Отчёт по лабораторной работе №5

Операционные системы

Терещенкова М.В.

Содержание

Список иллюстраций

# 1 Цель работы

Познакомиться с pass, gopass, native messaging, chezmoi. Научиься пользоваться этими утилитами, синхранизировать их c git.

# 2 Задание

* Установить дополнительное ПО ﻿﻿﻿- Установить и настроить pass ﻿﻿﻿- Настроить интерфейс с браузером ﻿﻿﻿- Сохранить пароль ﻿﻿﻿- Установить и настроить chezmoi ﻿﻿﻿- Настроить chezmoi на новой машине ﻿﻿﻿- Выполнить ежедневные операции с chezmoi

# 3 Теоретическое введение

**Менеджер паролей pass** Менеджер паролей pass — программа, сделанная в рамках идеологии Unix.Также носит название стандартного менеджера паролей для Unix (The standard Unix password manager).

**Основные свойства**

Данные хранятся в файловой системе в виде каталогов и файлов.Файлы шифруются с помощью GPG-ключа.

**Структура базы паролей** Структура базы может быть произвольной, если Вы собираетесь использовать её напрямую, без промежуточного программного обеспечения. Тогда семантику структуры базы данных Вы держите в своей голове.Если же необходимо использовать дополнительное программное обеспечение, необходимо семантику заложить в структуру базы паролей.

**Семантическая структура базы паролей** Рассмотрим пользователя user в домене example.com, порт 22.Отсутствие имени пользователя или порта в имени файла означает, что любое имя пользователя и порт будут совпадать:

**example.com.pgp** Соответствующее имя пользователя может быть именем файла внутри каталога, имя которого совпадает с хостом. Это полезно, если в базе есть пароли для нескольких пользователей на одном хосте:

**example.com/user.pgp** Имя пользователя также может быть записано в виде префикса, отделенного от хоста знаком @:

**user@example.com.pgp** Соответствующий порт может быть указан после хоста, отделённый двоеточием (:):

example.com:22.pgp example.com:22/user.pgp user@example.com:22.pgp Эти все записи могут быть расположены в произвольных каталогах, задающих Вашу собственную иерархию.

