

Индивидуальный проект: Персональный сайт научного работника

Первый этап реализации

Ермишина Мария Кирилловна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение этапа реализации проекта	7
4	Выводы	17
	Список литературы	18

Список иллюстраций

3.1	Скачиваем файл	7
3.2	Разархивируем и переносим	8
3.3	Добавляем репозиторий	9
3.4	Клонируем данные	10
3.5	Запуск сервера	11
3.6	Проверка сайта	12
3.7	Новый репозиторий	13
3.8	Клонирование данных	13
3.9	Создание файла	14
3.10	Обновление информации на git	14
3.11	Саб-модуль	15
3.12	Запуск сервера	15
3.13	Обновление информации	16
3.14	Обновление информации	16

Список таблиц

1 Цель работы

Целью данного этапа индивидуального проекта является размещение на Github pages заготовки для персонального сайта. [1]

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 этапа реализации проекта

Проект мы будем реализовывать с помощью hugo.

1. Для начала выполнения проекта нам необходимо установить hugo. Устанавливать файлы мы будем с сайта GitHub вручную. Выбираем необходимый файл и скачиваем его. (рис. 3.1)

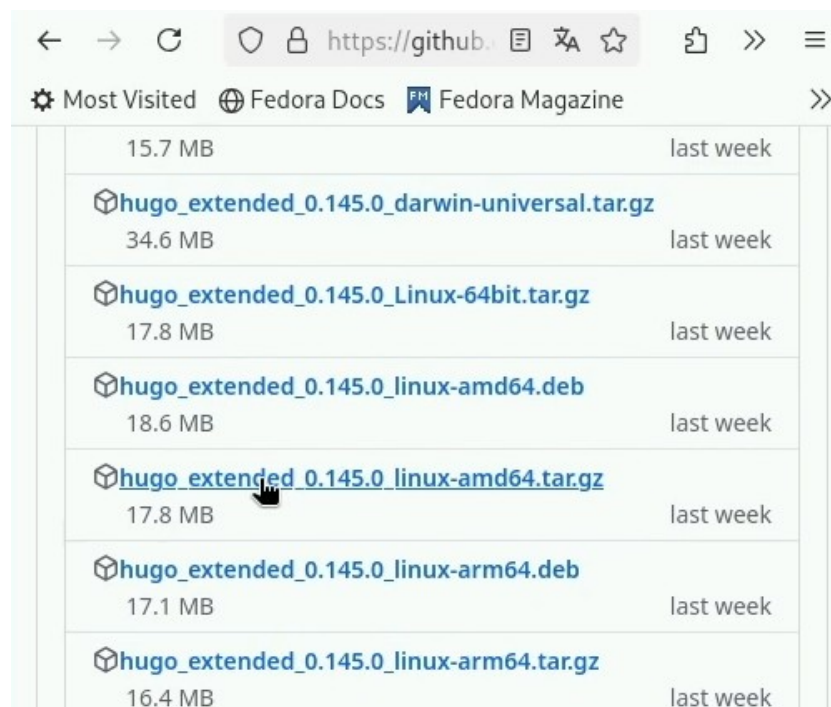


Рис. 3.1: Скачиваем файл

2. После загрузки файла нам необходимо разархивировать файл с помощью следующей команды: (рис. 3.2)

- `tar -xvf` (название скаченного файла) (название нового файла) В моём случае новый файл будет называться `hugo`.

3. Далее мы переносим новый файл в `/usr/local/bin`: (рис. 3.2)

- `sudo mv /usr/local/bin` (в данном случае перенос осуществляется только с помощью супер-пользователя) И проверяем наличие файла с помощью `ls`.

```
[ermimash@ermimash ~]$ cd Загрузки
[ermimash@ermimash Загрузки]$ ls
hugo_extended_0.145.0_linux-amd64.tar.gz  pandoc-crossref-Linux.tar.xz
pandoc-crossref.1
[ermimash@ermimash Загрузки]$ tar -xvf hugo_extended_0.145.0_linux-amd64.tar.gz
hugo
README.md
LICENSE
[ermimash@ermimash Загрузки]$ ls
hugo                                pandoc-crossref.1
hugo_extended_0.145.0_linux-amd64.tar.gz  pandoc-crossref-Linux.tar.xz
LICENSE                                README.md
[ermimash@ermimash Загрузки]$ sudo mv hugo /usr/local/bin
[sudo] пароль для ermimash:
[ermimash@ermimash Загрузки]$ ls /usr/local/bin
hugo  pandoc-crossref
[ermimash@ermimash Загрузки]$
```

