Отчёт по индивидуальному проекту. Этап 1

Размещение на Github pages заготовки для персонального сайта

Головина Мария Игоревна

Содержание

# 1 Цель работы

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

# 2 Задание

1. Установить необходимое программное обеспечение.
2. Скачать шаблон темы сайта.
3. Разместить его на хостинге git.
4. Установить параметр для URLs сайта.
5. Разместить заготовку сайта на Github pages.

# 3 Теоретическое введение

Примеры использования git:

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

1. Загружаем последнюю версию hugo (рис. 4.1).



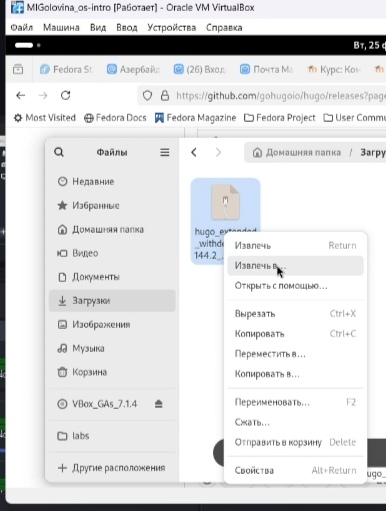
Загрузка нужной нам версии

1. Файл скачивается в папку “Загрузки” (рис. 4.2).

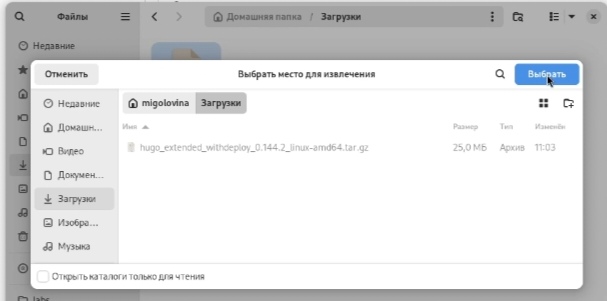


Скачивание

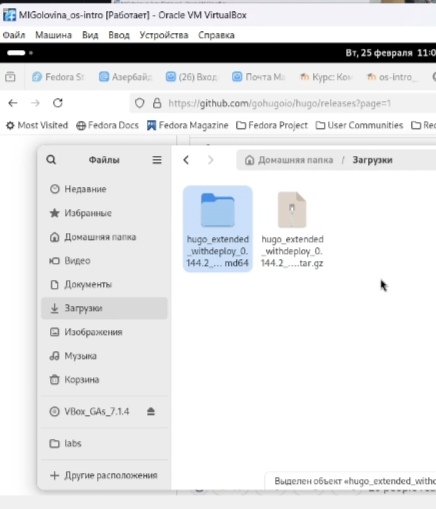
1. По завершении скачивания извлекаем архив в ту же папку, в которой мы находимся (рис. 4.3-4.5).



Извлечение

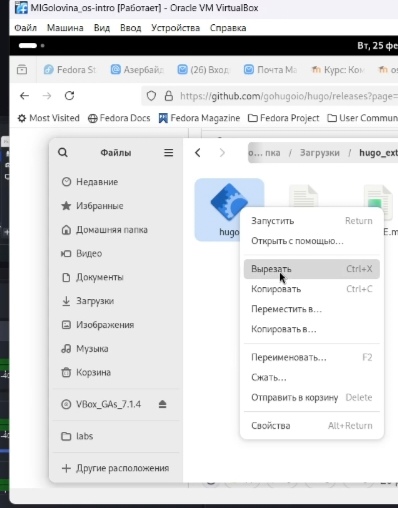


Извлечение - выбор папки

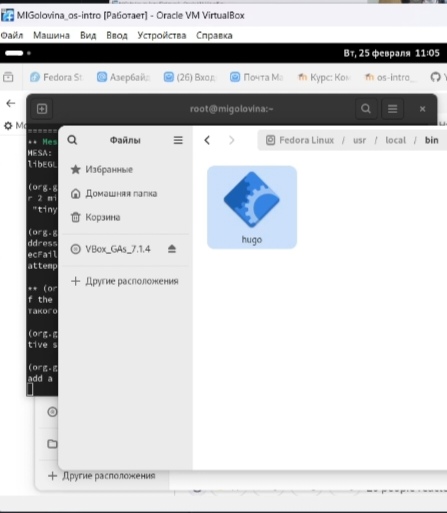


Проверка извлечения

1. После извлечения файла, нам его необходимо вырезать и вставить в папку /usr/local/bin (рис. 4.6-4.7).



