Отчёт по 2 этапу реализации проекта

Операционные системы

Самсонова Мария Ильинична

Содержание

Цель работы

Приобрести практические навыки размещения информации о себе и выкладывания постов на собственном сайте путем редактирования файлов в каталогах, создания новых файлов и каталогов и обновления через Терминал файлов.

Задание

Добавить к сайту данные о себе:

- Разместить фотографию владельца сайта.
- Разместить краткое описание владельца сайта (Biography).
- Добавить информацию об интересах (Interests).
- Добавить информацию об образовании (Education).
- Сделать пост по прошедшей неделе.
- Добавить пост на тему по выбору:
  - Управление версиями. Git.
  - Непрерывная интеграция и непрерывное развертывание (CI/CD).

Реализация 2 этапа проекта

1. Для начала запустим созданный сайт: (рис. [-@fig:001])

```
[misamsonova@fedora blog]$ hugo server
hugo: downloading modules …
hugo: collected modules in 17168 ms
Start building sites …
hugo v0.98.0-165d299cde259c8b801abadc6d3405a229e449f6+extended linux/amd64 BuildDate=2022-04-28T10:23:30Z VendorInfo=gohugoio

                   | EN
-------------------+-----
  Pages            | 53
  Paginator pages  |  0
  Non-page files   |  7
  Static files     |  9
  Processed images | 16
  Aliases          | 11
  Sitemaps         |  1
  Cleaned          |  0

Built in 1312 ms
Watching for changes in /home/misamsonova/blog/{assets,content,data,static}
Watching for config changes in /home/misamsonova/blog/config/_default, /home/misamsonova/blog/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
```

Запуск работы сайта

{ #fig:001 width=70% }

2. Изменим информацию о себе, отыскав файл _index.md в каталоге *~/blog/content/authors/admin* и внеся в него основные сведения (образование, имя и фамилия, сайт университета, интересы и пр.):(рис. [-@fig:002]) и (рис. [-@fig:003])

Открыть

_index.md
~/blog/content/authors/admin

```
---
# Display name
title: Samsonova Maria

# Is this the primary user of the site?
superuser: true

# Role/position/tagline
role: Student of the direction of physical, mathematical and natural sciences

# Organizations/Affiliations to show in About widget
organizations:
  - name: RUDN University
    url: https://www.rudn.ru/

# Short bio (displayed in user profile at end of posts)
bio: My research interests include distributed robotics, mobile computing and programmable mat

# Interests to show in About widget
interests:
  - Programming
  - Mathematical analysis
  - Discrete mathematics
  - History
  - Mathematical logic

# Education to show in About widget
education:
  courses:
    - course: 1 course
      institution: RUDN Moscow
      year: 2022

# Social/Academic Networking
# For available icons, see: https://wowchemy.com/docs/getting-started/page-builder/#icons
#   For an email link, use "fas" icon pack, "envelope" icon, and a link in the
#   form "mailto:your-email@example.com" or "/#contact" for contact widget.
social:
  - icon: envelope
    icon_pack: fas
    link: '/#contact'
  - icon: twitter
    icon_pack: fab
    link: https://twitter.com/GeorgeCushen
  - icon: graduation-cap # Alternatively, use `google-scholar` icon from `ai` icon pack
```

