

Отчёт по индивидуальному проекту.

Этап 1

Размещение на Github pages заготовки для персонального сайта

Головина Мария Игоревна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение индивидуального проекта	9
5	Выводы	32
	Список литературы	33

Список иллюстраций

4.1	Загрузка нужной нам версии	9
4.2	Скачивание	9
4.3	Извлечение	10
4.4	Извлечение - выбор папки	11
4.5	Проверка извлечения	11
4.6	Вырезание файла	12
4.7	Перенос файла	13
4.8	Создание репозитория	14
4.9	Создание репозитория	14
4.10	Проверка	15
4.11	Клонирование репозитория	16
4.12	Проверка	17
4.13	Hugo	18
4.14	Проверка	19
4.15	Запуск hugo server	19
4.16	Шаблон сайта	20
4.17	Github	20
4.18	Клонирование репозитория	21
4.19	Проверка	22
4.20	Переключаемся на ветку “main”	23
4.21	Создание пустого файла и отправка изменений	24
4.22	Проверка	25
4.23	Создаем новую ветку	26
4.24	HUGO	27
4.25	Проверка	28
4.26	Загружаем обновления	29
4.27	Загружаем обновления	30
4.28	Проверка github	31
4.29	Шаблон сайта готовый	31

Список таблиц

3.1	Описание некоторых команд системы контроля версий Git	7
-----	---	---

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. 3.1 приведено краткое описание основных команд Git.

Таблица 3.1: Описание некоторых команд системы контроля версий Git

Команда	Описание команды
<code>git init</code>	Создание основного дерева репозитория
<code>git pull</code>	Получение обновлений(изменений текущего дерева из центрального репозитория
<code>git push</code>	Отправка всех произведённых изменений локального дерева в центральный репозиторий
<code>git status</code>	Просмотр списка изменённых файлов в текущей директории
<code>git diff</code>	Просмотр текущих изменений
<code>git add .</code>	Добавление все изменённые и/или созданные файлы и/или каталоги

Команда	Описание команды
git rm имя_файлов	Удаление файлов и/или каталогов из индекса репозитория
git commit -am	Сохранение всех добавленных изменений и всех изменённых файлов
‘Описание коммита’	
git commit	Сохранение добавленных 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://github.com)

4 Выполнение индивидуального проекта

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

 hugo_extended_withdeploy_0.144.2_Linux-64bit.tar.gz	23.9 MB	last week
 hugo_extended_withdeploy_0.144.2_linux-amd64.deb	24.9 MB	last week
 hugo_extended_withdeploy_0.144.2_linux-amd64.tar.gz	23.9 MB	last week
 hugo_extended_withdeploy_0.144.2_linux-arm64.deb	22.8 MB	last week
 hugo_extended_withdeploy_0.144.2_linux-arm64.tar.gz	22 MB	last week