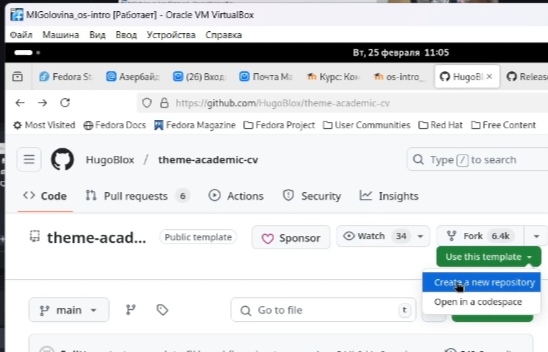
Вырезание файла



Перенос файла

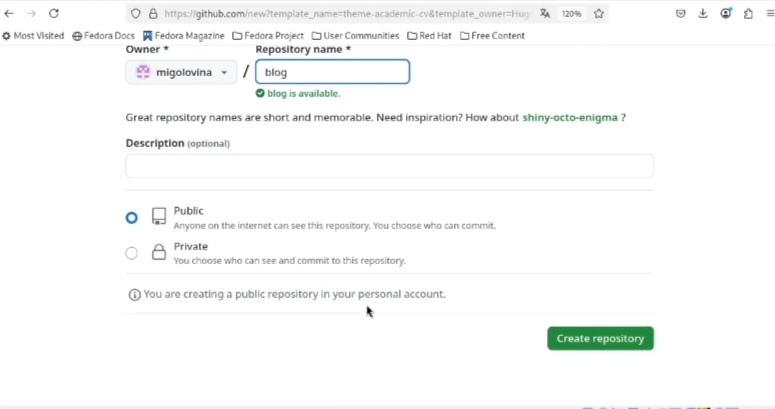
1. Далее открываем наш github и создаем репозиторий на основе данного нам (рис. 4.8).

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



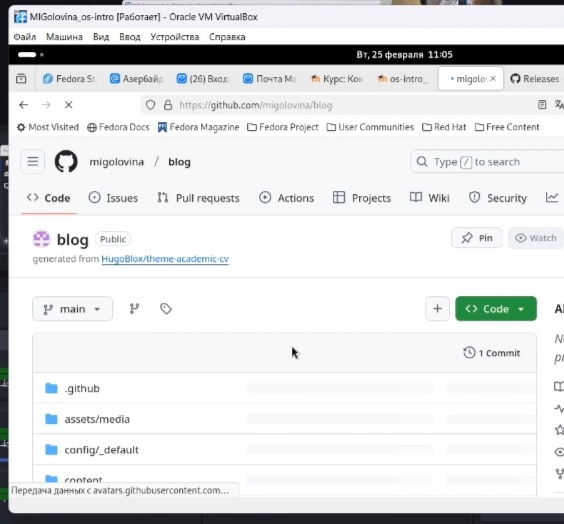
Создание репозитория

1. При создании даём ему имя blog (рис. 4.9).



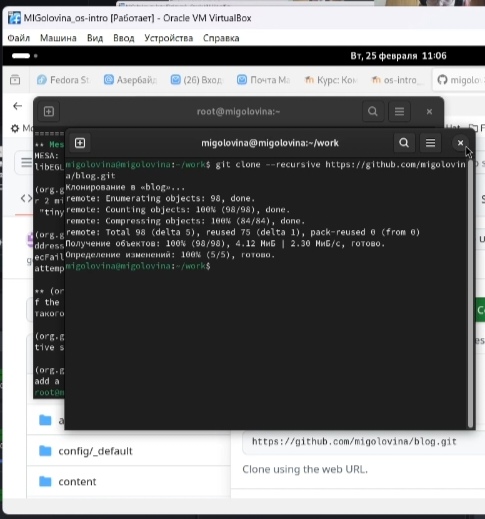
Создание репозитория

1. Проверка (рис. 4.10).

