

Отчёт по 1-му этапу персонального проекта

Размещение на Github pages заготовки для персонального сайта.

Ганина Таисия, НКАбд-01-22

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	17
	Список литературы	18

Список иллюстраций

4.1	Скачивание нужного архива	9
4.2	Распаковка архива	9
4.3	Создание папки bin для использования hugo	10
4.4	Перенос исполняемого файла в папку bin	10
4.5	Нужный репозиторий	11
4.6	Клонирование репозитория	11
4.7	Установка модуля golang	11
4.8	Bulding site	12
4.9	Удаление каталога public	12
4.10	Подключение к серверу	13
4.11	Локальная страничка	13
4.12	Создание специального репозитория	14
4.13	Клонирование пока что пустого репозитория	14
4.14	Переключение на новую ветку и создание файла README	14
4.15	Создание модуля в папке public	15
4.16	Убрать public из списка игнорируемых	15
4.17	Добавление новых данных на Github	16
4.18	Готовый URL	16

Список таблиц

1 Цель работы

Создать репозиторий для дальнейшей работы с проектом. Создать URL для сайта и оформить шаблон.

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

В ходе выполнения первого этапа проекта было произведено много работы с использованием системы контроля версий Git. Так что, как я считаю, будет уместно привести часть команд и подробнее рассказать про эту систему.

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется.

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

Системы контроля версий поддерживают возможность отслеживания и разре-

шения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.

Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

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.

4 Выполнение лабораторной работы

1. Установить необходимое программное обеспечение (рис. 4.1, 4.2, 4.3, 4.4).