Рис. 4.1: Загрузка нужной нам версии

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



Рис. 4.2: Скачивание

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

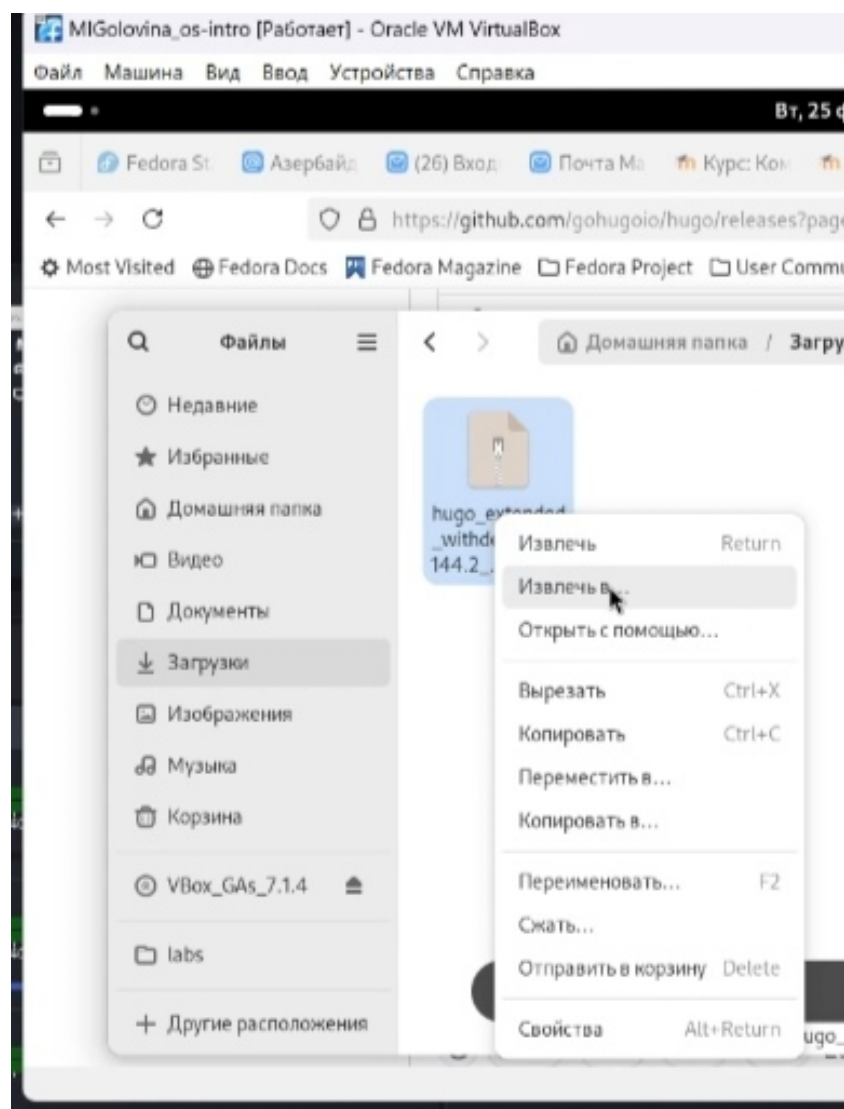


Рис. 4.3: Извлечение

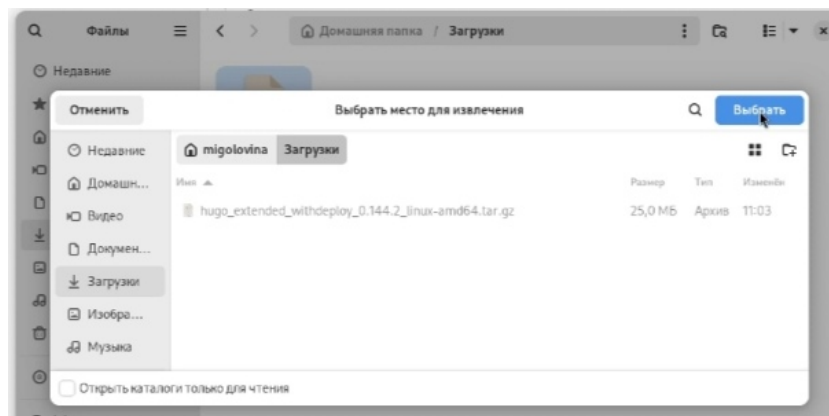


Рис. 4.4: Извлечение - выбор папки

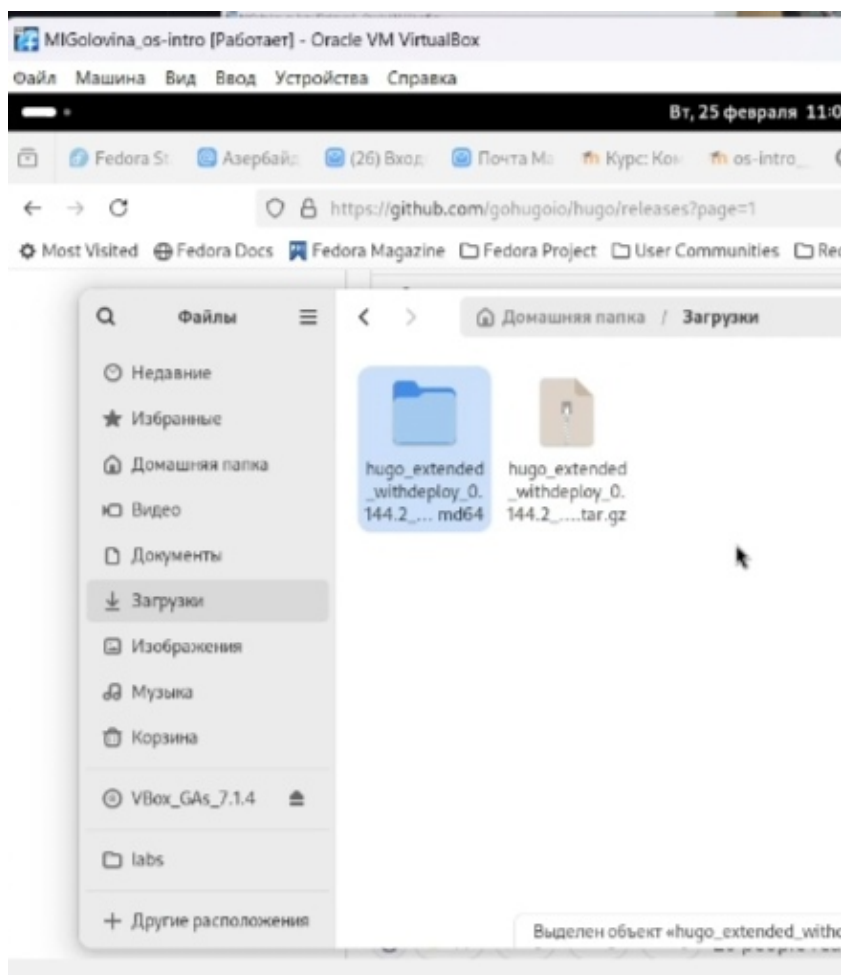


