Отчёт по индивидуальному проекту. Этап 1.

Архитектура компьютера и операционные системы

Рогожина Надежда Александровна, НКАбд-02-22

Содержание

1 Цель работы **6**

2 Задание **7**

3 Теоретическое введение **8**
3.1 Примеры использования git: 8
3.2 Основные команды git: 8

4 Выполнение лабораторной работы **10**

5 Выводы **26**

Список литературы **27**

Список иллюстраций

4.1 Загрузка нужной нам версии 10
4.2 Скачивание 10
4.3 Извлечение 11
4.4 Извлечение - выбор папки 11
4.5 Проверка извлечения 12
4.6 Вырезание файла 12
4.7 Перенос файла 13
4.8 Создание репозитория 13
4.9 Создание репозитория 14
4.10 Проверка 14
4.11 Клонирование репозитория 15
4.12 Проверка 15
4.13 Hugo 15
4.14 Доустановка модуля “GO” 16
4.15 HUGO 16
4.16 Проверка 16
4.17 Удаляем public 17
4.18 Запуск hugo server 17
4.19 Запуск hugo server 17
4.20 Шаблон сайта 18
4.21 Закрытие hugo server 18
4.22 Github 18
4.23 Создаем новый репозиторий 19
4.24 Клонирование репозитория 19
4.25 Проверка 19
4.26 Переключаемся на ветку “main” 20
4.27 Создание пустого файла и отправка изменений 20
4.28 Проверка 20
4.29 Создаем новую ветку 21
4.30 Исправление ошибки .gitignore 21
4.31 Проверка 22
4.32 Повтор введенной команды 22
4.33 HUGO 22
4.34 Проверка 23
4.35 Загружаем обновления 23
4.36 Загружаем обновления 23
4.37 Проверка github 24

4.38 Шаблон сайта готовый 25

Список таблиц

3.1 Описание некоторых команд системы контроля версий Git 8

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. 3.1 приведено краткое описание основных команд Git.

Таблица 3.1: 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

Команда	Описание команды
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

4 Выполнение лабораторной работы

Загружаем последнюю версию hugo (рис. 4.1):