Рис. 3.2: Разархивируем и переносим

4. Нам необходимо добавить на наш аккаунт `git` репозиторий, который указан в технической реализации проекта. Я назвала его `Blog`. (рис. 3.3)

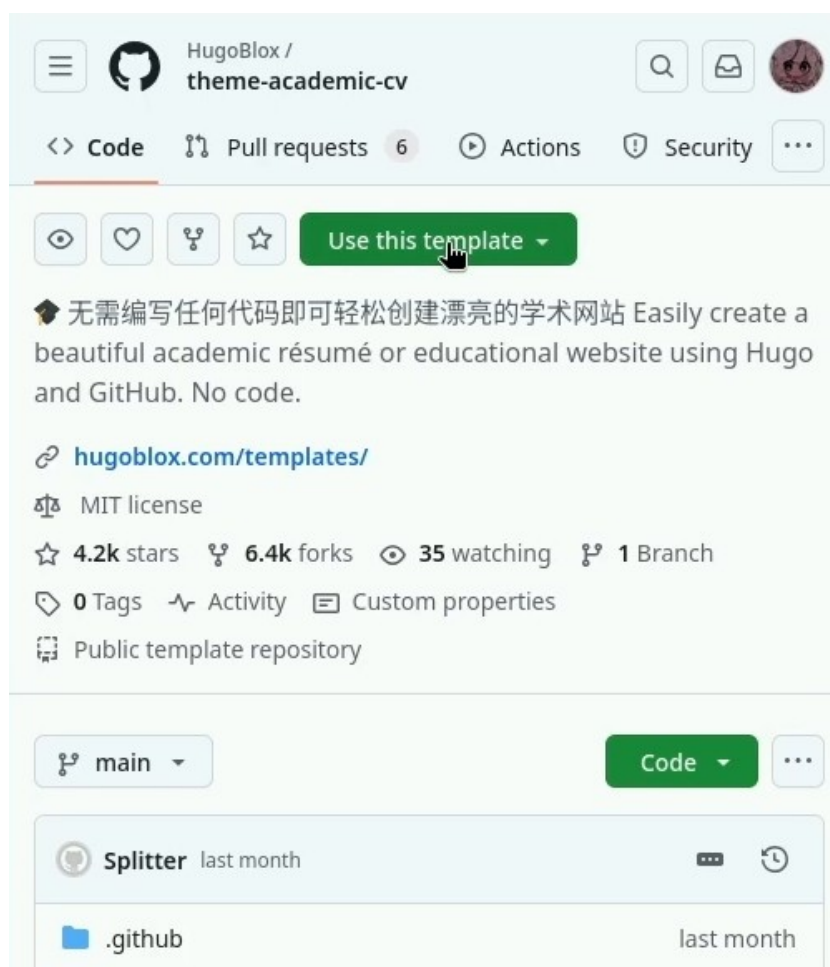


Рис. 3.3: Добавляем репозиторий

5. Далее мы переходим в каталог `work`, в который нам необходимо клонировать данные из нашего нового репозитория. Воспользуемся следующими командами: (рис. 3.4)

- `cd work` (переходим в каталог)
- `git clone -recursive (SSH нашего репозитория)`

```
[root@ermimash ~]# cd work/
[root@ermimash work]# git clone --recursive git@github.com:ErmiMash/blog.git
Клонирование в «blog»...
Enter passphrase for key '/root/.ssh/id_rsa':
Enter passphrase for key '/root/.ssh/id_rsa':
remote: Enumerating objects: 98, done.
remote: Counting objects: 100% (98/98), done.
remote: Compressing objects: 100% (84/84), done.
remote: Total 98 (delta 5), reused 75 (delta 1), pack-reused 0 (from 0)
Получение объектов: 100% (98/98), 4.12 МиБ | 6.23 МиБ/с, готово.
Определение изменений: 100% (5/5), готово.
[root@ermimash work]# ls
```

Рис. 3.4: Клонирование данных

6. После добавления всех файлов мы переходим в каталог `blog`, который появился после клонирования файлов. Теперь мы можем запустить наш сервер, с помощью которого наш сайт начнёт работать. Воспользуемся следующими командами: (рис. 3.5)

