Отчёт по индивидуальному проекту. Этап 1

Дисциплина: Операционные системы

Батова Ирина Сергеевна, НММбд-01-22

Содержание

# 1 Цель работы

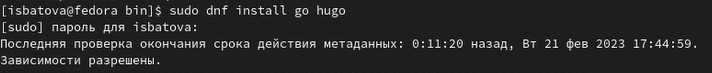
Целью первого этапа индивидуального проекта является размещение на Github pages заготовки для персонального сайта.

# 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.

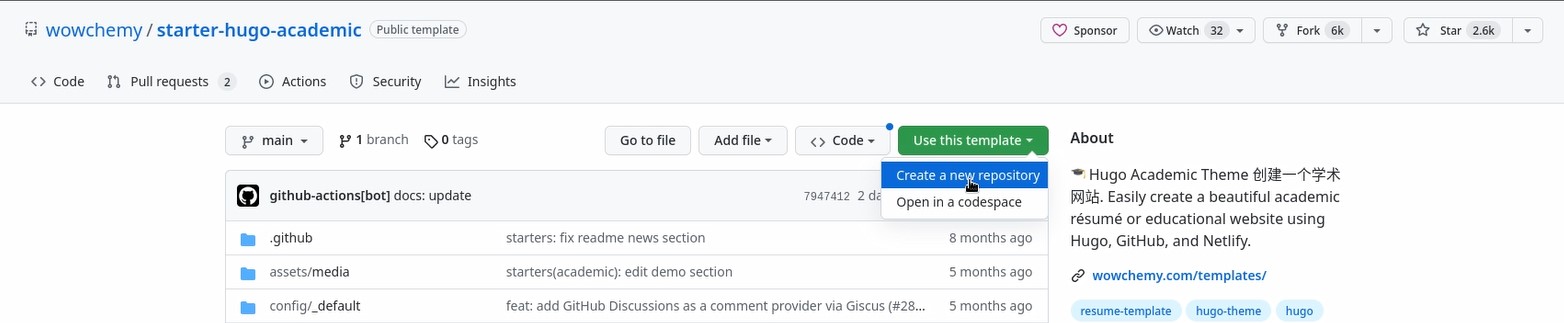
# 3 Выполнение работы

Для начала работы устанавливаем go hugo (рис. ??).



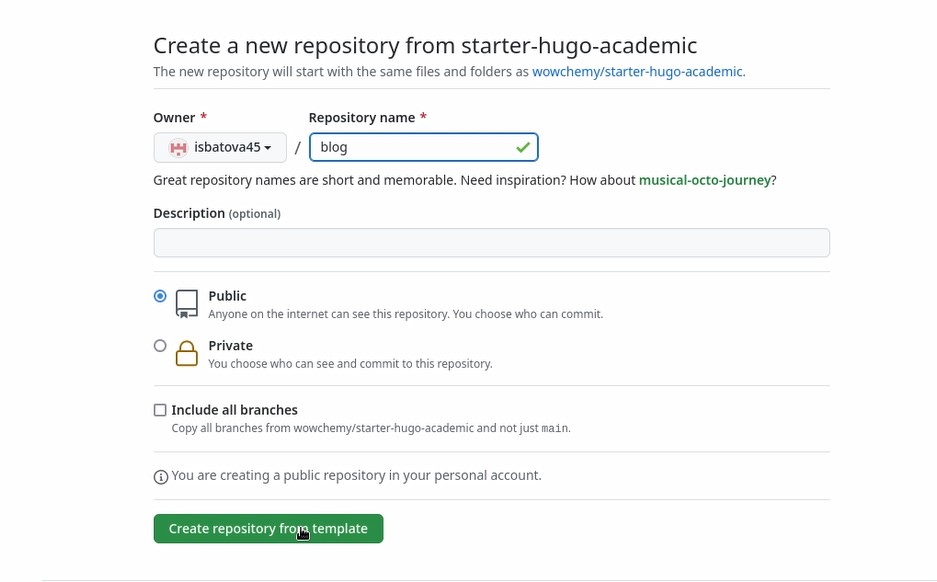
Установка go hugo

Далее в ТУИС берем ссылку на шаблон репозитория, открываем эту ссылку в браузере и генерируем репозиторий по этому шаблону (рис. ??).