Markdown

Изменение информации о себе

{ #fig:002 width=70% }

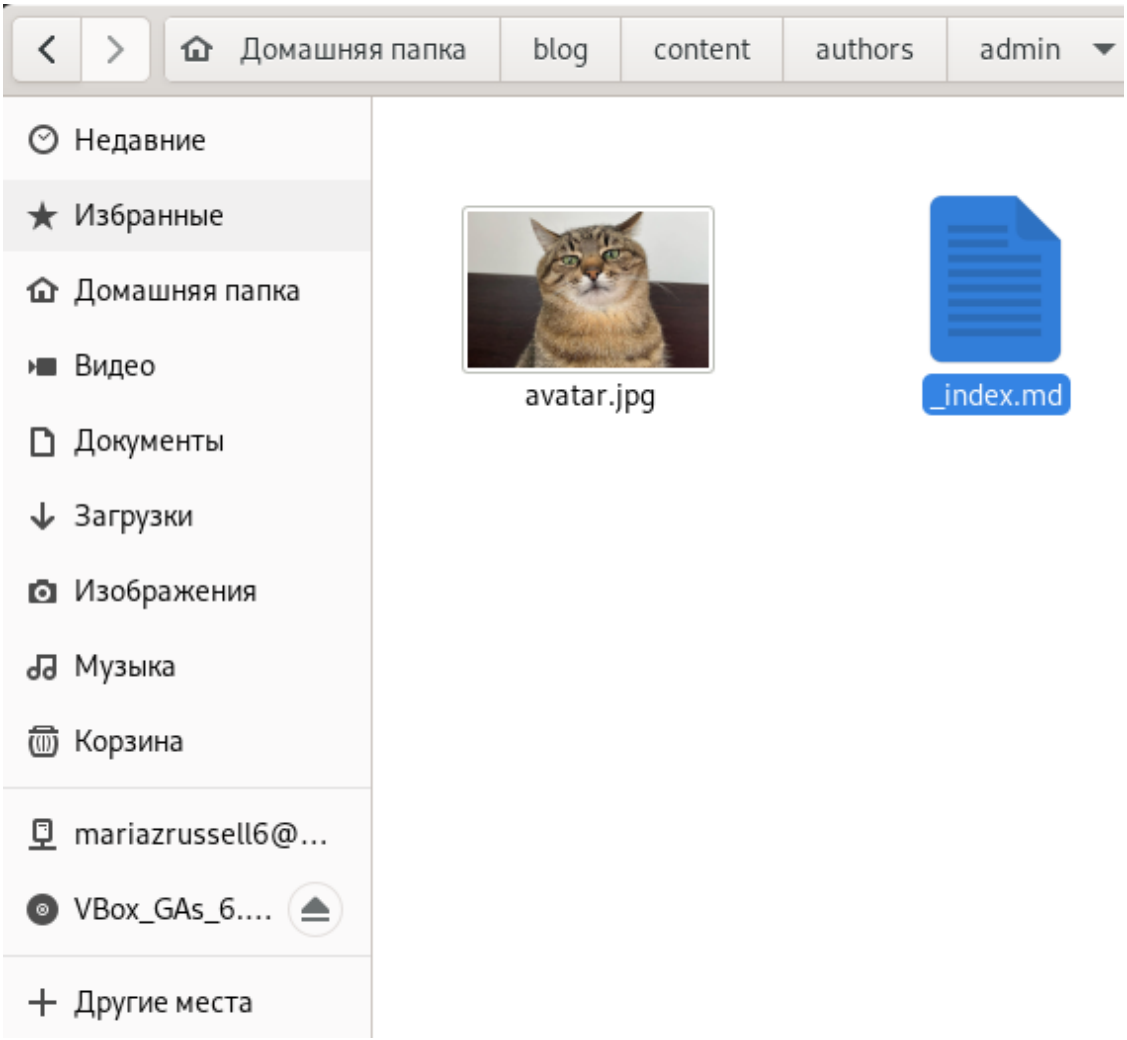
```
  courses:
    - course: 1 course
      institution: RUDN Moscow
      year: 2022

# Social/Academic Networking
# For available icons, see: https://wowchemy.com/docs/getting-started/page-builder/#icons
#   For an email link, use "fas" icon pack, "envelope" icon, and a link in the
#   form "mailto:your-email@example.com" or "/#contact" for contact widget.
social:
  - icon: envelope
    icon_pack: fas
    link: '/#contact'
  - icon: twitter
    icon_pack: fab
    link: https://twitter.com/GeorgeCushen
  - icon: graduation-cap # Alternatively, use `google-scholar` icon from `ai` icon pack
    icon_pack: fas
    link: https://scholar.google.co.uk/citations?user=sIwtMXoAAAAJ
  - icon: github
    icon_pack: fab
    link: https://github.com/gcushen
  - icon: linkedin
    icon_pack: fab
    link: https://www.linkedin.com/

# Link to a PDF of your resume/CV.
# To use: copy your resume to `static/uploads/resume.pdf`, enable `ai` icons in `params.toml`,
# and uncomment the lines below.
# - icon: cv
#   icon_pack: ai
#   link: uploads/resume.pdf

# Enter email to display Gravatar (if Gravatar enabled in Config)
email: ''

# Highlight the author in author lists? (true/false)
highlight_name: true
---

Welcome to my website. my name is Maria. I was born on December 6, 2003 in the first science city of the world, the city of Obninsk, Russia.
At the moment I am a student of the RUDN University. If you are interested in programming, robotics or foreign languages, I will be glad to talk t
you.

{{< icon name="download" pack="fas" >}} Download my {{< staticref "uploads/demo_resume.pdf" "newtab" >}}resumé{{< /staticref >}}.
```

Markdown Ширина табуляции: 8 Стр 9, Стлб 1 ВС

Изменение информации о себе

{ #fig:003 width=70% }

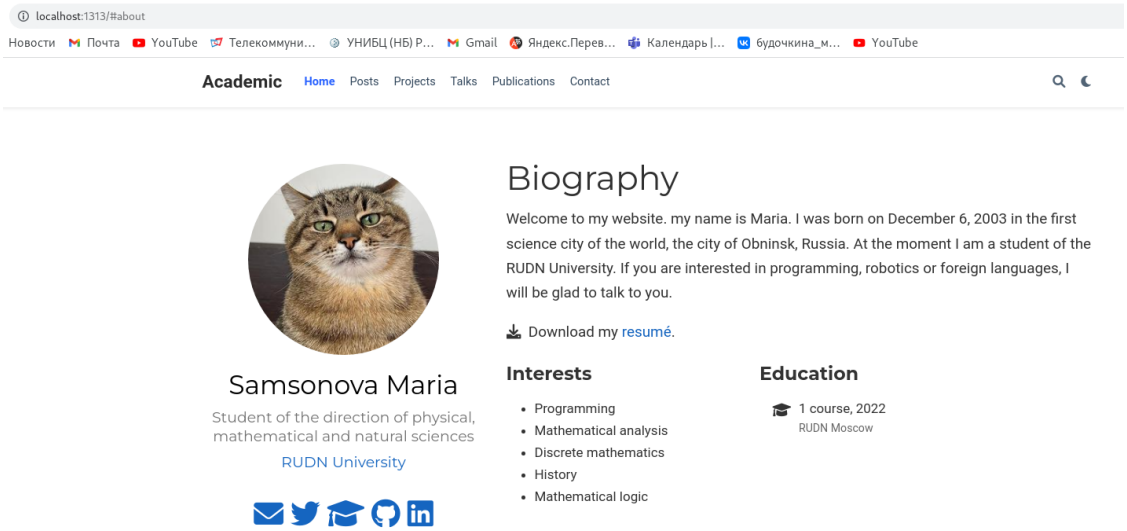
3. Меняем фотографию на странице, загрузив фотографию в каталог *~/blog/content/authors/admin* и переименовав в avatar.jpg: (рис. [-@fig:004])



