

Министерство цифрового развития, связи и массовых коммуникаций РФ
Уральский технический институт связи и информатики (филиал) ФГБОУ ВО
"Сибирский государственный университет телекоммуникаций и информатики" в
г. Екатеринбурге
(УрТИСИ СибГУТИ)

КАФЕДРА
ИСТ

ОТЧЕТ

По дисциплине «Сетевое программирование»
Практическое занятие № 1
«Работа с системой управления версиями»

Выполнил: студент гр. ПЕ-126
Камков Д.А.

Проверил: зав. каф. ИСТ,
Бурумбаев Д.И.

Ассистент:

Екатеринбург, 2024

1 Цель работы:

- 1.1 Научиться работать с системой управления версиями;
- 1.2 Закрепить знания по теме «Системы управления версиями».

2 Перечень оборудования:

- 2.1 Персональный компьютер;
- 2.2 Система управления версиями Git;
- 2.3 Visual Studio Code.

3.Ход работы:

3.1 Перед началом работы с Git необходимо создать папку с фамилией и номером группы, в которой будут храниться наши элементы. После этого, необходимо создать следующую структуру, как показано на рисунке 1.

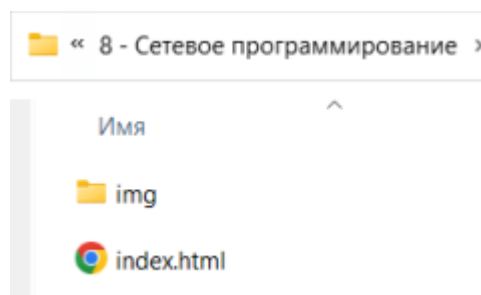


Рисунок 1 – Структура папки

Как видно из структуры, в корневой директории лежат:

- 1. Основная страница блога index.html
- 2. Папка img, в которой хранятся изображения (файлы *.jpg) сайта.

Тематика блога выбирается самостоятельно, картинки – тоже, но в рамках общественной нормы.

В качестве примера предполагается, что блог посвящен диким животным. Тогда структура файла index.html будет выглядеть как показано на рисунке 2.

```

<!DOCTYPE html>
<html>

  <head>
    <title>Wild Animals Blog</title>
    <meta charset="utf-8">
  </head>

  <body align=center>

    
    <h1>Wild Animals</h1>
    <small>- Blog about nature -</small>

    <br>
    <br>
    <br>
    <br>

    <p>Let's talk about wild animals around the world:</p>
    <h2>Giraffe</h2>
    
    <p><b>Area:</b> Africa</p>
    <p><b>Weight:</b> 900-1200kg</p>
    <p><b>Height:</b> 6m</p>
    <br>

    <h2>Elephant</h2>
    
    <p><b>Area:</b> Africa, Asia</p>
    <p><b>Weight:</b> 4000-7000kg</p>
    <p><b>Height:</b> 3m</p>
    <br>

  </body>
</html>

```

Рисунок 2 – Структура index.html

Если открыть данный файл в браузере, то выглядит основная (и пока единственная) страница блога показана на рисунке 3:



Thw BEST Cars

- Blog about cars in BMW concern -

Let's talk about bests cars around the world:

BMW M5 G30



HorsePower: 600 hp

TorquePower: 750 Nm

Acceleration : 3.4 sec

BMW_M5_G30



HorsePower: 510 hp

TorquePower: 650 Nm

Acceleration : 3.6 sec

Рисунок 3 – Страница блога в браузере

Далее требуется создать репозиторий для нашего блога. Для этого необходимо перейти в директорию будущего репозитория (или открыть при помощи терминала вашу папку)