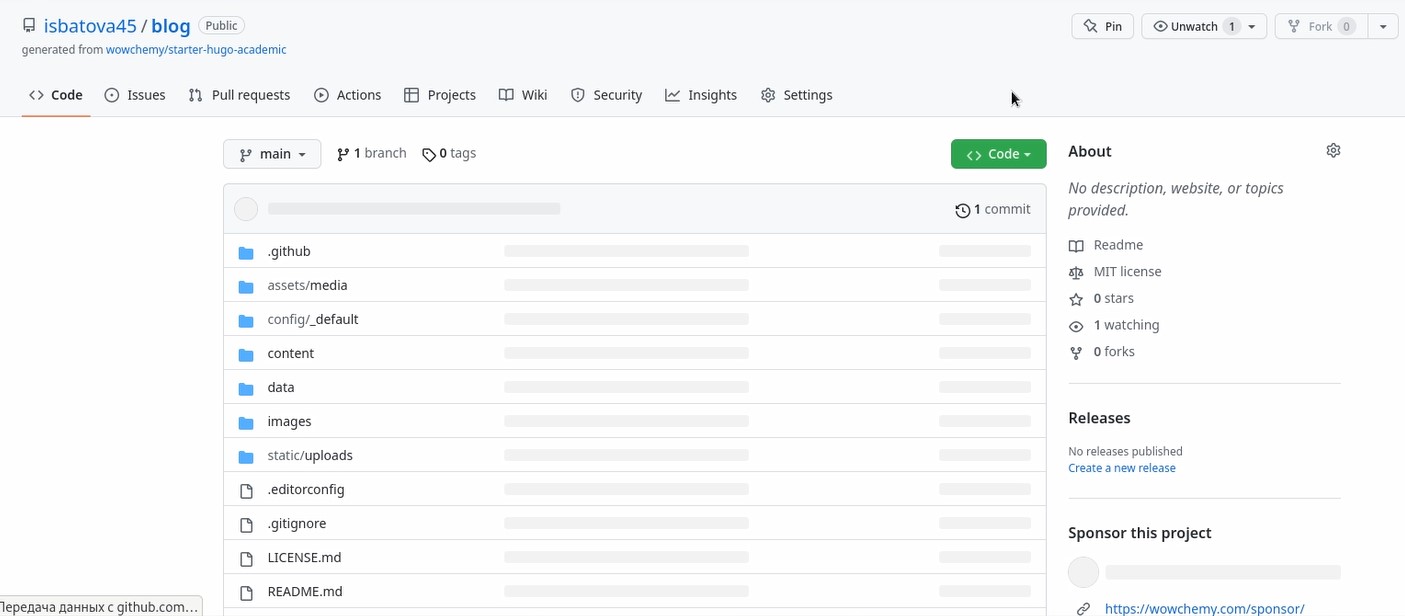


Генерация репозитория

Назовем этот репозиторий ‘blog’ (рис. ??). Создаем репозиторий (рис. ??).

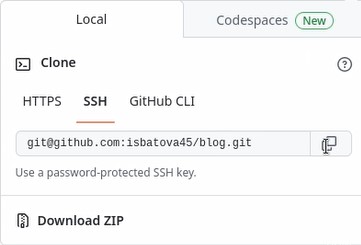


Название репозитория



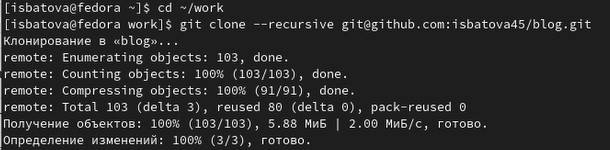
Создание репозитория

Далее копируем SSH-ключ созданного репозитория (рис. ??).



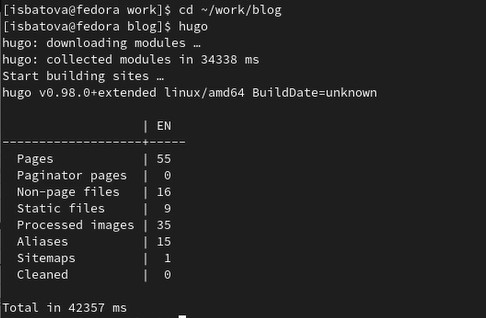
Копирование SSH-ключа

Открываем терминал, переходим в каталог ‘work’ и клонируем туда данный репозиторий (рис. ??).



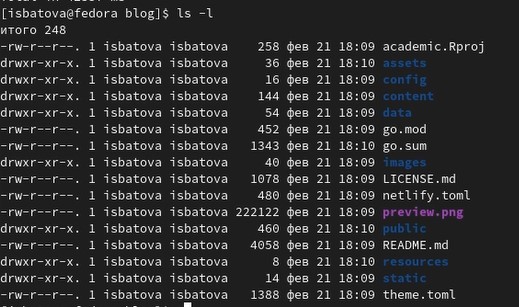
Клонирование репозитория

Далее переходим в создавшийся каталог ‘blog’ вводим команду ‘hugo’ (рис. ??).