Рис. 4.5: Проверка извлечения

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

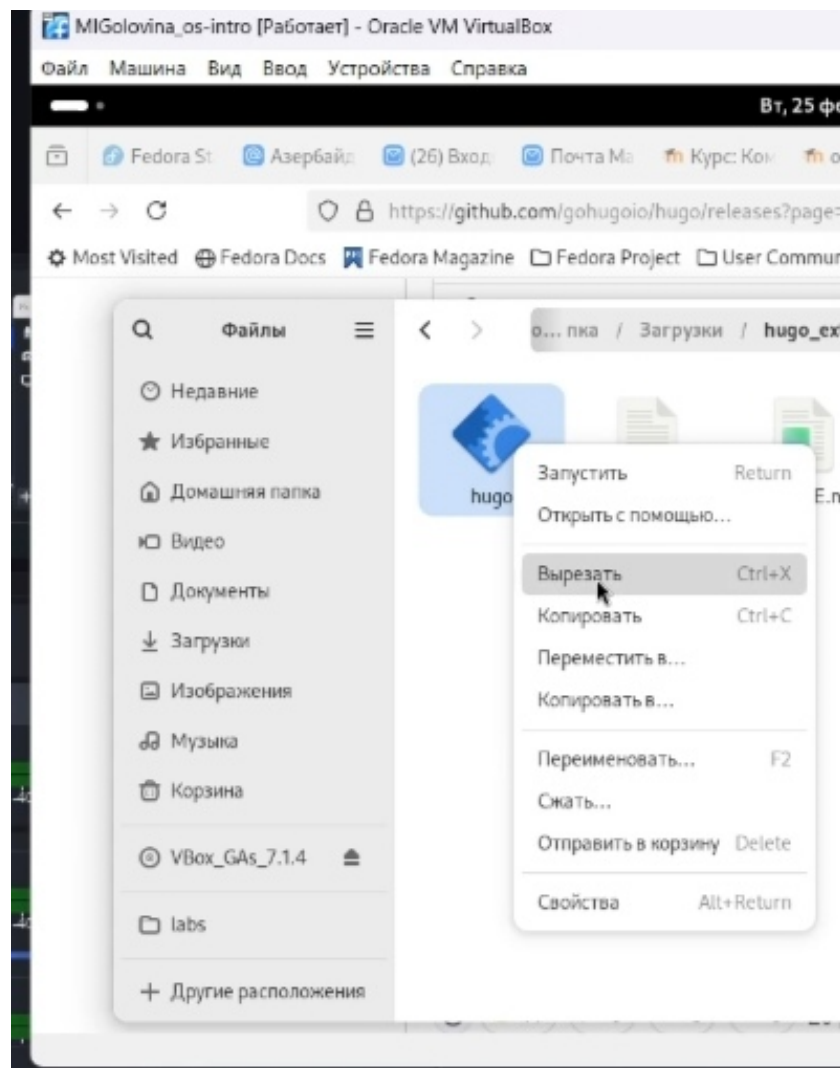


Рис. 4.6: Вырезание файла

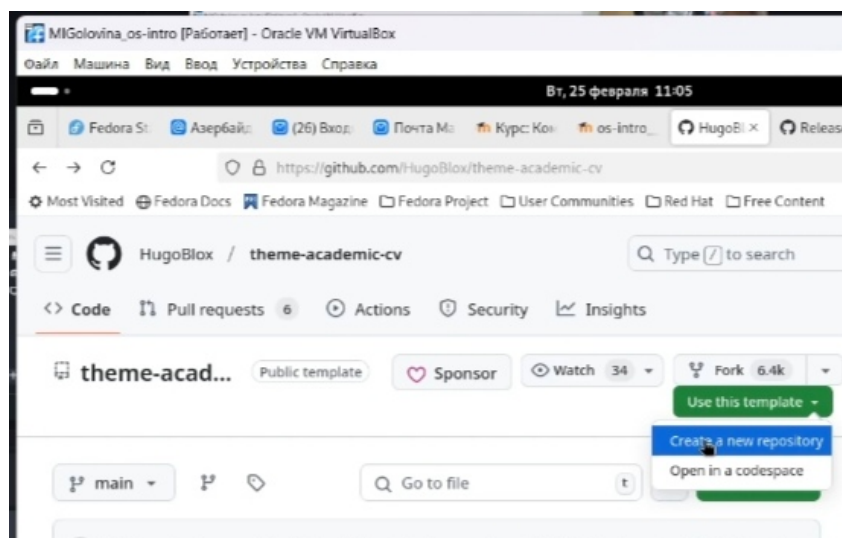


Рис. 4.8: Создание репозитория

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

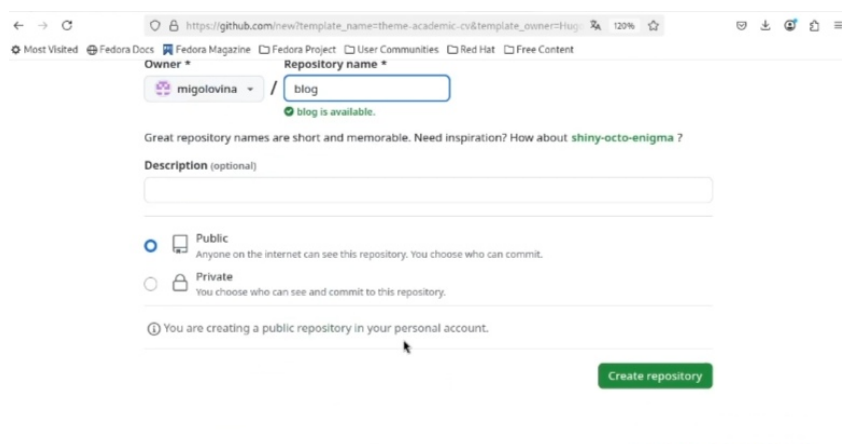


Рис. 4.9: Создание репозитория

7. Проверка (рис. 4.10).