# 4 Выполнение лабораторной работы

## 4.1 Менеджер паролей pass

### 4.1.1 Установка

Устанавливаю pass

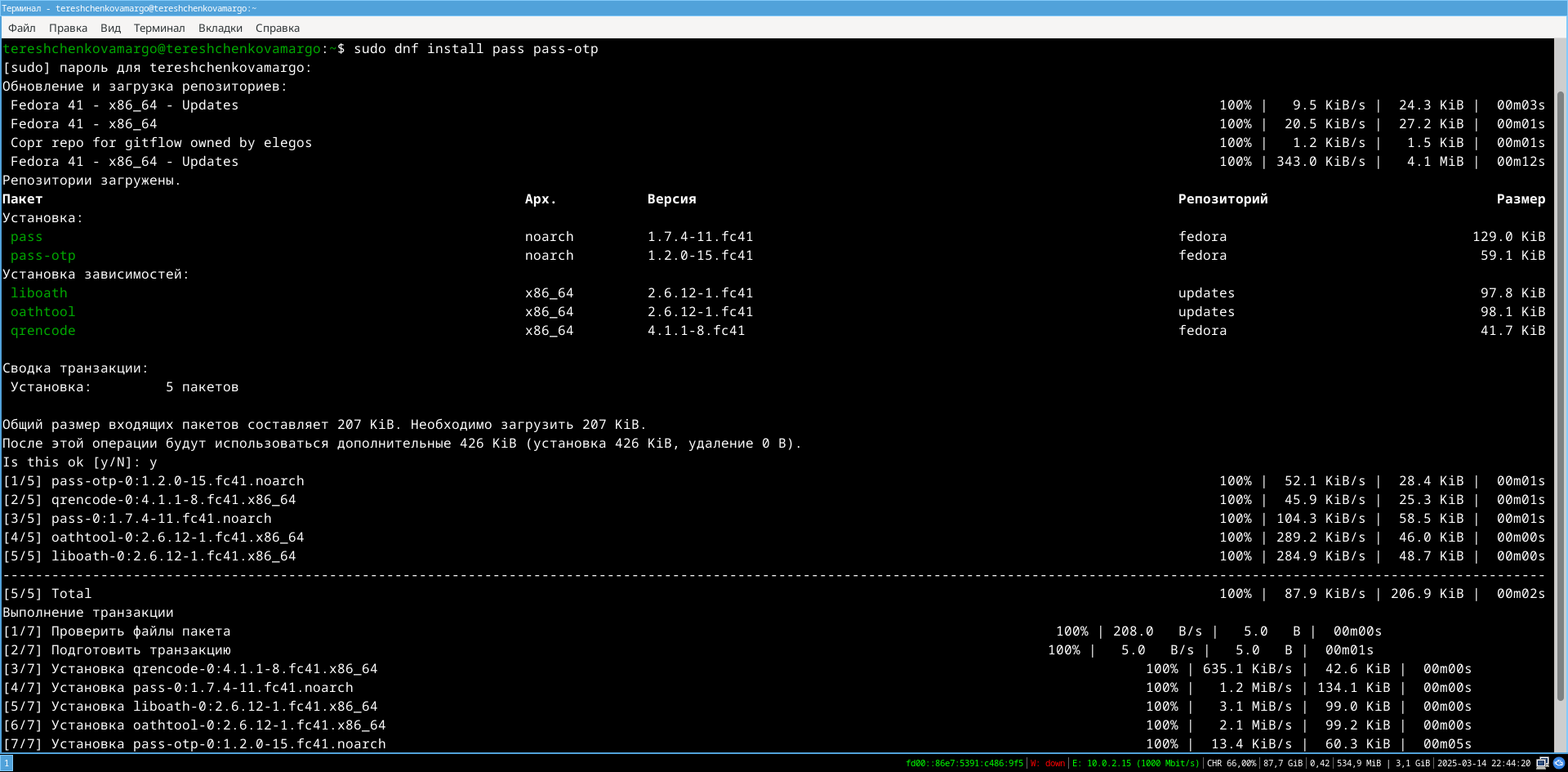


Рис. 1: Установка Pass

Устанавливаю gopass

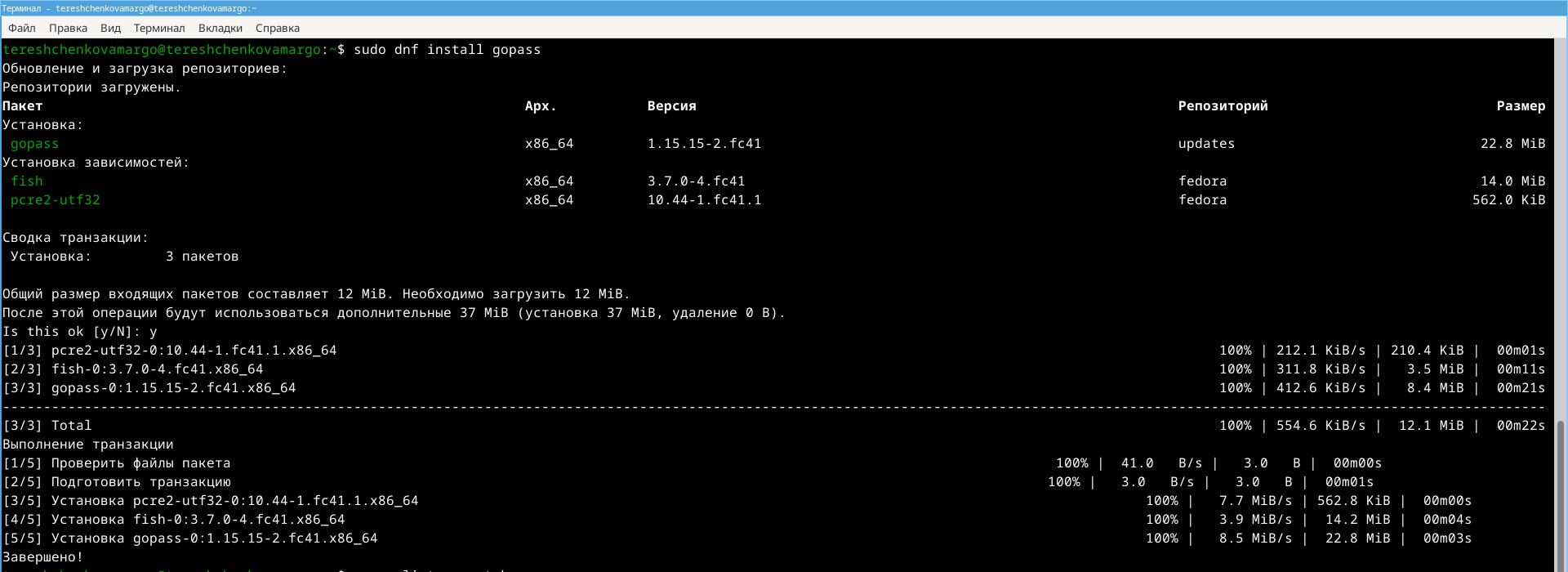


Рис. 2: Установка gopass

### 4.1.2 Настройка

Просмотриваю списки ключей gpg.

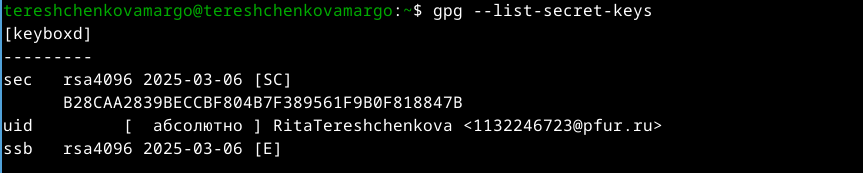