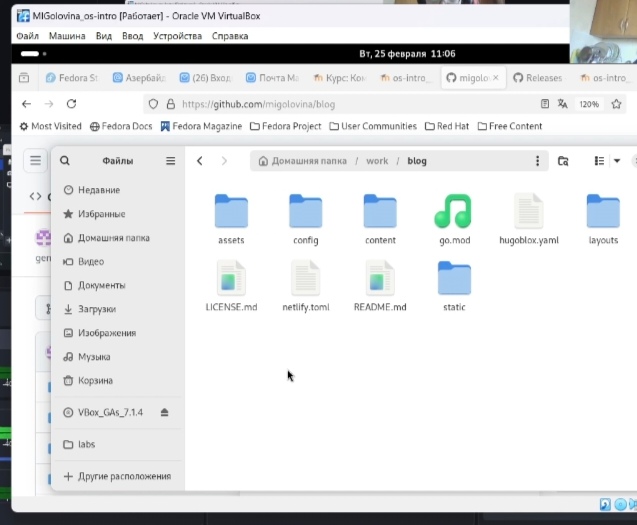


Проверка

1. После клонируем данный репозиторий в путь /home/migolovina/work (рис. 4.11-4.12).

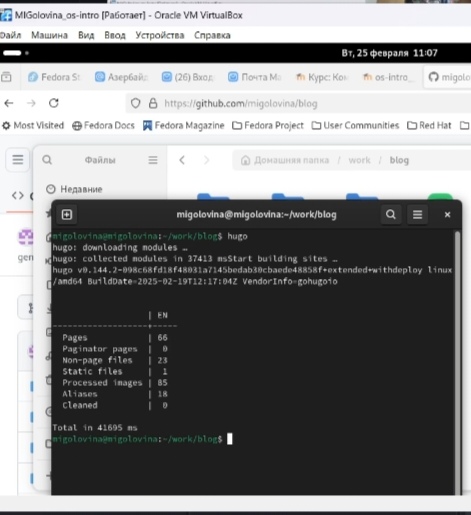


Клонирование репозитория



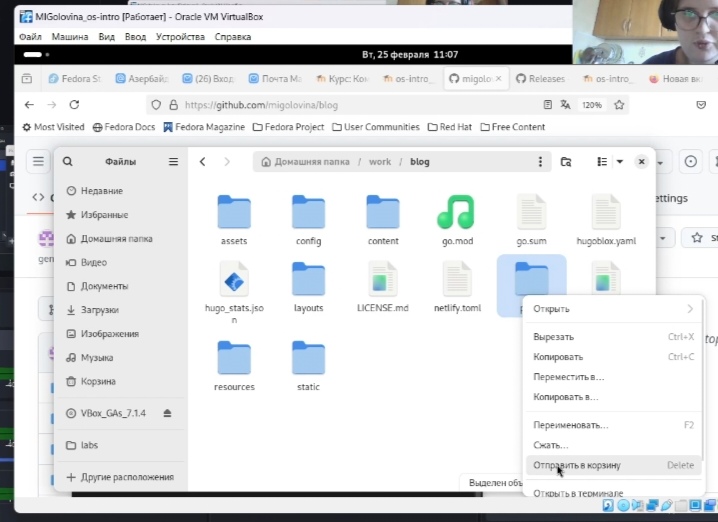
Проверка

1. Переходим в папку blog и запускаем hugo (рис. 4.13).