Далее необходимо прописать команду git. Если у Вас открывается сообщение, как показано на рисунке 3, то значит git установлен. Если подобное сообщение не открывается необходимо установить его в зависимости от Вашей операционной системы с сайта <https://git-scm.com/book/ru/v2/ВведениеУстановка-Git>

```
C:\Windows\system32\cmd.exe

C:\Users\dimon>git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      [--config-env=<name>=<envvar>] <command> [<args>]

These are common Git commands used in various situations:


start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one


work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm         Remove files from the working tree and from the index


examine the history and state (see also: git help revisions)
  bisect    Use binary search to find the commit that introduced a bug
  diff      Show changes between commits, commit and working tree, etc
  grep      Print lines matching a pattern
  log       Show commit logs
```

Рисунок 4 – Сообщение, если git установлен

Для создания репозитория необходимо прописать команду `git init`, после чего появится сообщение:

```
C:\Users\dimon>cd C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126
C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126>git init
Initialized empty Git repository in C:/Users/dimon/_Valhalla/_Lern/Сетевое_программирование/Камков_ПЕ-126/.git/
```

Рисунок 5 – Результат выполнения “git init”

После такого сообщения, можно проверить структуру `git` файла, как показано на рисунке 5:

```
C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126>tree .git
Folder PATH listing
Volume serial number is 000000E0 CC86:B2C2
C:\USERS\DIMON\_VALHALLA\_LERN\Сетевое_программирование\Камков_ПЕ-126\GIT
├── hooks
├── info
├── objects
│   ├── info
│   └── pack
├── refs
│   ├── heads
│   └── tags
```

Рисунок 6 – Проверка структуры файла `.git`

Далее необходимо проверить текущее состояние рабочей директории, как показано на рисунке 7:

```
C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)
```

Рисунок 7 – Проверка состояния директории

Как видно из листинга, есть не отслеживаемые файлы. Как сделать отслеживаемыми – будет чуть-чуть попозже.

Далее необходимо выполнить следующее задание – настроить пользователя Git на уровне локального репозитория:

1) Изучите содержимое файла конфигурации Git для текущего репозитория (.git/config)

2) Настройте имя и email пользователя для текущего репозитория

3) Убедитесь, что файл .git/config изменился соответствующим образом

Для вывода содержимого файла конфигурации Git для текущего репозитория необходимо воспользоваться командой, как показано на рисунке 8.

```
cat .git/config // команда
[core] // отсюда начинается результат
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true
```

Рисунок 8 – Вывод содержимого конфигурации

Далее необходимо задать имя и email пользователя для текущего репозитория при помощи команды. Значение name и your@email.ru указываются студентами самостоятельно.

```
C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126>git config user.name Kamkov_PE-12b
C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126>git config user.email dimon.kamkov@mail.ru
```

Рисунок 9 – Добавление имени и email пользователя

После этого, необходимо проверить содержимое файла конфигурации при помощи команды, которая была использована на рисунке 7. Результат выполнения для тестового задания представлен на рисунке 10.

```
C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126>cd .git
C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126\.git>type config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true
[user]
    name = Kamkov_PE-12b
    email = dimon.kamkov@mail.ru
```

Рисунок 10 – Вывод содержимого файла конфигурации

Следующим заданием будет необходимо создать первый коммит. Для этого необходимо посмотреть состояние рабочей директории, как объяснялось ранее.

После того, как было проверено текущее состояние директории, необходимо добавить все файлы в индекс при помощи команды `git add *`.

После этого, снова необходимо проверить статус репозитория. Если все сделано верно, то файлы попали в индекс и теперь готовы к коммиту, как показано на рисунке 11.

```
C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   img/BMW_M3_G81.jpg
    new file:   img/BMW_M5_G30.jpg
    new file:   img/paw_print.jpg
    new file:   index.html
```

Рисунок 11 – Статус репозитория после добавления в индекс

Далее необходимо выполнить коммит при помощи команды `git commit -m "Project initial"`. После этого должен быть ответ, как показан на рисунке 12.

```
C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126>git commit -m "G-02: Initial"
[master (root-commit) 7e41cd6] G-02: Initial
4 files changed, 33 insertions(+)
create mode 100644 img/BMW_M3_G81.jpg
create mode 100644 img/BMW_M5_G30.jpg
create mode 100644 img/paw_print.jpg
create mode 100644 index.html
```