Изменение фотографии на сайте

{ #fig:004 width=70% }

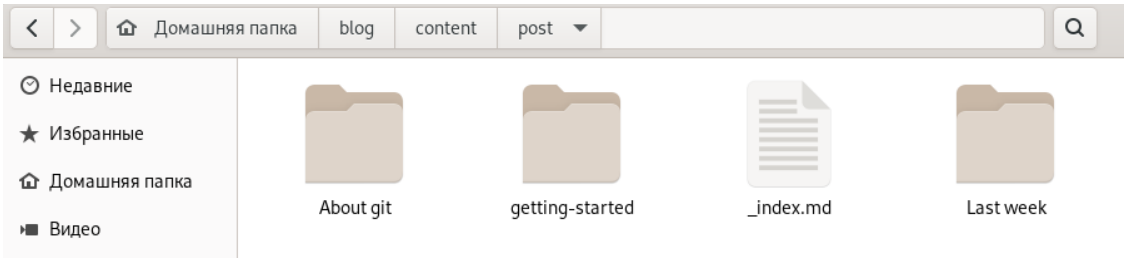
4. Открываем сайт, чтобы удостовериться, что изменения применены: (рис. [-@fig:005])



Изменение информации на сайте

{ #fig:005 width=70% }

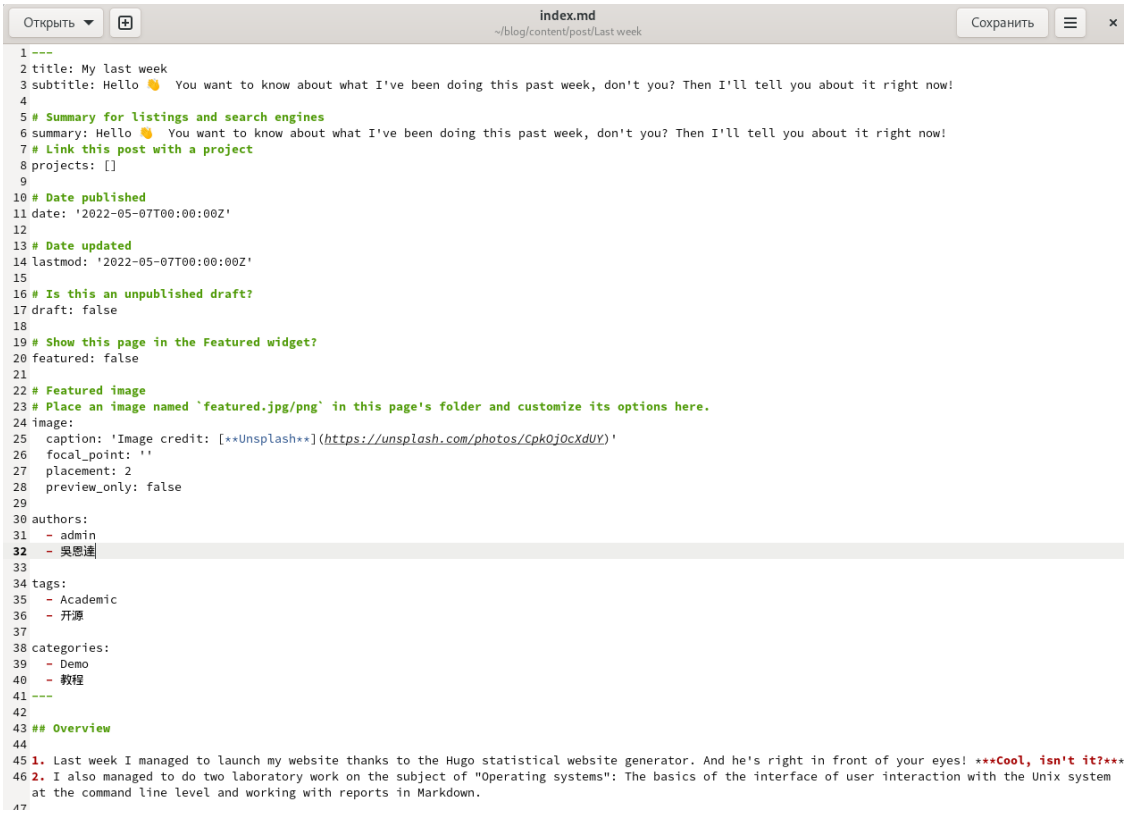
5. Далее мы переходим в каталог *~/blog/content/post*, создаём новые каталоги “my last week” и “About git” и копируем в них файл _index.md: (рис. [-@fig:006])



Создание новых папок

{ #fig:006 width=70% }

6. Теперь переходим в каталог “my last week” и в файле _index.md добавляем информацию о прошедшей неделе. Также добавляем в данный каталог картинку: (рис. [-@fig:007])



Редактирование информации в файле

{ #fig:007 width=70% }

7. После этого переходим в каталог “About git” и в файле _index.md добавляем информацию об управлении версиями git, также добавляем в данный каталог картинку: (рис. [-@fig:008]), (рис. [-@fig:009]) и (рис. [-@fig:010])

