



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

О Т Ч Е Т

по лабораторной работе № 1

Название: Основы Git & GitHub

Дисциплина: Языки интернет-программирования

Студент

ИУ6-33Б
(Группа)

(Подпись, дата)

С.В. Сонин
(И.О. Фамилия)

Преподаватель

(Подпись, дата)

В.Д. Шульман
(И.О. Фамилия)

Москва, 2024

Цель работы:

Научиться использовать Git - консольную утилиту, для отслеживания и ведения истории изменения файлов.

Задание:

Установите Git, зарегистрируйтесь на GitHub, сгенерируйте SSH-ключ, сделайте форк репозитория, клонируйте его на свою локальную машину, создайте ветку dev от master, замените "world" на своё имя в text.txt, добавьте и закоммитьте изменения, добавьте отчет в docs, запушьте dev в GitHub и создайте Pull Request в master.

Ход работы:

1) Установка Git.

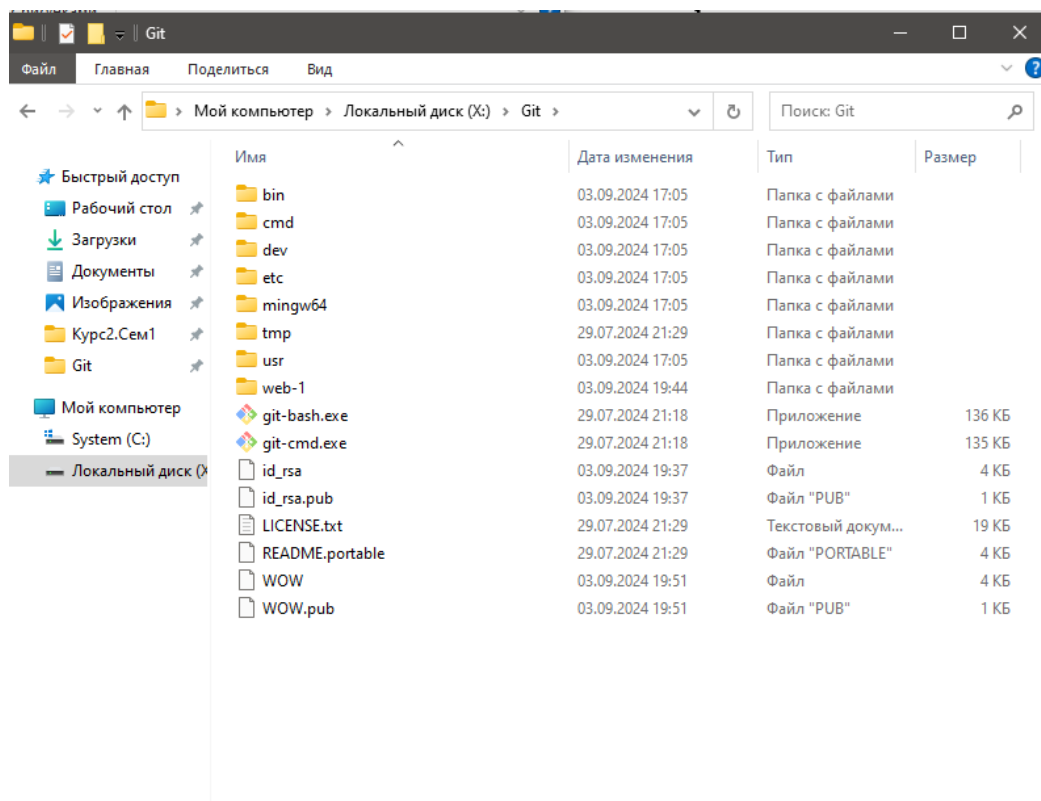


Рисунок 1 – корневой каталок с программой Git.

2) Генерация SSH-ключа и добавление его в аккаунт на GitHub.

```
MINGW64:/
$ ssh-keygen -t rsa -b 4096 -C "sonin.serej@yandex.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/РСТ_ЕВАНГЕЛИСТ/.ssh/id_rsa): WOW
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in WOW
Your public key has been saved in WOW.pub
The key fingerprint is:
SHA256:LS6LF0z/SB0rq10zP+jIRPkVQszQogVWskFdGk5vdkE sonin.serej@yandex.ru
The key's randomart image is:
+---[RSA 4096]-----+
| ."+=BE          |
| .O++O=         |
| O.+ ....       |
| O.+... O       |
| O=. S .        |
| O OOO .        |
| +.O+O .        |
| OOOO=O. O      |
| .+==++OO       |
+---[SHA256]-----+
$
```

Рисунок 2 – Вывод консоли при генерации SSH-ключа.