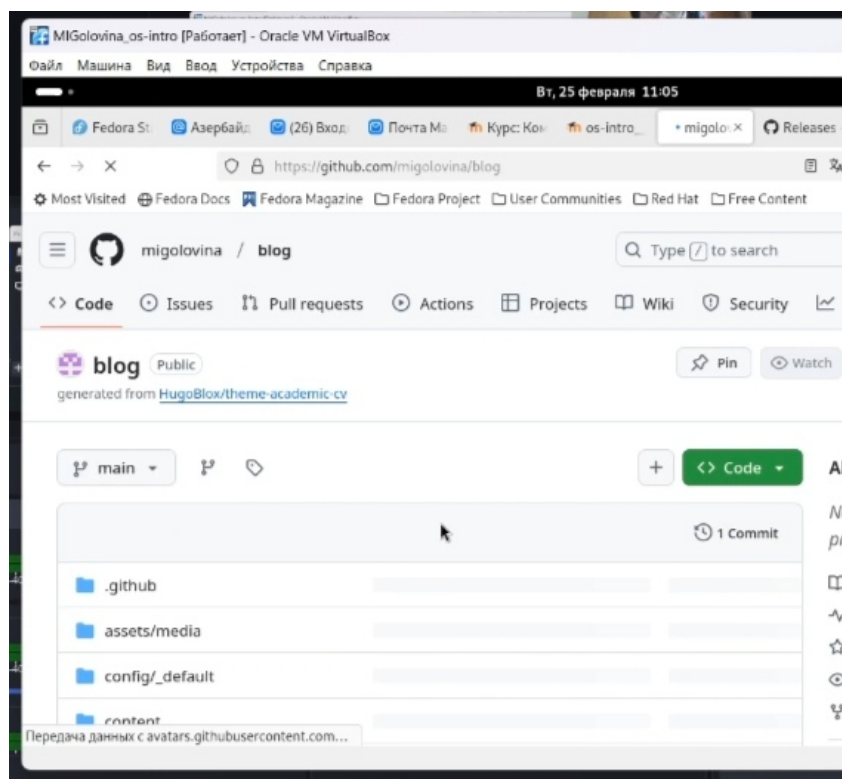


Рис. 4.10: Проверка

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

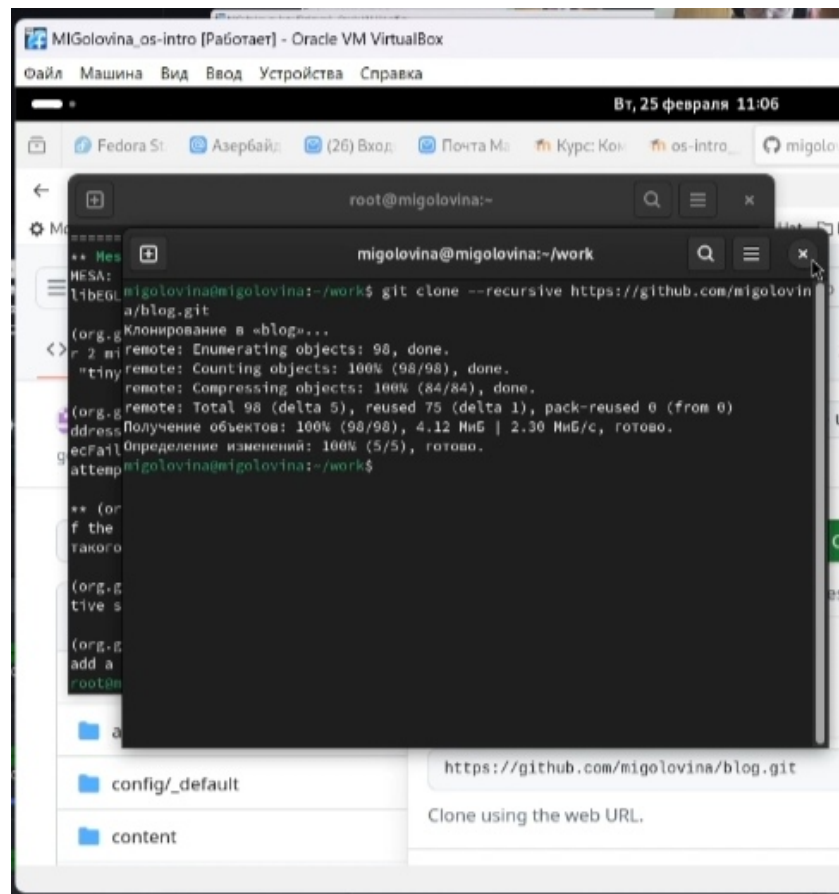


Рис. 4.11: Клонирование репозитория

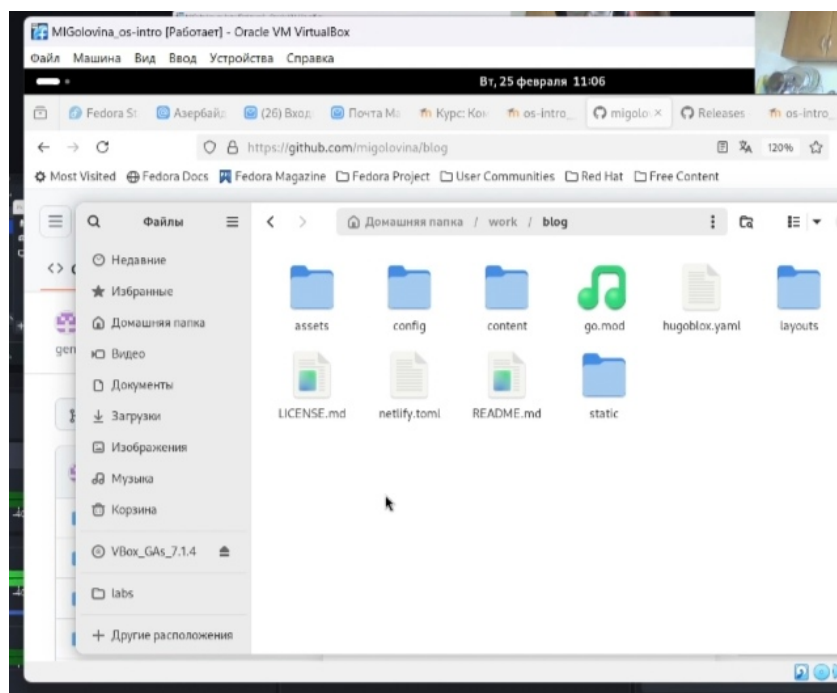


Рис. 4.12: Проверка

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

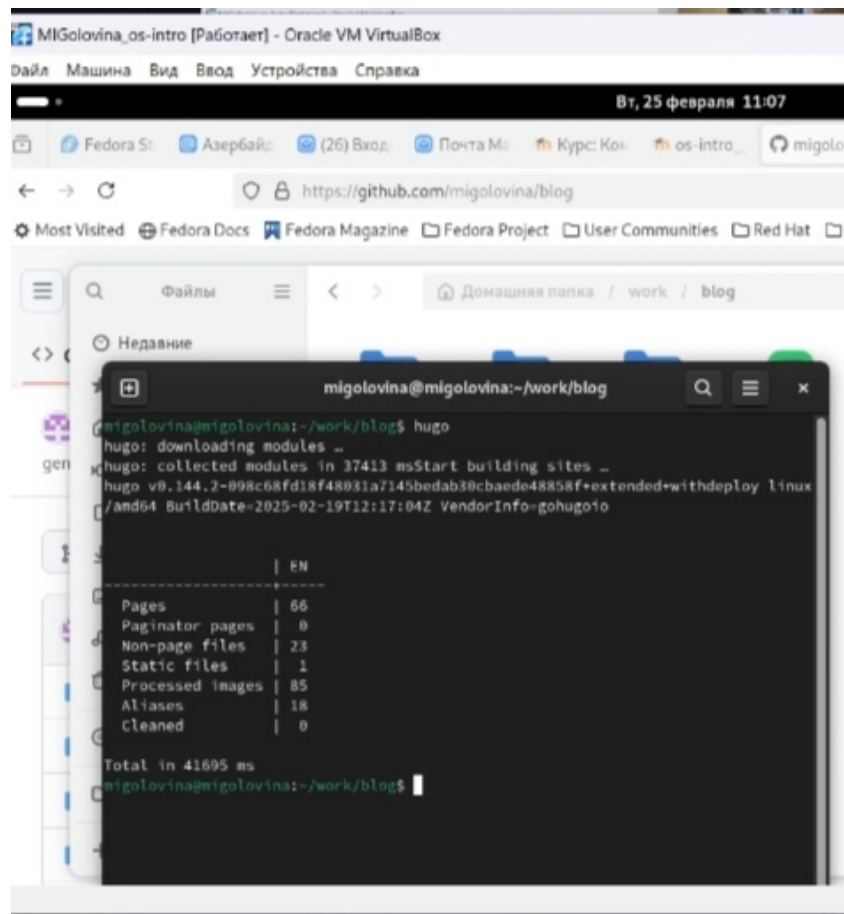


Рис. 4.13: Hugo

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

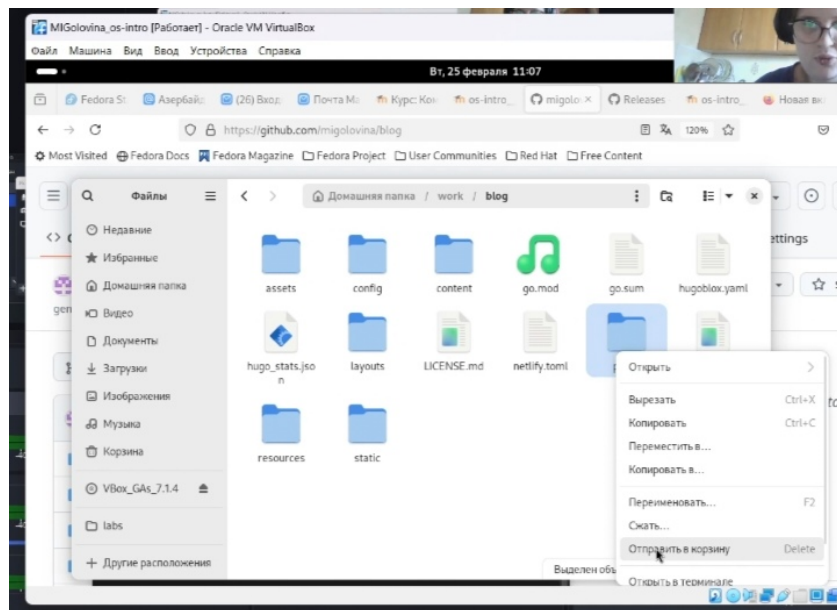


Рис. 4.14: Проверка

11. Запускаем hugo server (рис. 4.15).

