

Отчёт по реализации индивидуального проекта. Часть 5

Дисциплина: Архитектура ЭВМ

Перегудов Александр Вадимович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	19
	Список литературы	20

Список иллюстраций

4.1	Папка admin	8
4.2	Сборка сайта	8
4.3	Запуск процесса для сайта	9
4.4	Папка project	9
4.5	Переименованные папки	9
4.6	Изменённый файл index.md	10
4.7	Заменённая картинка	10
4.8	Заменённая картинка	11
4.9	Изменённый файл index.md	11
4.10	Проекты	12
4.11	Проект	12
4.12	Проект	13
4.13	Папка post	13
4.14	Смена имени для директории	14
4.15	Изменённый файл index.md	14
4.16	Смена имени для директории	15
4.17	Первая часть изменённого файла index.md	15
4.18	Вторая часть изменённого файла index.md	16
4.19	Третья часть изменённого файла index.md	16
4.20	Заменённая картинка	16
4.21	Посты	17
4.22	Часть поста	17
4.23	Часть поста	18

Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . .	7
-----	---	---

1 Цель работы

Приобрести навыки добавления на сайт дополнительных элементов.

2 Задание

Добавить к сайту все остальные элементы. Сделать пост по прошедшей неделе.
Добавить пост по теме “Языки научного программирования”.

3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux

Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую систему
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно про Unix см. в [1–4].

4 Выполнение лабораторной работы

Перешёл в директорию с сайтом (рис. 4.1).

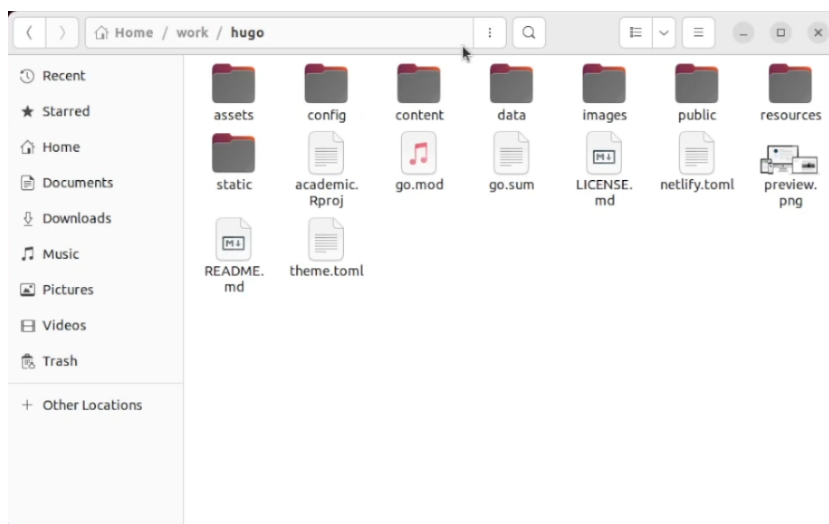


Рис. 4.1: Папка admin

Собрал сайт (рис. 4.2).

```
avperegudov@avperegudov:~/work/hugo$ ~/bin/hugo
hugo: collected modules in 826 ms
Start building sites ...
hugo v0.124.1-db083b05f16c945fec04f745f0ca8640560cf1ec+extended linux/amd64 BuildDate=2024-03-20T11:40:10Z VendorInfo=gohugoio

-----+ EN
Pages           | 60
Paginator pages | 0
Non-page files  | 34
Static files    | 9
Processed images | 76
Aliases         | 15
Cleaned         | 0
Total in 1404 ms
```

Рис. 4.2: Сборка сайта

Запустил процесс для работы сайта (рис. 4.3).

```
avperegudov@avperegudov:~/work/hugo$ ~/bin/hugo server
Watching for changes in /home/avperegudov/{.cache,work}
Watching for config changes in /home/avperegudov/work/hugo/config/_default, /home/avperegudov/.cache/hugo_cache/modules/filecache/modules/pkg/mod/github.com/hugo!blox/hugo-blox-builder/modules/blox-plugin-netlify@v1.1.2-0.20231108141515-0478cf6921f9/config.yaml, /home/avperegudov/.cache/hugo_cache/modules/filecache/modules/pkg/mod/github.com/hugo!blox/hugo-blox-builder/modules/blox-plugin-reveal@v1.1.2/config.yaml, /home/avperegudov/.cache/hugo_cache/modules/filecache/modules/pkg/mod/github.com/hugo!blox/hugo-blox-builder/modules/blox-bootstrap/v5@v5.9.7/hugo.yaml, /home/avperegudov/work/hugo/go.mod
Start building sites ...
hugo v0.124.1-d8083b05f16c945fec04f745f0ca8640560cf1ec+extended linux/amd64 BuildDate=2024-03-20T11:40:10Z VendorInfo=gohugoio
```

Рис. 4.3: Запуск процесса для сайта

Перешёл в директорию content/project (рис. 4.4).