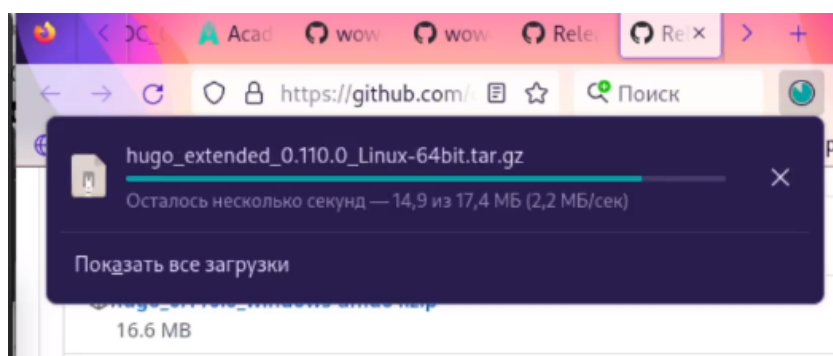


Рис. 4.1: Скачивание нужного архива

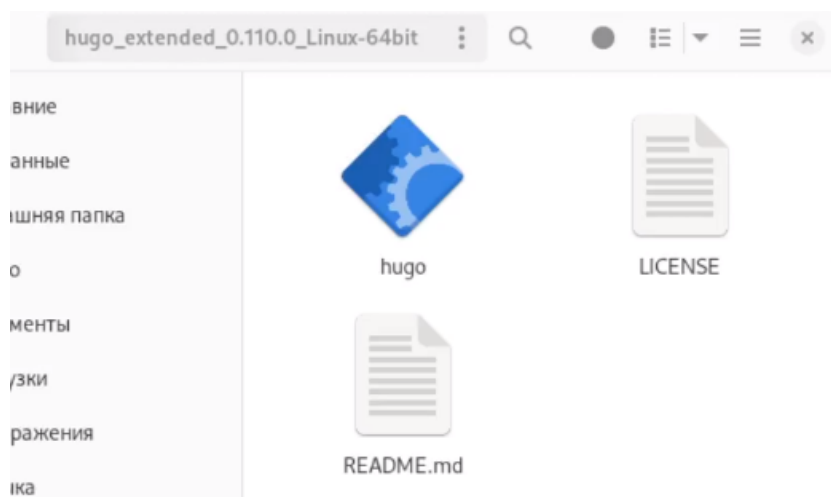


Рис. 4.2: Распаковка архива

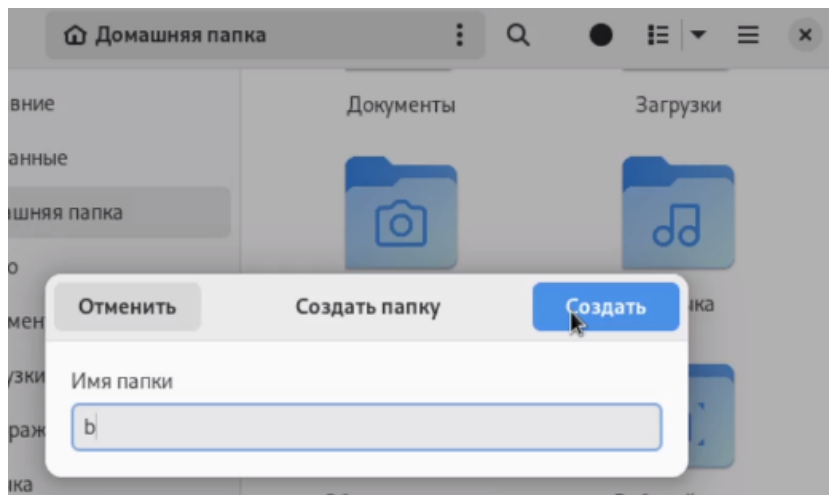


Рис. 4.3: Создание папки bin для использования hugo

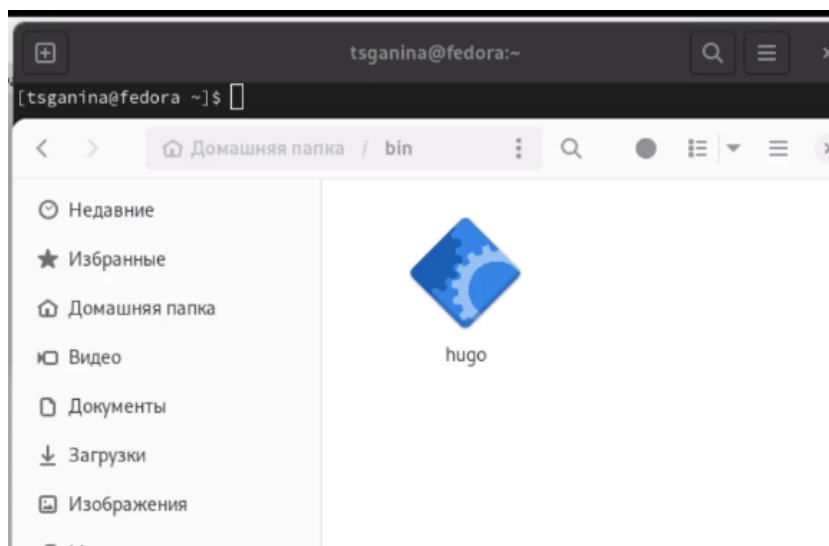


Рис. 4.4: Перенос исполняемого файла в папку bin

2. Скачать шаблон темы сайта (рис. 4.5)