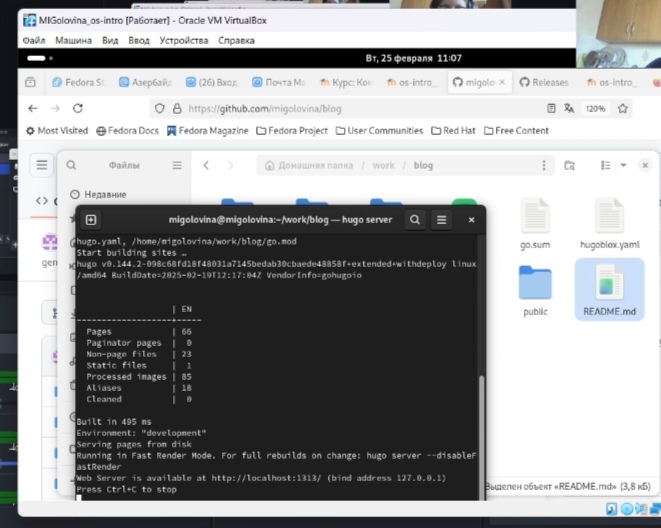
Hugo

1. После установки необходимых модулей проверяем создание папок и файлов и удаляем каталог public (рис. 4.14).



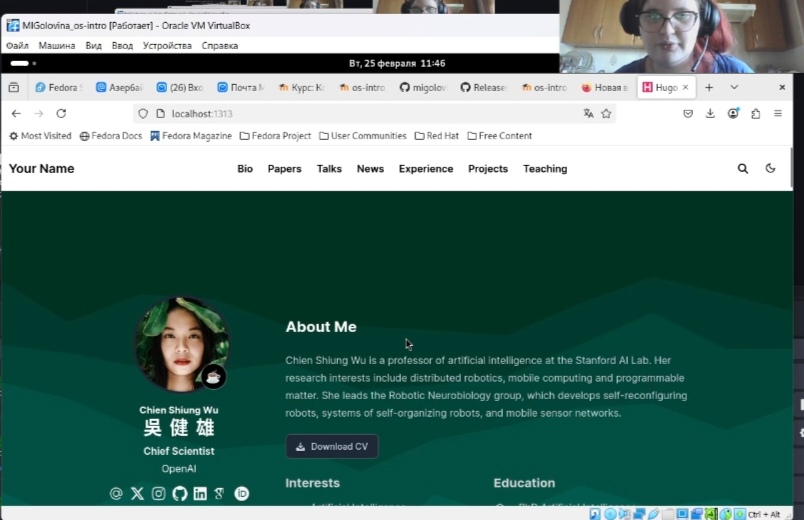
Проверка

1. Запускаем hugo server (рис. 4.15).



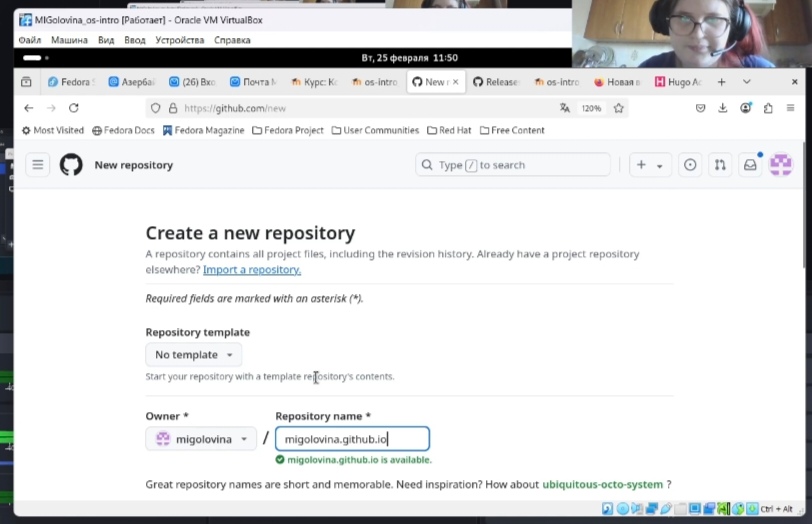
Запуск hugo server

1. Открываем ссылку в браузере и видим сайт (рис. 4.16).