Открыть | index.md ~/blog/content/post/About git | Сохранить

```
---
title: Version control. Git.
subtitle: Hi! 👋 Today I will tell you about the git version control system. Let's get started!

# Summary for listings and search engines
summary: Hi! 👋 Today I will tell you about the git version control system. Let's get started!
# Link this post with a project
projects: []

# Date published
date: '2022-05-07T00:00:00Z'

# Date updated
lastmod: '2022-05-07T00:00:00Z'

# Is this an unpublished draft?
draft: false

# Show this page in the Featured widget?
featured: false

# Featured image
# Place an image named `featured.jpg/png` in this page's folder and customize its options here.
image:
  caption: 'Image credit: [**Unsplash**](https://unsplash.com/photos/CpkOjOcXdUY)'
  focal_point: ''
  placement: 2
  preview_only: false

authors:
  - admin
  - 吴恩達

tags:
  - Academic
  - 开源

categories:
  - Demo
  - 教程
---

# About the version control system

What is a "version control system" and why is it important? A version control system is a system that records changes to a file or set of files over time
and allows you to return later to a specific version. For file version control, this book will use the source code of the software as an example,
although in fact you can use version control for almost any type of file.

```

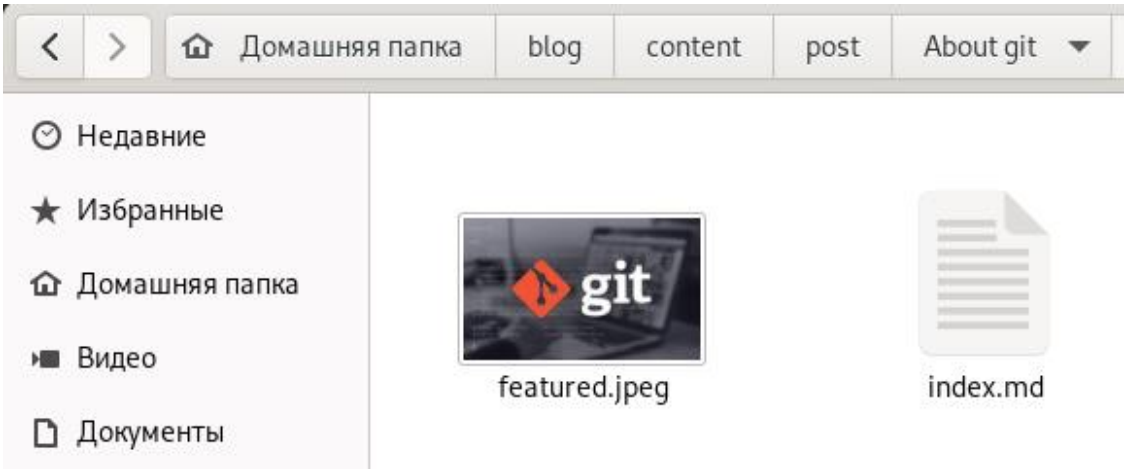
Редактирование информации в файле

{ #fig:008 width=70% }

```
# About the version control system

What is a "version control system" and why is it important? A version control system is a system that records changes to a file or set of files over time
and allows you to return later to a specific version. For file version control, this book will use the source code of the software as an example,
although in fact you can use version control for almost any type of file.

If you are a graphic or web designer and you want to save every version of an image or layout (most likely you will), the version control system
(hereinafter referred to as SLE) – just what you need. It allows you to return files to the state they were in before the changes, return the project to
its original state, see the changes, see who last changed something and caused the problem, who set the task and when, and much more. Using SLE also
means in general that if you have broken something or lost files, you can safely fix everything. In addition to everything, you will get it all without
any extra effort.

# Local version control systems
Many people use copying files to a separate directory as a version control method (perhaps even a directory with a time stamp, if they are smart enough).
This approach is very common because of its simplicity, but it is incredibly prone to errors. You can easily forget which directory you are in and
accidentally change the wrong file or copy the wrong files that you wanted.

In order to solve this problem, programmers have long ago developed local SLE with a simple database that stores records of all changes in files, thereby
monitoring revisions.

One of the most popular SLE was the RCS system, which is still distributed with many computers today. RCS stores on disk sets of patches (differences
between files) in a special format, using which it can recreate the state of each file at a given time.

# Centralized version control systems
The next major problem that people face is the need to interact with other developers. In order to deal with it, centralized version control systems
(CSCs) were developed. Systems such as CVS, Subversion, and Perforce use a single server containing all versions of files, and a number of clients that
receive files from this centralized repository. The use of CSCS has been the standard for many years.

This approach has many advantages, especially over local SLE. For example, all project developers know to a certain extent what each of them is doing.
Administrators have full control over who can do what, and it is much easier to administer the CSCS than to operate local databases on each client.

Despite this, this approach also has serious disadvantages. The most obvious disadvantage is a single point of failure represented by a centralized
server. If this server goes down for an hour, then during this time no one will be able to use version control to save the changes they are working on,
and no one will be able to share these changes with other developers. If the hard disk on which the central database is stored is damaged, and there are
no timely backups, you will lose everything – the entire history of the project, not counting single repository snapshots that have been saved on local
developer machines. Local SLE suffer from the same problem: when the entire project history is stored in one place, you risk losing everything.

# Distributed version control systems
This is where distributed version control systems (RCS) come into play. In RSKV (such as Git, Mercurial, Bazaar or Darcs), clients do not just download a
snapshot of all files (the state of files at a certain point in time) – they completely copy the repository. In this case, if one of the servers through
which the developers exchanged data dies, any client repository can be copied to another server to continue working. Each copy of the repository is a
complete backup of all data.

Moreover, many RSCs can simultaneously interact with several remote repositories, thanks to this you can work with different groups of people using
different approaches at the same time within the same project. This allows you to apply several approaches to development at once, for example,
hierarchical models, which is completely impossible in centralized systems.
```