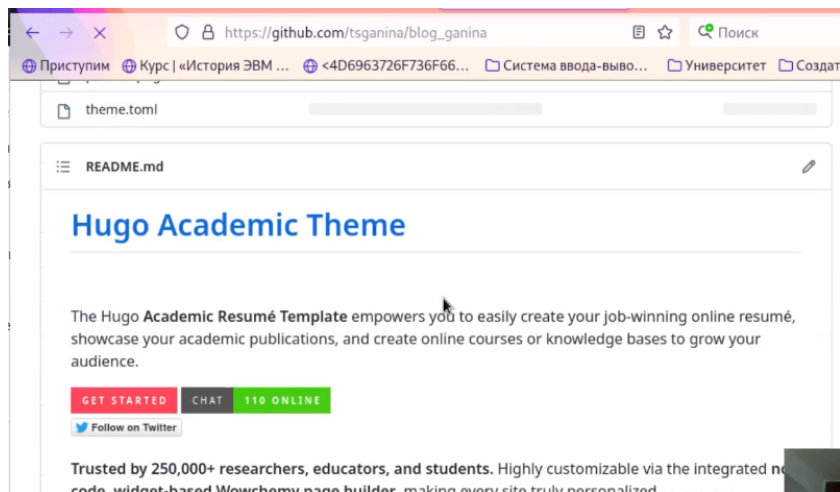


Рис. 4.5: Нужный репозиторий

3. Разместить его на хостинге git (рис. 4.6, 4.7, 4.8, 4.9, 4.10, 4.11).

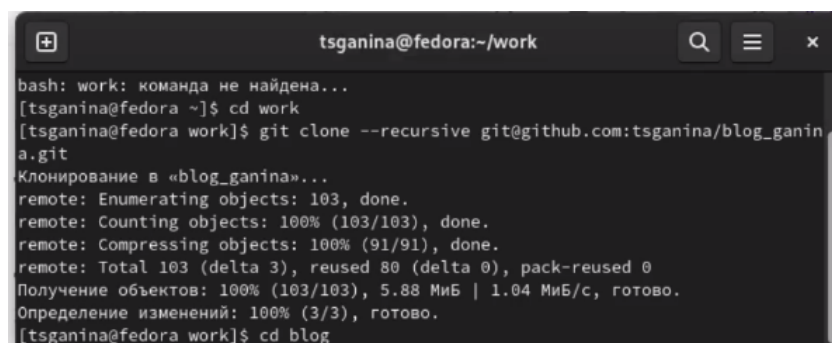


Рис. 4.6: Клонирование репозитория

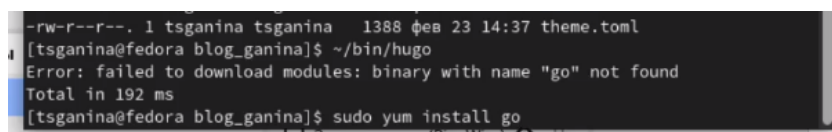


Рис. 4.7: Установка модуля golang

```
Выполнено!  
[tsganina@fedora blog_ganina]$ ~/bin/hugo  
hugo: downloading modules ...  
hugo: collected modules in 25488 ms  
Start building sites ...  
hugo v0.110.0-e32a493b7826d02763c3b79623952e625402b168+extended linux/amd64 BuildDate=2023-01-17T12:16:09Z VendorInfo=gohugoio
```