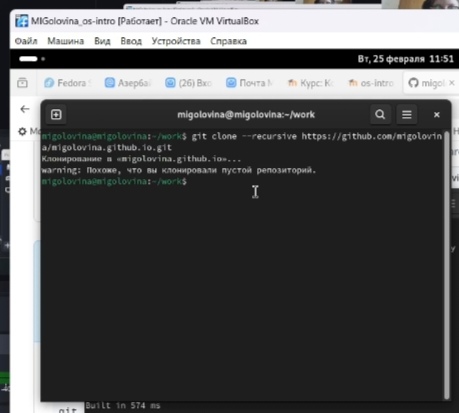
Шаблон сайта

1. Создание нового репозитория. Название репозитория должно полностью совпадать с именем владельца + github.io (рис. 4.17).

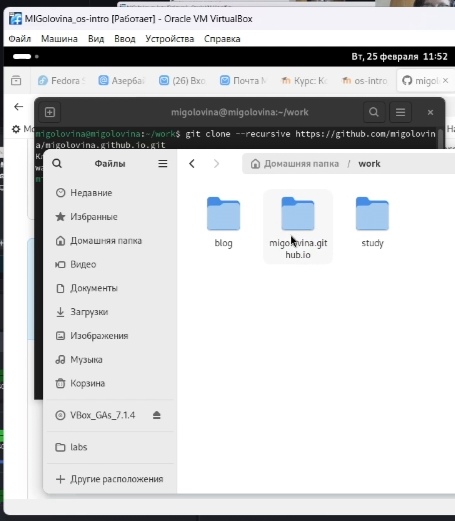


Github

1. Возвращаемся в терминал, в папку work и клонируем туда наш репозиторий (свежесозданный) и проверяю. (рис. 4.18-4.19).

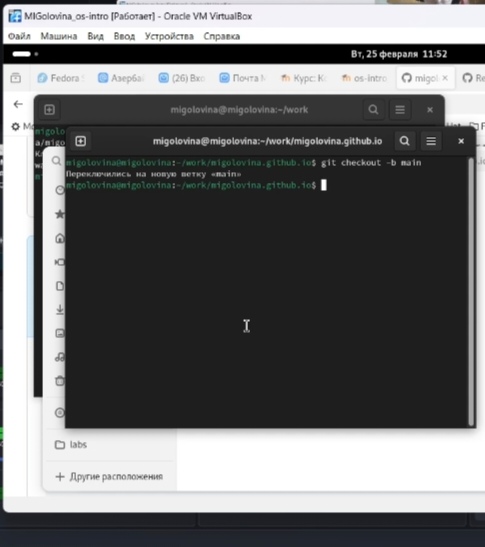


Клонирование репозитория



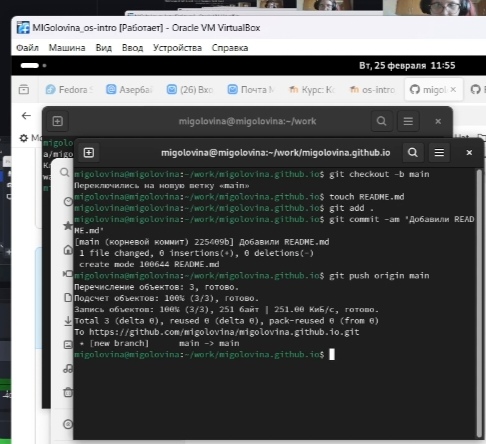
Проверка

1. Переключаемся на ветку “main” (рис. 4.20).

