Отчёт по индивидуальному проекту. Этап 1.

Архитектура компьютера и операционные системы

Елисейкина Надежда Михайловна НММбд-02-24

Содержание

# 1 Цель работы

* Создать свой сайт (разместить на Github pages заготовки для персонального сайта)

# 2 Задание

* Установить необходимое программное обеспечение.
* Скачать шаблон темы сайта.
* Разместить его на хостинге git.
* Установить параметр для URLs сайта.
* Разместить заготовку сайта на Github pages.

# 3 Теоретическое введение

## 3.1 Примеры использования git:

* Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды git с различными опциями.
* Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

## 3.2 Основные команды git:

Например, в табл. [1](#tbl:std-dir) приведено краткое описание основных команд Git.

Table 1: Описание некоторых команд системы контроля версий Git

| Команда | Описание команды |
| --- | --- |
| git init | Создание основного дерева репозитория |
| git pull | Получение обновлений(изменений текущего дерева из центрального репозитория |
| git push | Отправка всех произведённых изменений локального дерева в центральный репозиторий |
| git status | Просмотр списка изменённых файлов в текущей директории |
| git diff | Просмотр текущих изменений |
| git add . | Добавление все изменённые и/или созданные файлы и/или каталоги |
| git rm имена\_файлов | Удаление файлов и/или каталогов из индекса репозитория |
| git commit -am ‘Описание коммита’ | Сохранение всех добавленных изменений и всех изменённых файлов |
| git commit | Сохранение добавленный изменений с внесением комментария через встроенный редактор |
| git checkout -b имя\_ветки | Создание новой ветки, базирующейся на текущей |
| git branch -d имя\_ветки | Удаление локальной уже слитой с основным деревом ветки |
| git branch -D имя\_ветки | Принудительное удаление локальной ветки |

Полный список команд можно посмотреть на официальном сайте: [Github.com](https://docs.github.com/en/get-started/using-github/github-command-palette)

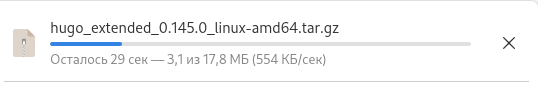
# 4 Выполнение лабораторной работы

Загружаем последнюю версию hugo :

Загрузка нужной нам версии

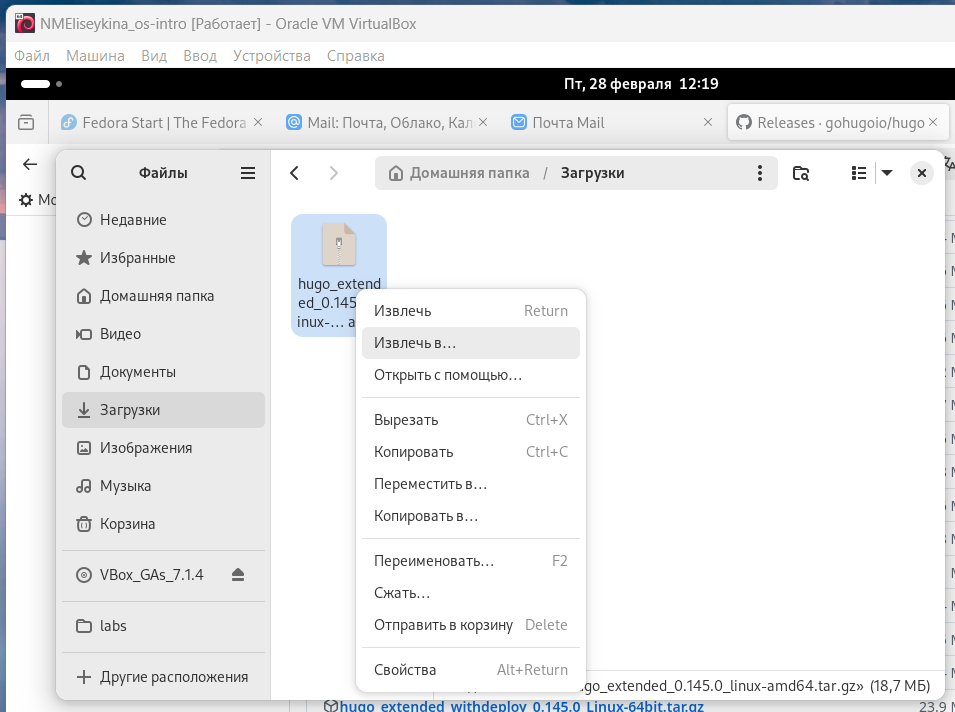
Загрузка нужной нам версии

Файл скачивается в папку *“Загрузки”* :

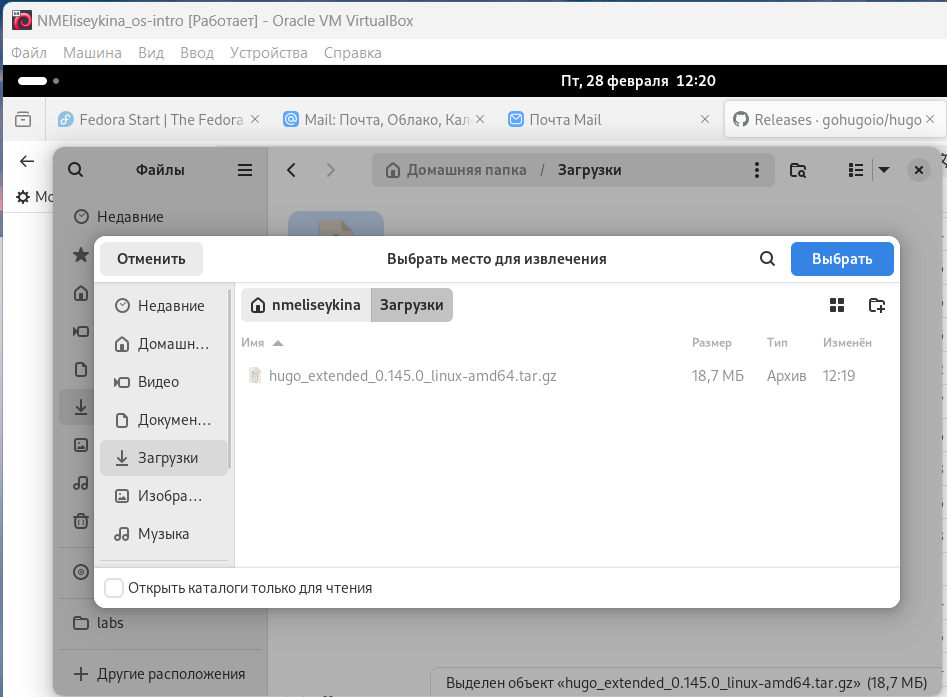


Скачивание

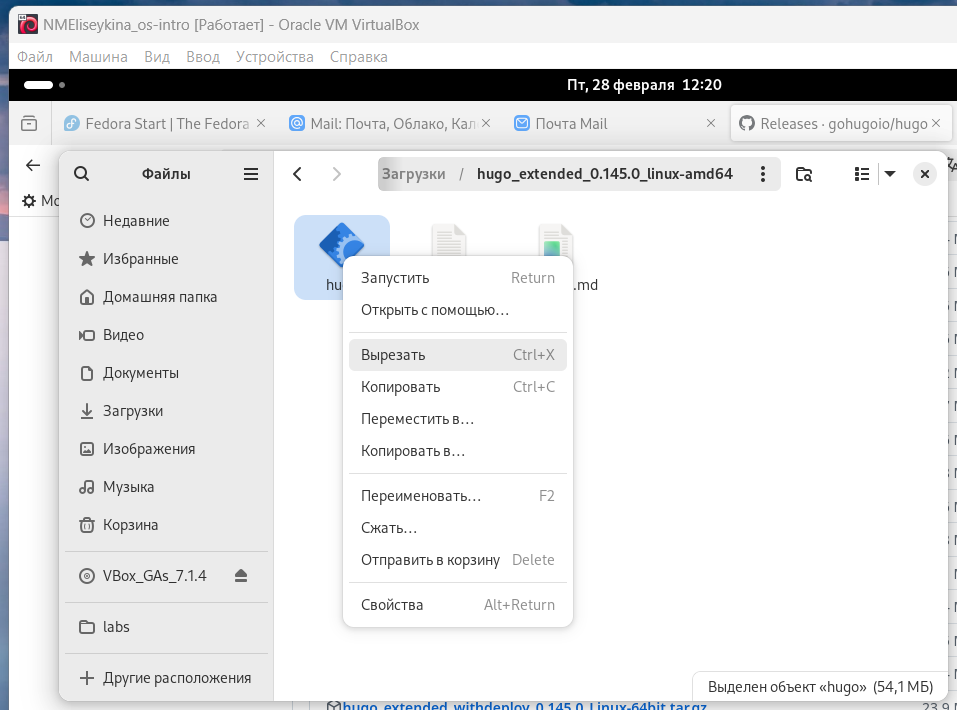
По завершении скачивания извлекаем архив в ту же папку, в которой мы находимся:



Извлечение

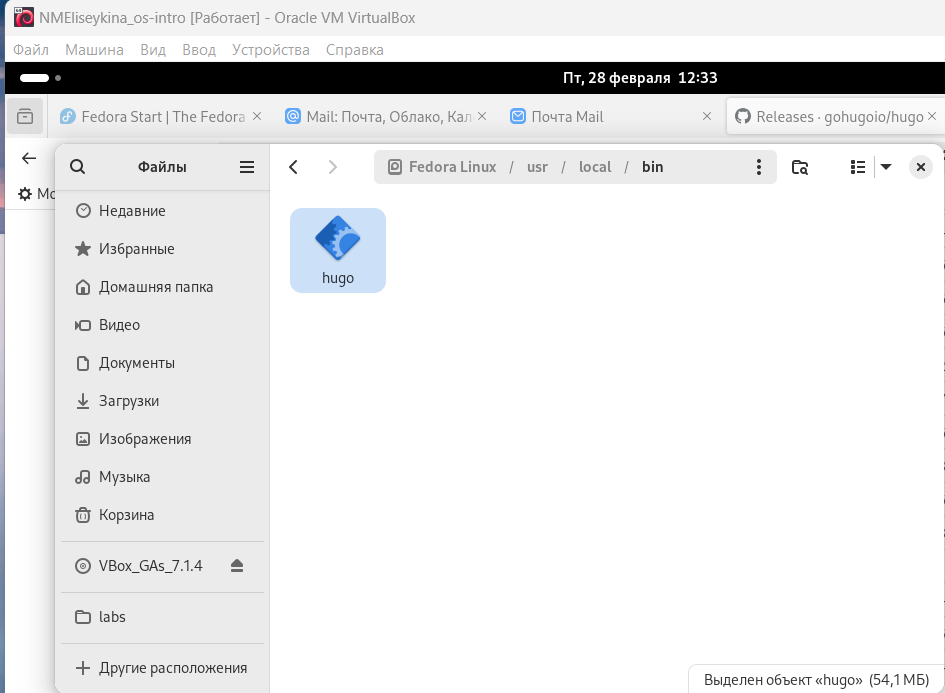


Извлечение - выбор папки

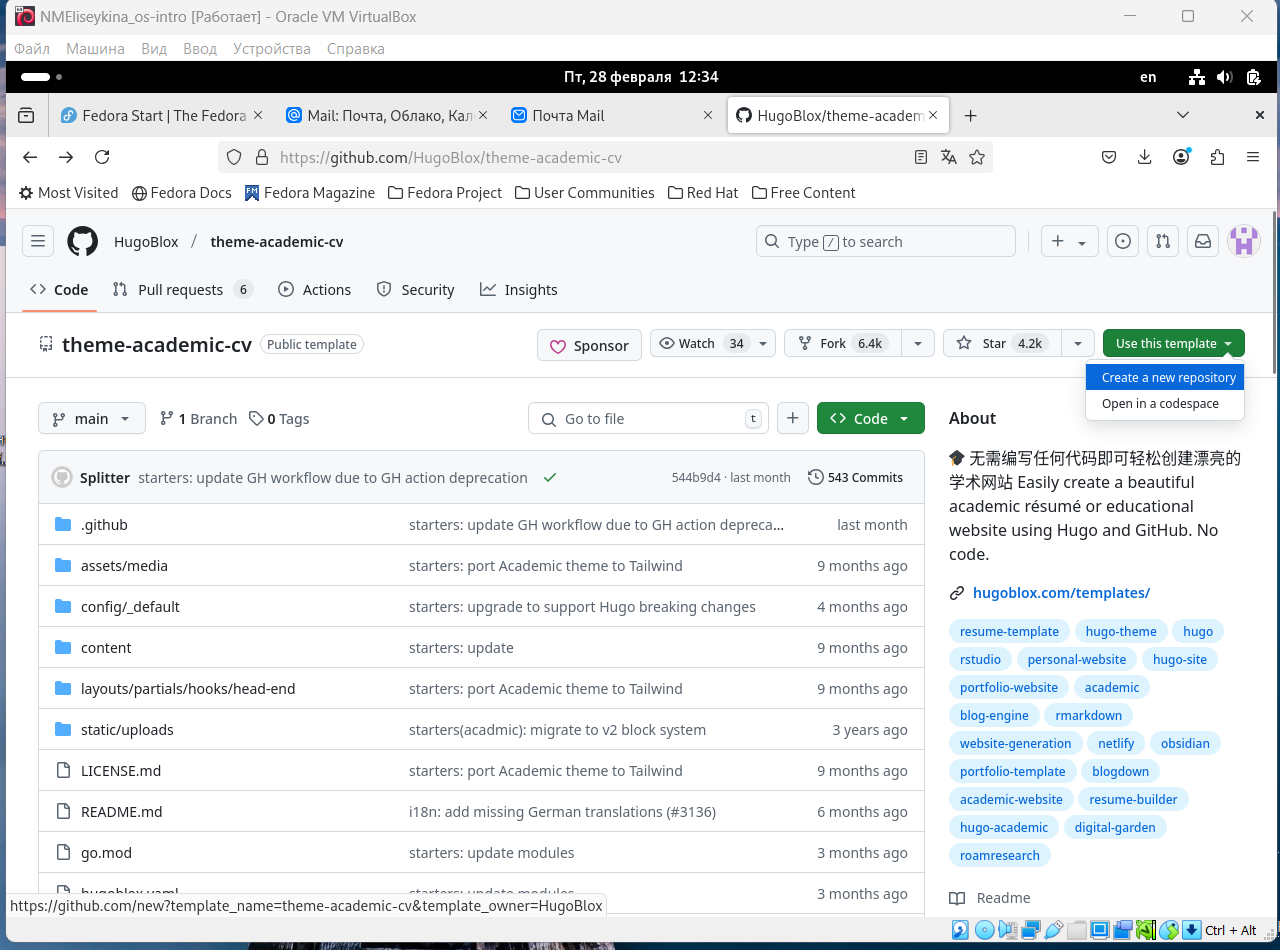


Проверка извлечения

После извлечения файла, нам его необходимо вырезать и вставить в папку */usr/local/bin*:



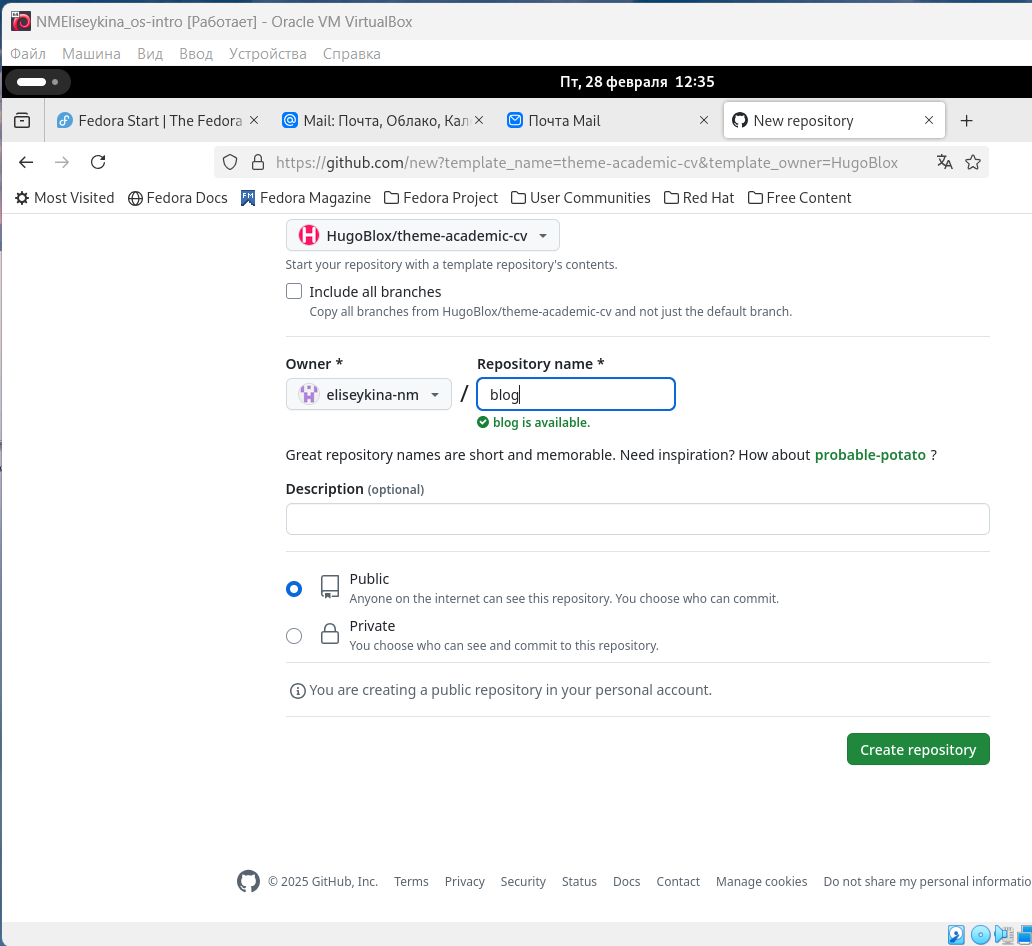
Вырезание файла



Перенос файла

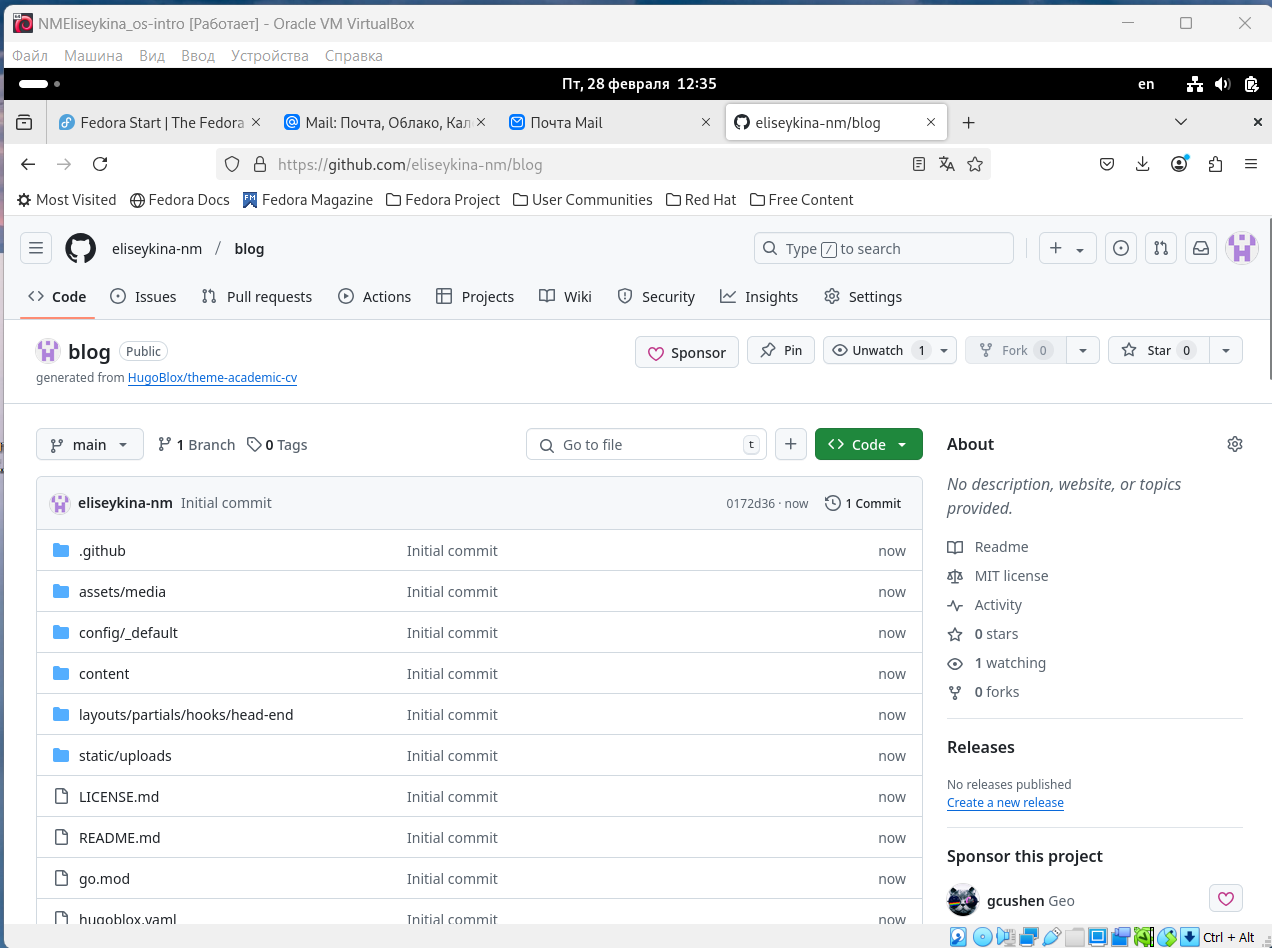
Далее открываем наш github и создаем репозиторий на основе данного нам:

[Репозиторий](https://github.com/wowchemy/starter-hugo-academic)



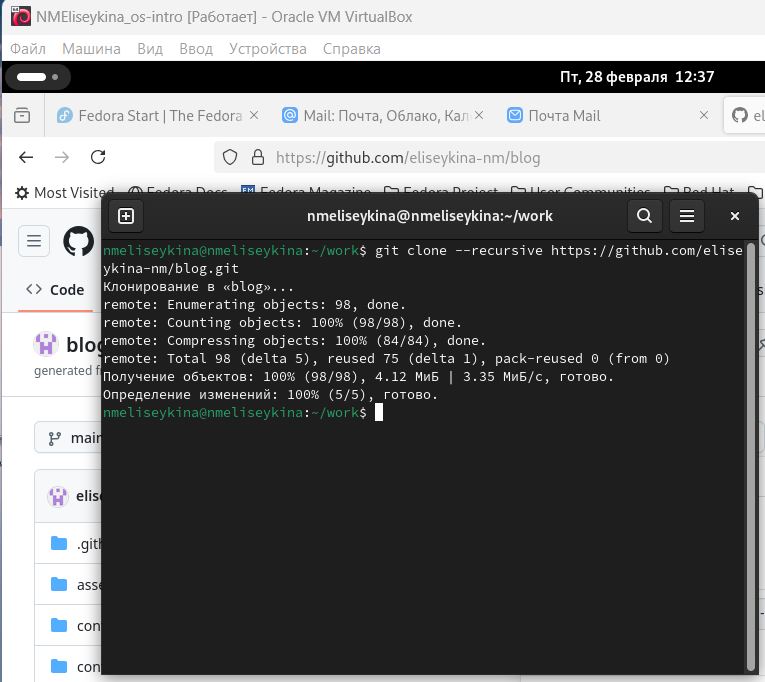
Создание репозитория

При создании даём ему имя blog:

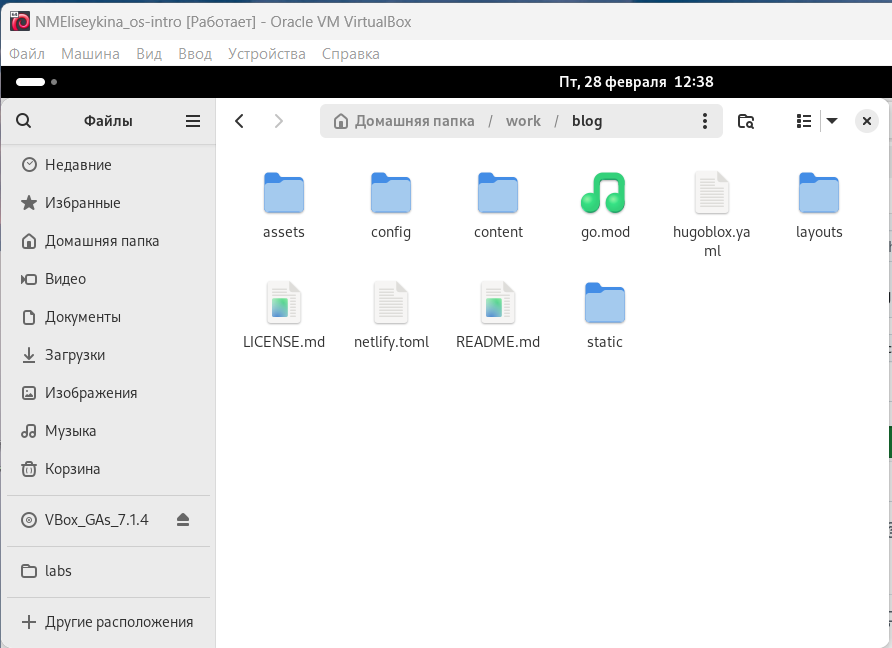


Создание репозитория

После клонируем данный репозиторий в путь */home/nmeliseykina/work*:

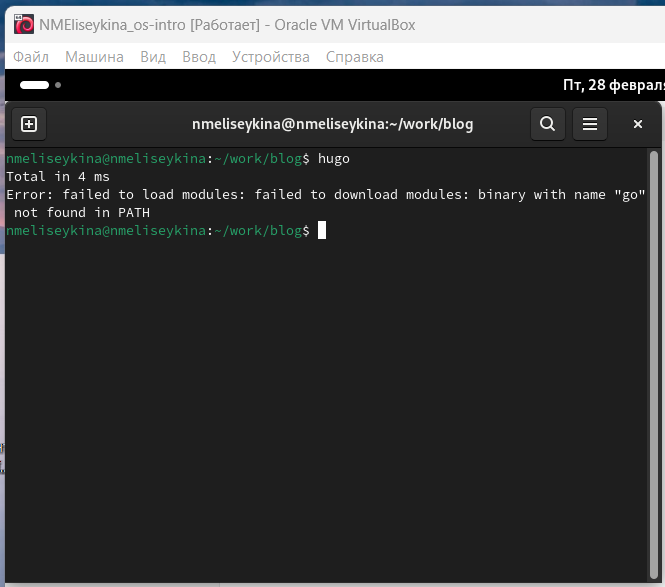


Клонирование репозитория



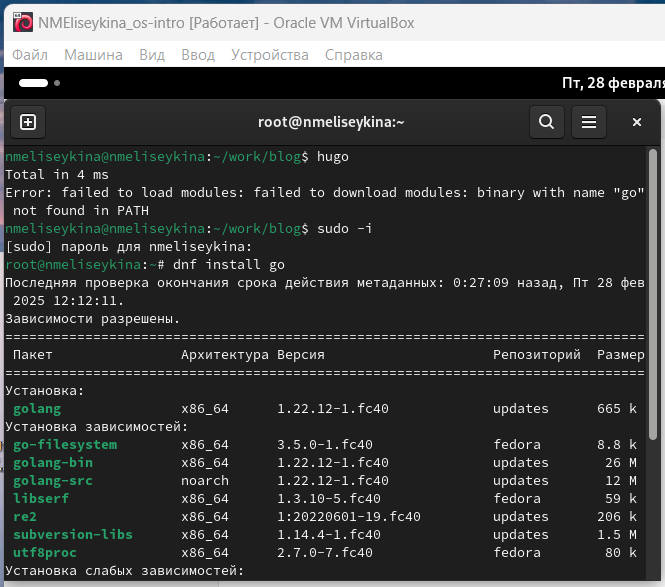
Проверка

Переходим в папку *blog* и запускаем *hugo*:



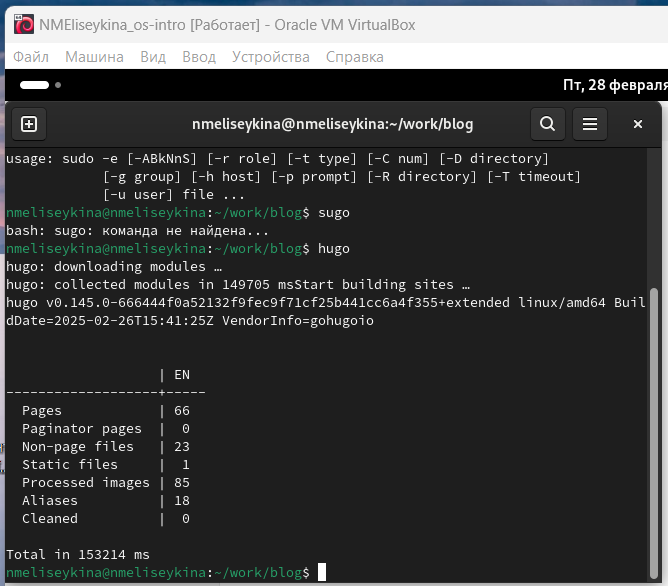
Hugo

Так как выдало ошибку, доустановим модуль go из-под суперпользователя:



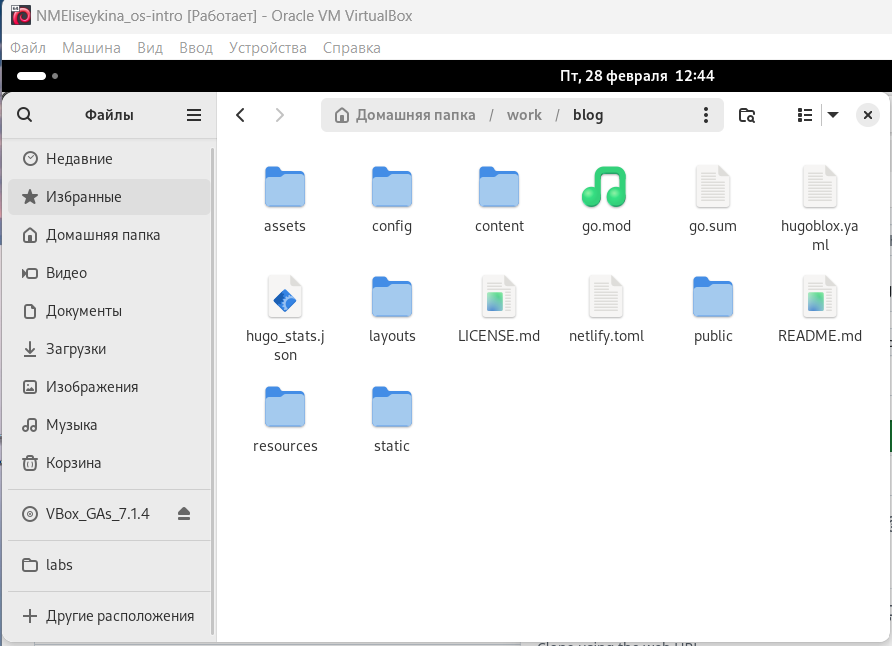
Доустановка модуля “GO”

Возвращаемся в папку blog и запускаем *hugo*:



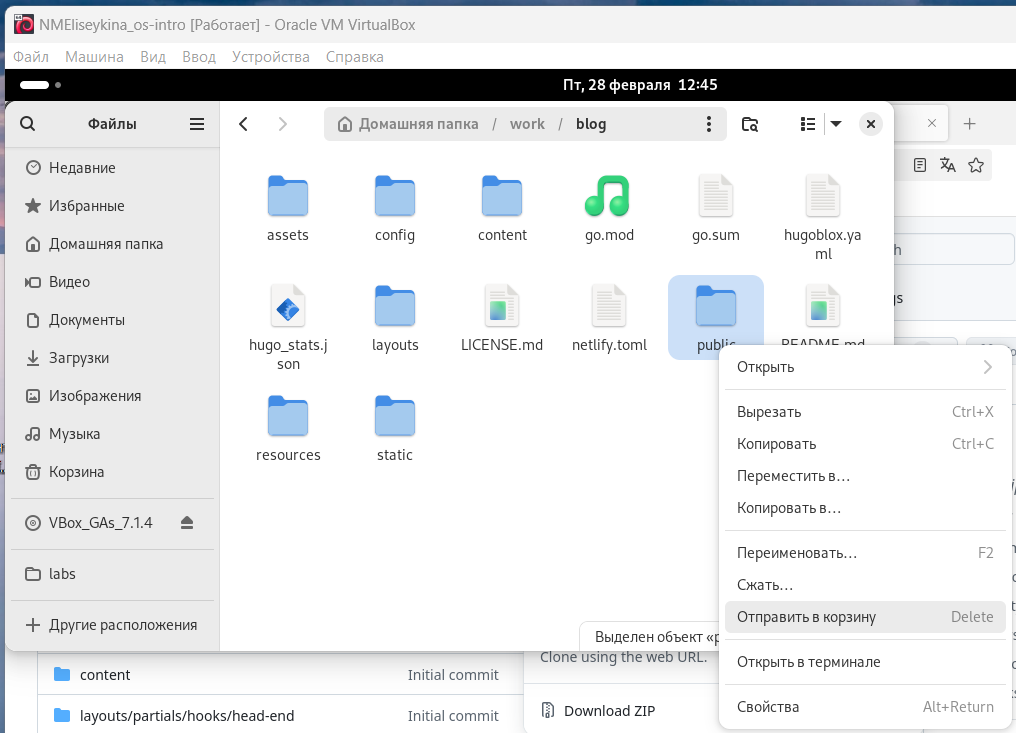
HUGO

После установки необходимых модулей проверяем создание папок и файлов:



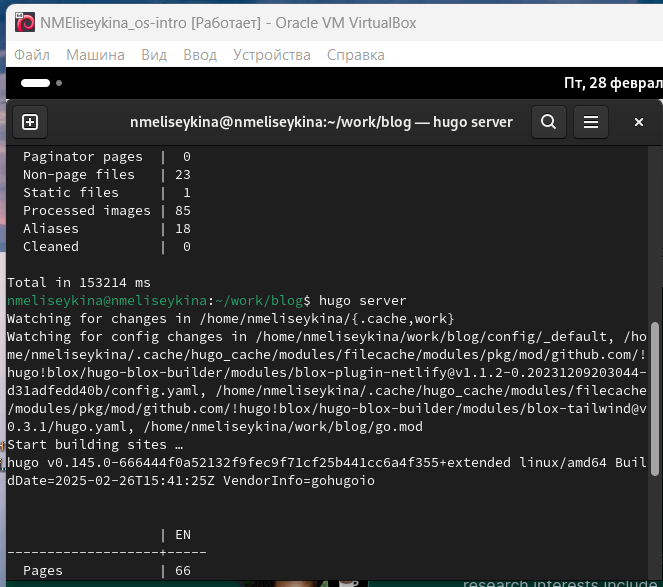
Проверка

Удаляем каталог public:

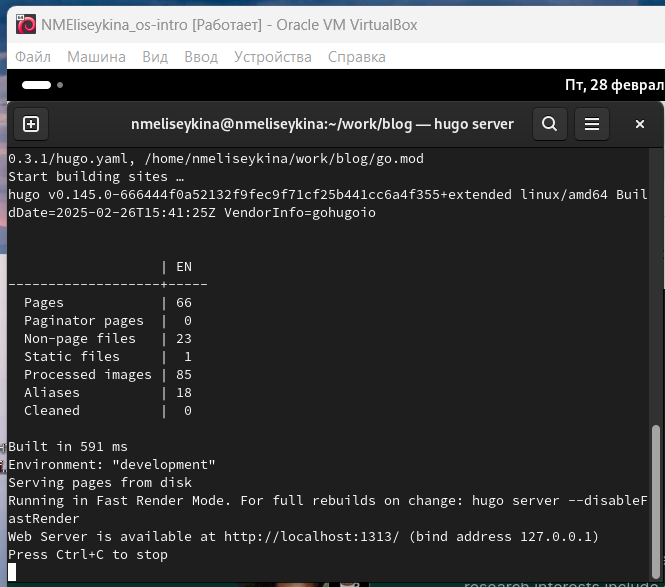


Удаляем public

Запускаем hugo server:

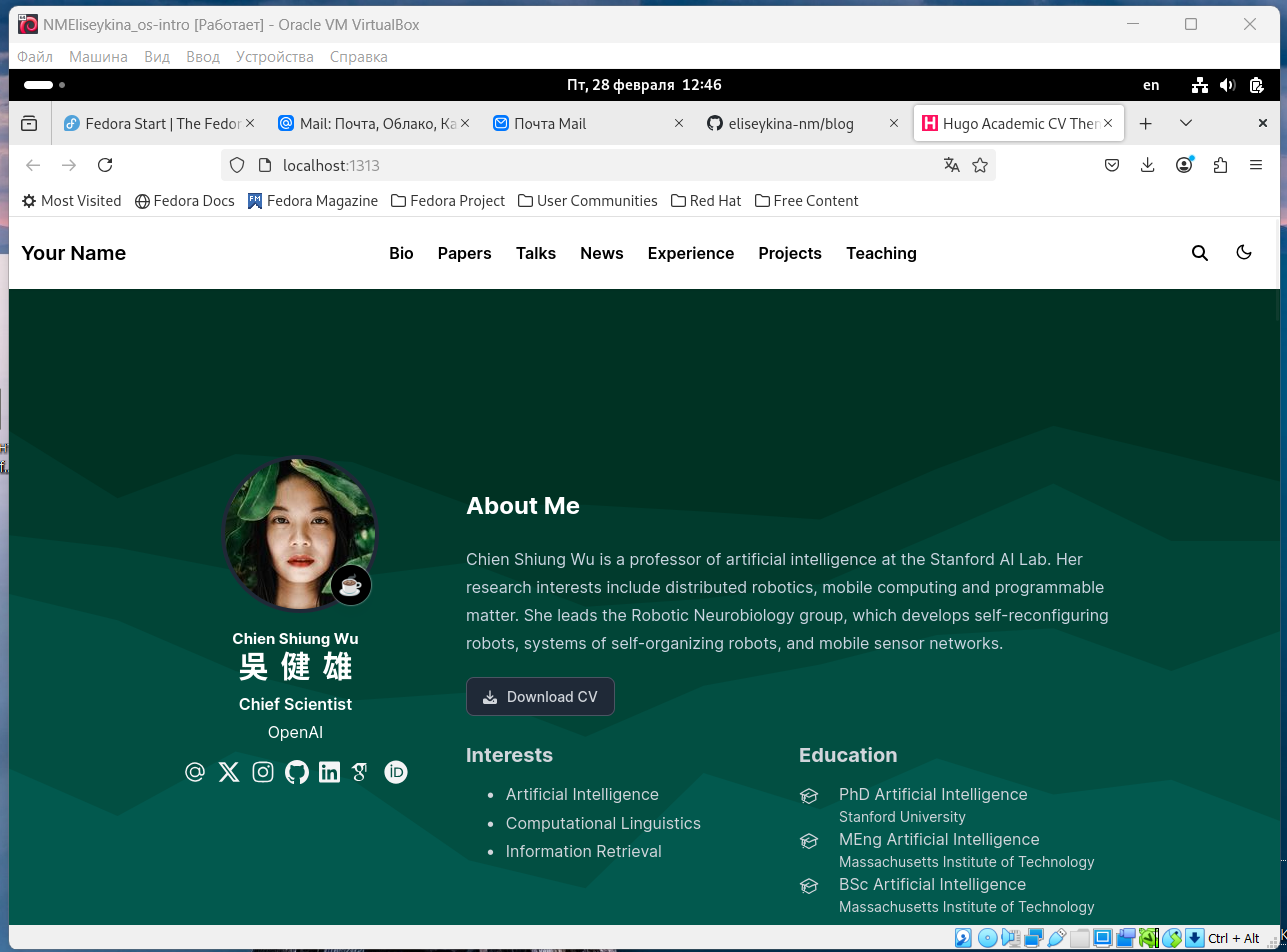


Запуск hugo server



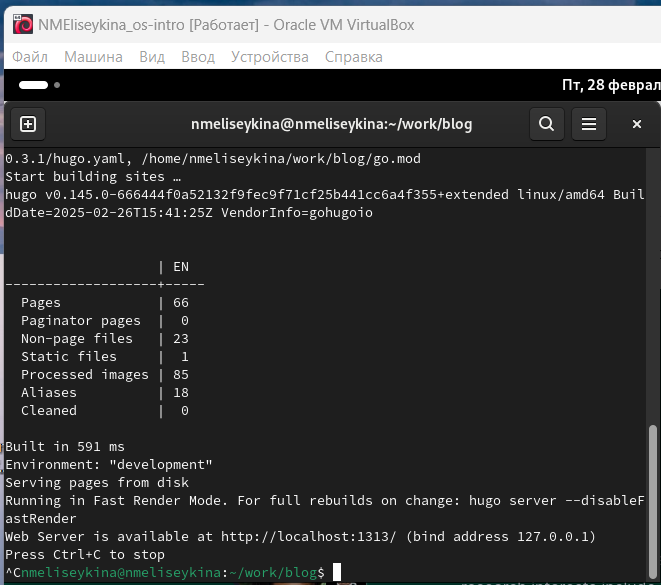
Запуск hugo server

Открываем ссылку в браузере и видим сайт:



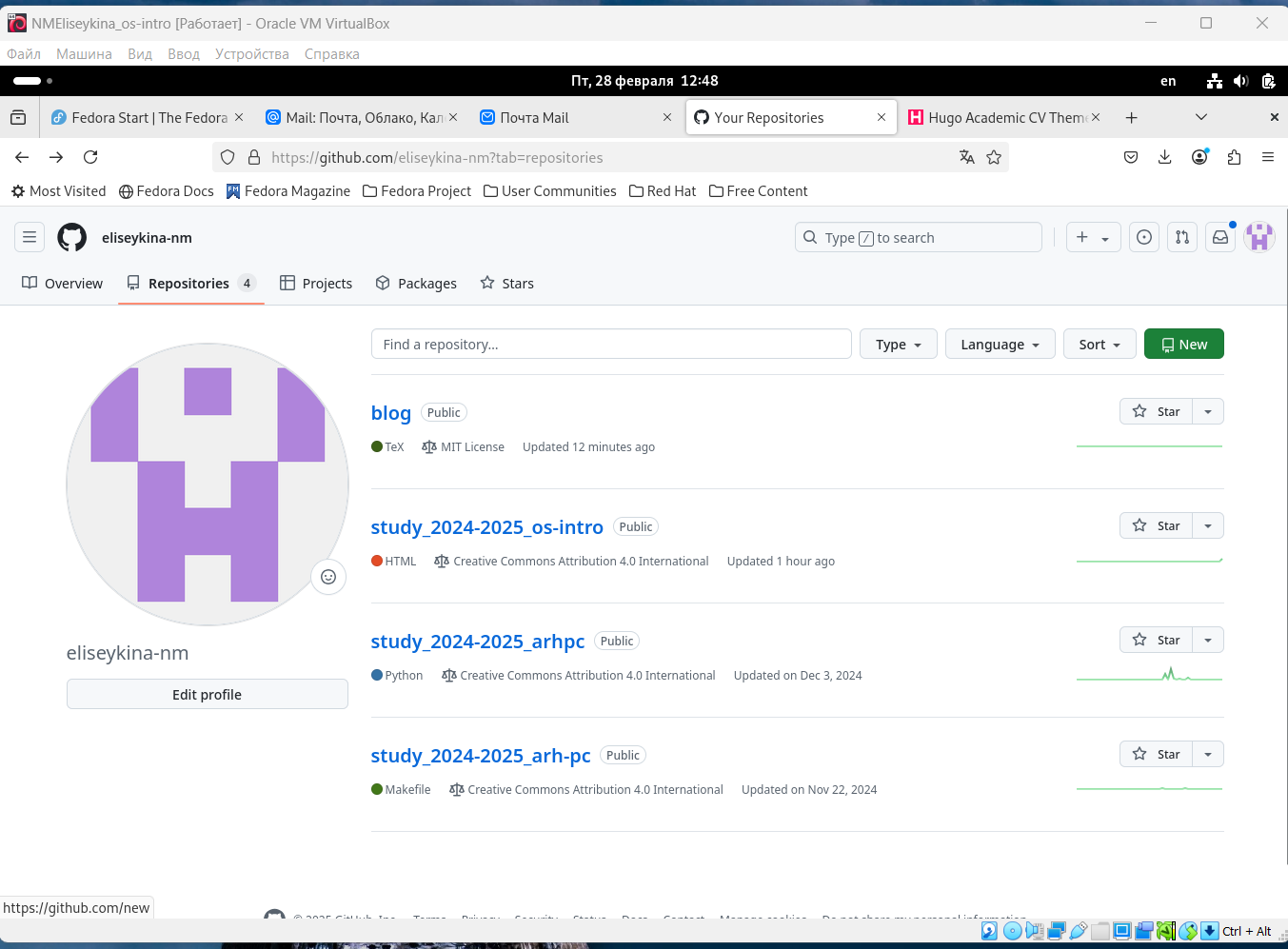
Шаблон сайта

После проверки в браузере закроем сервер:



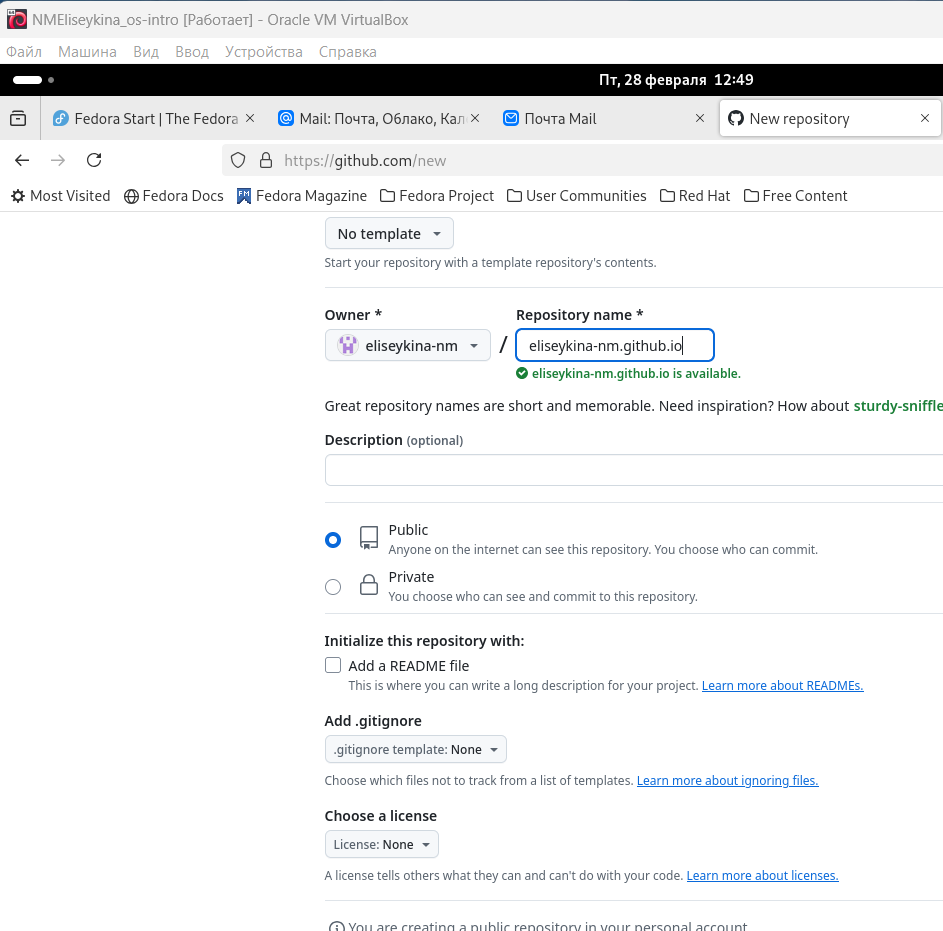
Закрытие hugo server

Создание нового репозитория:



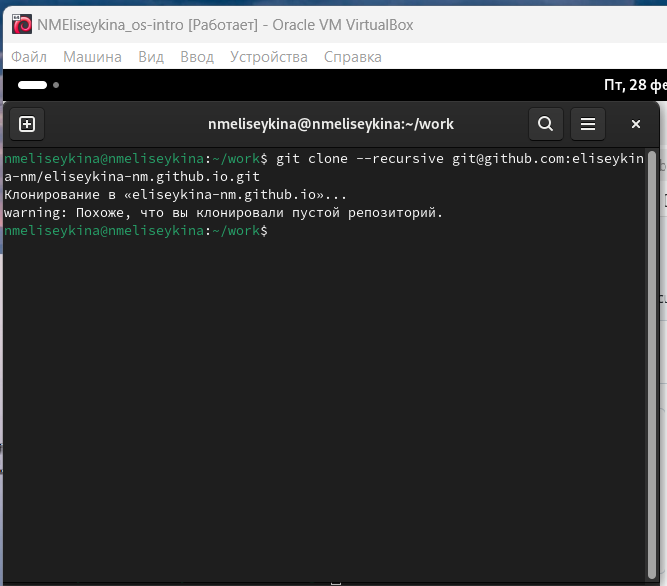
Github

Название репозитория должно полностью совпадать с именем владельца + github.io:

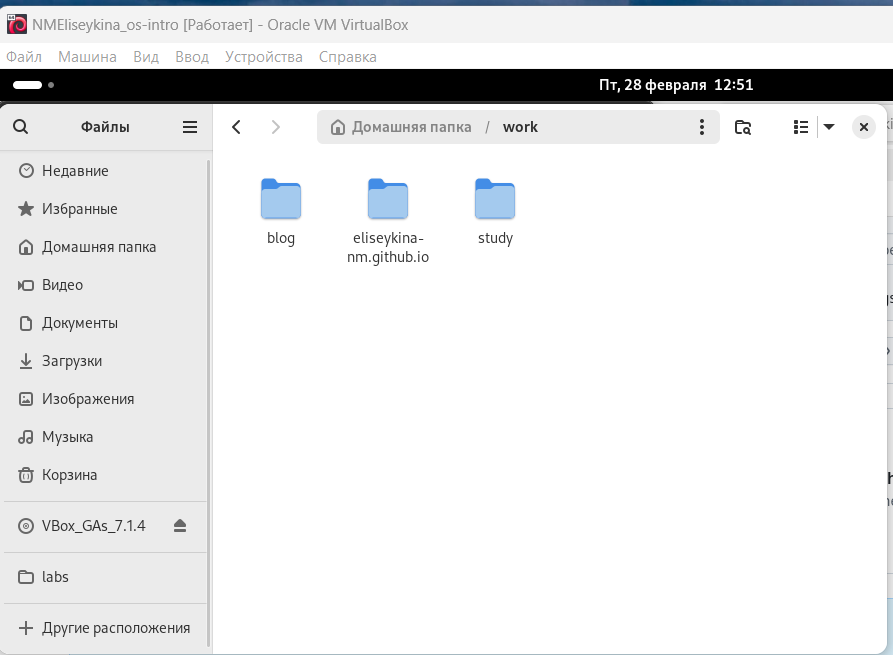


Создаем новый репозиторий

Возвращаемся в терминал, в папку work и клонируем туда наш репозиторий (свежесозданный):

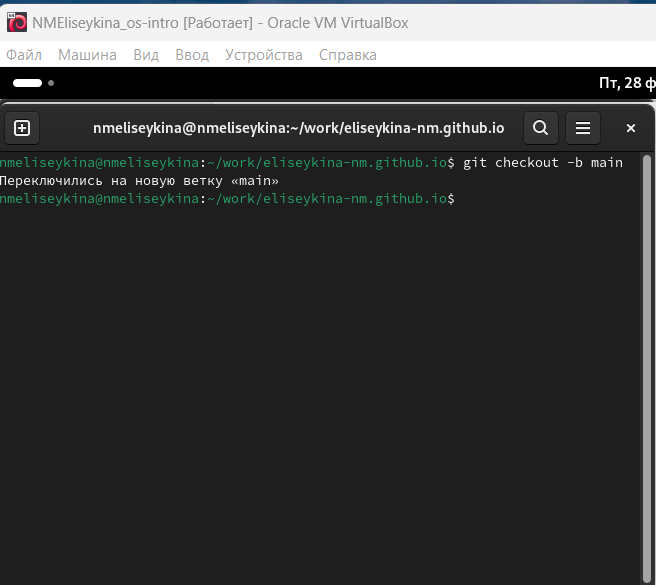


Клонирование репозитория



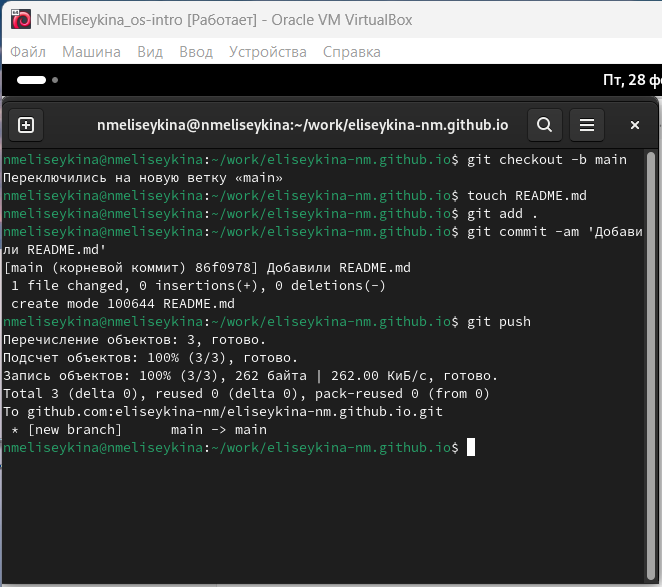
Проверка

Переключаемся на ветку “main”:

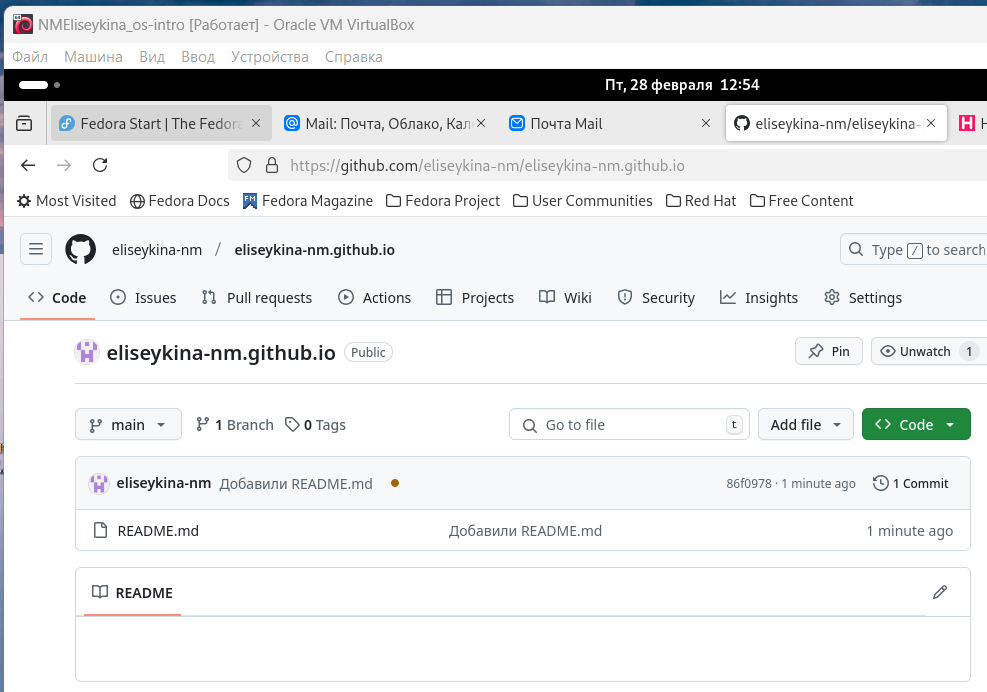


Переключаемся на ветку “main”

Создаем пустой файл README.md, а затем коммитим все изменения и отправляет на github:

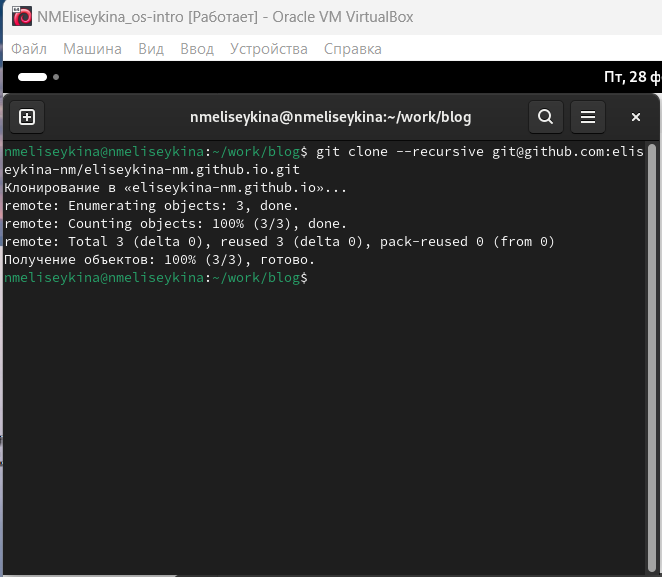


Создание пустого файла и отправка изменений



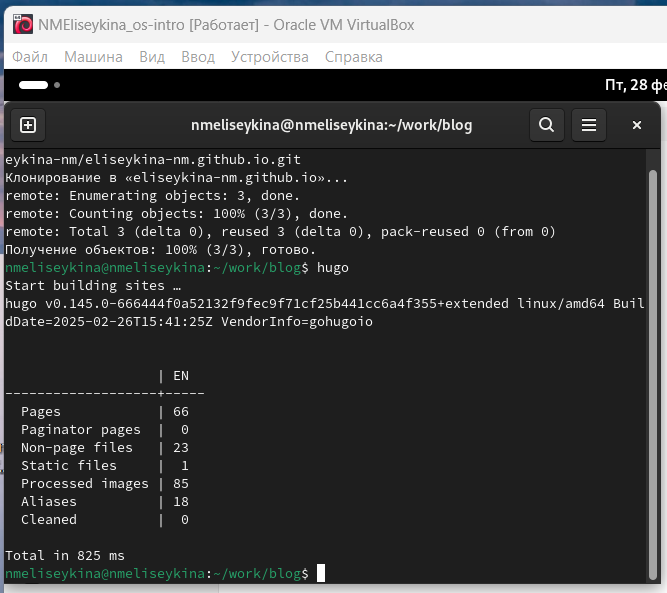
Проверка

Создаем ветку подмодуля, клонируя репозиторий с нашего Github:



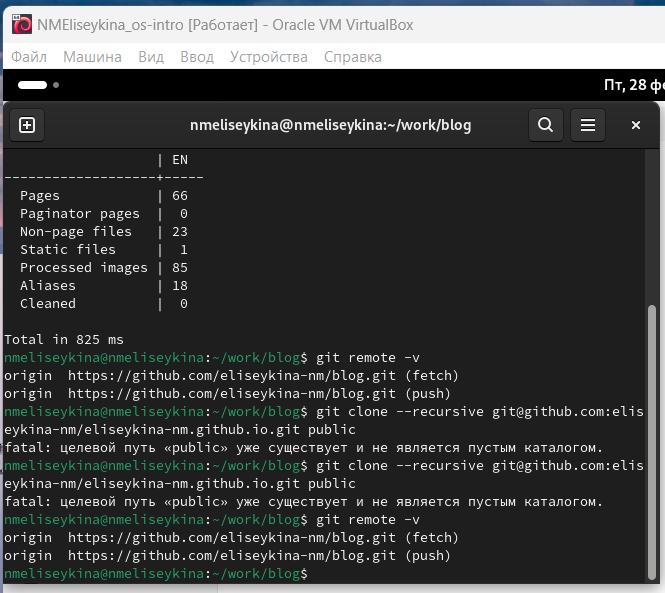
Создаем новую ветку

После выполнения запускаем hugo:



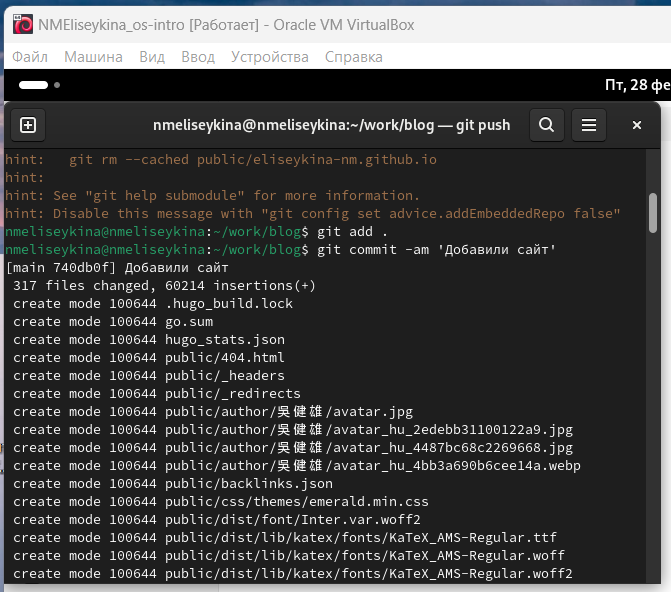
HUGO

Проверим подключение каталога к репозиторию командой git remote -v:

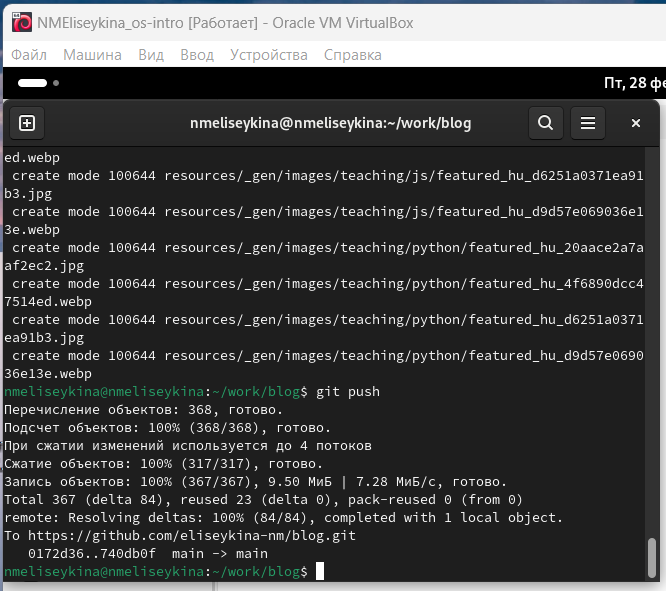


Проверка

Добавим изменения на github:

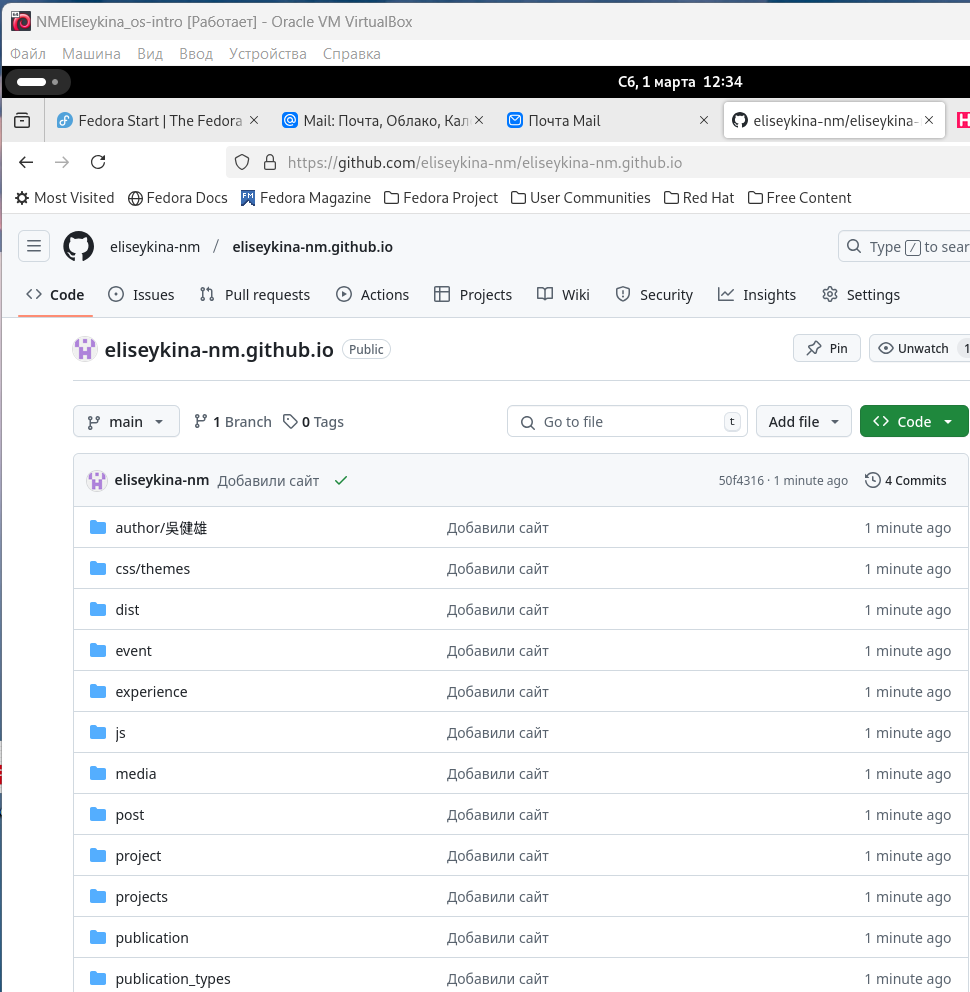


Загружаем обновления



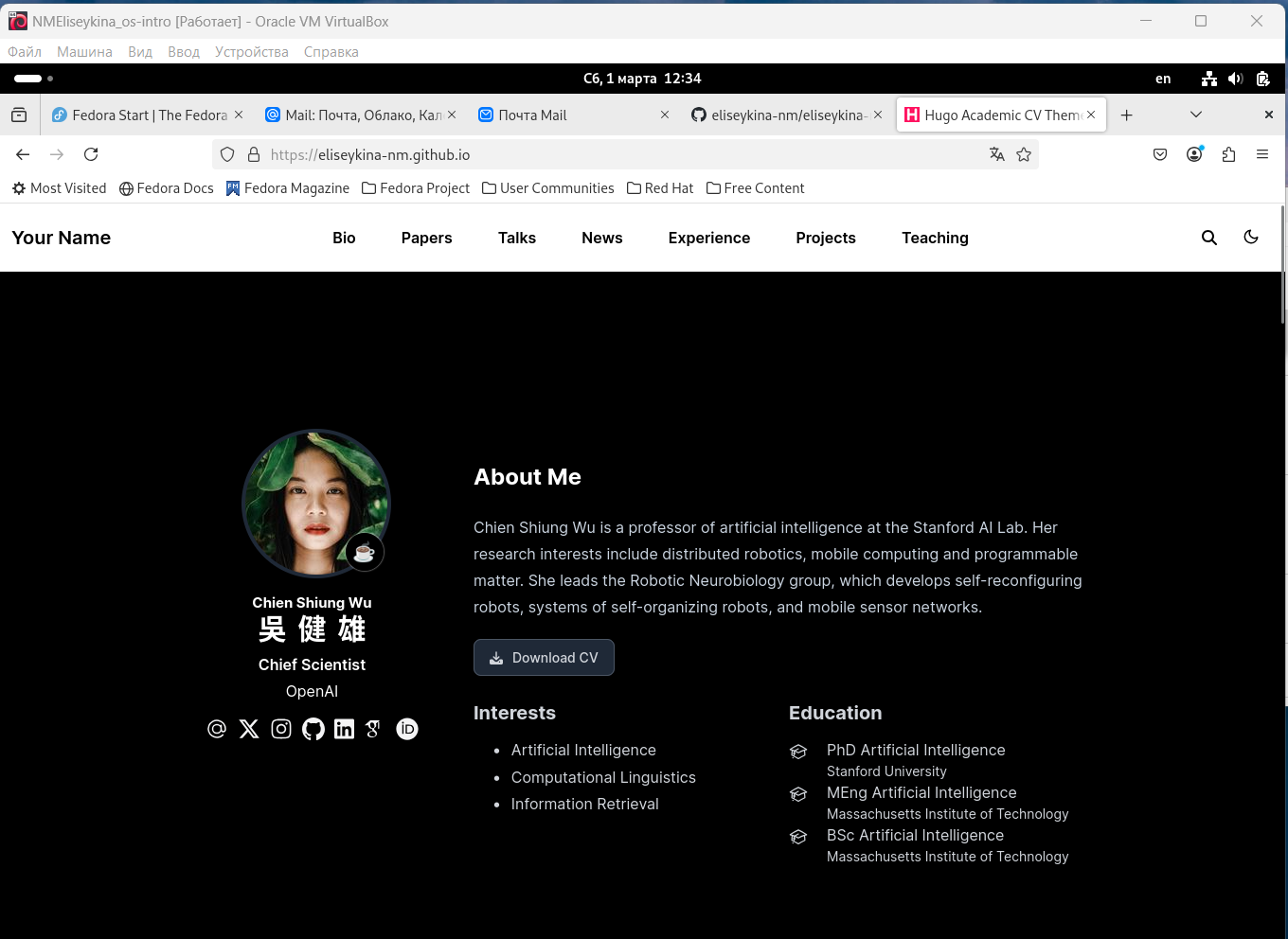
Загружаем обновления

Проверка обновлений:



Проверка github

Открываем наш сайт:



Шаблон сайта готовый

# 5 Выводы

В ходе данной работы я создала шаблон своего сайта, который в будущем буду дорабатывать, а также запрепила навыки работы с системой контроля версий Git.

# Список литературы

1. [Этапы реализации проекта](https://esystem.rudn.ru/mod/page/view.php?id=970806&forceview=1)
2. [Техническая реализация проекта](https://esystem.rudn.ru/mod/page/view.php?id=970807&forceview=1)
3. [Руководство по выполнению первого этапа индивидуального проекта](https://esystem.rudn.ru/mod/url/view.php?id=980904&forceview=1)
4. [Инструменты Git - Подмодули](https://git-scm.com/book/ru/v2/%D0%98%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D1%8B-Git-%D0%9F%D0%BE%D0%B4%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D0%B8)