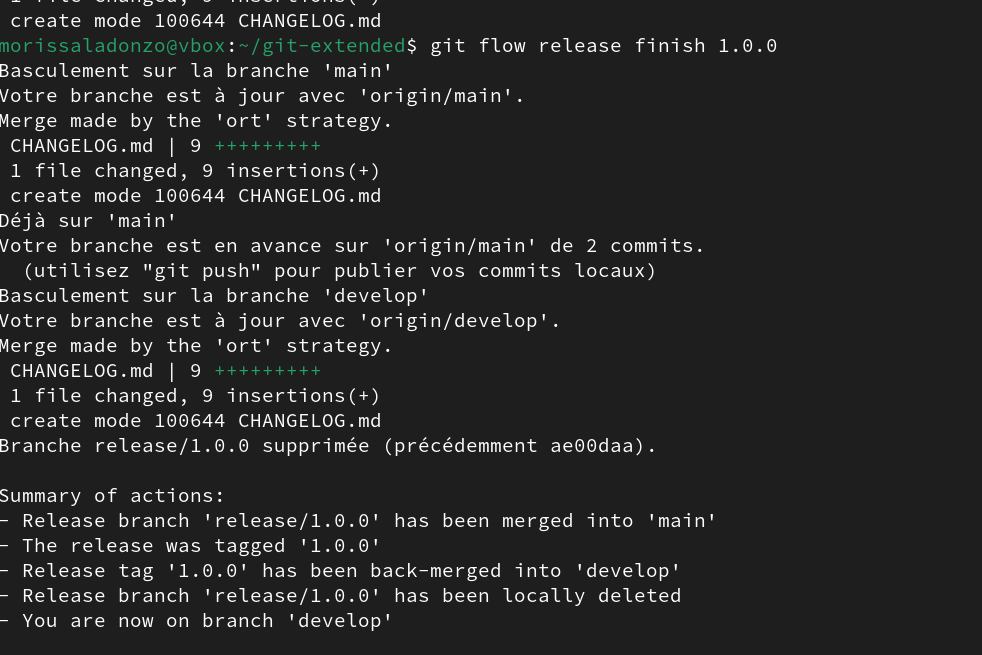
Создание релиза

Добавляю журнал изменений в индекс:



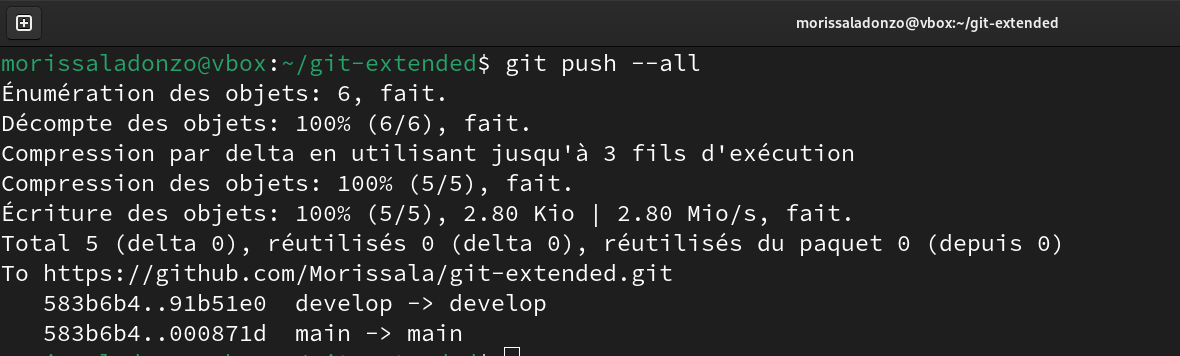
Добавление журнала изменений

Залью релизную ветку в основную ветку:



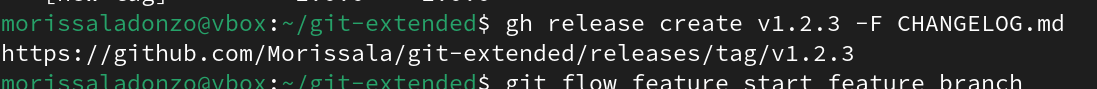
Замена ветки

Отправляю данные на github:



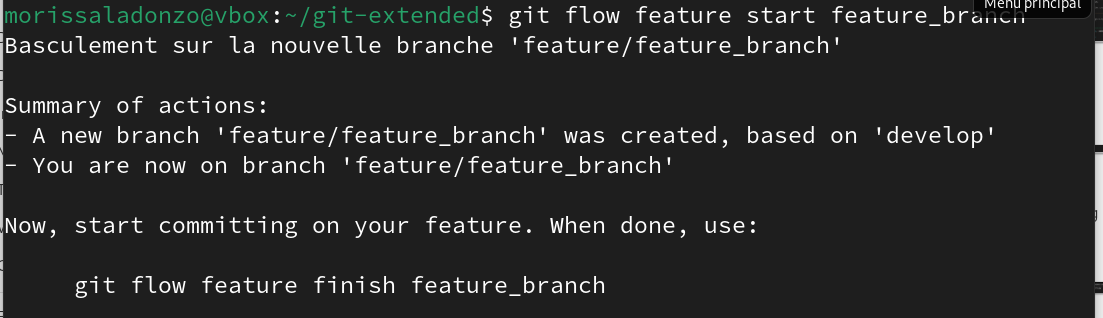
Отправка на гит

Создаю релиз на github. Для этого использую утилиты работы с github gh (gh release create):