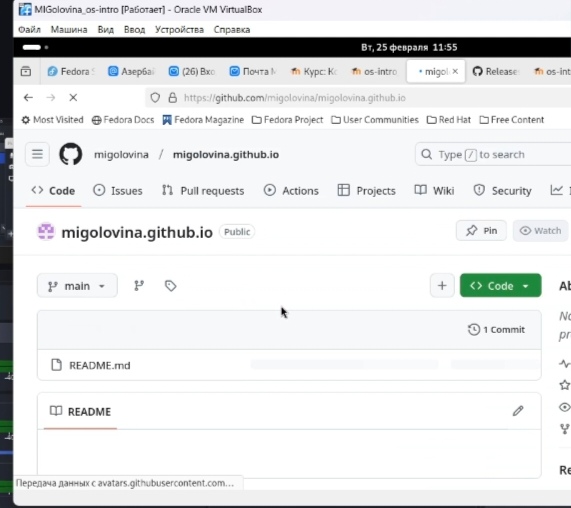


Переключаемся на ветку “main”

1. Создаем пустой файл README.md, а затем коммитим все изменения и отправляет на github. (рис. 4.21-4.22).

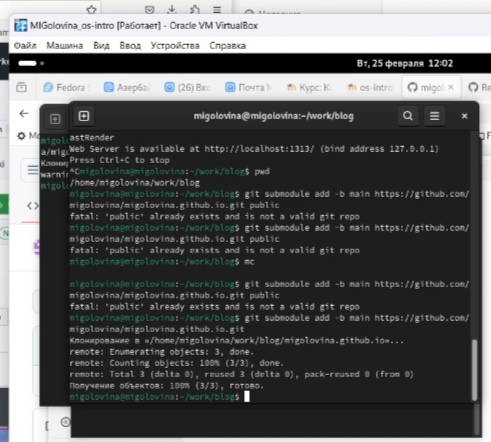


Создание пустого файла и отправка изменений



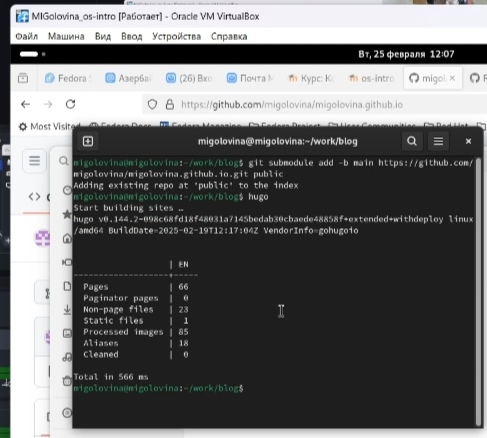
Проверка

1. Создаем ветку подмодуля, клонируя репозиторий с нашего Github (рис. 4.23).



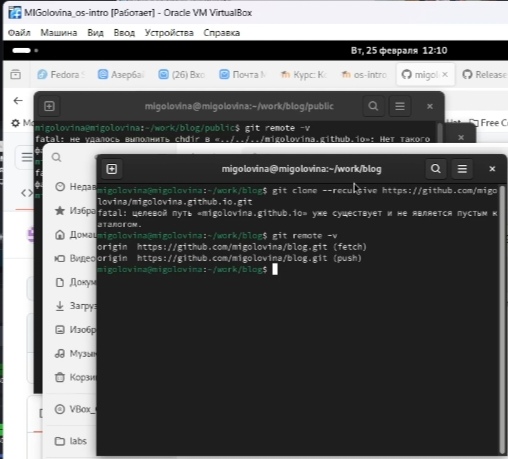
Создаем новую ветку

1. После выполнения запускаем hugo (рис. 4.24).



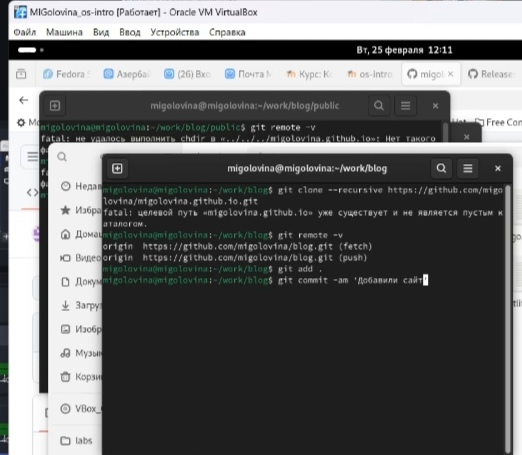
HUGO

1. Проверим подключение каталога к репозиторию командой git remote -v (рис. 4.25).