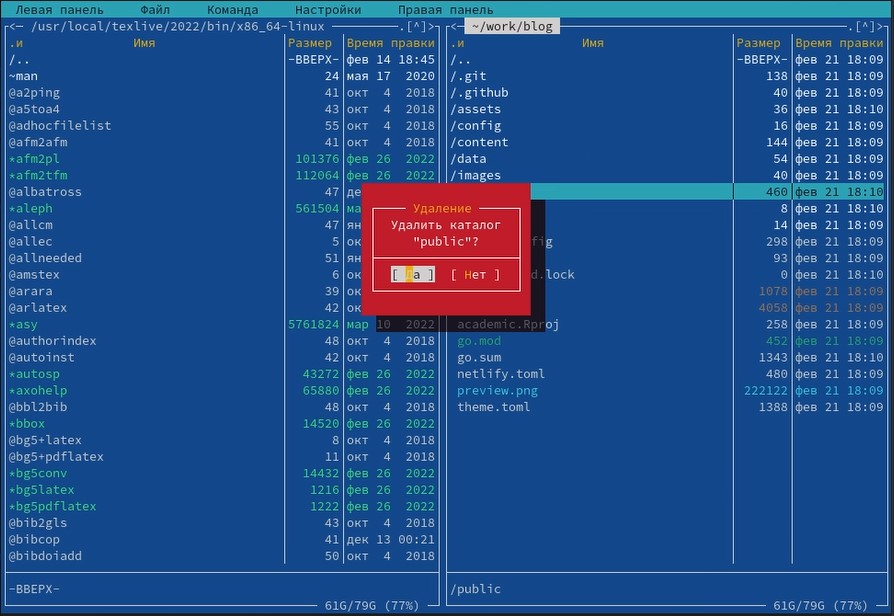
Команда ‘hugo’

При проверке содержимого каталога командой ‘ls -l’ видим, что у нас появился каталог ‘public’ (рис. ??).



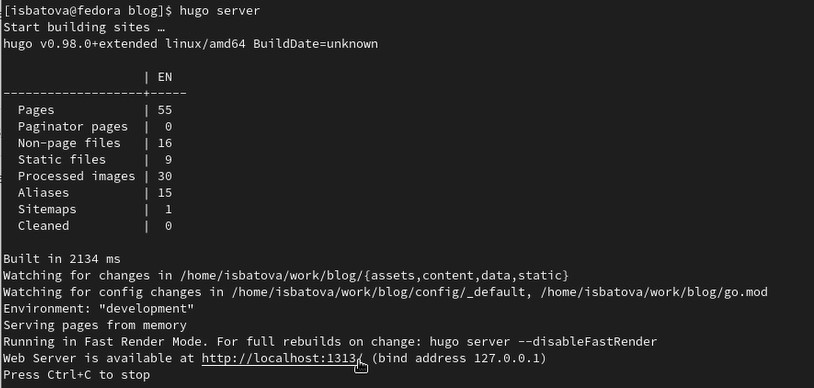
Просмотр содержимого каталога blog

Нам необходимо удалить данный каталог для дальнейшей работы. Для этого переходим в mc и удаляем там каталог ‘public’ (рис. ??).