Рис. 4.1: Загрузка нужной нам версии

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

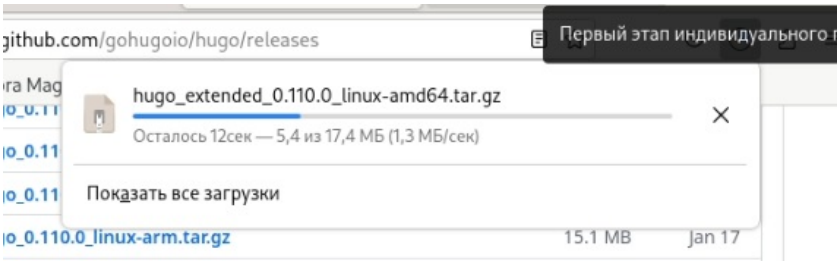

Рис. 4.2: Скачивание

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

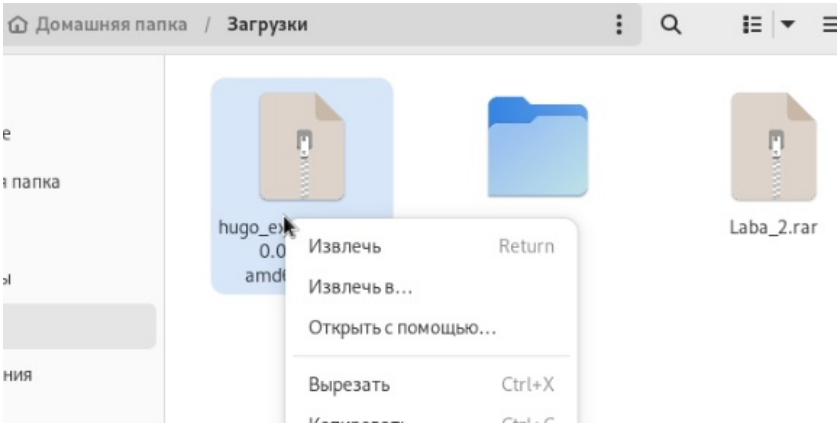

Рис. 4.3: Извлечение

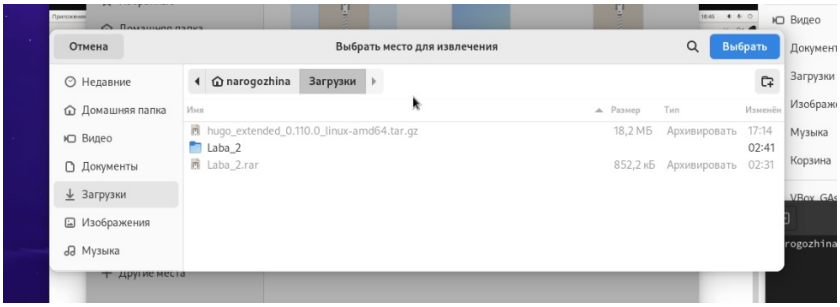

Рис. 4.4: Извлечение - выбор папки

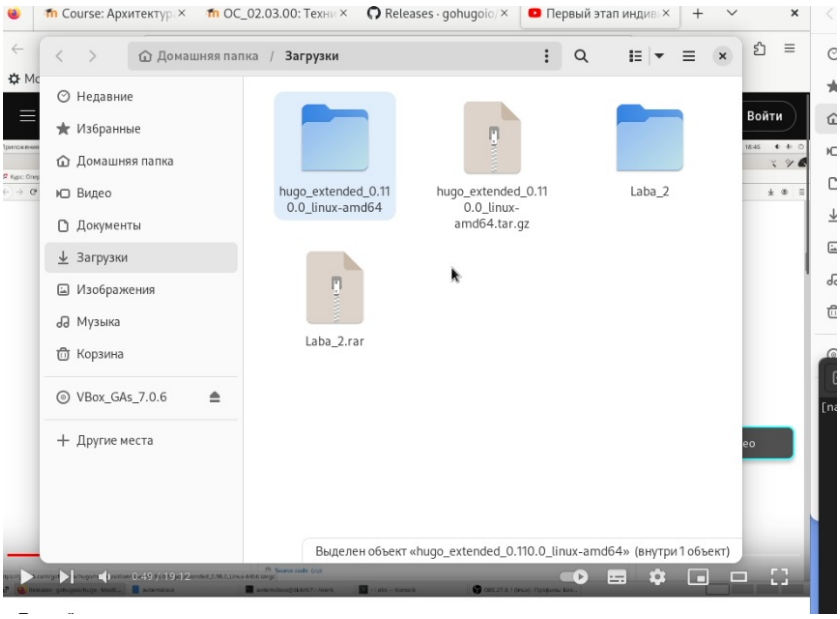

Рис. 4.5: Проверка извлечения

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

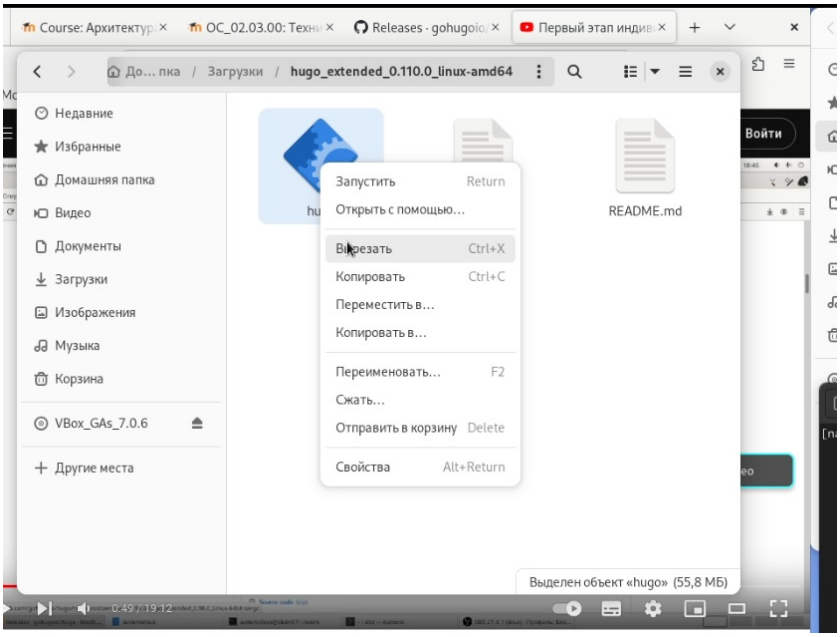

Рис. 4.6: Вырезание файла

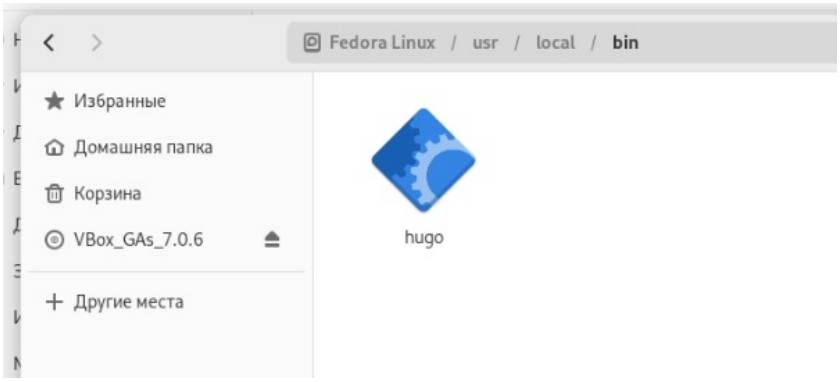

Рис. 4.7: Перенос файла

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

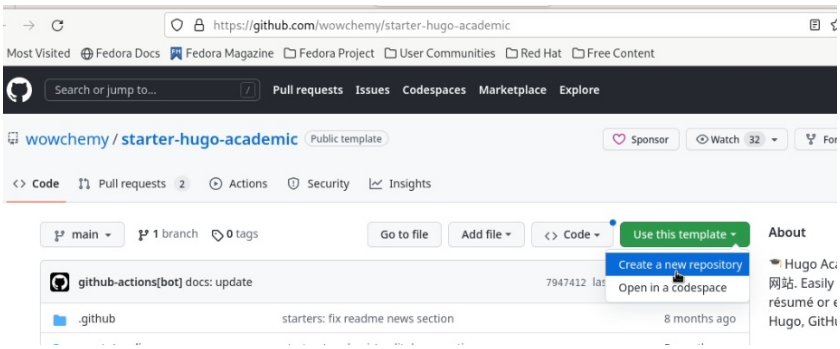

Рис. 4.8: Создание репозитория

При создании даём ему имя ***blog***:

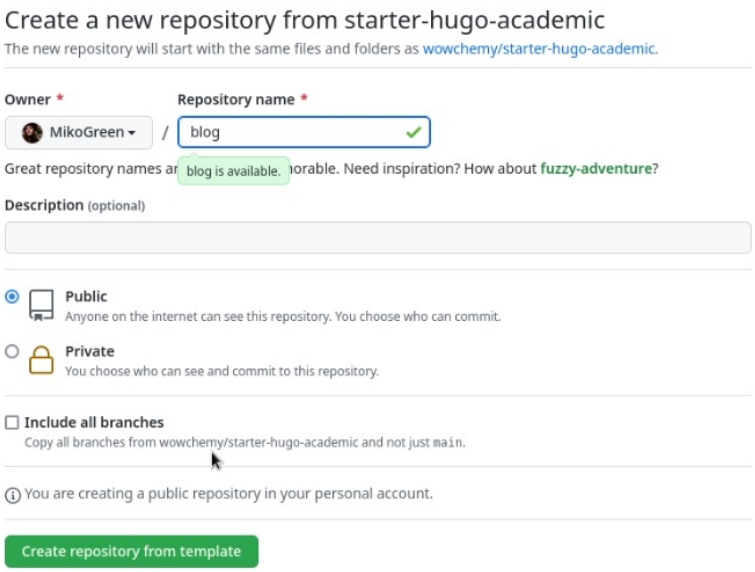

Рис. 4.9: Создание репозитория

Проверка:

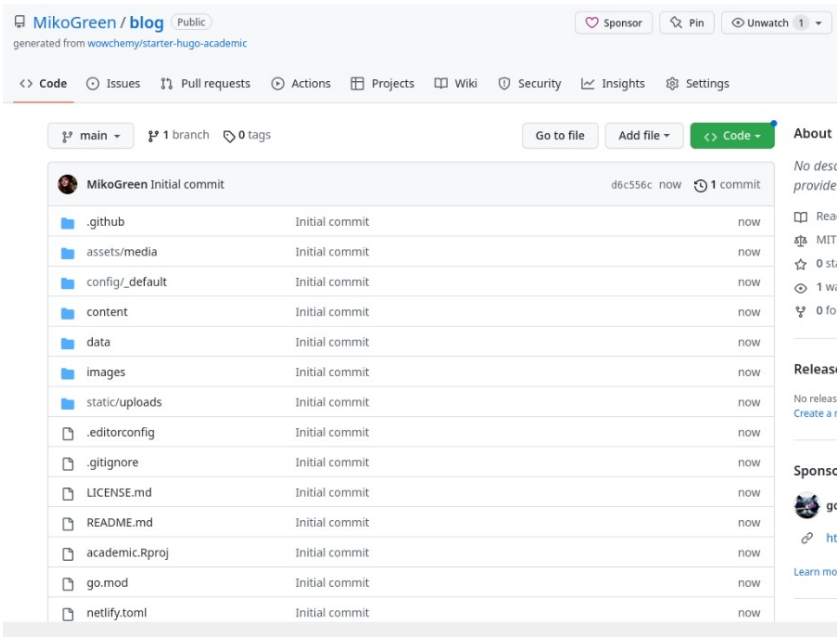

Рис. 4.10: Проверка

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

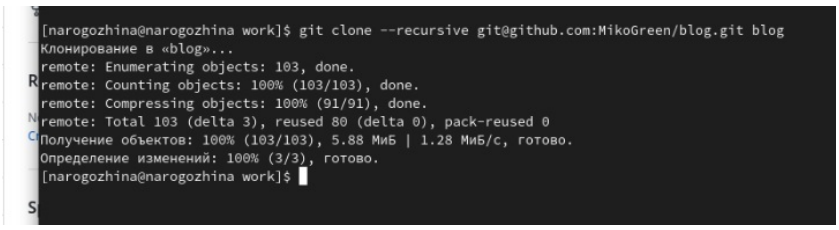

Рис. 4.11: Клонирование репозитория

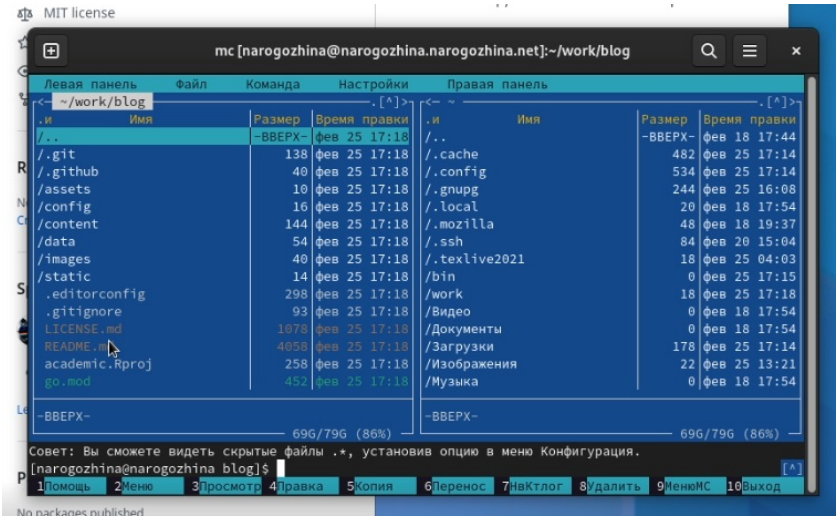

Рис. 4.12: Проверка

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

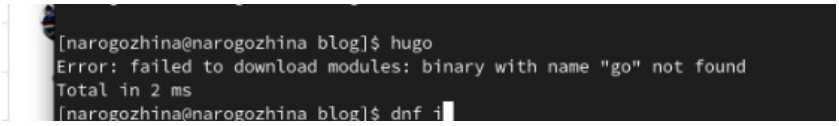

Рис. 4.13: Hugo

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

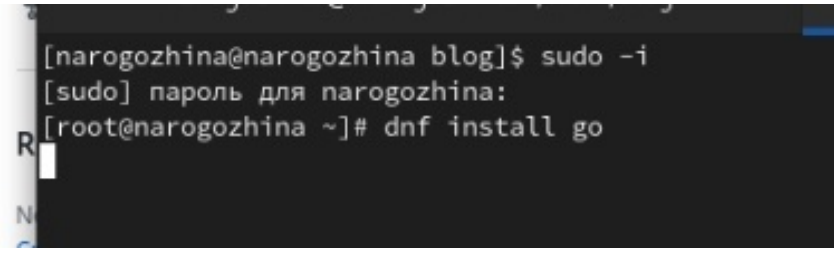

Рис. 4.14: Доустановка модуля “GO”

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

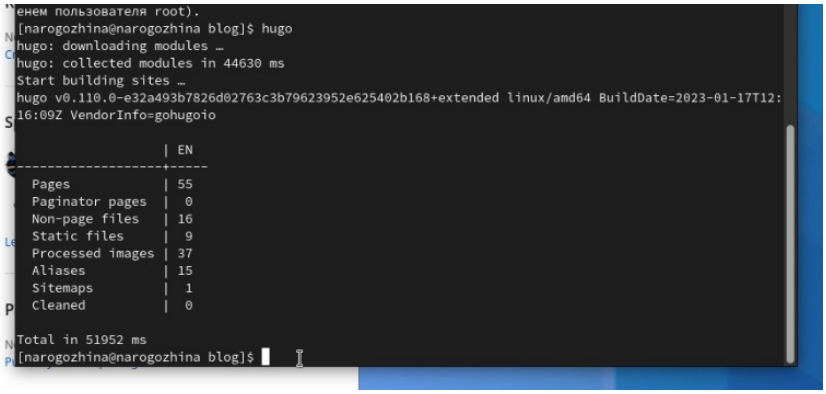

Рис. 4.15: HUGO

После установки необходимых модулей проверяем создание папок и файлов:

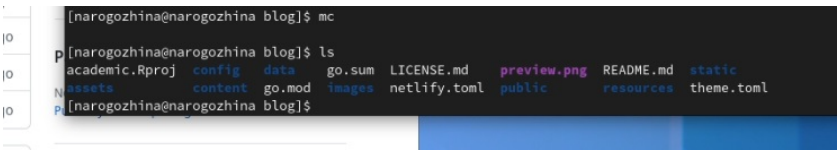

Рис. 4.16: Проверка

Удаляем каталог *public* с помощью *mc*:

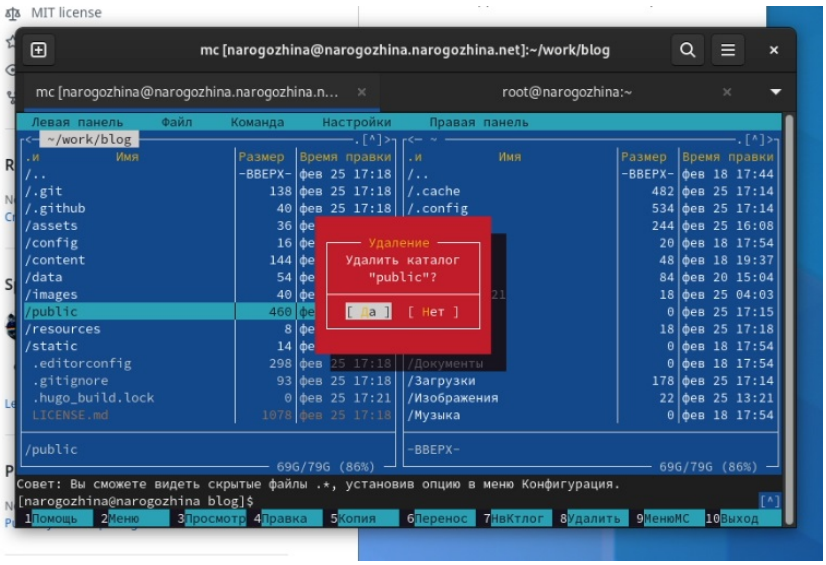

Рис. 4.17: Удаляем public

Запускаем *hugo server*:

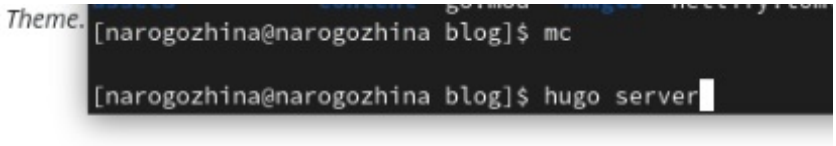

Рис. 4.18: Запуск hugo server

Рис. 4.19: Запуск hugo server

Открываем ссылку в браузере и видим сайт:

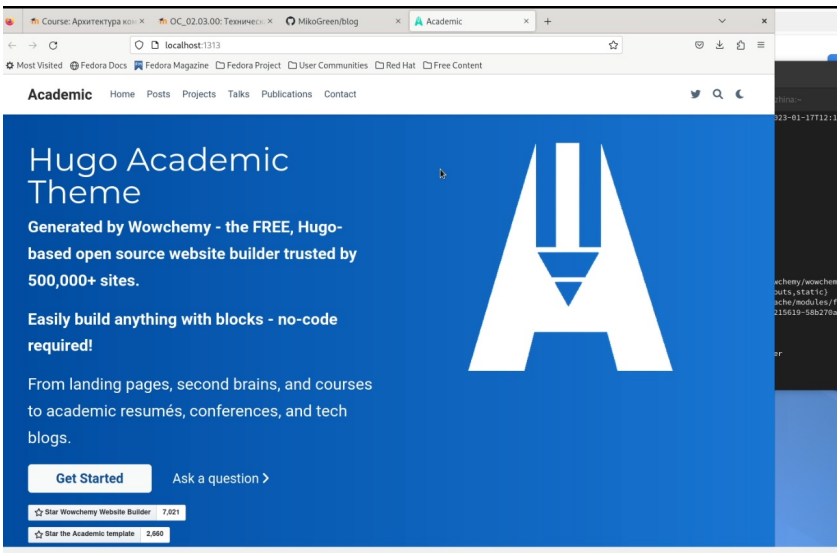

Рис. 4.20: Шаблон сайта

После проверки в браузере закроем сервер:

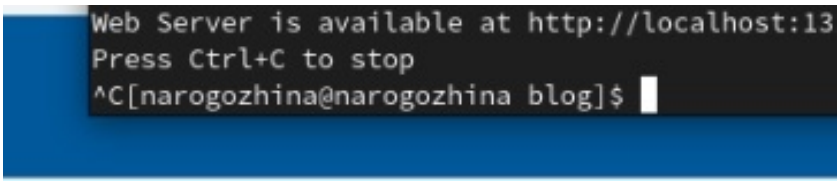

Рис. 4.21: Закрытие hugo server

Создание нового репозитория:

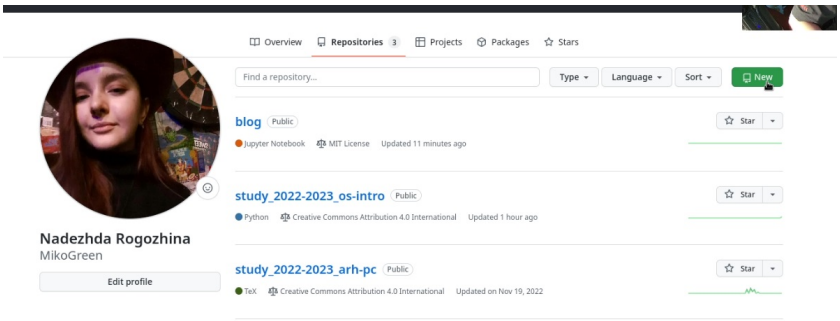

Рис. 4.22: Github

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

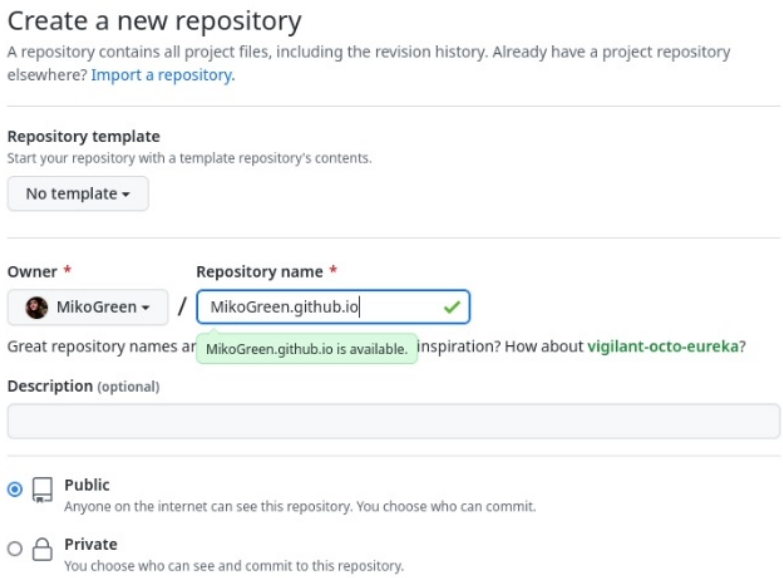

Рис. 4.23: Создаем новый репозиторий

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

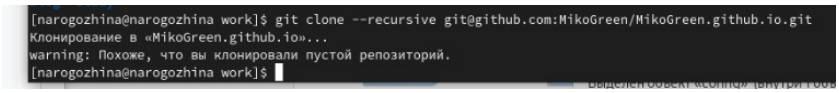

Рис. 4.24: Клонирование репозитория

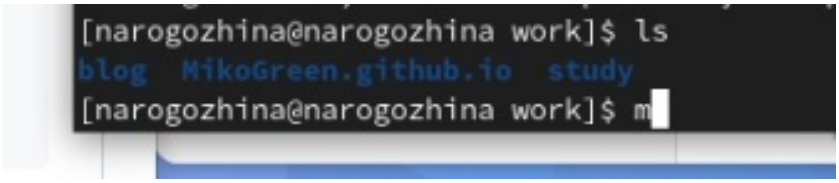

Рис. 4.25: Проверка

Переключаемся на ветку “main”:

Рис. 4.26: Переключаемся на ветку “main”

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

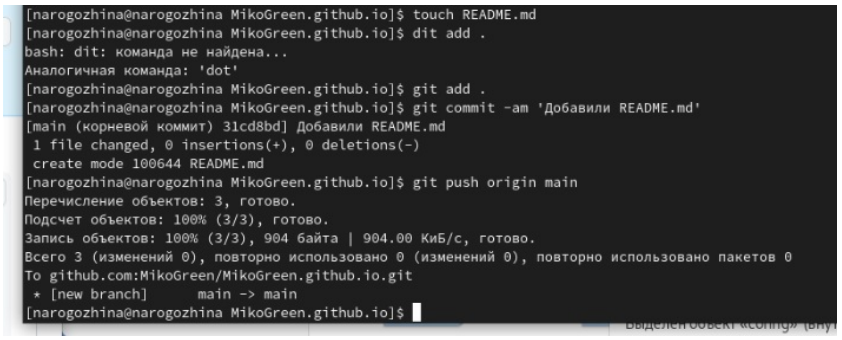

Рис. 4.27: Создание пустого файла и отправка изменений

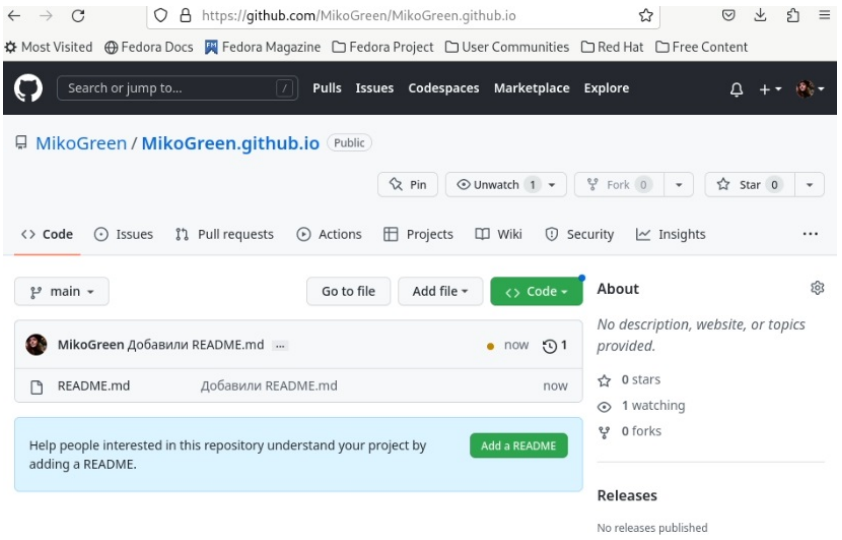

Рис. 4.28: Проверка

Создаем ветку подмодуля, клонируя репозиторий с нашего Github:

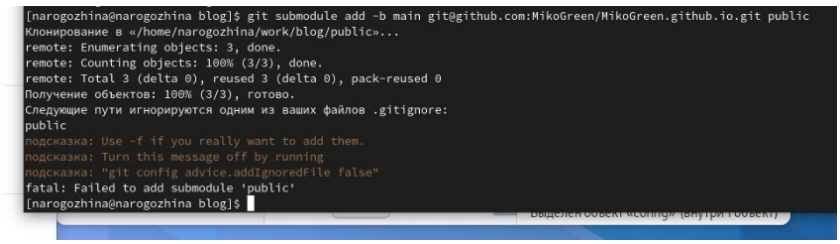

Рис. 4.29: Создаем новую ветку

Так как вылезает ошибка *.gitignore*, то нужно её исправить прежде чем идти дальше - в файле .gitignore закомментим папку public, чтобы этот путь не игнорировался:

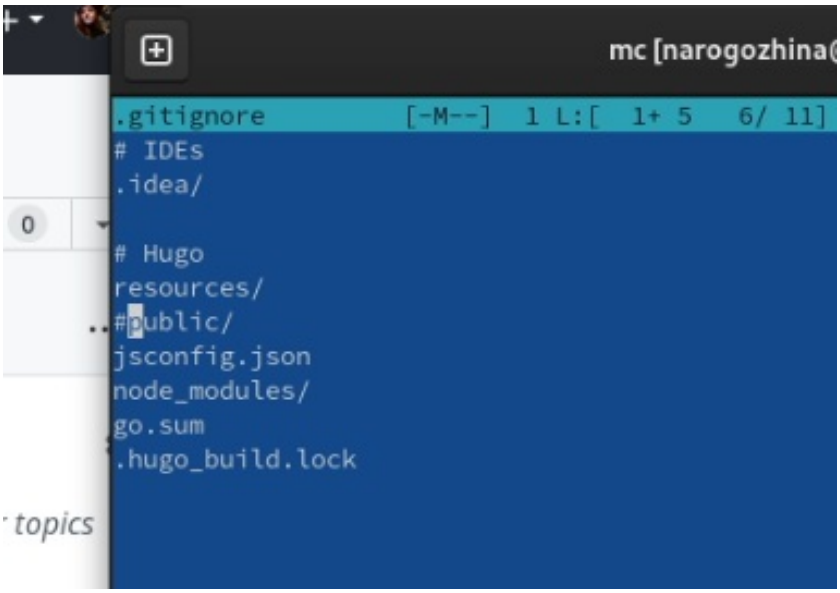

Рис. 4.30: Исправление ошибки .gitignore

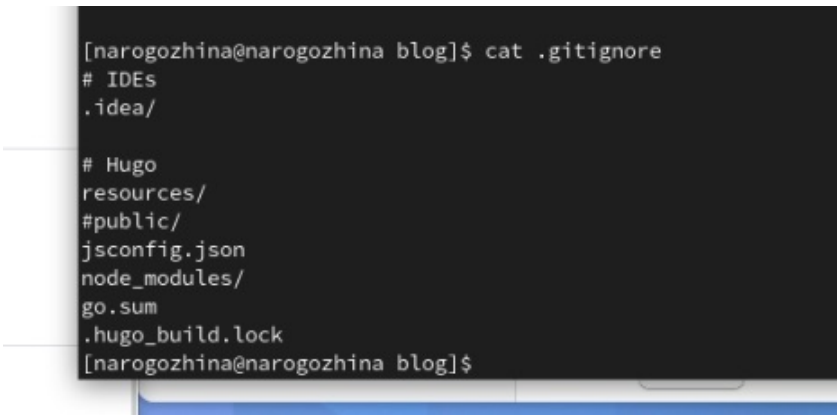

Рис. 4.31: Проверка

Повторяем команду:

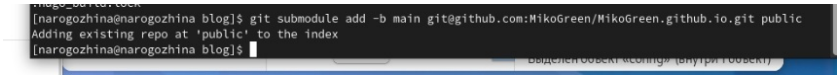

Рис. 4.32: Повтор введенной команды

После выполнения запускаем hugo:

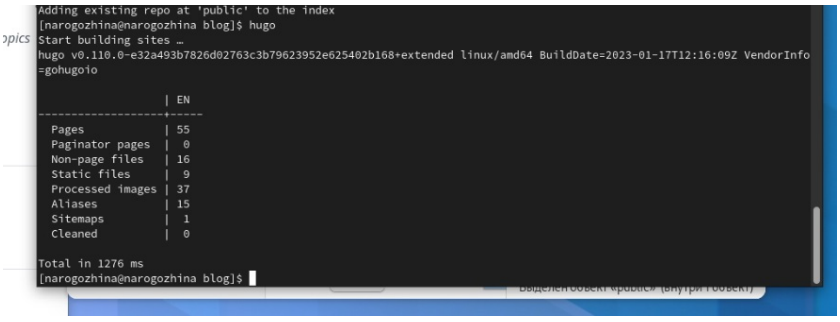

Рис. 4.33: HUGO

Проверим подключение каталога к репозиторию командой **git remote -v**:

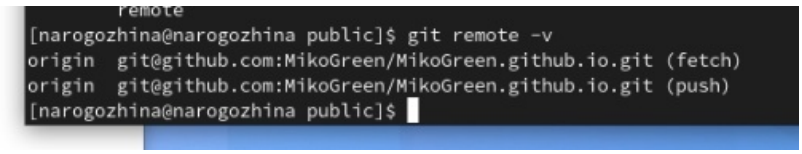

Рис. 4.34: Проверка

Добавим изменения на github:

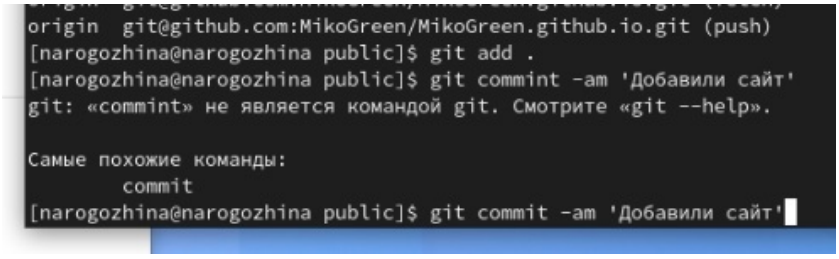

Рис. 4.35: Загружаем обновления

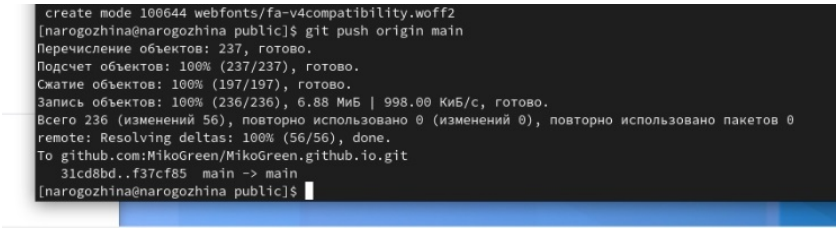

Рис. 4.36: Загружаем обновления

Проверка обновлений:

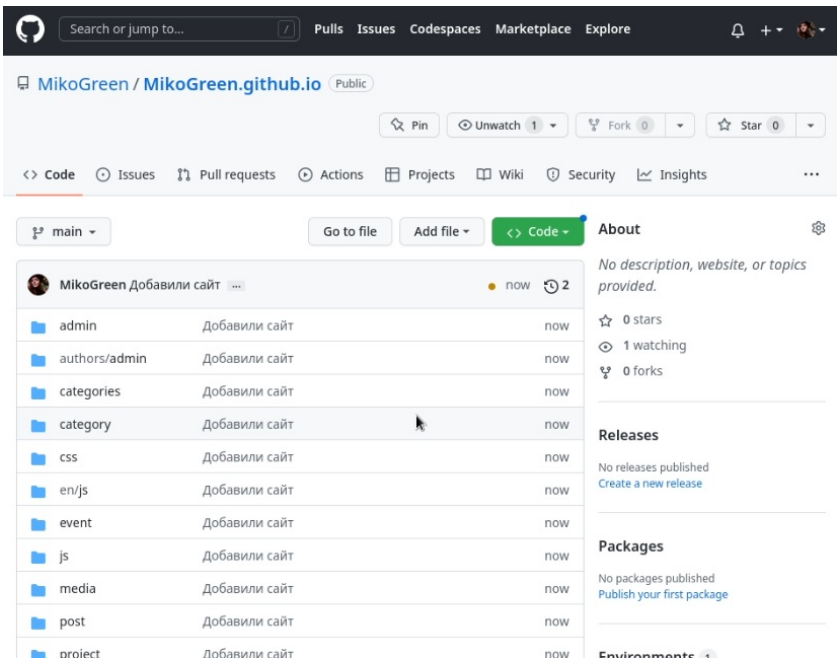

Рис. 4.37: Проверка github

Открываем наш сайт:

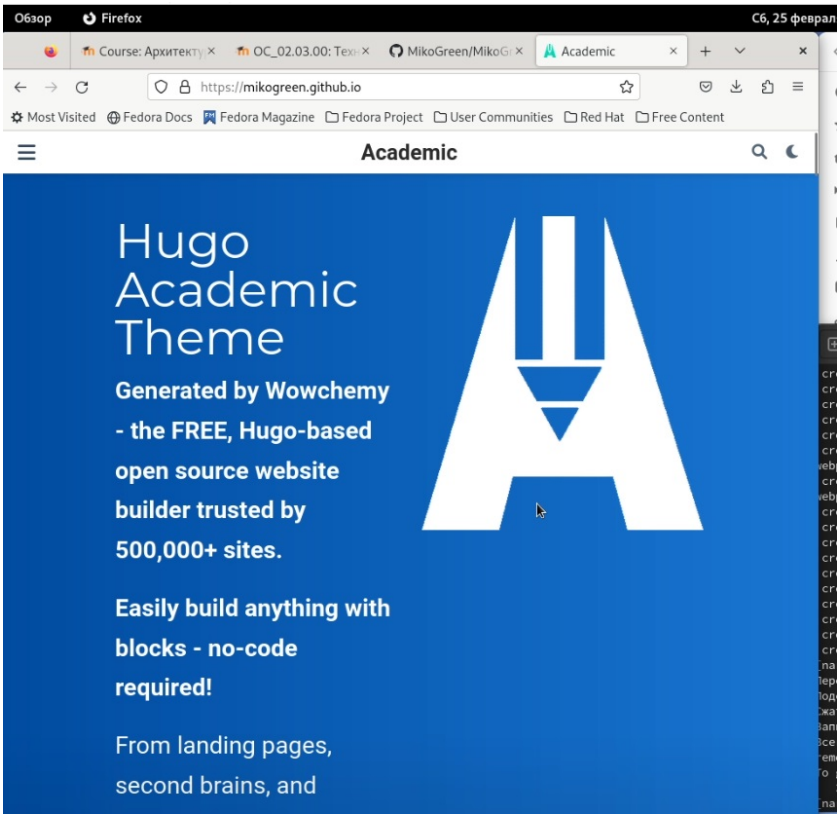

Рис. 4.38: Шаблон сайта готовый

5 Выводы

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