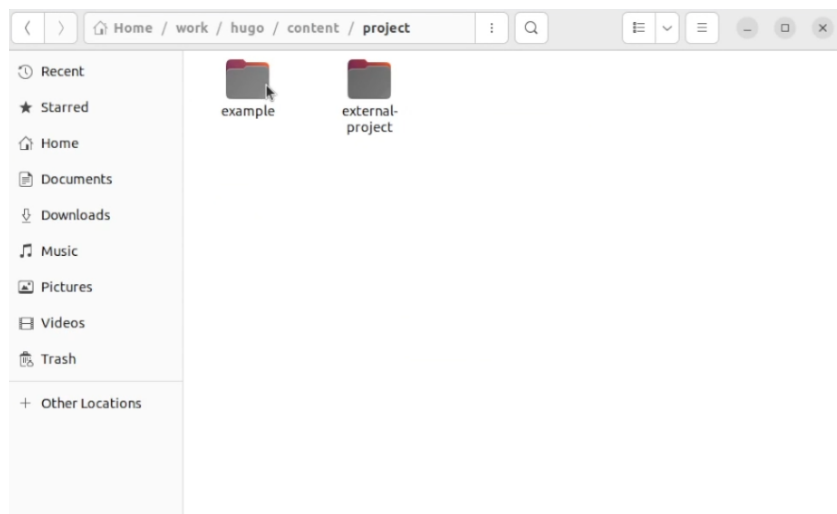


Рис. 4.4: Папка project

Переименовал директории (рис. 4.5).

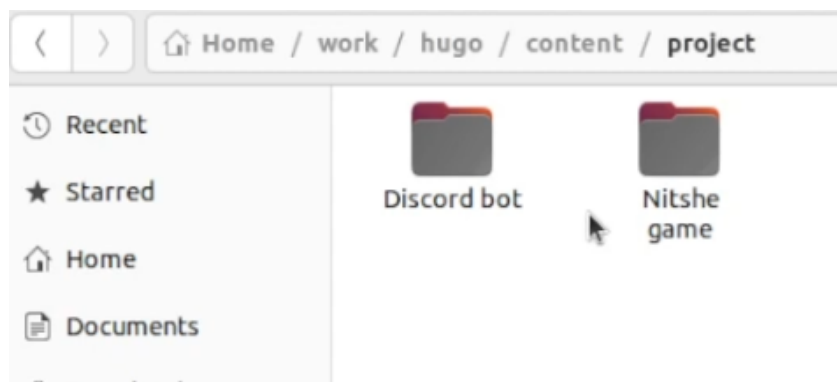
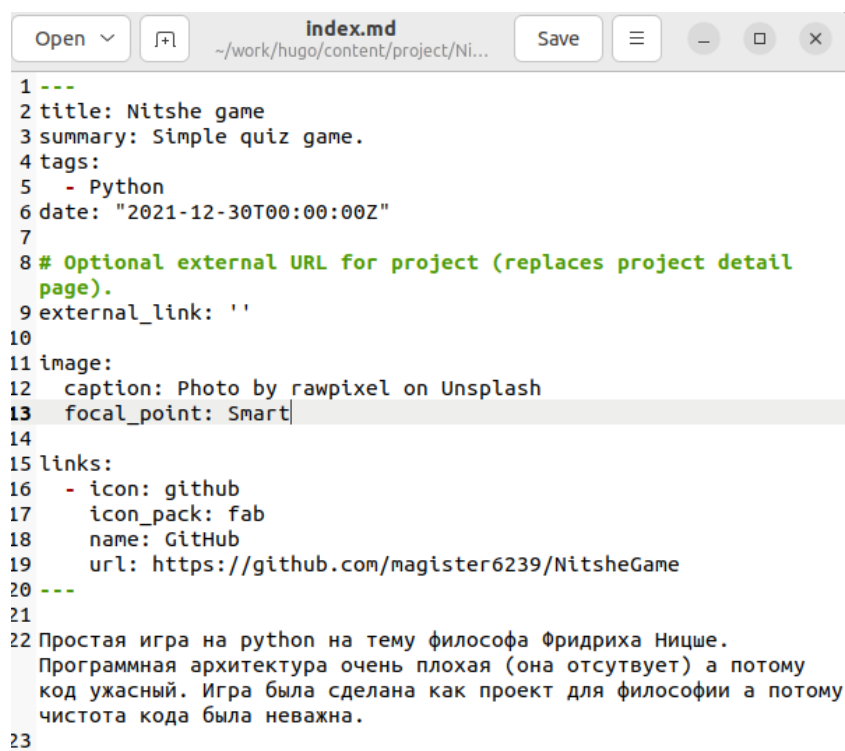