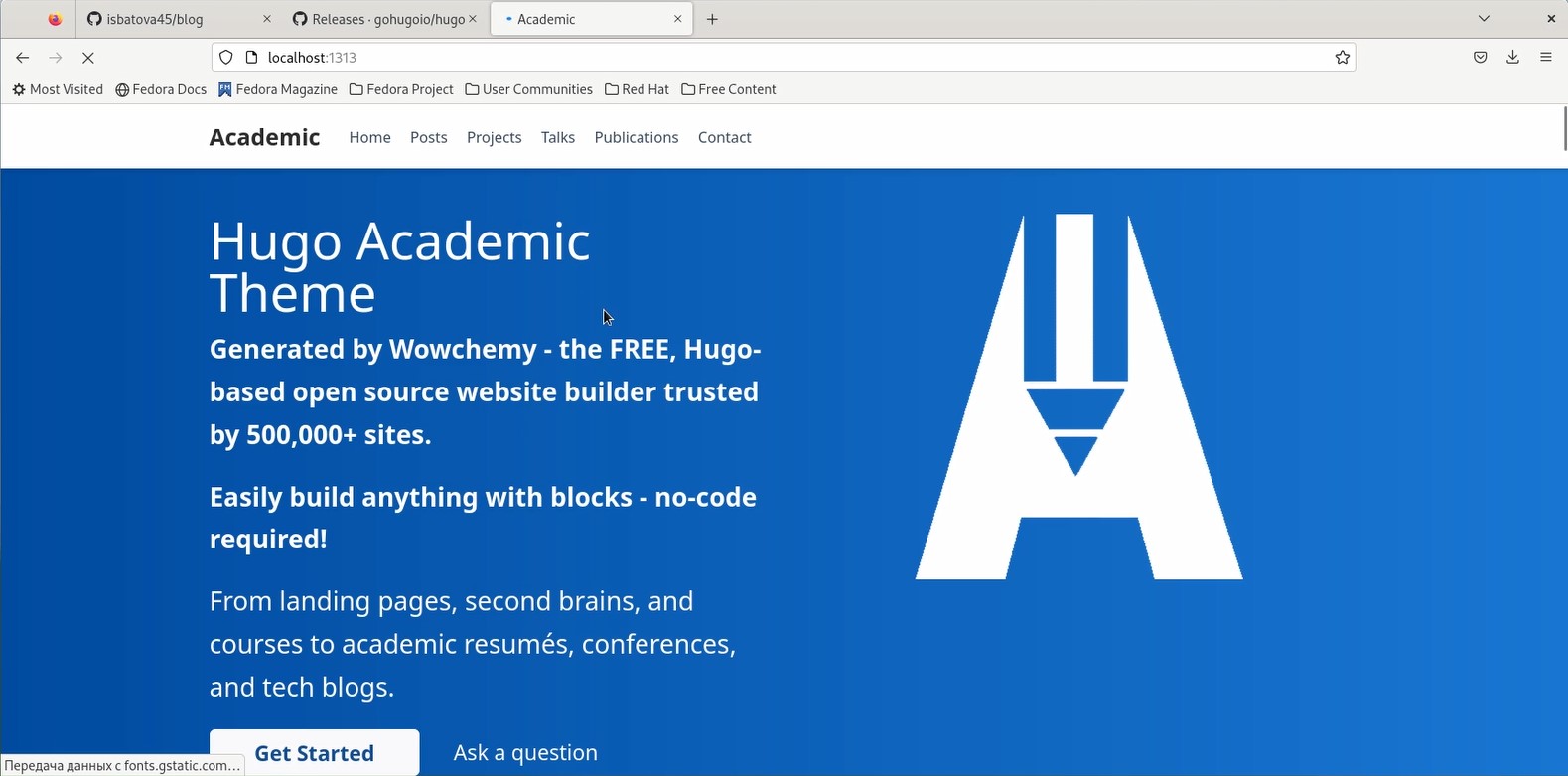
Удаление каталога ‘public’

Далее возвращаемся в терминал и вводим команду ‘hugo server’ (рис. ??). В результате выполнения данной команды мы получаем ссылку на наш сайт, которая работает только на локальном устройстве.



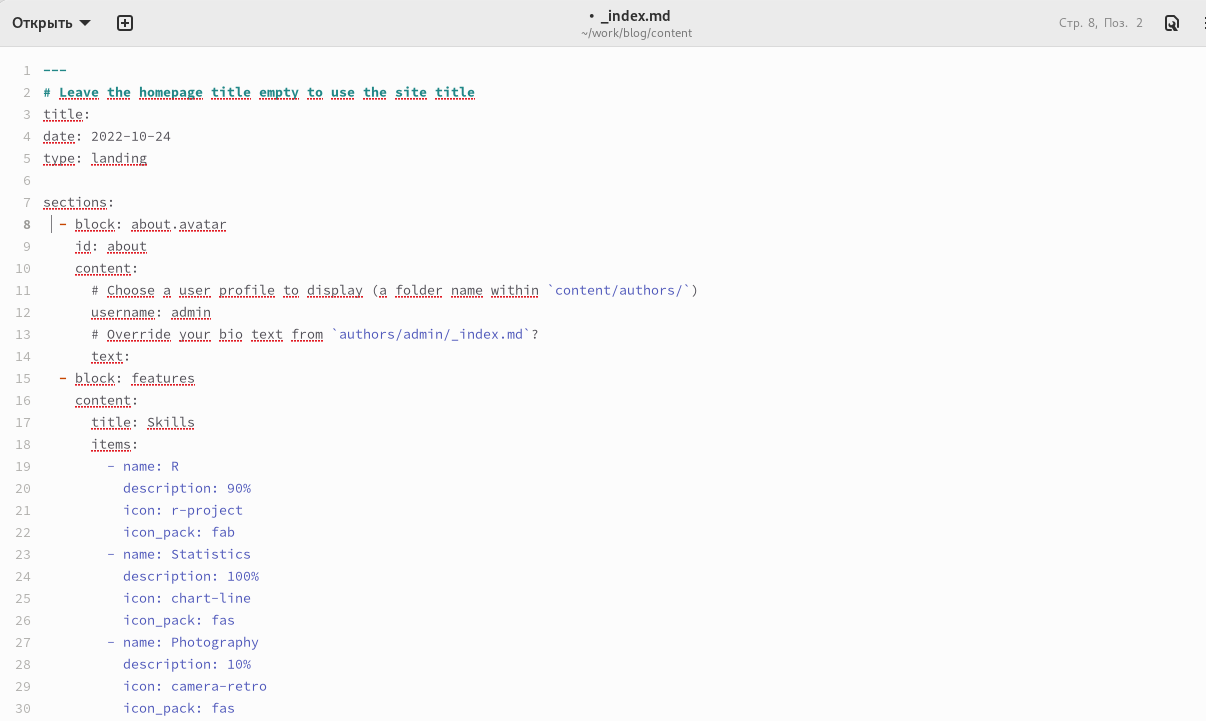
Команда ‘hugo server’