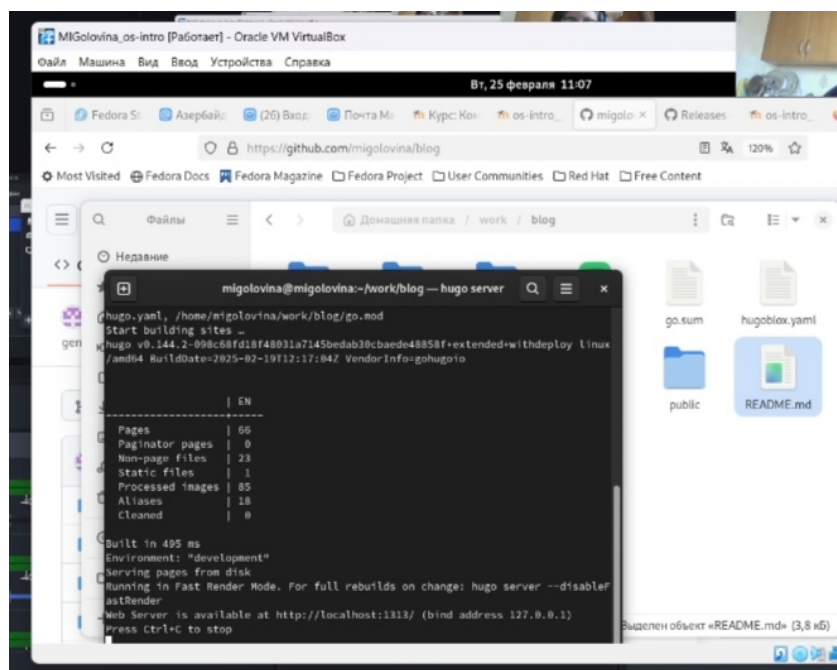


Рис. 4.15: Запуск hugo server

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

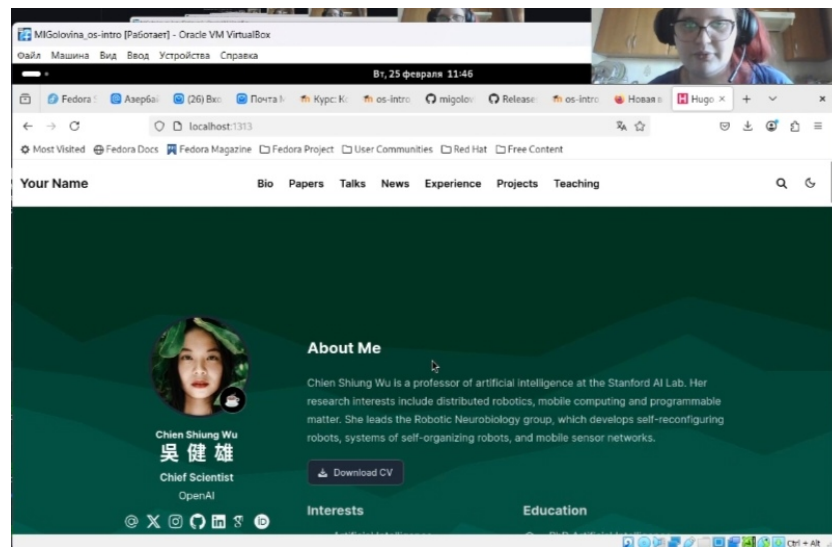


Рис. 4.16: Шаблон сайта

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

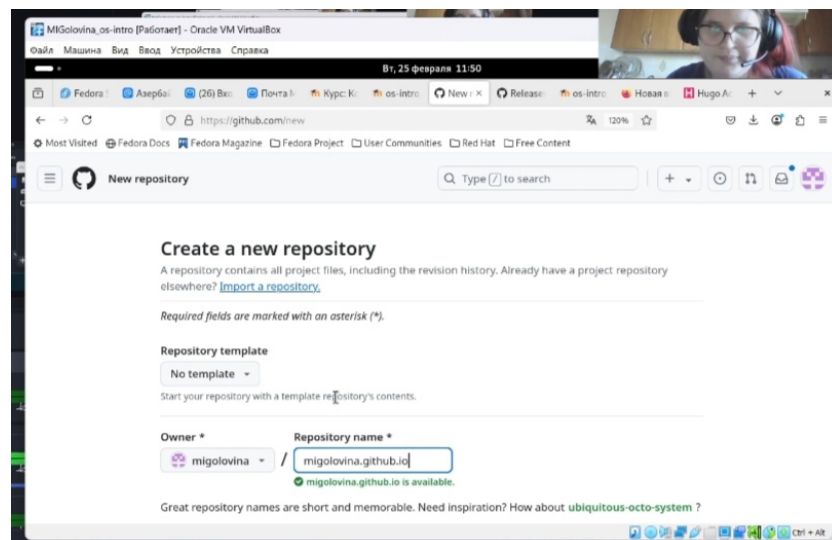
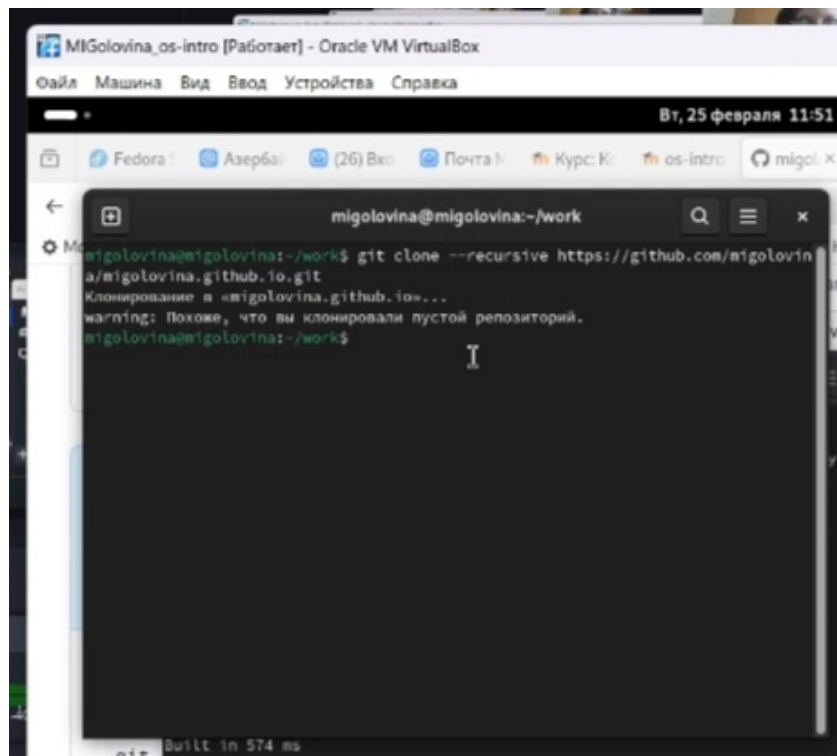


Рис. 4.17: Github

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



The screenshot shows a terminal window titled "migolovina@migolovina:~/work" with a search icon, menu icon, and close button. The terminal output is as follows:

```

migolovina@migolovina:~/work$ git clone --recursive https://github.com/migolovina/migolovina.github.io.git
Клонирование в «migolovina.github.io»...
warning: Похоже, что вы клонировали пустой репозиторий.
migolovina@migolovina:~/work$
```

The terminal window is part of a larger interface showing a browser window with tabs for "Fedora", "Азербайджан", "(26) Вх...", "Почта", "Курс: К...", "os-intro", and "migol x". The system clock at the top right indicates "Вт, 25 февраля 11:51".