Рис. 4.5: Переименованные папки

Изменил файл index.md в директории Nitshe game (рис. 4.6).



```
1 ---
2 title: Nitshe game
3 summary: Simple quiz game.
4 tags:
5   - Python
6 date: "2021-12-30T00:00:00Z"
7
8 # Optional external URL for project (replaces project detail
9   page).
10 external_link: ''
11
12 image:
13   caption: Photo by rawpixel on Unsplash
14   focal_point: Smart
15
16 links:
17   - icon: github
18     icon_pack: fab
19     name: GitHub
20     url: https://github.com/magister6239/NitsheGame
21
22 ---
23 Простая игра на python на тему философа Фридриха Ницше.
24 Программная архитектура очень плохая (она отсутствует) а потому
25 код ужасный. Игра была сделана как проект для философии а потому
26 чистота кода была неважна.
```

Рис. 4.6: Изменённый файл index.md

Заменил картинку в директории Nitshe game (рис. 4.7).

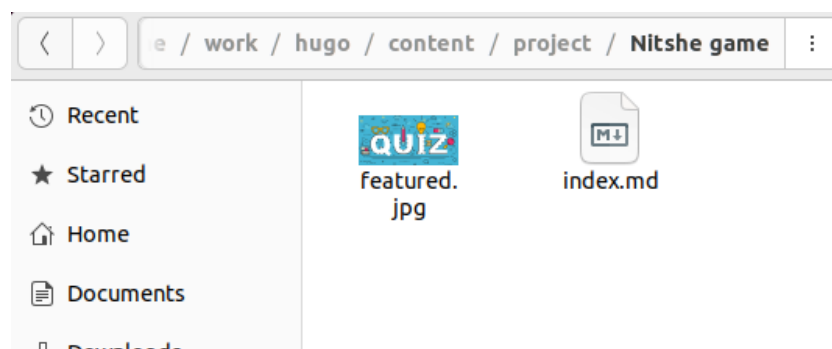


Рис. 4.7: Заменённая картинка

Заменил картинку в директории Discord bot (рис. 4.8).

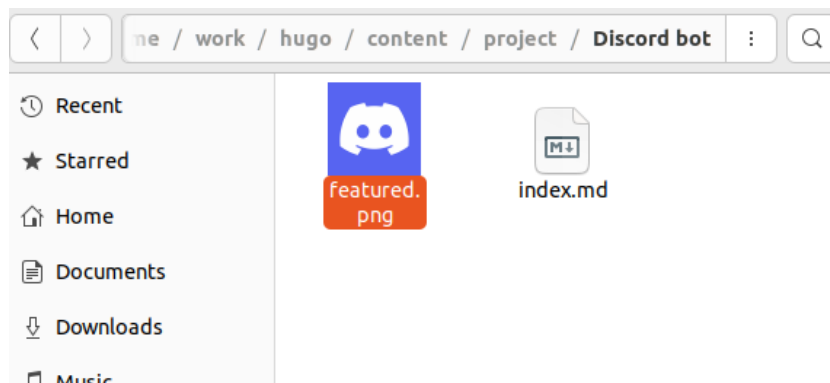


Рис. 4.8: Заменённая картинка

Изменил файл index.md в директории Nitshe game (рис. 4.9).

```

Open  [+] index.md ~/work/hugo/content/project/Di... Save [Menu] [Close] [Maximize]
1 ---
2 title: Discord bot
3 summary: Simple discord bot.
4 tags:
5   - Python
6 date: "2021-06-26T00:00:00Z"
7
8 # Optional external URL for project (replaces project detail
  page).
9 external_link: ''
10
11 image:
12   caption: Photo by Toa Heftiba on Unsplash
13   focal_point: Smart
14
15 links:
16   - icon: github
17     icon_pack: fab
18     name: GitHub
19     url: https://github.com/magister6239/DiscordBot
20 ---
21
22 Простой бот на python написанный в качестве эксперимента. У него
  маленький функционал, но он умеет, то что не умеют другие боты.

```

Рис. 4.9: Изменённый файл index.md

Открыл сайт и проверил список проектов (рис. 4.10).