Копируем ссылку и вставляем ее в браузер (рис. ??).



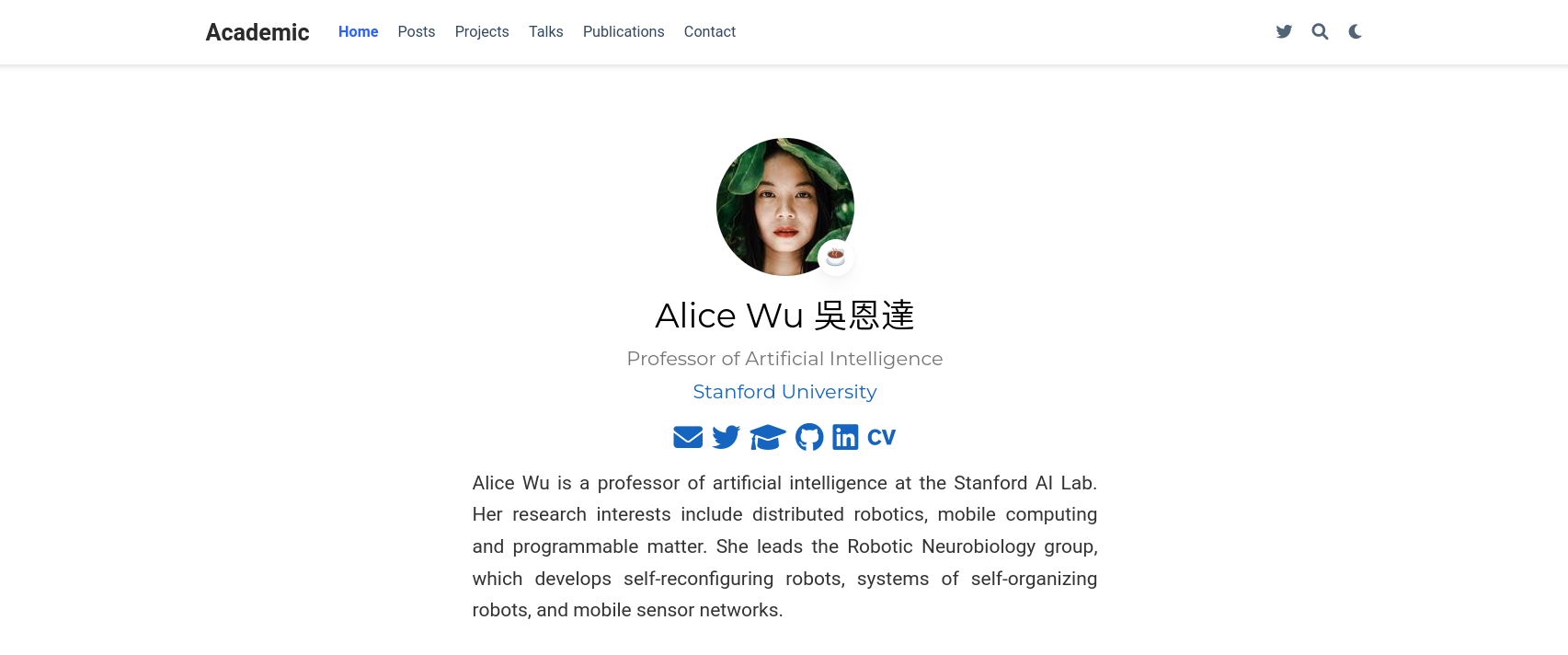
Локальная ссылка на сайт

Следующим шагом нам необходимо удалить синее поле наверху сайта. Для этого переходим в каталог work -> blog -> content, открываем файл с названием ’\_index.md’ и удаляем первый блок, располагающийся после слова ‘sections’ (рис. ??).