Рис. 4.18: Клонирование репозитория

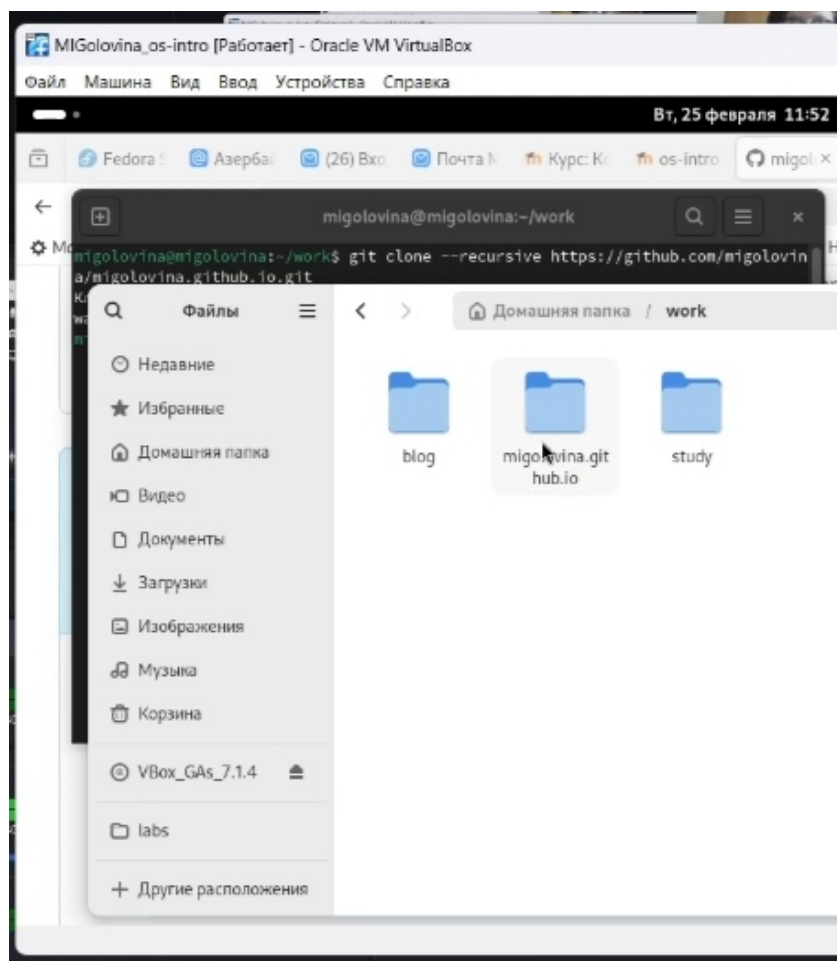


Рис. 4.19: Проверка

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

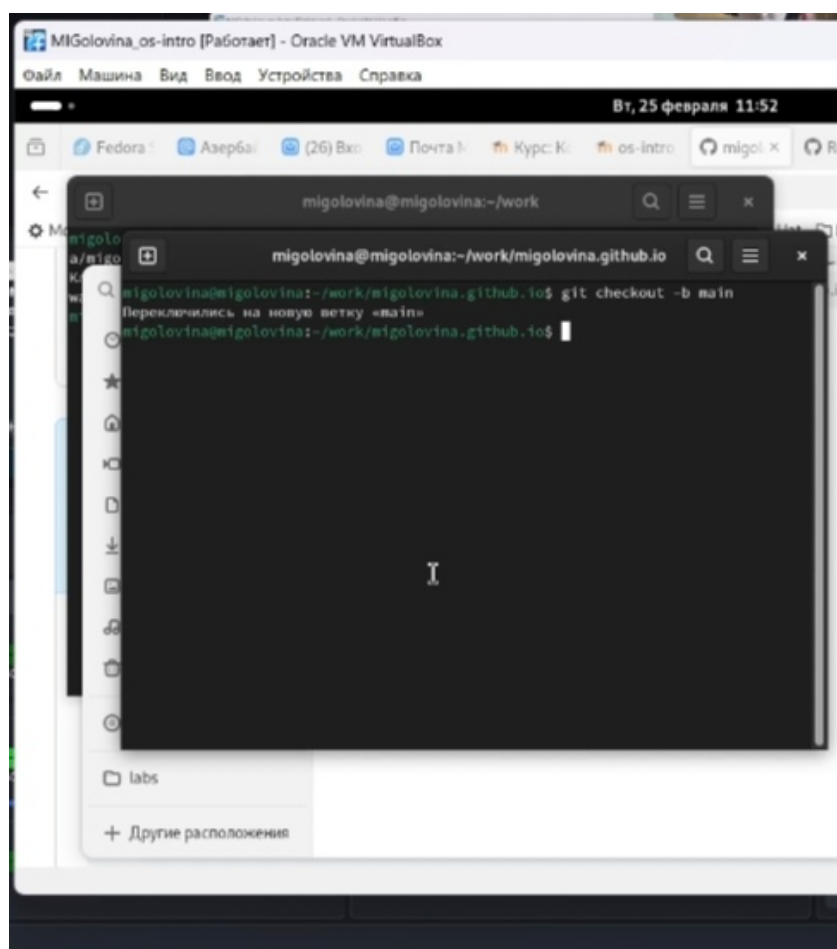


Рис. 4.20: Переключаемся на ветку “main”

