

Презентация по второму этапу индивидуального проекта

Операционные системы

Арбатова В. П., НКАбд-01-23

11 марта 2024

Российский университет дружбы народов, Москва, Россия

Цель работы

Добавить к сайту информацию о себе

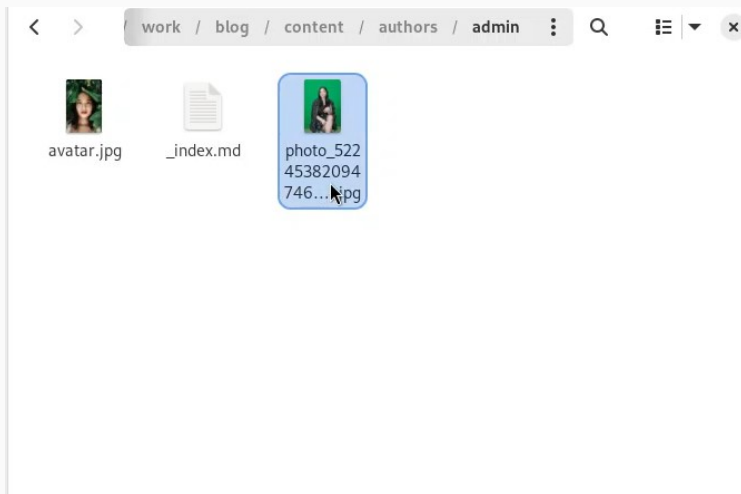
Задание

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 управление версиями. Git.

Выполнение лабораторной работы

Добавление фото

Скачиваю свою фотографию (взяла из выпускного альбома, может потом найду что-то лучше), добавляю его в папку `content/authors/admin`



Переименовываю фотографию и удаляю старую

```
vparbatova@varbatova:~/work/blog/content/authors/admin$ mv photo_5224538209474693555_y.jpg avatar.jpg
```

Рис. 2: Переименование

Меняю имя на своё

```
first_name: Varvara  
last_name: Arbatova
```

Рис. 3: Новое имя

Добавляю краткую биографию и интересы

```
# Short bio (displayed in user profile at end of posts)
bio: Мне интересно рисование, писательство, конный спорт

# Interests to show in About widget
interests:
  - Рисование
  - Конный спорт
  - Писательство
  - 3D моделирование
  - Путешествия
```

Рис. 4: Название рисунка

Добавляю образование (что вспомнила и закончила, остальное пока не указывала, когда появится необходимость и возможность, добавлю)

```
# Education to show in About widget
education:
  courses:
    - course: Дизайнерское искусство
      institution: Империя дизайна К
      year: 2012-2020
    - course: Театральное искусство
      institution: ЛИК
      year: 2012-2014
    - course: Программирование
      institution: Яндекс практикум
      year: 2020|
```

Добавляю информацию о сфере деятельности сейчас и об организации

```
# Role/position/tagline
role: Студентка ФФМиЕН РУДН

# Organizations/Affiliations to show in About widget
organizations:
  - name: РУДН
    url: https://www.rudn.edu/
```

Рис. 6: Сфера деятельности

Добавляю имя, которое будет на титульном листе

```
# Name pronunciation (optional)
name_pronunciation: Арбатова Варвара Петровна
```

Рис. 7: Имя

Добавляю полную информацию

```
# Highlight the author in author lists? (true/false)
```

```
highlight_name: true
```

```
---
```

Арбатова Варвара – студентка Российского университета дружбы народов факультета журналистики, занимаюсь спортом самостоятельно.

```
{style="text-align: justify;"}
```

Рис. 8: Полная информация

Запускаю синхронизацию с сайтом

```
vparbatova@vparbatova:~/work/blog$ ~/bin/hugo server
Watching for changes in /home/vparbatova/{.cache,work}
Watching for config changes in /home/vparbatova/work/blog/config/_default, /home/vparbatova/.c
78cf6921f9/config.yaml, /home/vparbatova/.cache/hugo_cache/modules/filecache/modules/pkg/mod/g
es/pkg/mod/github.com/hugo!blox/hugo-blox-builder/modules/blox-bootstrap/v5@v5.9.7/hugo.yaml,
Start building sites ...
hugo v0.123.6-92684f9a26838a46d1a81e3c250fef5207bcb735+extended linux/amd64 BuildDate=2024-02-
```

	EN
Pages	54
Paginator pages	0
Non-page files	16

Рис. 9: Запуск

Проверяю, как выглядит сайт



Арбатова Варвара Петровна
吳 健 雄
Студентка ФФМиЕН РУДН
РУДН

Biography

Арбатова Варвара - студентка Российского университета дружбы народов факультета физико-математических и естественных наук. Стараюсь много времени уделять самореализации в разных сферах, рисую, пишу, читаю, состою в сборной вуза по чирлидингу, занимаюсь спортом самостоятельно.