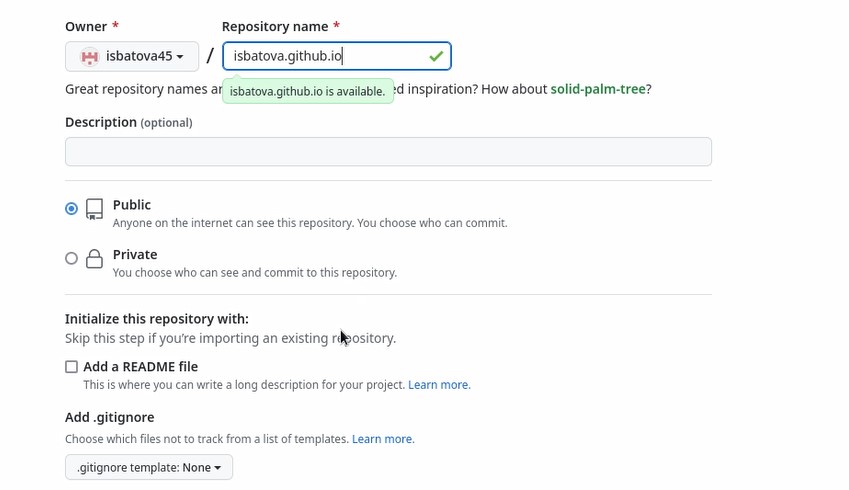
Удаление синего поля

Далее возвращаемся на сайт и проверяем - синее поле исчезло (рис. ??).



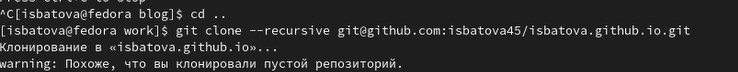
Сайт без синего поля

Остаемся в браузере и заходим на github. Создаем новый репозиторий (рис. ??).



Создание нового репозитория

После этого возвращаемся в терминал, переходим на ступень выше (в каталог ‘work’) и клонируем туда только что созданный репозиторий (рис. ??).



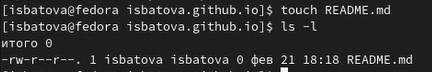
Клонирование нового репозитория

Переходим в создавшийся после клонирования каталог. Так как репозиторий пустой, создаем в нем главную ветку (рис. ??).

Создание главной ветки

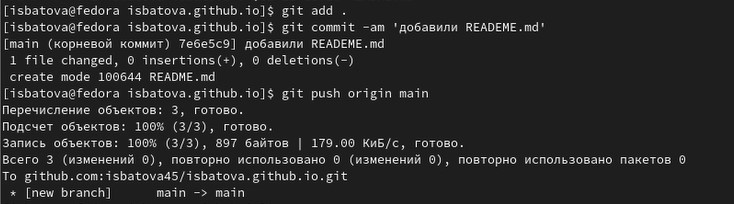
Создание главной ветки

Следующим шагом создаем пустой файл README.md, проверяем, что он создался командой ‘ls -l’ (рис. ??).



Создание файла README.md

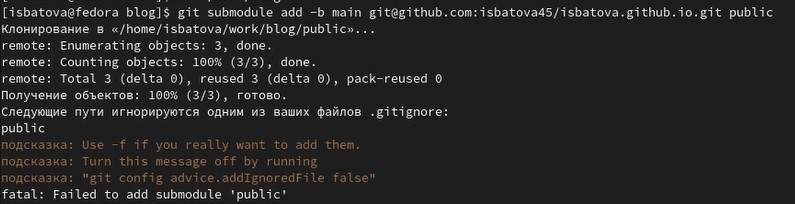
Следующим шагом отправляем изменения в репозиторий, используя команды ‘git add .’, ‘git commit -am’ и ‘git push origin main’ (рис. ??).



Отправление изменений в репозиторий

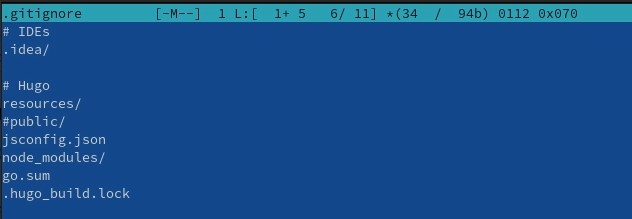
Через браузер проверяем, что наш файл был опубликован.

Далее возвращаемся в терминал и вводим команду ‘git submodule add -b main (ссылка на репозиторий) public’. Команда нам показывает, что ‘public’ находится в файле .gitignore, то есть игнорируется при выполнении команды (рис. ??).