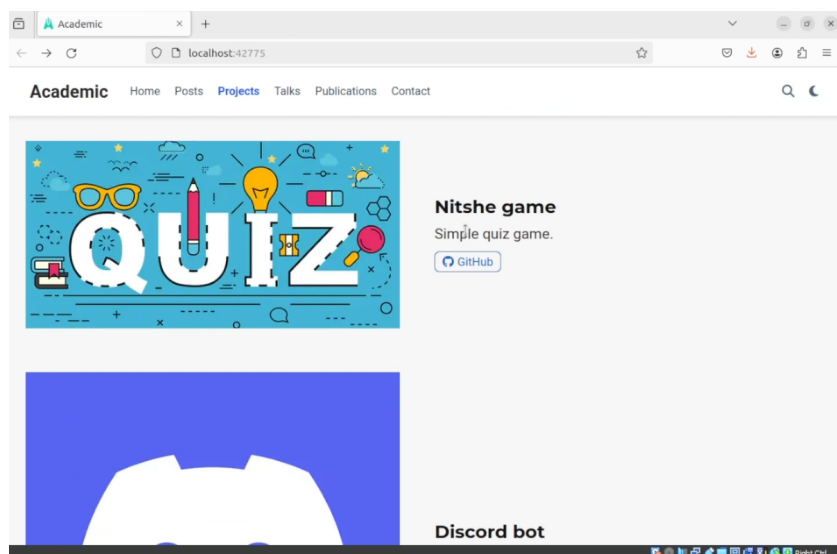


Рис. 4.10: Проекты

Проверил проект Nitshe game (рис. 4.11).

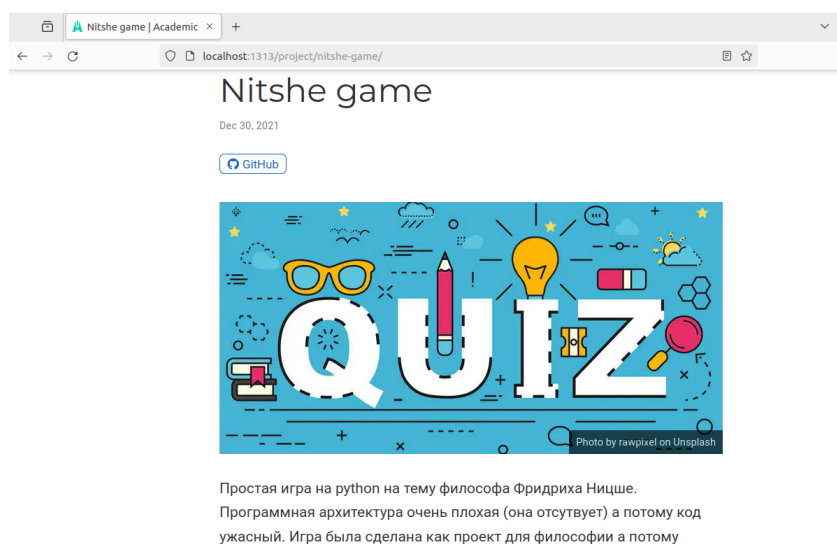


Рис. 4.11: Проект

Проверил проект Discord bot (рис. 4.12).

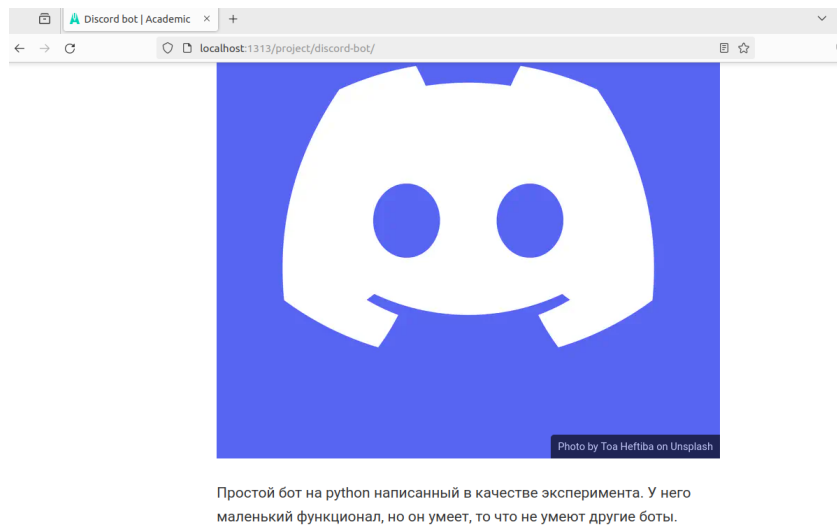


Рис. 4.12: Проект

Перешёл в директорию content/post (рис. 4.13).