- `cd blog` (переходим в каталог)
- `hugo server` (запускаем сервер)

```

[root@ermimash ~]# ls
anaconda-ks.cfg  blog  work
[root@ermimash ~]# cd work
[root@ermimash work]# ls
blog  study
[root@ermimash work]# cd work/blog
-bash: cd: work/blog: Нет такого файла или каталога
[root@ermimash work]# cd blog
[root@ermimash blog]# ls
assets  content  hugoblox.yaml  LICENSE.md  README.md
config  go.mod    layouts        netlify.toml  static
[root@ermimash blog]# hugo server
hugo: downloading modules ...
hugo: collected modules in 9479 msWatching for changes in /root/./.cache/work}
Watching for config changes in /root/work/blog/config/_default, /root/./.cache/hugo_cache/modules/filecache/modules/pkg/mod/github.com/hugo!blox/hugo-blox-builder/module
s/blox-plugin-netlify@v1.1.2-0.20231209203044-d31adfedd40b/config.yaml, /root/./.cache
/hugo_cache/modules/filecache/modules/pkg/mod/github.com/hugo!blox/hugo-blox-builder/modules/blox-tailwind@v0.3.1/hugo.yaml, /root/work/blog/go.mod
Start building sites ...
hugo v0.145.0-666444f0a52132f9fec9f71cf25b441cc6a4f355+extended linux/amd64 BuildDate=2025-02-26T15:41:25Z VendorInfo=gohugoio

| EN
-----+-----
Pages | 66
Paginator pages | 0
Non-page files | 23
Static files | 1
Processed images | 85
Aliases | 18
Cleaned | 0

Built in 4042 ms
Environment: "development"
Serving pages from disk
Running in Fast Render Mode. For full rebuilds on change: hugo server --disableFastRender
Web Server is available at http://localhost:1313/ (bind address 127.0.0.1)
Press Ctrl+C to stop

```

Рис. 3.5: Запуск сервера

Переходим по ссылке, которая появилась в терминале, и проверяем работу нашего сайта. Всё работает, на сайте видна информация, указанная в репозитории. (рис. 3.6)