Рисунок 12 – Ответ после выполнения команды

Далее можно проверить последнее изменение при помощи команды `git log`, как показано на рисунке 13.

```
C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126>git log
commit 7e41cd624f88f1c6cb10d7c0686f2407f1c16a62 (HEAD -> master)
Author: Kamkov_PE-12b <dimon.kamkov@mail.ru>
Date:   Sun Feb 25 17:57:19 2024 +0500

    G-02: Initial

C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126>
```

Рисунок 13 – Результат выполнения команды `git log`

В конце необходимо проверить статус репозитория, результат выполнения представлен на рисунке 14.

```
C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126>git status
On branch master
nothing to commit, working tree clean

C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126>_
```

Рисунок 14 – Результат выполнения `git status`

Индивидуальное задание по пункту 3.1. В существующий файл `index.html` необходимо добавить еще одну картинку и дополнительный текст и сделать второй коммит самостоятельно. Результат выполнения может быть представлен в форме листинга/скриншота.

```
C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126>git init
Reinitialized existing Git repository in C:/Users/dimon/_Valhalla/_Lern/Сетевое_программирование/Камков_ПЕ-126/.git/

C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126>git add *

C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   img/BMW_X5M_G05.jpg
        modified:   index.html

C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Камков_ПЕ-126>
```

Рисунок 15 – Скриншот листинга индивидуального задания

3.2 Индивидуальное задание студента.

- 1) Зарегистрируйтесь на GitHub и подтвердите свою почту.
- 2) Настройте SSH или HTTPS подключение (на ваш выбор). Для того, чтобы настроить SSH, можно прочитать здесь:

<https://timeweb.cloud/tutorials/windows/kak-sgenerirovat-ssh-klyuch-dlya-windows>

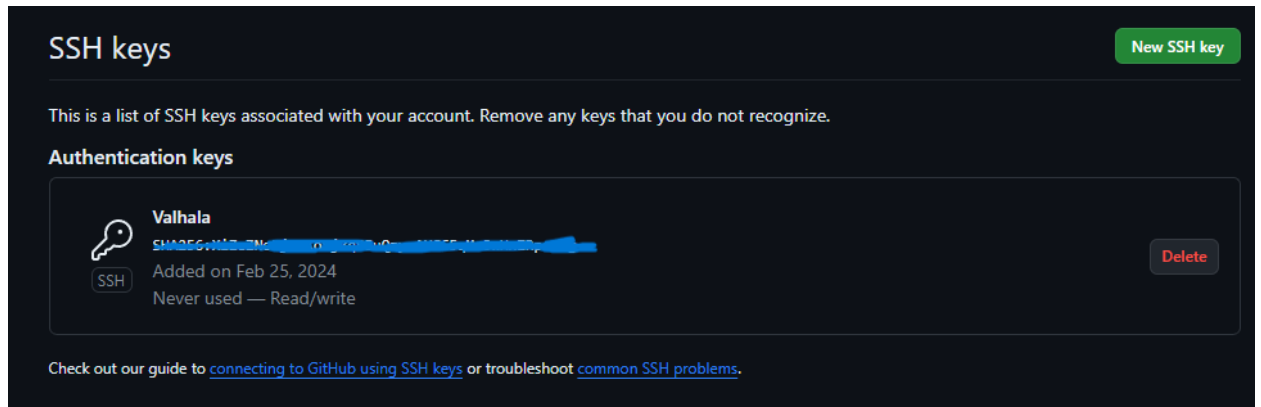


Рисунок 16 – SSH-key задания 3.2.2

- 3) Отправьте ваш локальный репозиторий (созданный в задании 3.1) на аккаунт GitHub (при помощи командной строки или вспомогательного ПО, не имеет значения).