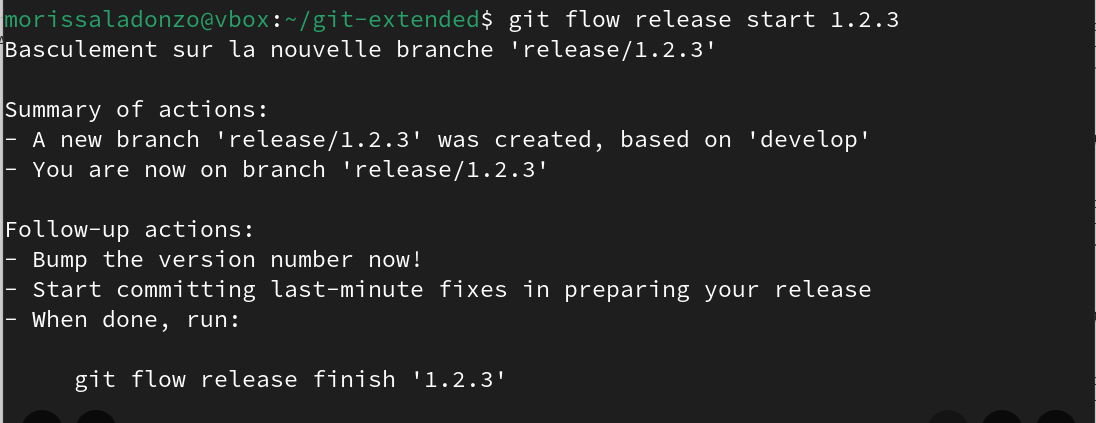
Создание релиза

Создаю ветку для новой функциональности:



Создание ветки feature\_branch

Далее, продолжаю работу c git как обычно. Создаю релиз с версией 1.2.3:



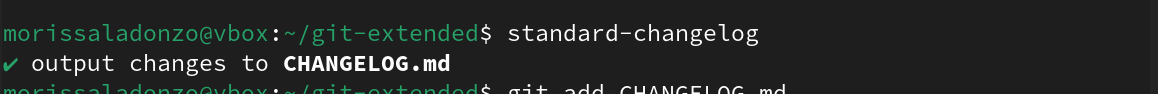
Создание релиза с версией 1.2.3

Обновляю номер версии в файле package.json в 1.2.3:



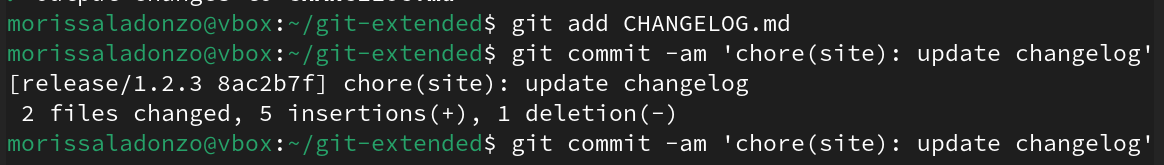
Обновление номера версии

Создаю журнал изменений (standard-changelog):



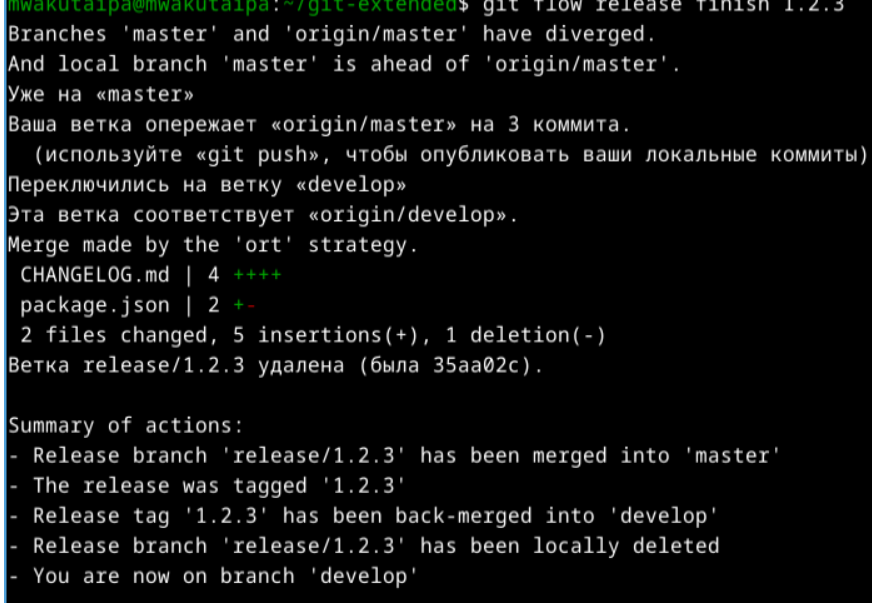
Создание нового журнала изменений

Добавляю журнал изменений в индекс:



Добавление журнала

Залью релизную ветку в основную ветку:



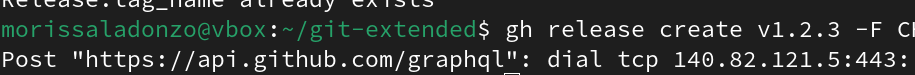
Замена ветки

Отправляю данные на github:



Отправка данных

Создаю релиз на github с комментарием из журнала изменений:



Cоздание релиза

# Выводы

При выполнение работы я получил навыки правильной работы с репозиториями git.

# Список литературы

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