Добавлении картинки

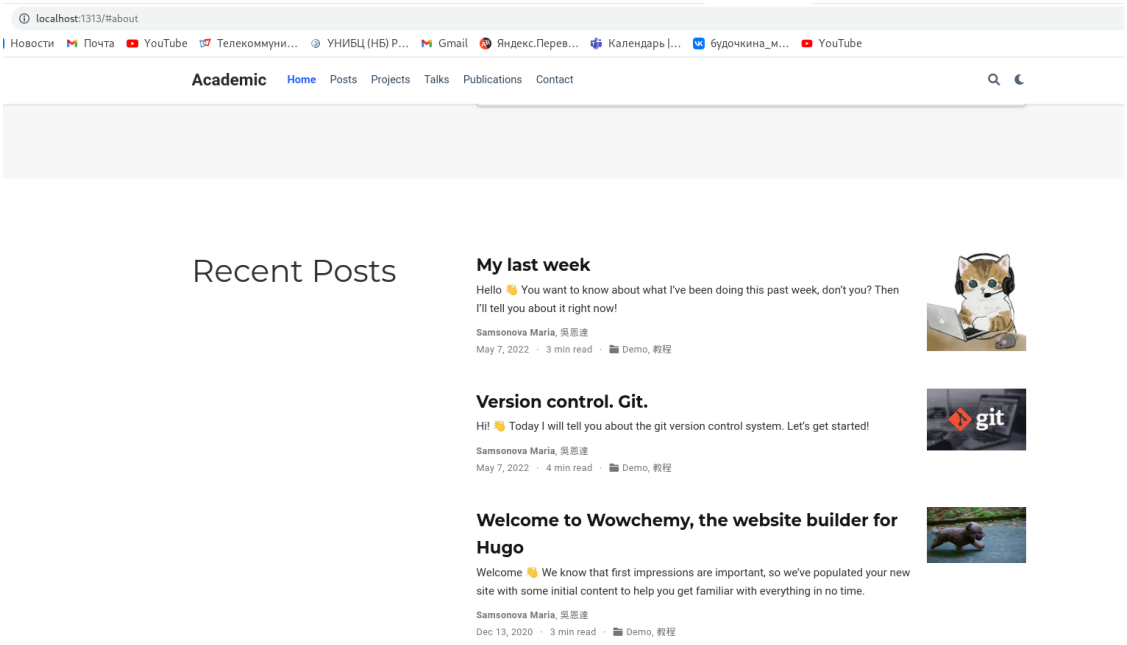
{ #fig:009 width=70% }



Добавлении картинки

{ #fig:010 width=70% }

8. Открываем сайт, чтобы удостовериться, что изменения применены: (рис. [-@fig:011])



Добавление постов на сайте

{ #fig:011 width=70% }

9. Также замечаем, что наши изменения фиксируются в Терминале: (рис. [-@fig:012])

```
Change detected, rebuilding site.
2022-05-07 16:54:34.613 +0300
Source changed "/home/misamsonova/blog/content/post/getting-started (копия)": RENAME
Source changed "/home/misamsonova/blog/content/post/Last week/featured.jpg": CREATE
Source changed "/home/misamsonova/blog/content/post/Last week/index.md": CREATE
Total in 878 ms

Change detected, rebuilding site.
2022-05-07 16:57:08.613 +0300
Source changed "/home/misamsonova/blog/content/post/Last week/file_269448.webp": WRITE
Total in 130 ms

Change detected, rebuilding site.
2022-05-07 16:58:01.613 +0300
Source changed "/home/misamsonova/blog/content/post/Last week/file_269448.webp": RENAME
Total in 79 ms

Change detected, rebuilding site.
2022-05-07 16:58:26.634 +0300
Source changed "/home/misamsonova/blog/content/post/Last week/file_269448 (1).jpg": CREATE
Total in 215 ms

Change detected, rebuilding site.
2022-05-07 16:58:36.613 +0300
Source changed "/home/misamsonova/blog/content/post/Last week/file_269448 (1).jpg": RENAME
Source changed "/home/misamsonova/blog/content/post/Last week/featured1.jpg": CREATE
Total in 87 ms

Change detected, rebuilding site.
2022-05-07 16:58:37.613 +0300
Source changed "/home/misamsonova/blog/content/post/Last week/featured.jpg": RENAME
Total in 452 ms

Change detected, rebuilding site.
2022-05-07 16:58:42.114 +0300
Source changed "/home/misamsonova/blog/content/post/Last week/featured1.jpg": RENAME
Source changed "/home/misamsonova/blog/content/post/Last week/featured.jpg": CREATE
Total in 303 ms

Change detected, rebuilding site.
2022-05-07 17:01:58.116 +0300
Source changed "/home/misamsonova/blog/content/post/Last week/index.md": CREATE
Total in 309 ms

Change detected, rebuilding site.
```