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

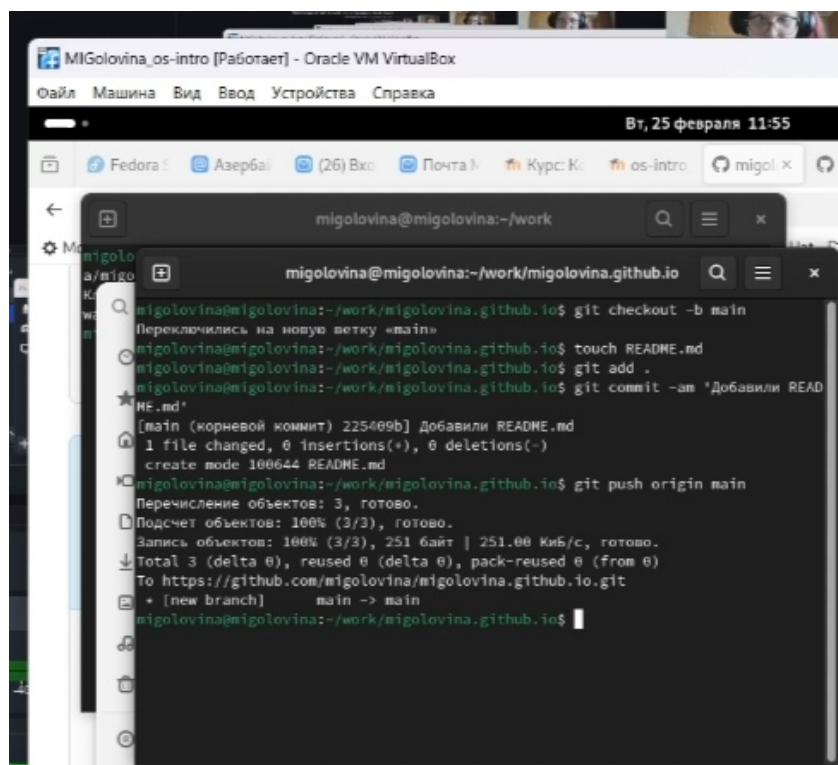


Рис. 4.21: Создание пустого файла и отправка изменений

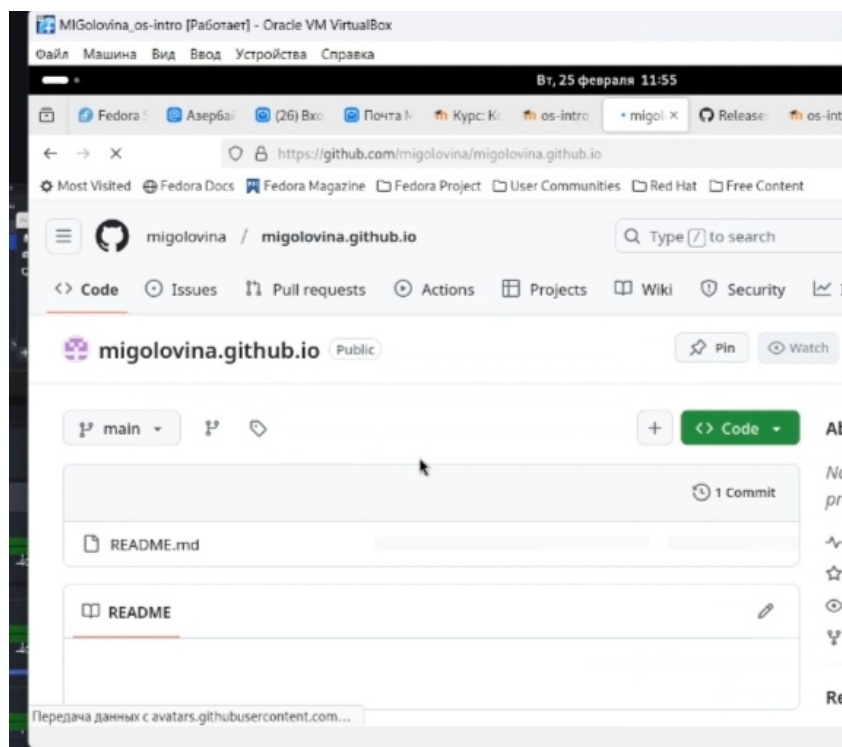


Рис. 4.22: Проверка

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

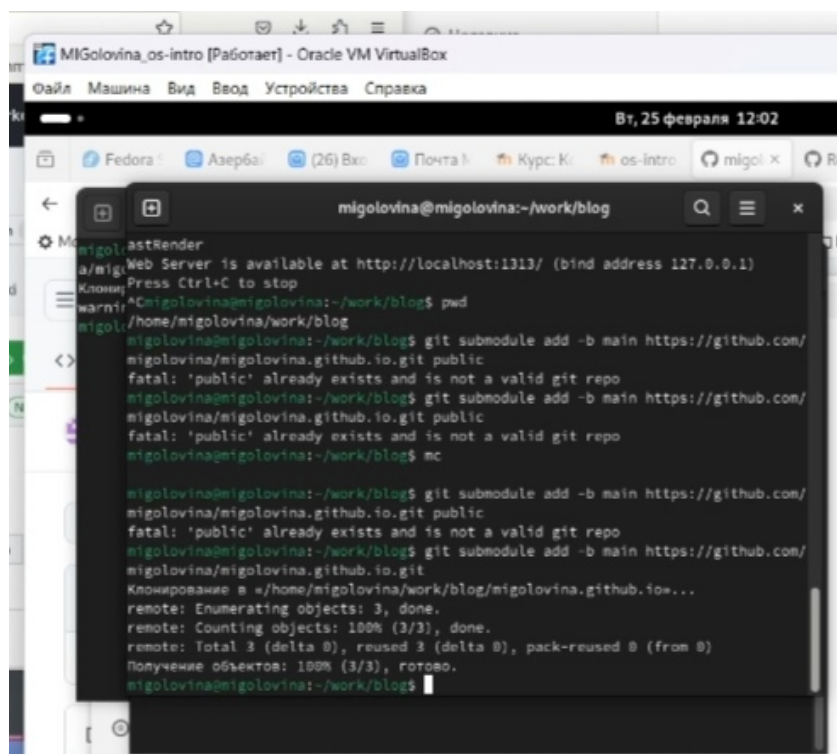


Рис. 4.23: Создаем новую ветку

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

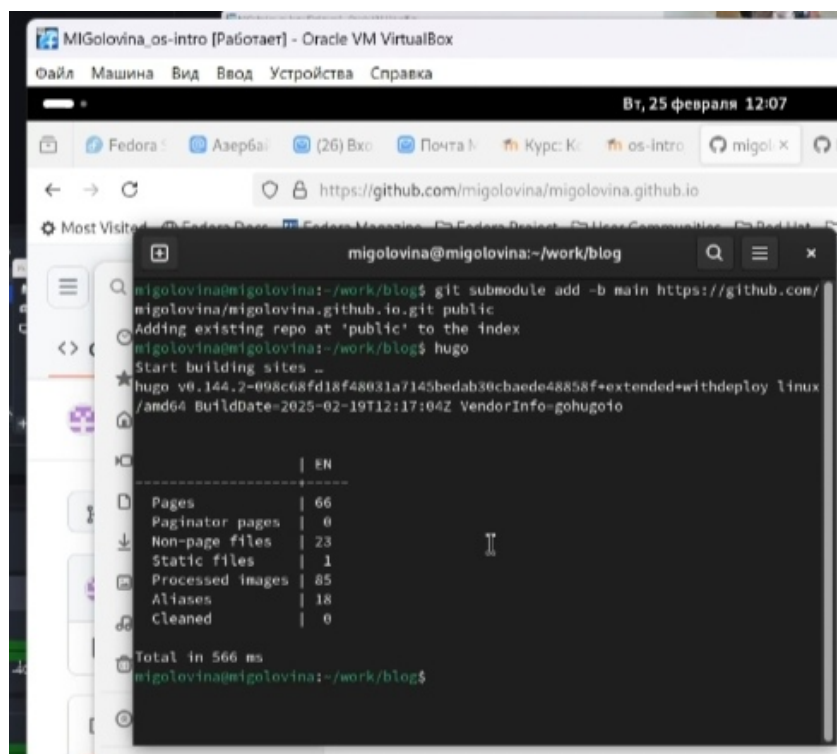


Рис. 4.24: HUGO

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

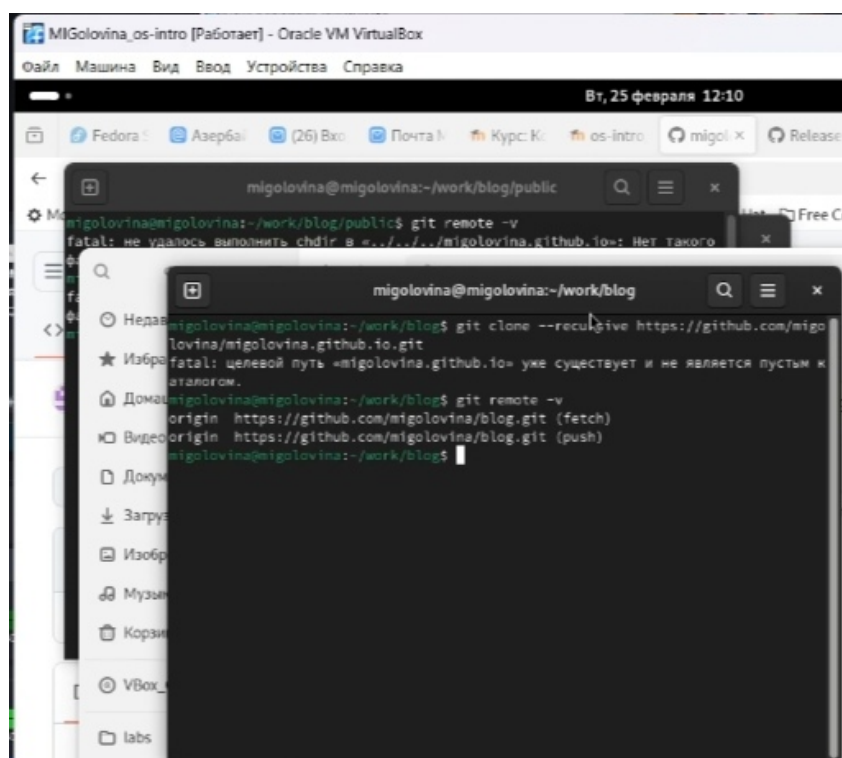


Рис. 4.25: Проверка