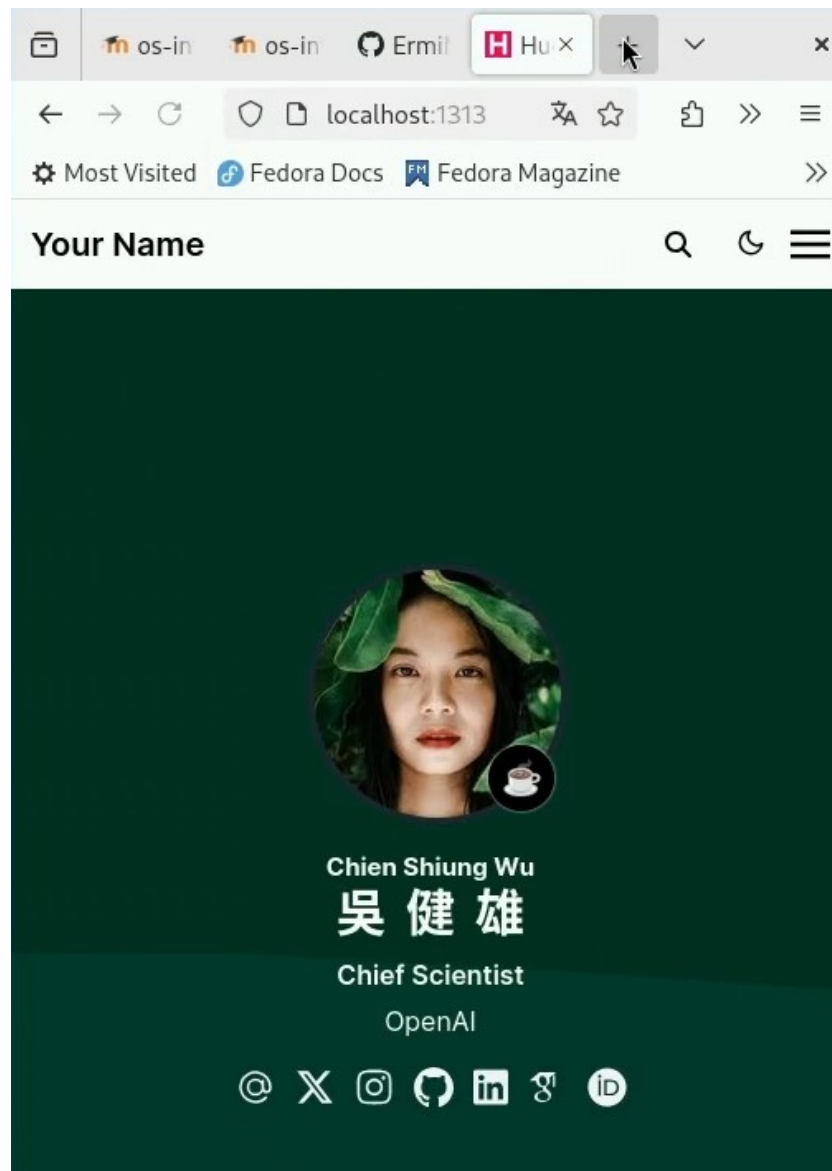


Рис. 3.6: Проверка сайта

7. Далее мы создаем новый пустой репозиторий. Его необходимо назвать следующим образом: ник_git.github.io - это параметр URLs сайта. (рис. 3.7)

Owner *

ErmiMash

Repository name *

ErmiMash.github.io

✓ ErmiMash.github.io is available.

Great repository names are short and memorable. Need inspiration?
How about **probable-broccoli** ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Рис. 3.7: Новый репозиторий

8. После создания нового репозитория нам необходимо клонировать его данные в наш каталог work. Воспользуемся следующими командами: (рис. 3.8)

- `cd ..` (возвращаемся в каталог work)
- `git clone --recursive git@github.com:ErmiMash/ErmiMash.github.io.git` И проверяем наличие нового каталога с помощью `ls`.

```
[root@ermimash blog]# cd ..
[root@ermimash work]# git clone --recursive git@github.com:ErmiMash/ErmiMash.github.io.git
Клонирование в «ErmiMash.github.io»...
Enter passphrase for key '/root/.ssh/id_rsa':
warning: Похоже, что Вы клонировали пустой репозиторий.
[root@ermimash work]# ls
blog  ErmiMash.github.io  study
[root@ermimash work]#
```

Рис. 3.8: Клонирование данных

Переходи в новый каталог и переключаемся на ветку “main”. После этого создаём текстовый файл README.md. Воспользуемся следующими командами: (рис.

3.9) - cd ErmiMash.github.io - git checkout -b main - touch README.md

Так как мы создали новый файл - нам необходимо обновить информацию на GitHub: (рис. 3.10) - git add . - git commit -am "Create README.md"

```
[root@ermimash work]# cd ErmiMash.github.io
[root@ermimash ErmiMash.github.io]# git checkout -b main
Переключились на новую ветку «main»
[root@ermimash ErmiMash.github.io]# touch README.md
[root@ermimash ErmiMash.github.io]# ls
README.md
```

Рис. 3.9: Создание файла

```
[root@ermimash ErmiMash.github.io]# git add .
[root@ermimash ErmiMash.github.io]# git commit -am "Create README.md"
[main (корневой коммит) ceddde3] Create README.md
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
[root@ermimash ErmiMash.github.io]# git push
Enter passphrase for key '/root/.ssh/id_rsa':
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 866 байтов | 866.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:ErmiMash/ErmiMash.github.io.git
 * [new branch]      main -> main
```

Рис. 3.10: Обновление информации на git

9. Нам необходимо указать пустой репозиторий как саб-модуль для нашего сервера - это необходимо для нашего удобства: (рис. 3.11)