Вывод об изменениях сайта в Терминале

{ #fig:012 width=70% }

10. Теперь следующими командами мы обновляем файлы на github и запускаем сервер для просмотра другими пользователями сайта с уже изменённой информацией:(рис. [-@fig:013]),(рис. [-@fig:014]),(рис. [-@fig:015])

```
[misamsonova@fedora blog]$ git add .
[misamsonova@fedora blog]$ git commit -am 'Everything is new'
^[[A[main 53d9249] Everything is new
 7 files changed, 241 insertions(+), 20 deletions(-)
 rewrite content/authors/admin/avatar.jpg (98%)
 create mode 100644 content/post/About git/featured.jpeg
 create mode 100644 content/post/About git/index.md
 create mode 100644 content/post/Last week/featured.jpg
 create mode 100644 content/post/Last week/index.md
[misamsonova@fedora blog]$ git push
Перечисление объектов: 21, готово.
Подсчет объектов: 100% (21/21), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (14/14), готово.
Запись объектов: 100% (14/14), 171.33 КиБ | 6.34 МиБ/с, готово.
Всего 14 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 3 local objects.
To https://github.com/misamsonova/blog.git
   0bcf1b0..53d9249  main -> main
```

Ввод команд для обновления файлов на github

{ #fig:013 width=70% }

```
^C[misamsonova@fedora blog]$ hugo
Start building sites …
hugo v0.98.0-165d299cde259c8b801abadc6d3405a229e449f6+extended linux/amd64 BuildDate=2022-04-28T10:23:30Z VendorInfo=gohugoio

                   | EN
-------------------+-----
  Pages            | 55
  Paginator pages  |  0
  Non-page files   |  9
  Static files     |  9
  Processed images | 19
  Aliases          | 11
  Sitemaps         |  1
  Cleaned          |  0

Total in 1353 ms
[misamsonova@fedora blog]$ cd public/
[misamsonova@fedora public]$ git add .
[misamsonova@fedora public]$ git commit -am 'Everything is new'
[main 2ddf607] Everything is new
 39 files changed, 5238 insertions(+), 113 deletions(-)
 rewrite authors/admin/avatar.jpg (98%)
 create mode 100644 authors/admin/avatar_hu669edb64a2fa6a04680eb1267b076714_156571_270x270_fill_q75_lanczos_center.jpg
 rewrite index.json (100%)
 create mode 100644 post/about-git/featured.jpeg
 create mode 100644 post/about-git/featured_hu44aaeb6db78b663ce8664da97ba699c8_5558_1200x2500_fit_q75_h2_lanczos.webp
 create mode 100644 post/about-git/featured_hu44aaeb6db78b663ce8664da97ba699c8_5558_150x0_resize_q75_h2_lanczos.webp
 create mode 100644 post/about-git/index.html
 create mode 100644 post/last-week/featured.jpg
 create mode 100644 post/last-week/featured_hu69ff2d5a7c8989295ba4849735939a5b_15796_1200x2500_fit_q75_h2_lanczos.webp
 create mode 100644 post/last-week/featured_hu69ff2d5a7c8989295ba4849735939a5b_15796_150x0_resize_q75_h2_lanczos.webp
 create mode 100644 post/last-week/index.html
[misamsonova@fedora public]$ git commit -am 'Everything is new'
На ветке main
Ваша ветка опережает «origin/main» на 1 коммит.
  (используйте «git push», чтобы опубликовать ваши локальные коммиты)

нечего коммитить, нет изменений в рабочем каталоге
[misamsonova@fedora public]$ git push
Перечисление объектов: 119, готово.
Подсчет объектов: 100% (119/119), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (63/63), готово.
Запись объектов: 100% (66/66), 244.53 КиБ | 6.43 МиБ/с, готово.
Всего 66 (изменений 33), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
```