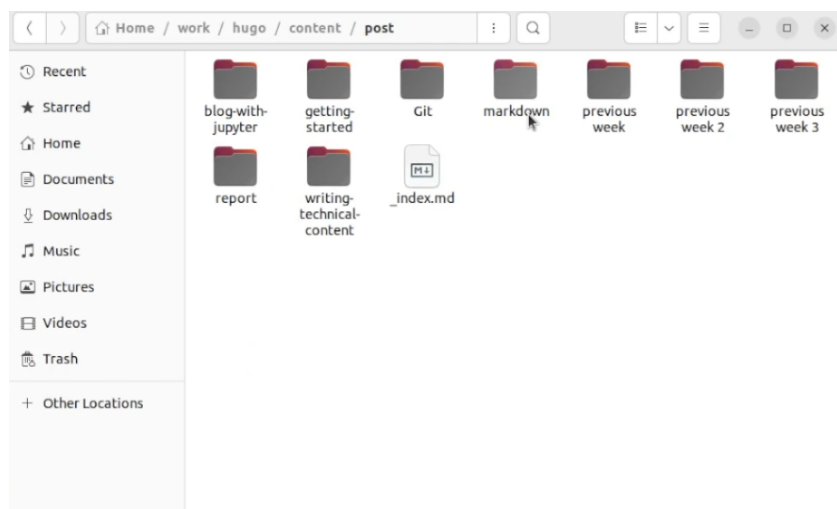


Рис. 4.13: Папка post

Скопировал директорию previous week 3 и переименовал её в previous week 4 (рис. 4.14).

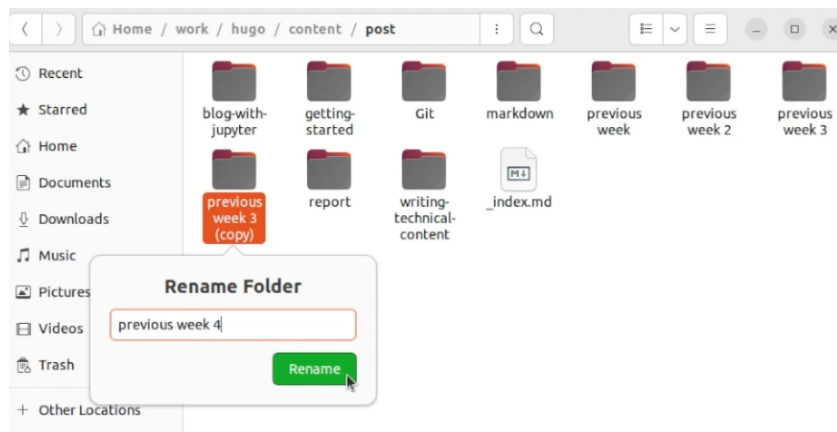


Рис. 4.14: Смена имени для директории

Изменил файл index.md в директории previous week 4 (рис. 4.15).

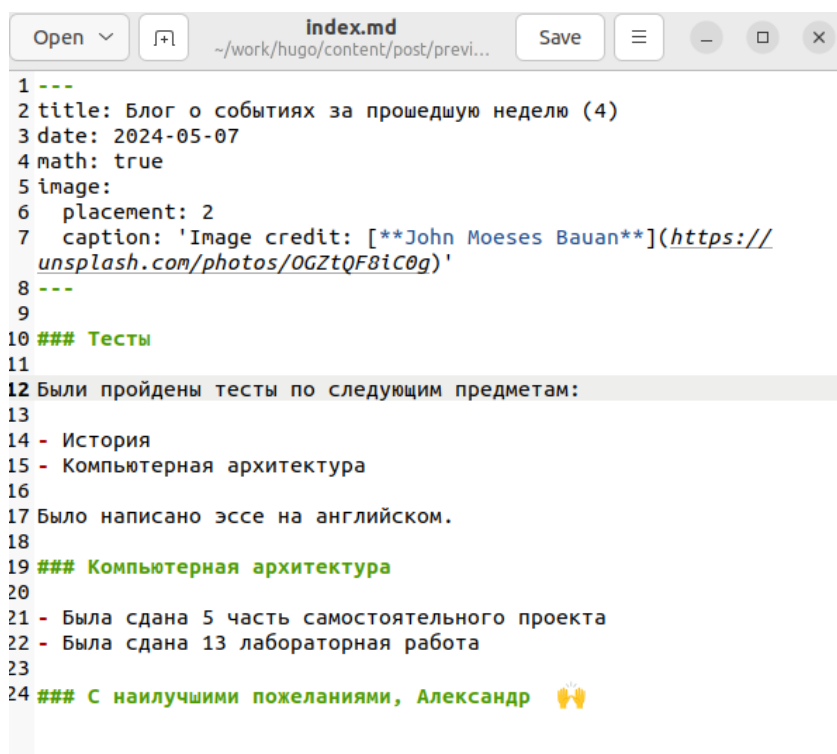


Рис. 4.15: Измененный файл index.md

Скопировал директорию Git и переименовал её в Scintific Programing languages (рис. 4.16).