Рис. 3: Просмотр списка ключей

Вижу, что ключ есть, поэтому инициализирую хранилище.

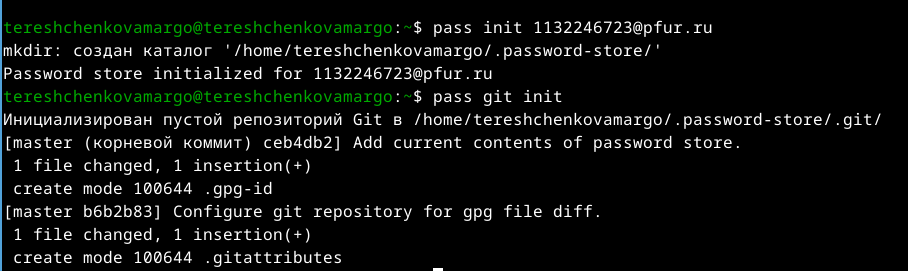


Рис. 4: Инициализация хранилища

### 4.1.3 Настройка формата коммитов

Был инициализирован package.json с настройками:

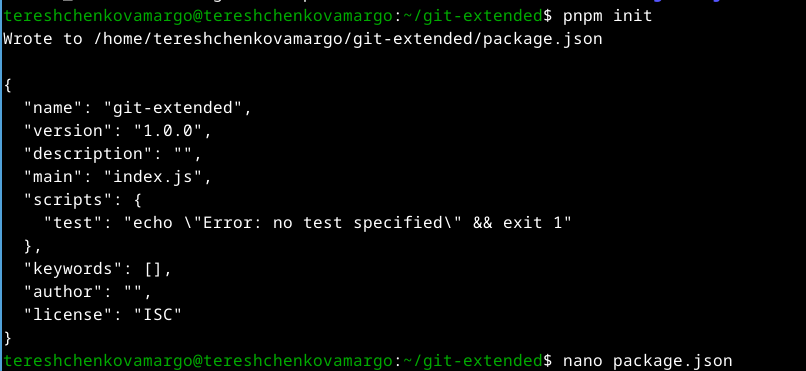


Рис. 5: Инициализация package.json

Добавлена конфигурация для commitizen:

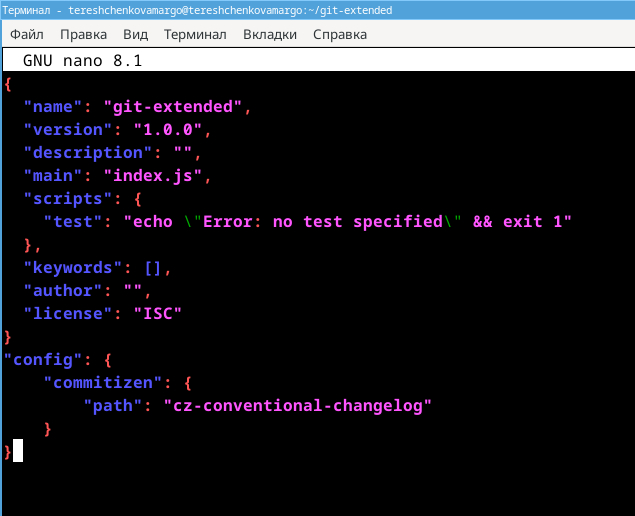


Рис. 6: Добавление конфигурации

После этого были добавлены файлы и выполнен коммит через git cz:

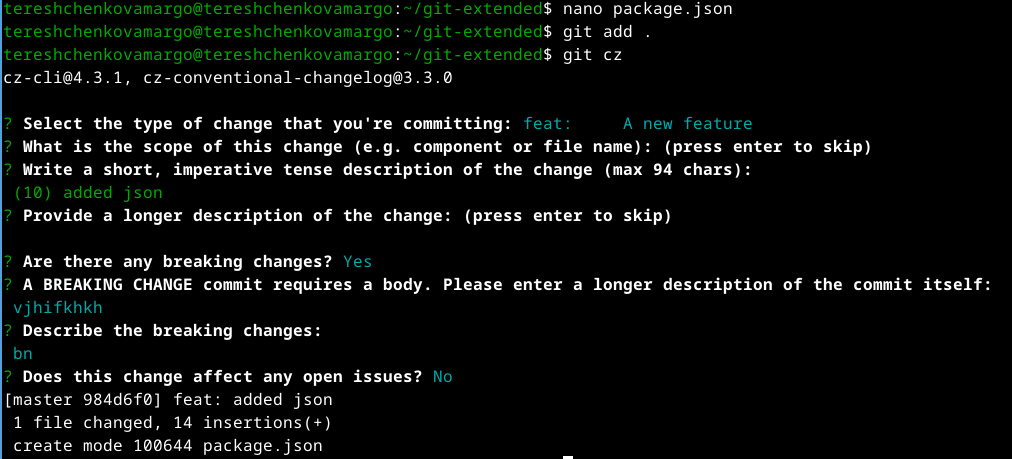


Рис. 7: Добавление файлов и выполнение коммита

## 4.2 Конфигурация git-flown

Выполняю инициализацию.Установливаю префикс для версий v. Проверяю, что активная ветка — develop и загружаю весь репозиторий.