Interests

- Рисование
- Конный спорт
- Писательство
- 3D моделирование
- Путешествия

Education

- 🎓 Дизайнерское искусство, 2012-2020
Империя дизайна K
- 🎓 Театральное искусство, 2012-2014
ЛИК
- 🎓 Программирование, 2020
Яндекс практикум

Рис. 10: Сайт

Создаю папки post1 и post2 в папке content/post

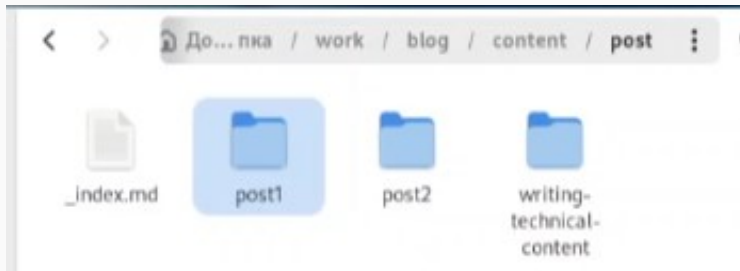


Рис. 11: Папки

Создаю пост про прошлую неделю



```
• index.md x
- Academic
- 开源

categories:
- Demo
- 教程
---

# Приключения
Прошлая неделя была ст|
```

Рис. 12: Недельный пост

Создаю пост про управление версиями git

О системе контроля версий

Что такое «система контроля версий» и почему это важно? Система контроля версий—это система, записывающая изменения в файл или набор файлов. Она используется для хранения исходного кода программного обеспечения, хотя на самом деле вы можете использовать контроль версий практически для всего.

Локальные системы контроля версий

Многие люди в качестве метода контроля версий применяют копирование файлов в отдельный каталог (возможно даже, каталог с датой и временем). Можно легко забыть в каком каталоге вы находитесь и случайно изменить не тот файл или скопировать не те файлы, к которым нужно. Для того, чтобы решить эту проблему, программисты давно-давно разработали локальные VCS с простой базой данных, которая хранит записи об изменениях. Одной из популярных VCS была система RCS, которая и сегодня распространяется со многими компьютерами. RCS хранит на диске наборы данных в определённые моменты времени.

Централизованные системы контроля версий

Следующая серьёзная проблема, с которой сталкиваются люди,—это необходимость взаимодействовать с другими разработчиками. Для того, чтобы решить эту проблему, системы, как CVS, Subversion и Perforce, используют единственный сервер, содержащий все версии файлов, и некоторое количество клиентов.

Такой подход имеет множество преимуществ, особенно перед локальными VCS. Например, все разработчики проекта в определённой степени могут работать с файлами, чем оперировать локальными базами данных на каждом клиенте.

Несмотря на это, данный подход тоже имеет серьёзные минусы. Самый очевидный минус—это единая точка отказа, представленная центральным сервером. Если сервер выйдет из строя, то все изменения, над которыми работает, а также никто не сможет обмениваться этими изменениями с другими разработчиками. Если считать единичных снимков репозитория, которые сохранились на локальных машинах разработчиков. Локальные VCS страдают от той же самой проблемы.

Распределённые системы контроля версий

Здесь в игру вступают распределённые системы контроля версий (Distributed Version Control System, далее DVCS). В DVCS (таких как Git, Mercurial, Bazaar) полностью копируют репозиторий. В этом случае, если один из серверов, через который разработчики обменивались данными, умрёт, любой из клиентов сможет восстановить репозиторий.

Более того, многие DVCS могут одновременно взаимодействовать с несколькими удалёнными репозиториями, благодаря этому вы можете использовать различные подходы в разработке, например, иерархические модели, что совершенно невозможно в централизованных системах.

Выводы

Я добавила на сайт информацию о себе

Список литературы