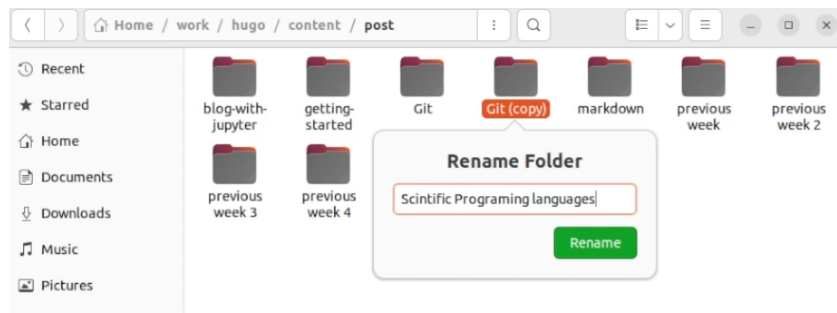


Рис. 4.16: Смена имени для директории

Изменил файл `index.md` в директории `Scintific Programing languages` (рис. 4.17, 4.18, 4.19).

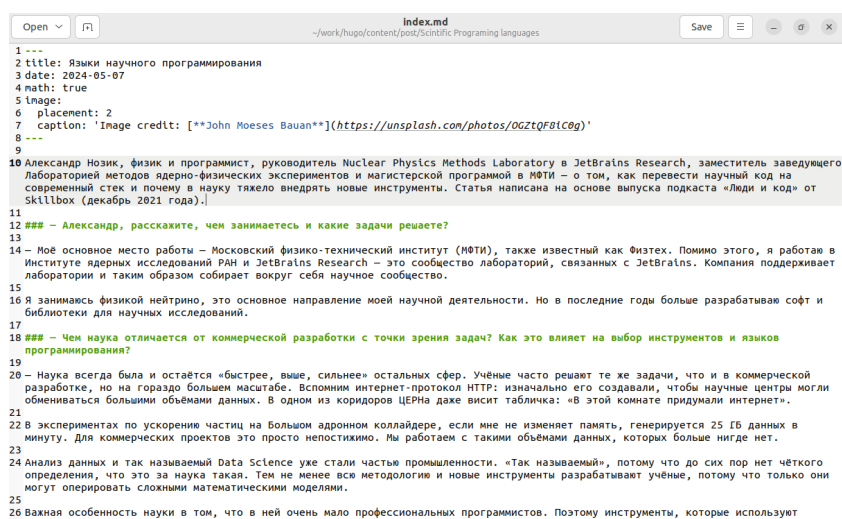


Рис. 4.17: Первая часть измененённого файла `index.md`

Важная особенность науки в том, что в ней очень мало профессиональных программистов. Поэтому инструменты, которые используют учёные, должны быть не только самими гибкими и быстрыми, но и простыми. Иначе учёные-непрограммисты, такого наводят, что потом никто не сможет это разобрать. Собственно, нечто подобное уже случилось в физике, и теперь научное сообщество решает эту проблему.

Дело в том, что физики пишут вычислительный код на C++, а потом оборачивают его в код на Python. И мне кажется, этот подход себя изжил: на Python очень сложно поддерживать большую кодовую базу, а тут размеры проектов растут именно на стороне пользователей кода – учёных. Поэтому такие системы потихоньку рассыпаются и последние пять лет инженеры и учёные ищут более гибкие и простые альтернативы.

– А в других науках используют такую же связку языков?

– Да, везде примерно то же самое, но с небольшой разницей. Например, биоинформатика как отдельная область появилась относительно недавно. В отличие от физиков, которым пришлось «пересаживаться» на Python, там сразу стали работать на современных языках. Но в большинстве наук всё ещё используют библиотеки C, Fortran и C++, а над ними пишут надстройку на Python. И там сейчас те же самые проблемы с гибкостью.

Кроме Python в разных областях науки пишут или пытались писать на других языках.

R. Его используют в статистике. Это узкоспециализированный язык, который отлично подходит для решения статистических задач. Но часто нам надо не только получить данные, но и сделать веб-сервис, чтобы пользователь имел доступ к этим данным. Написать его на R – дело непростое.

Julia. Это довольно интересный язык со множеством конструктивных особенностей. Попробуйте его, если вам не хватает скорости или гибкости в Python. Хотя и у Julia есть недостаток: его набор инструментов всё ещё нестабильный.

Julia придумали как альтернативу Python и MATLAB. Последний до сих пор используют, но это проприетарная, «неживая» система. Как только вы выходите за рамки привычных задач, то сразу ощущаете его ограниченность.

Swift. Из Swift тоже пытались сделать универсальный язык, но он так и не вышел за рамки iOS. А потом появился Kotlin, который по синтаксису сильно напоминает Swift, но при этом подходит для решения более широкого спектра задач и позволяет работать с библиотеками из Java, JavaScript и C.

Java. Раньше я программировал на Java – это классный язык, который часто несправедливо ругают. Его создавали для энтерпрайза, поэтому там чересчур затянута «церемония»: чтобы собрать простое приложение, надо написать много дополнительного кода. Да, это упрощает поддержку и повышает стабильность приложения, но сильно усложняет сам процесс программирования.