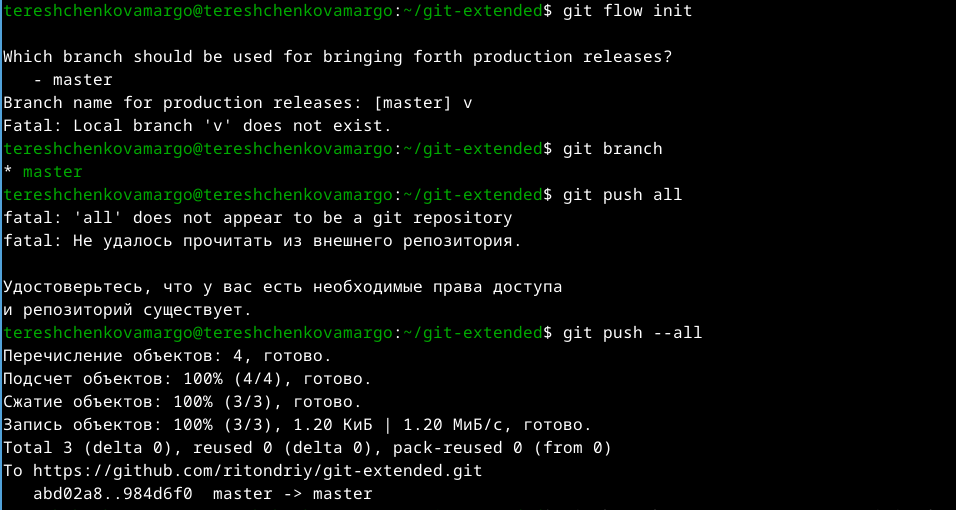


Рис. 8: Добавление файлов и выполнение коммита

## 4.3 Создание релиза

Создаю релиз 1.0.0;