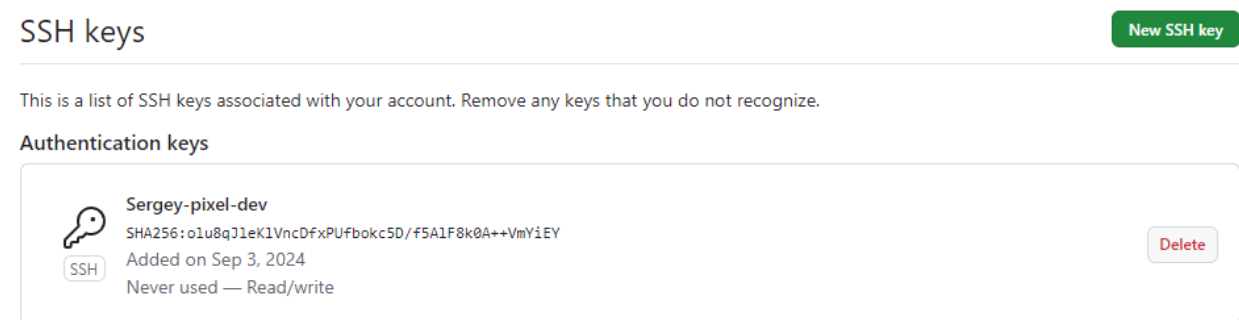


Рисунок 3 – SSH-ключ добавлен в аккаунт на GitHub.

3) Создание локального репозитория путём клонирование удаленного, изменение локального репозитория согласно заданию.

```
MINGW64:/web-1
$ cd web-1

MINGW64 /web-1 (master)
$ git checkout -b
error: switch 'b' requires a value

MINGW64 /web-1 (master)
$ git checkout -b dev
Switched to a new branch 'dev'

MINGW64 /web-1 (dev)
$ git commit -m "hello"
On branch dev
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   text.txt

no changes added to commit (use "git add" and/or "git commit -a")

MINGW64 /web-1 (dev)
$ git add text.txt

MINGW64 /web-1 (dev)
$ git commit -m "hello"
[dev 850dc63] hello
1 file changed, 1 insertion(+), 1 deletion(-)

MINGW64 /web-1 (dev)
$ |
```

Рисунок 4 – Выполнение задания в консоли.

Вывод:

Научился использовать Git - консольную утилиту, для отслеживания и ведения истории изменения файлов.

Список использованных источников:

- <https://dev.to/joshhortt/how-to-generate-and-add-an-ssh-key-to-github-1fe1>
- <https://github.com/ValeryBMSTU/web-1>
- https://learngitbranching.js.org/?locale=ru_RU

Контрольные вопросы:

1. В чём разница между Git и Github?

Git – система контроля версий, в то время как GitHub - хостинговая платформа для Git-репозитория.

2. Как можно объединить несколько коммитов в один коммит?

Необходимо использовать `git rebase -i`

3. Для чего нужен `git rebase`, если есть `git merge`?

`git rebase` нужен для создания чистой и линейной истории, переписывая коммиты на основе другой ветки, тогда как `git merge` сохраняет все ветвления и добавляет коммит слияния.

`rebase` удобен, когда важно упростить историю, убрав ненужные разветвления, особенно перед публикацией изменений.

4. Опишите назначение команд: `clone`, `add`, `pull`, `commit`, `push`, `merge`, `rebase`.

`Git clone` – копирование репозитория с удаленного сервера на локальный.

`Git add` – добавление объектов в систему контроля версий.

`Git pull` - загрузка изменения из удалённого репозитория и автоматически объединяет их с локальной веткой.

`Git merge` – объединение изменения из одной ветки в другую, создавая коммит слияния.

`Git rebase` - перенос коммитов из одной ветки на вершину другой ветки, переписывая историю коммитов.

5. Как локально создать Git-репозиторий с нуля?

Список необходимых команд:

- `mkdir my_project`
- `cd my_project`
- `git init`