

# Отчёт по 2 этапу индивидуального проекта

## Операционные системы

Арбатова Варвара Петровна

### Содержание

1	Цель работы .....	1
2	Задание .....	1
3	Выполнение лабораторной работы.....	1
3.1	Добавление фото.....	1
3.2	Описание владельца сайта.....	2
3.3	Краткая биография и интересы .....	2
3.4	Образование .....	3
4	Выводы.....	5
	Список литературы .....	5

## 1 Цель работы

Добавить к сайту информацию о себе

## 2 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.

## 3 Выполнение лабораторной работы

### 3.1 Добавление фото

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

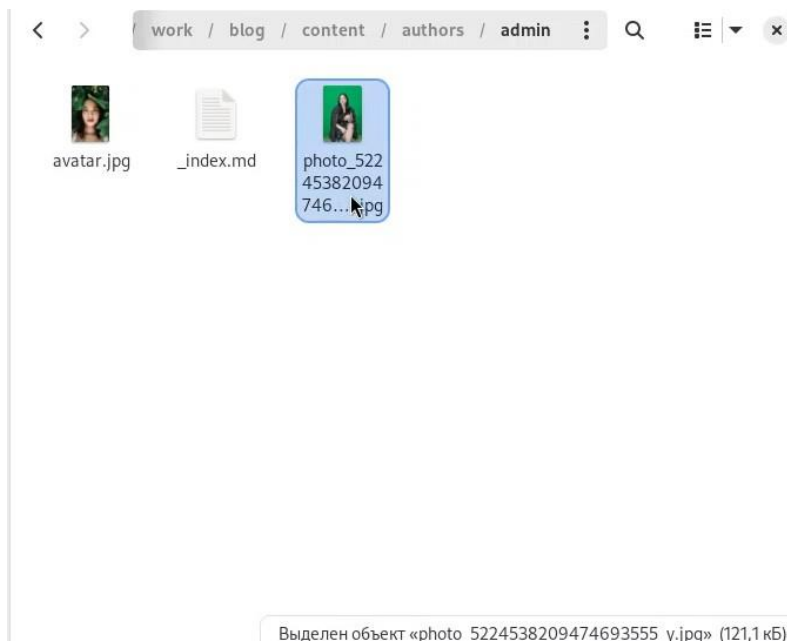


Рис. 1: Фото

Переименовываю фотографию и удаляю старую

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

Рис. 2: Переименование

## 3.2 Описание владельца сайта

Меняю имя на своё

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

Рис. 3: Новое имя

## 3.3 Краткая биография и интересы

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

### 3.4 Образование

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|
```

Рис. 5: Образование

Добавляю информацию о сфере деятельности сейчас и об организации

```
# Role/position/tagline
role: Студентка ФФМФН 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 факул-
тет юриспруден-
ции, 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



Рис. 10: Сайт

Создаю папки post1 и post2 в папке content/post

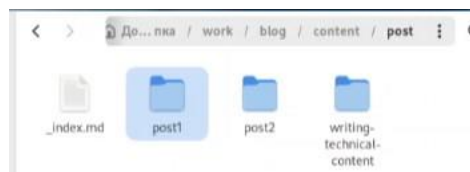


Рис. 11: Папки

Создаю пост про прошлую неделю



Рис. 12: Недельный пост

Создаю пост про управление версиями git

#### # 0 системы контроля версий

Что такое «система контроля версий» и почему это важно? Система контроля версий—это система, записывающая изменения в файл или набор файлов. Обычно используется исходный код 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, и так далее. Для того, чтобы решить эту проблему, программисты давным-давно разработали локальные VCS с простой базой данных, которая хранит историю изменений. 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, были разработаны централизованные VCS, такие как CVS, Subversion и Perforce, которые 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 кодом, что 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 клиенты потеряют доступ к коду. Кроме того, централизованные VCS требуют постоянного подключения к серверу, что может быть неудобно для разработчиков, работающих в автономном режиме. Локальные VCS страдают от той же проблемы.

#### # Распределённые системы контроля версий

Здесь в игру вступает распределённые системы контроля версий (Distributed Version Control System, далее DVCS). В DVCS (таких как Git, Mercurial, Bazaar и др.) каждая копия репозитория является полной копией репозитория. В этом случае, если один из серверов, через который разработчики обменивались данными, выйдет из строя, то все данные будут сохранены на других серверах.

Более того, многие DVCS могут одновременно взаимодействовать с несколькими удалёнными репозиториями, благодаря этому вы можете работать с кодом в нескольких местах, например, иерархические модели, что совершенно невозможно в централизованных системах.

Рис. 13: Пост 2

## 4 Выводы

Я добавила на сайт информацию о себе

## Список литературы