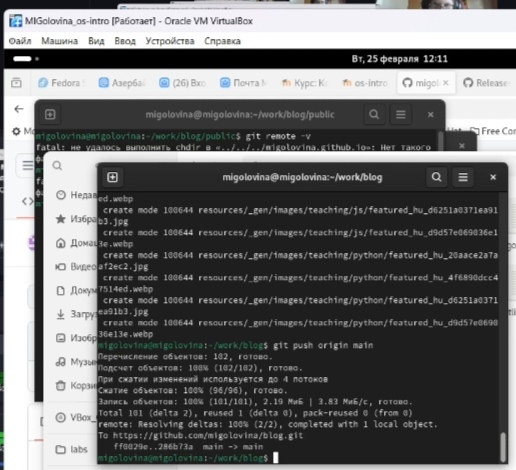


Проверка

1. Добавим изменения на github (рис. 4.26-4.27).

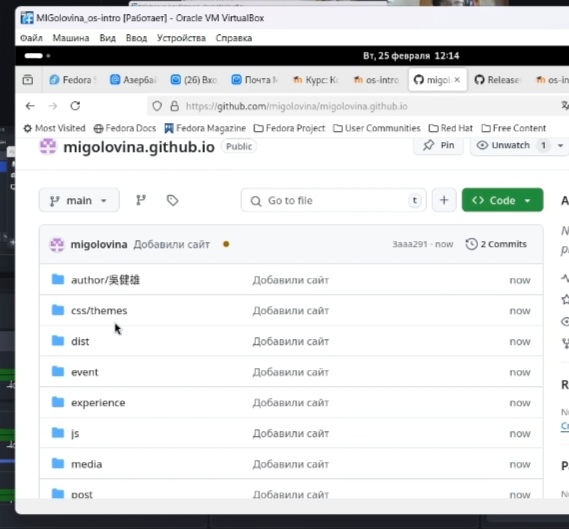


Загружаем обновления



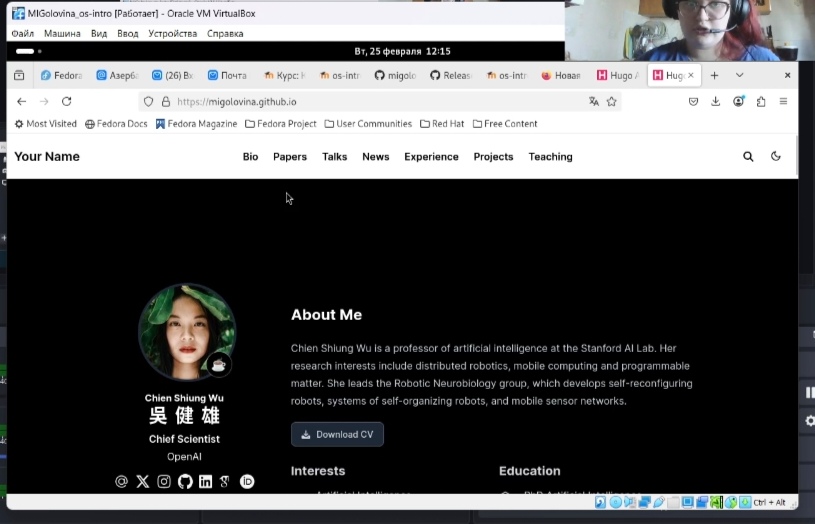
Загружаем обновления

1. Проверка обновлений (рис. 4.28).



Проверка github

1. Открываем наш сайт (рис. 4.29).



Шаблон сайта готовый

# 5 Выводы

В ходе данной работы я создала шаблон своего сайта, который в будущем буду дорабатывать, а также закрепила 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)