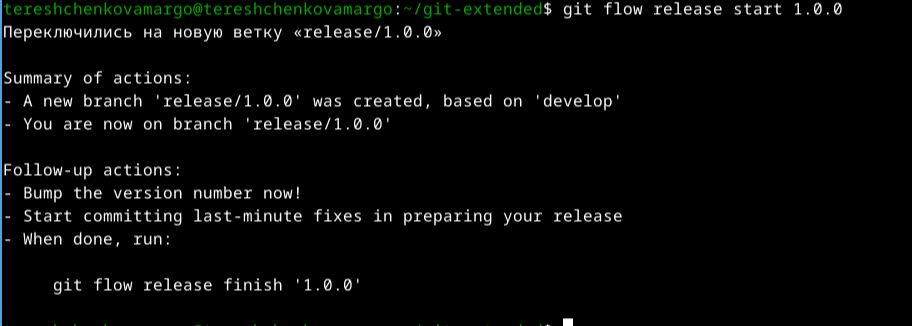


Рис. 9: Создание релиза 1.0.0

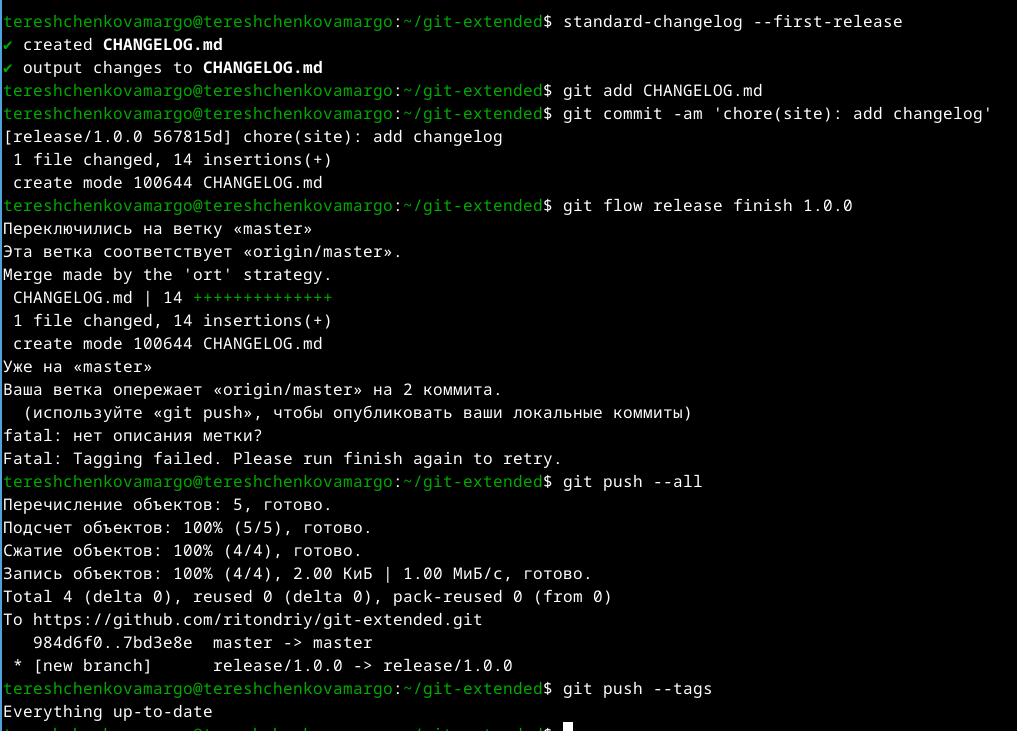


Рис. 10: Создание релиза 1.0.0

Создаю релиз на GitHub с помощью команды: **gh release create v1.0.0 -F CHANGELOG.md**

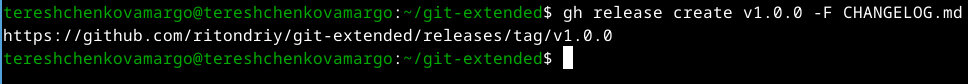


Рис. 11: Создание релиза на GitHub

## 4.4 Работа с репозиторием

### 4.4.1 Разработка новой функциональности

Создана ветка новой функциональности с помощью команды: **git flow feature start feature\_branch**. После внесения изменений выполнено объединение с develop **git flow feature finish feature\_branch**. Создаю новый релиз 1.2.3: **git flow release start 1.2.3**. Обновляю package.json, затем выполняю генерацию журнала изменений и коммит Завершаю релиз и отправляю на GitHub.

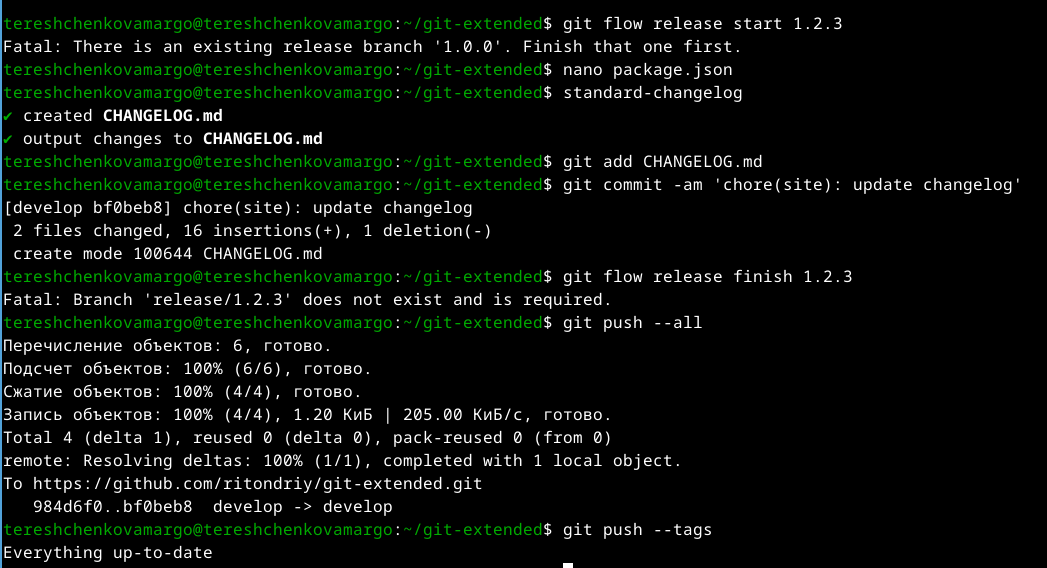


Рис. 12: Работа с репозиторием

# 5 Выводы

Познакомилась с pass, gopass, native messaging, chezmoi. Научилась пользоваться этими утилитами, синхранизировать их c git.

# Список литературы