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

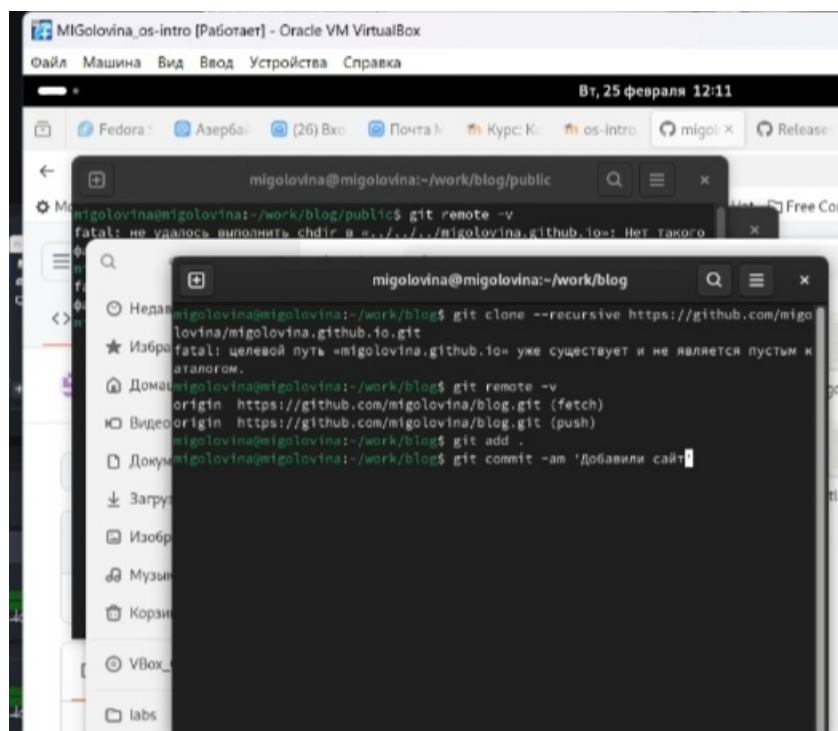


Рис. 4.26: Загружаем обновления

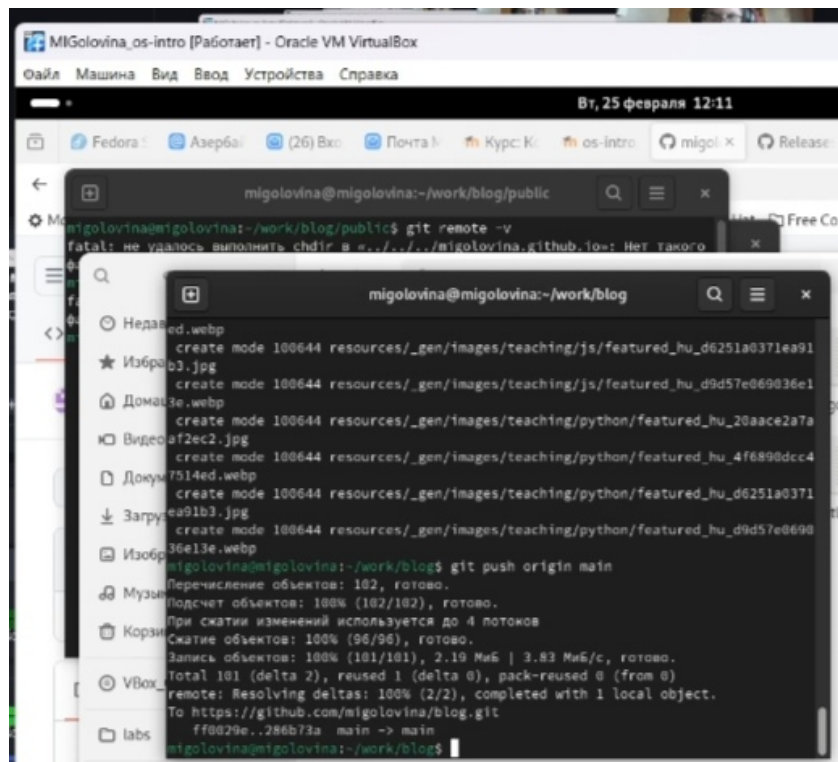


Рис. 4.27: Загружаем обновления

21. Проверка обновлений (рис. 4.28).

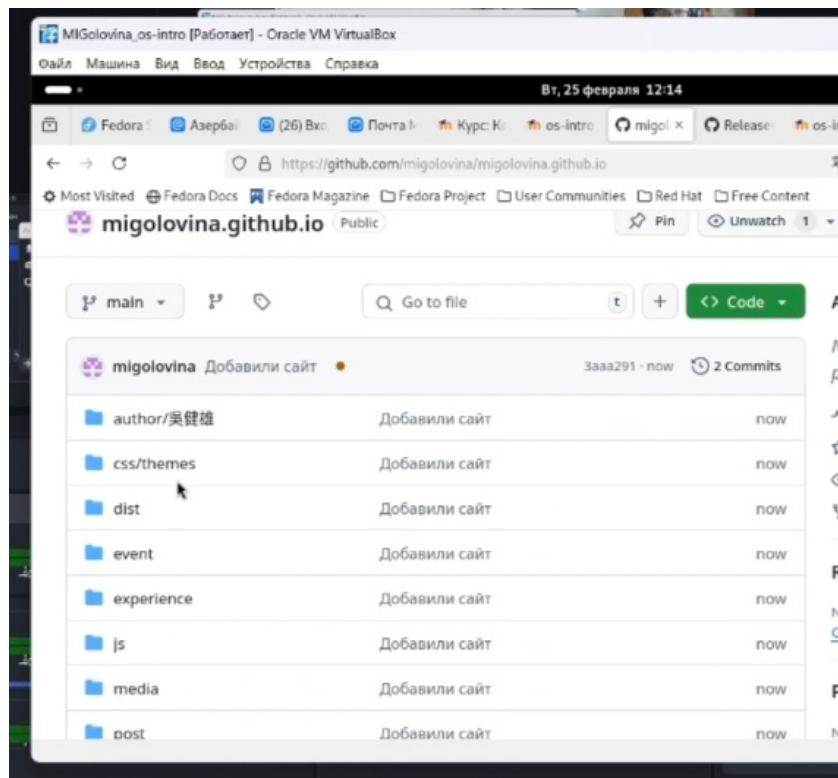


Рис. 4.28: Проверка github

22. Открываем наш сайт (рис. 4.29).

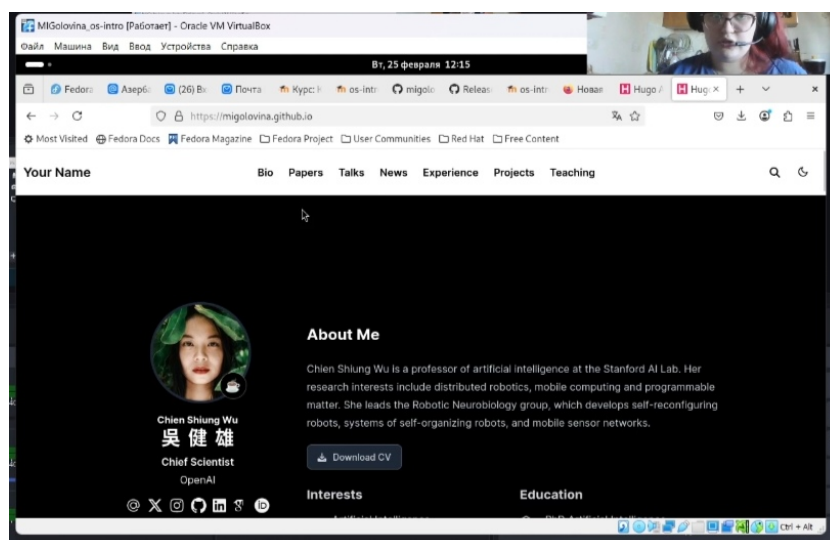


Рис. 4.29: Шаблон сайта готовый

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
2. Техническая реализация проекта
3. Руководство по выполнению первого этапа индивидуального проекта
4. Инструменты Git - Подмодули