Выполнение команды ‘git submodule add’

Для дальнейшей корректной работы открываем с помощью mc файл .gitignore и символом ‘#’ комментируем ‘public’ (рис. ??).



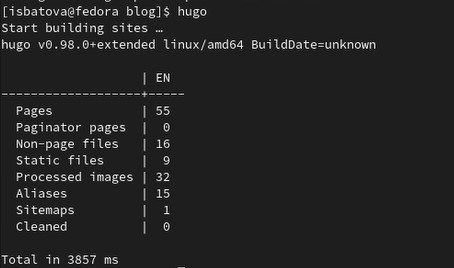
Исправление файла .gitignore

Вновь вводим команду ‘git submodule add -b main (ссылка на репозиторий) public’. В этот раз она выполнена успешно (рис. ??).

Успешное выполнение команды ‘git submodule add’

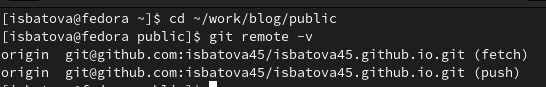
Успешное выполнение команды ‘git submodule add’

Следующим шагом вводим команду ‘hugo’ (рис. ??).



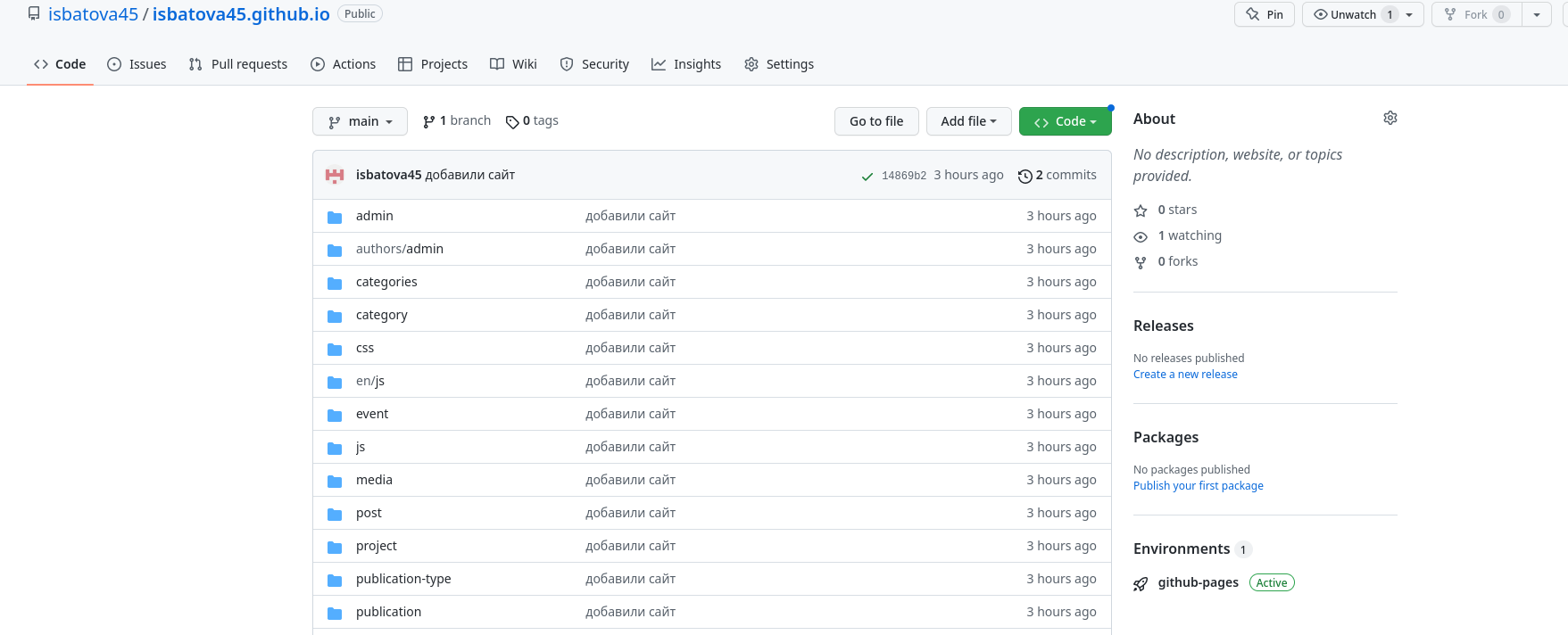
Команда ‘hugo’

Переходим в каталог ‘public’ и вводим команду ‘git remote -v’. Видим, что каталог подключен к правильному репозиторию, то есть все предыдущие действия были выполнены корректно (рис. ??).