Kotlin. Сейчас я много пишу на Kotlin. Он обладает достоинствами Java, но избавляет программиста от доброй половины «церемоний», и потому, на мой взгляд, у него большие перспективы.

47 Какие основные проблемы в научном IT и как их пытаются решить

48

49 **### – Вы упомянули, что в науке было много всего наворочено. Интересно, это относится только к библиотекам на Fortran и C++ или к**

50 **цену-то ещё? И в чём там главные проблемы?**

51

52 Это очень сильно отстали, потому что продолжали разрабатывать ПО в том же стиле, в котором делали это на допотопных мейнфреймах. Они придумывали замечательные алгоритмы, но совсем не думали о поддержке кода, тестировании, CI/CD и тем более об архитектуре.

53 Например, системы сбора данных совершенно не умеют взаимодействовать друг с другом. Их писали разные программисты, которые иногда специально делали так, чтобы системы не могли общаться между собой. Если два эксперимента выполнены на разных системах, то учёному придётся обрабатывать их результаты по отдельности. Более того, на изучение каждой системы может уйти несколько лет – настолько они сложные.

54

55 Из-за того, что в науке не внедряли современные подходы к проектированию ПО и не использовали новые инструменты, у нас накопилось миллионы строк легаси. Там можно встретить хорошие и даже гениальные алгоритмы, но понять их будет практически невозможно. Чтобы почувствовать всю боль, найдите программу на Fortran 77 и попробуйте понять, что там происходит.

56

57 **### Для справки: в Fortran 77 длина имени переменной ограничена восемью символами. Это значит, что, когда заканчиваются все**

58 **подходящие имена переменных, программа становится нечитаемой. В код идут числовые идентификаторы, с которыми код становится**

59 **только запутанней. А ведь большая часть программ в науке всё ещё написана на Fortran.**

60

61 Относительно недавно учёные поняли, что нужно инвестировать в software engineering, уделять внимание архитектуре и осваивать современные инструменты, а не гордиться свои велосипеды. К моему удивлению, за последние два года появилось много групп инженеров, которые целенаправленно продвигают современный подход в науке.

62 **### С наилучшими пожеланиями. Александр 🍌**

Рис. 4.18: Вторая часть изменённого файла index.md

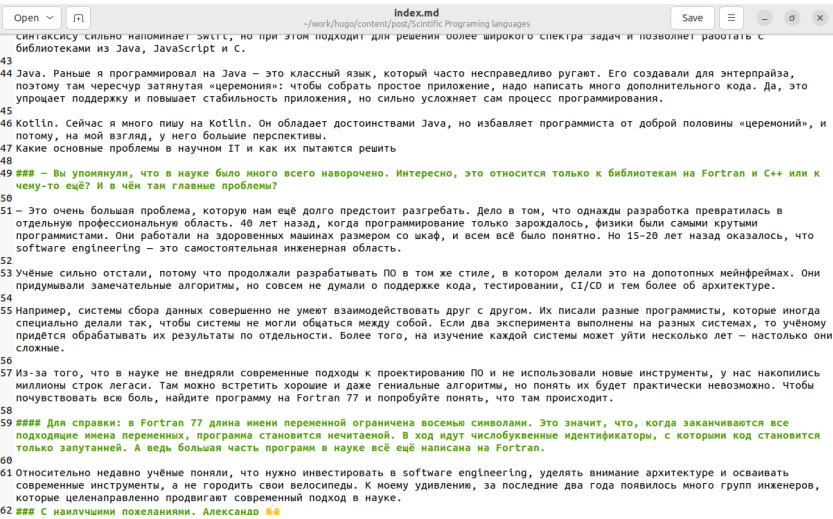


Рис. 4.19: Третья часть изменённого файла index.md

Заменяю картинку в директории Scintific Programing languages (рис. 4.20).

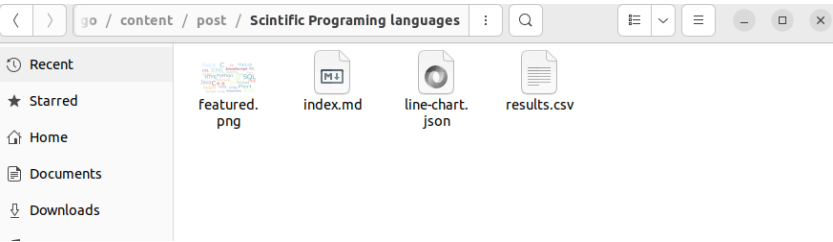


Рис. 4.20: Заменённая картинка

Открыл сайт и проверил список постов (рис. 4.21).