Запуск сервера и ввод команд для обновления файлов на github

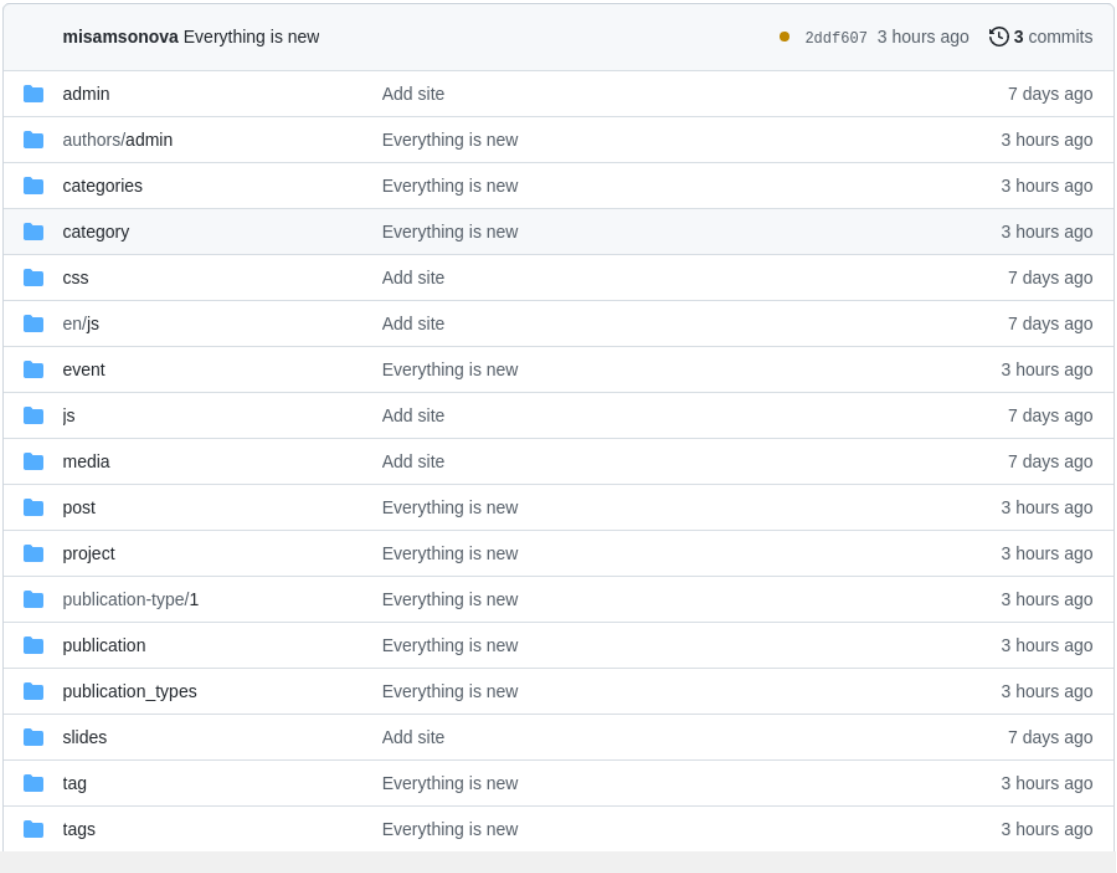
{ #fig:014 width=70% }

```
Подсчет объектов: 100% (119/119), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (63/63), готово.
Запись объектов: 100% (66/66), 244.53 КиБ | 6.43 МиБ/с, готово.
Всего 66 (изменений 33), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (33/33), completed with 25 local objects.
To https://github.com/misamsonova/misamsonova.github.io.git
   778fa7f..2ddf607  main -> main
[misamsonova@fedora public]$ cd ..
[misamsonova@fedora blog]$ git add .
[misamsonova@fedora blog]$ git commit -am 'Everything is new'
[main 076af18] Everything is new
 1 file changed, 1 insertion(+), 1 deletion(-)
[misamsonova@fedora blog]$ git push
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (2/2), 914 байтов | 914.00 КиБ/с, готово.
Всего 2 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/misamsonova/blog.git
   53d9249..076af18  main -> main
```

Загрузка обновлённых файлов на github

{ #fig:015 width=70% }

11. Открываем github, чтобы удостовериться, что изменения применены: (рис. [-@fig:016])