Рис. 4.8: Bulding site

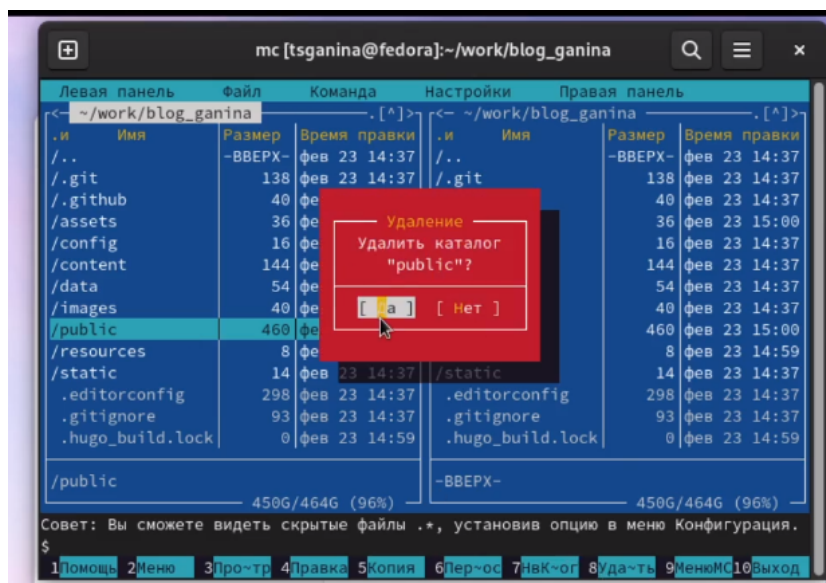


Рис. 4.9: Удаление каталога public

```
tsganina@fedora:~/work/blog_ganina
Total in 58659 ms
[tsganina@fedora blog_ganina]$ ls -l
итого 248
-rw-r--r--. 1 tsganina tsganina 258 фев 23 14:37 academic.Rproj
drwxr-xr-x. 1 tsganina tsganina 36 фев 23 15:00 assets
drwxr-xr-x. 1 tsganina tsganina 16 фев 23 14:37 config
drwxr-xr-x. 1 tsganina tsganina 144 фев 23 14:37 content
drwxr-xr-x. 1 tsganina tsganina 54 фев 23 14:37 data
-rw-r--r--. 1 tsganina tsganina 452 фев 23 14:37 go.mod
-rw-r--r--. 1 tsganina tsganina 1343 фев 23 14:59 go.sum
drwxr-xr-x. 1 tsganina tsganina 40 фев 23 14:37 images
-rw-r--r--. 1 tsganina tsganina 1078 фев 23 14:37 LICENSE.md
-rw-r--r--. 1 tsganina tsganina 480 фев 23 14:37 netlify.toml
-rw-r--r--. 1 tsganina tsganina 222122 фев 23 14:37 preview.png
drwxr-xr-x. 1 tsganina tsganina 460 фев 23 15:00 public
-rw-r--r--. 1 tsganina tsganina 4058 фев 23 14:37 README.md
drwxr-xr-x. 1 tsganina tsganina 8 фев 23 14:59 resources
drwxr-xr-x. 1 tsganina tsganina 14 фев 23 14:37 static
-rw-r--r--. 1 tsganina tsganina 1388 фев 23 14:37 theme.toml
[tsganina@fedora blog_ganina]$ mc
[tsganina@fedora blog_ganina]$ ~/bin/hugo server
```

Рис. 4.10: Подключение к серверу

```
tsganina@fedora:~/work/blog_ganina — /home/tsganina/bin/...
Static files | 9
Processed images | 41
Aliases | 15
Sitemaps | 1
Cleaned | 0

Built in 3935 ms
Watching for changes in /home/tsganina/work/blog_ganina/{assets,content,data,static}
Watching for changes in /tmp/hugo_cache/modules/filecache/modules/pkg/mod/github.com/wowchemy/wowchemy-hugo-themes/modules/wowchemy/v5@v5.7.1-0.20221127215619-58b270a3e103/{archetypes,assets,data,layouts,static}
Watching for config changes in /home/tsganina/work/blog_ganina/config/_default, /tmp/hugo_cache/modules/filecache/modules/pkg/mod/github.com/wowchemy/wowchemy-hugo-themes/modules/wowchemy/v5@v5.7.1-0.20221127215619-58b270a3e103/config.yaml, /home/tsganina/work/blog_ganina/go.mod
Environment: "development"
Serving pages from memory
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. 4.11: Локальная страничка

4. Установить параметр для URLs сайта (рис. 4.12, 4.13, 4.14, 4.15, 4.16).

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner * Repository name *

tsganina / tsganina.github. ✓

Great repository names tsganina.github. is available. Need inspiration? How about [cuddly-fishstick?](#)

Description (optional)

Рис. 4.12: Создание специального репозитория

```
^C[tsganina@fedora blog_ganina]$ cd ..
[tsganina@fedora work]$ git clone --recursive git@github.com:tsganina/tsganina.github.io.git
Клонирование в «tsganina.github.io»...
warning: Похоже, что вы клонировали пустой репозиторий.
[tsganina@fedora work]$
```

Рис. 4.13: Клонирование пока что пустого репозитория