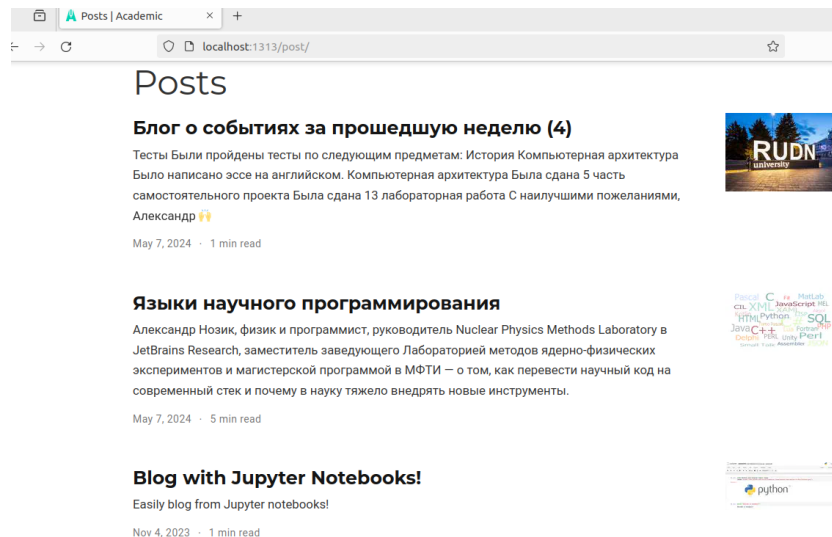


Рис. 4.21: Посты

Проверил пост за прошедшую неделю (4) (рис. 4.22).

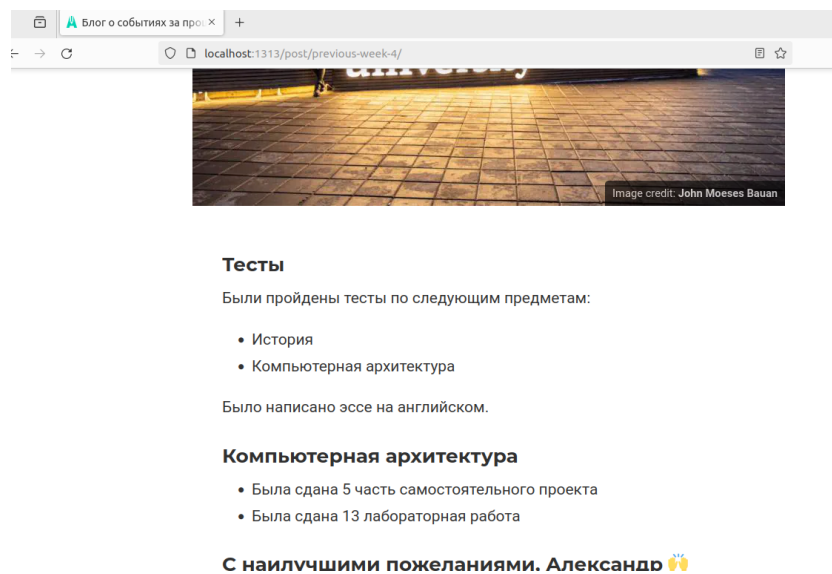


Рис. 4.22: Часть поста

Проверил пост на тему языки научного программирования (рис. 4.23).

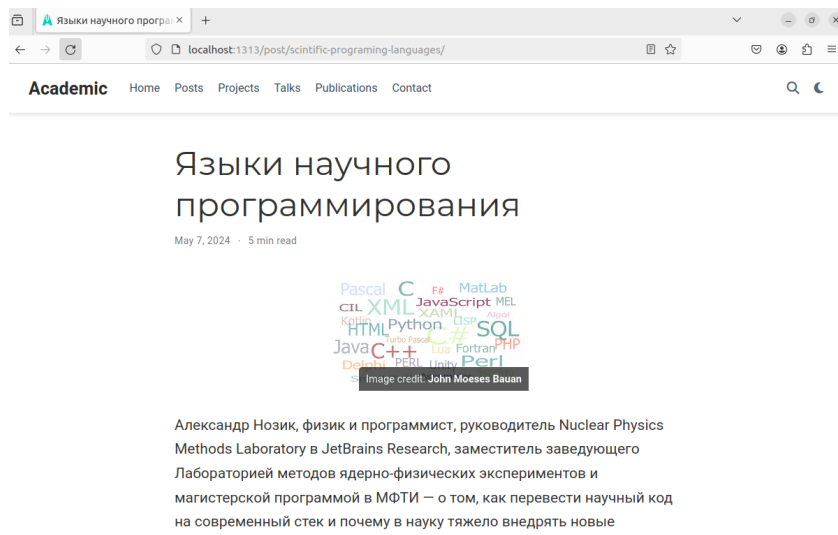


Рис. 4.23: Часть поста

5 Выводы

Были приобретены навыки изменения блока проектов.

Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
2. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.