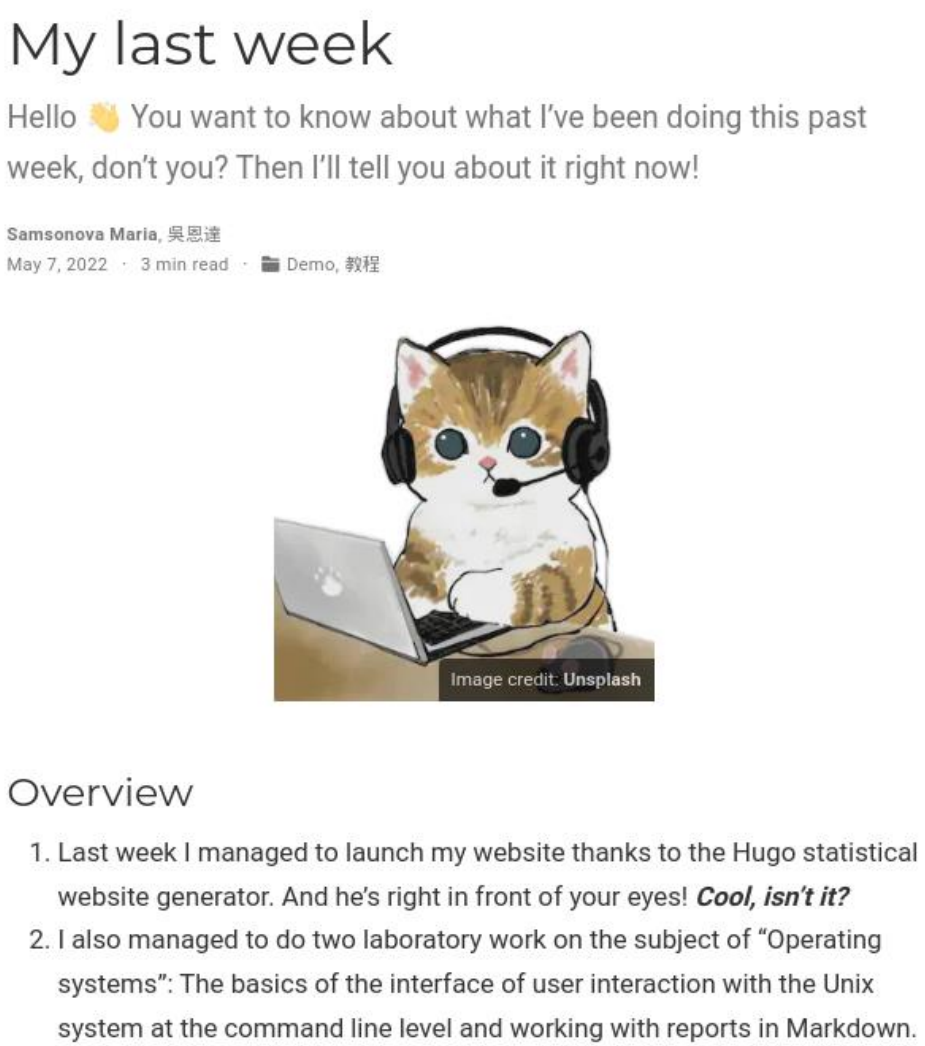


misamsonova Everything is new	2ddf607 3 hours ago	**3** commits
admin	Add site	7 days ago
authors/admin	Everything is new	3 hours ago
categories	Everything is new	3 hours ago
category	Everything is new	3 hours ago
css	Add site	7 days ago
en/js	Add site	7 days ago
event	Everything is new	3 hours ago
js	Add site	7 days ago
media	Add site	7 days ago
post	Everything is new	3 hours ago
project	Everything is new	3 hours ago
publication-type/1	Everything is new	3 hours ago
publication	Everything is new	3 hours ago
publication_types	Everything is new	3 hours ago
slides	Add site	7 days ago
tag	Everything is new	3 hours ago
tags	Everything is new	3 hours ago

Обновлённые папки на github

{ #fig:016 width=70% }

12. Запускаем сайт misamsonova.github.io через терминал командой hugo и просматриваем посты, убеждаемся, что всё имеется:(рис. [-@fig:017]),(рис. [-@fig:018]), (рис. [-@fig:019]),(рис. [-@fig:020])



Просмотр поста "my last week" на сайте

{ #fig:017 width=70% }

Version control. Git.

Hi! 👋 Today I will tell you about the git version control system.
Let's get started!

Samsonova Maria, 吳恩達
May 7, 2022 · 6 min read · Demo, 教程

About the version control system

What is a "version control system" and why is it important? A version control system is a system that records changes to a file or set of files over time and allows you to return later to a specific version. For file version control, this book will use the source code of the software as an example, although in fact you can use version control for almost any type of file.

If you are a graphic or web designer and you want to save every version of an image or layout (most likely you will), the version control system (hereinafter referred to as SLE) — just what you need. It allows you to return files to the state they were in before the changes, return the project to its original state, see the changes, see who last changed something and caused the problem, who set the task and when, and much more. Using SLE also means in general that if you have broken something or lost files, you can safely fix everything. In addition to everything, you will get it all without any extra effort.

Просмотр поста "Version control.Git." на сайте

{ #fig:017 width=70% }

Local version control systems

Many people use copying files to a separate directory as a version control method (perhaps even a directory with a time stamp, if they are smart enough). This approach is very common because of its simplicity, but it is incredibly prone to errors. You can easily forget which directory you are in and accidentally change the wrong file or copy the wrong files that you wanted.

In order to solve this problem, programmers have long ago developed local SLE with a simple database that stores records of all changes in files, thereby monitoring revisions.

One of the most popular SLE was the RCS system, which is still distributed with many computers today. RCS stores on disk sets of patches (differences between files) in a special format, using which it can recreate the state of each file at a given time.

Centralized version control systems

The next major problem that people face is the need to interact with other developers. In order to deal with it, centralized version control systems (CSCs) were developed. Systems such as CVS, Subversion, and Perforce use a single server containing all versions of files, and a number of clients that receive files from this centralized repository. The use of CSCS has been the standard for many years.

This approach has many advantages, especially over local SLE. For example, all project developers know to a certain extent what each of them is doing. Administrators have full control over who can do what, and it is much easier to administer the CSCS than to operate local databases on each client.

Despite this, this approach also has serious disadvantages. The most obvious disadvantage is a single point of failure represented by a centralized server. If this server goes down for an hour, then during this time no one will be able to use version control to save the changes they are working on, and no one will be able to share these changes with other developers. If the hard

Просмотр поста “Version control.Git.” на сайте

{ #fig:017 width=70% }

to administer the CSCS than to operate local databases on each client.

Despite this, this approach also has serious disadvantages. The most obvious disadvantage is a single point of failure represented by a centralized server. If this server goes down for an hour, then during this time no one will be able to use version control to save the changes they are working on, and no one will be able to share these changes with other developers. If the hard disk on which the central database is stored is damaged, and there are no timely backups, you will lose everything — the entire history of the project, not counting single repository snapshots that have been saved on local developer machines. Local SLE suffer from the same problem: when the entire project history is stored in one place, you risk losing everything.

Distributed version control systems

This is where distributed version control systems (RCS) come into play. In RSKV (such as Git, Mercurial, Bazaar or Darcs), clients do not just download a snapshot of all files (the state of files at a certain point in time) — they completely copy the repository. In this case, if one of the servers through which the developers exchanged data dies, any client repository can be copied to another server to continue working. Each copy of the repository is a complete backup of all data.

Moreover, many RSCs can simultaneously interact with several remote repositories, thanks to this you can work with different groups of people using different approaches at the same time within the same project. This allows you to apply several approaches to development at once, for example, hierarchical models, which is completely impossible in centralized systems.

Просмотр поста “Version control.Git.” на сайте

{ #fig:017 width=70% }

Вывод

В процессе реализации 2-ого этапа мы приобрели практические навыки размещения информации о себе и выкладывания постов на собственном сайте путем редактирования файлов в каталогах, создания новых файлов и каталогов и обновления через Терминал файлов.