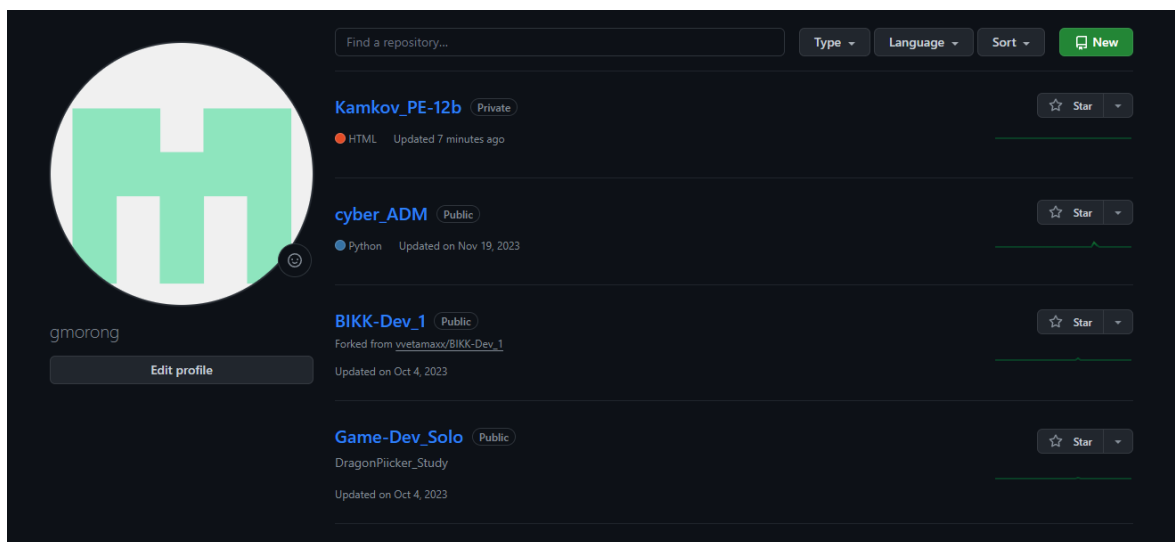


Рисунок 17– Репозиторий на GitHub'е

- 4) Внесите изменения в ваш репозиторий на локальном компьютере, сделайте коммит и отправьте его на GitHub.
- 5) Создайте ветку с указанием Вашей фамилии и инициалов, например, BurumbaevDI

```
C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Kamkov_PE12-b>git branch KamkovDA
C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Kamkov_PE12-b>git checkout KamkovDA
Switched to branch 'KamkovDA'

C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Kamkov_PE12-b>git branch
* KamkovDA
  master

C:\Users\dimon\_Valhalla\_Lern\Сетевое_программирование\Kamkov_PE12-b>_
```

Рисунок 18 – Задание 3.2.5

6) Далее необходимо переключиться на ветку с вашей фамилией, изменить ваш файл index.html и отправить его на GitHub (именно с второй ветки).

7) Необходимо выполнить Pull Request, указав комментарий.

8) После этого необходимо произвести слияние версий от главной ветки (master/main) и второстепенной (в случае примера, это BurumbaevDI). Данный раздел необходимо изучить самостоятельно.

Для того, чтобы выполнить индивидуальное задание No2 необходимо ознакомиться с приложением А к первой практической работе.

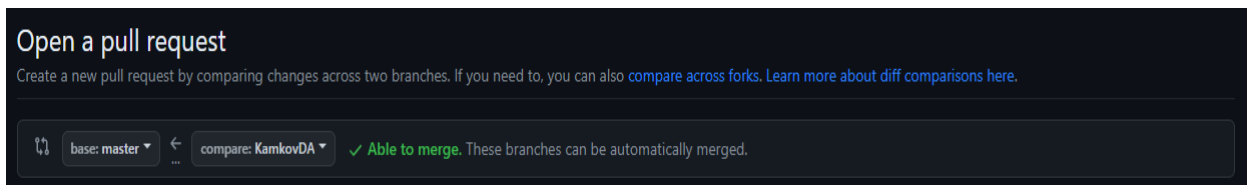


Рисунок 19 – Итог индивидуального задания