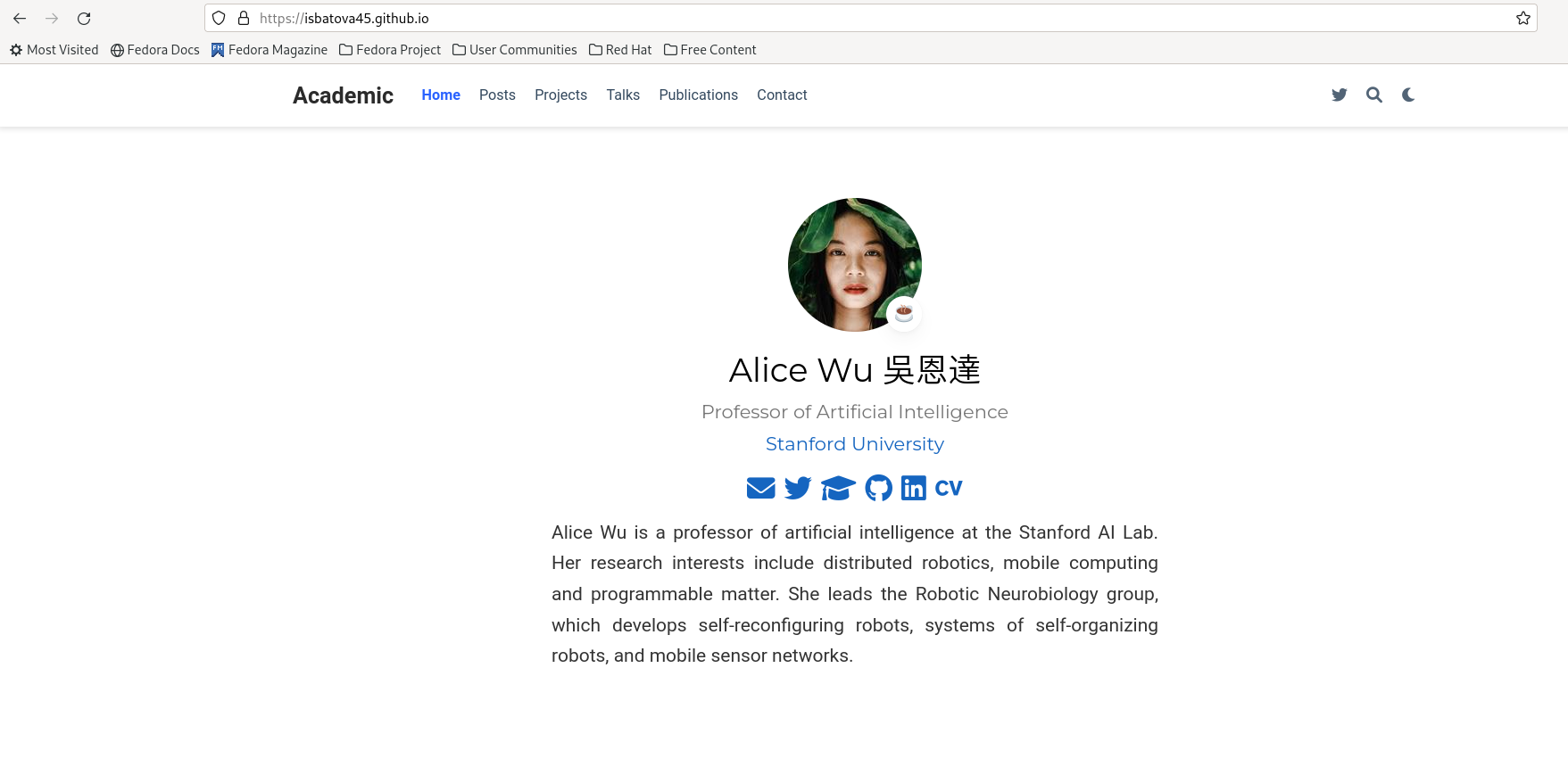
Проверка подключения ‘public’ к репозиторию

Далее отправляем изменения в репозиторий, используя команды ‘git add .’, ‘git commit -am’ и ‘git push origin main’. Заходим в браузер и проверяем, что все выполнено корректно (рис. ??).



Отправление изменений в новый репозиторий

Последним шагом копируем ссылку на наш сайт и вставляем в новую вкладку. Сайт создан (рис. ??).



Сайт создан

# 4 Выводы

В ходе выполнения данного этапа индивидуального проекта я разместила на Github pages заготовки для персонального сайта.