```
[tsganina@fedora work]$ cd tsganina.github.io
[tsganina@fedora tsganina.github.io]$ git checkout -b main
Переключились на новую ветку «main»
[tsganina@fedora tsganina.github.io]$ touch README.md
[tsganina@fedora tsganina.github.io]$ git add .
[tsganina@fedora tsganina.github.io]$ git commit -am "README add"
```

Рис. 4.14: Переключение на новую ветку и создание файла README

```

[tsganina@fedora tsganina.github.io]$ cd ..
[tsganina@fedora work]$ cd blog_ganina/
[tsganina@fedora blog_ganina]$ pwd
/home/tsganina/work/blog_ganina
[tsganina@fedora blog_ganina]$ git submodule add -b main git@github.com:tsganina
/tsganina.github.io.git public
Клонирование в «/home/tsganina/work/blog_ganina/public»...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Получение объектов: 100% (3/3), готово.
Следующие пути игнорируются одним из ваших файлов .gitignore:
public
подсказка: Use -f if you really want to add them.
подсказка: Turn this message off by running
подсказка: "git config advice.addIgnoredFile false"
fatal: Failed to add submodule 'public'
[tsganina@fedora blog_ganina]$

```

Рис. 4.15: Создание модуля в папке public

```

подсказка: Use -f if you really want to add them.
подсказка: Turn this message off by running
подсказка: "git config advice.addIgnoredFile false"
fatal: Failed to add submodule 'public'
[tsganina@fedora blog_ganina]$ mc

[tsganina@fedora blog_ganina]$ cat .gitignore
# IDEs
.idea/

# Hugo
resources/
#public/
jsconfig.json
node_modules/
go.sum
.hugo_build.lock
[tsganina@fedora blog_ganina]$ cat .gitignore

```

Рис. 4.16: Убрать public из списка игнорируемых

5. Разместить заготовку сайта на Github pages.

```
-----+-----
Pages | 55
Paginator pages | 0
Non-page files | 16
Static files | 9
Processed images | 42
Aliases | 15
Sitemaps | 1
Cleaned | 0

Total in 5214 ms
[tsganina@fedora blog_ganina]$ cd public/
[tsganina@fedora public]$ git remote -v
origin  git@github.com:tsganina/tsganina.github.io.git (fetch)
origin  git@github.com:tsganina/tsganina.github.io.git (push)
[tsganina@fedora public]$ git add .
[tsganina@fedora public]$ git commit -am "adding my website"
```

Рис. 4.17: Добавление новых данных на Github

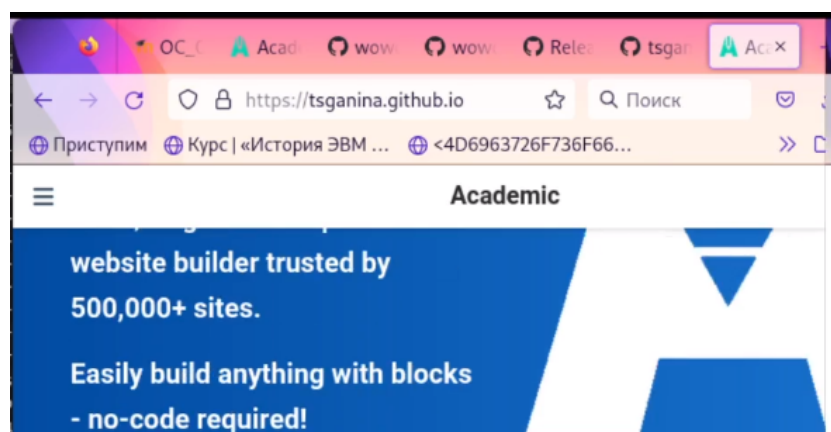


Рис. 4.18: Готовый URL

5 Выводы

В ходе этой работы был создан репозиторий для дальнейшей работы с проектом, а также URL для сайта.

Список литературы

1. Инструкция по выполнению первого этапа проекта