- git submodule -b main (SSH пустого репозитория)

```
[root@ermimash ErmiMash.github.io]# cd ..
[root@ermimash work]# cd blog
[root@ermimash blog]# git submodule -b main git@github.com:ErmiMash/ErmiMash.github.io.git
использование: git submodule [--quiet] [--cached]
    or: git submodule [--quiet] add [-b <branch>] [-f|--force] [--name <name>] [--reference <repository>] [--] <repository> [<path>]
    or: git submodule [--quiet] status [--cached] [--recursive] [--] [<path>...]
    or: git submodule [--quiet] init [--] [<path>...]
    or: git submodule [--quiet] deinit [-f|--force] [--all] [--] [<path>...]
    or: git submodule [--quiet] update [--init [--filter=<filter-spec>]] [--remote] [-N|--no-fetch] [-f|--force] [--checkout|--merge|--rebase] [--[no-]recommend-shallow] [--reference <repository>] [--recursive] [--[no-]single-branch] [--] [<path>...]
    or: git submodule [--quiet] set-branch (--default|--branch <branch>) [--] [<path>]
    or: git submodule [--quiet] set-url [--] [<path>] <newurl>
    or: git submodule [--quiet] summary [--cached|--files] [--summary-limit <n>] [commit] [--] [<path>...]
    or: git submodule [--quiet] foreach [--recursive] <command>
    or: git submodule [--quiet] sync [--recursive] [--] [<path>...]
    or: git submodule [--quiet] absorbgitdirs [--] [<path>...]
[root@ermimash blog]#
```

Рис. 3.11: Саб-модуль

Повторно запускаем наш сервер, чтобы проверить работоспособность. (рис. 3.12)

```
[root@ermimash blog]# hugo server
Watching for changes in /root/{.cache,work}
Watching for config changes in /root/work/blog/config/_default, /root/.cache/hugo_cache/modules/filecache/modules/pkg/mod/github.com/hugo!blox/hugo-blox-builder/modules/blox-plugin-netlify@v1.1.2-0.20231209203044-d31adfedd40b/config.yaml, /root/.cache/hugo_cache/modules/filecache/modules/pkg/mod/github.com/hugo!blox/hugo-blox-builder/modules/blox-tailwind@v0.3.1/hugo.yaml, /root/work/blog/go.mod
Start building sites ...
hugo v0.145.0-666444f0a52132f9fec9f71cf25b441cc6a4f355+extended linux/amd64 BuildDate=2025-02-26T15:41:25Z VendorInfo=gohugoio

| EN
-----+-----
Pages | 66
Paginator pages | 0
Non-page files | 23
Static files | 1
Processed images | 85
Aliases | 18
Cleaned | 0

Built in 685 ms
Environment: "development"
Serving pages from disk
Running in Fast Render Mode. For full rebuilds on change: hugo server --disableFastRender
Web Server is available at http://localhost:1313/ (bind address 127.0.0.1)
Press Ctrl+C to stop
```

Рис. 3.12: Запуск сервера

10. После проделанной работы нам необходимо загрузить всю информацию на сервер git. Делаем это с помощью следующих команд: (рис. 3.13), (рис. 3.14)

- `git add .`
- `git commit -am "added site"`
- `git push origin main`

```
[root@ermimash blog]# git commit -am "added site"
[main 11df9a7] added site
91 files changed, 666 insertions(+)
create mode 100644 .gitmodules
create mode 100644 .hugo_build.lock
create mode 160000 ErmiMash.github.io
create mode 100644 go.sum
create mode 100644 hugo_stats.json
create mode 160000 public
create mode 100644 resources/_gen/images/author/吴健雄/avatar_hu_2edebb31100122a9.jpg
create mode 100644 resources/_gen/images/author/吴健雄/avatar_hu_4487bc68c2269668.jpg
create mode 100644 resources/_gen/images/author/吴健雄/avatar_hu_4bb3a690b6cee14a.webp
create mode 100644 resources/_gen/images/event/example/featured_hu_1b041d148278d72.webp
create mode 100644 resources/_gen/images/event/example/featured_hu_23c116e63cb8c936.jpg
create mode 100644 resources/_gen/images/event/example/featured_hu_3c64af3e7ddfe5a0.jpg
```

Рис. 3.13: Обновление информации

```
[root@ermimash blog]# git push origin main
Enter passphrase for key '/root/.ssh/id_rsa':
Перечисление объектов: 102, готово.
Подсчет объектов: 100% (102/102), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (96/96), готово.
Запись объектов: 100% (101/101), 2.19 Миб | 3.08 Миб/с, готово.
Total 101 (delta 2), reused 1 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:ErmiMash/blog.git
 33a3a94..11df9a7 main -> main
[root@ermimash blog]#
```

Рис. 3.14: Обновление информации

4 Выводы

В рамках данного этапа индивидуального проекта разместили на Github pages заготовки для персонального сайта.

Список литературы

1. Д. К. Этапы реализации проекта. RUDN.