Отчёт по лабораторной №1. Работа с GIT

Подготовка, установка и приминение git

Шаян Фаисал НФИбд-02-19

Содержание

List of Figures

1 Цель лабораторной работы №1

Целью данной лабораторной работы является изучение пространства git, применение раздичных команд и отработка эллементарных действий с системой.

2 Ход работы лабораторной №1

Создаем учетную запись на github.com и репозиторий, в котором в дальнейшем будем работать

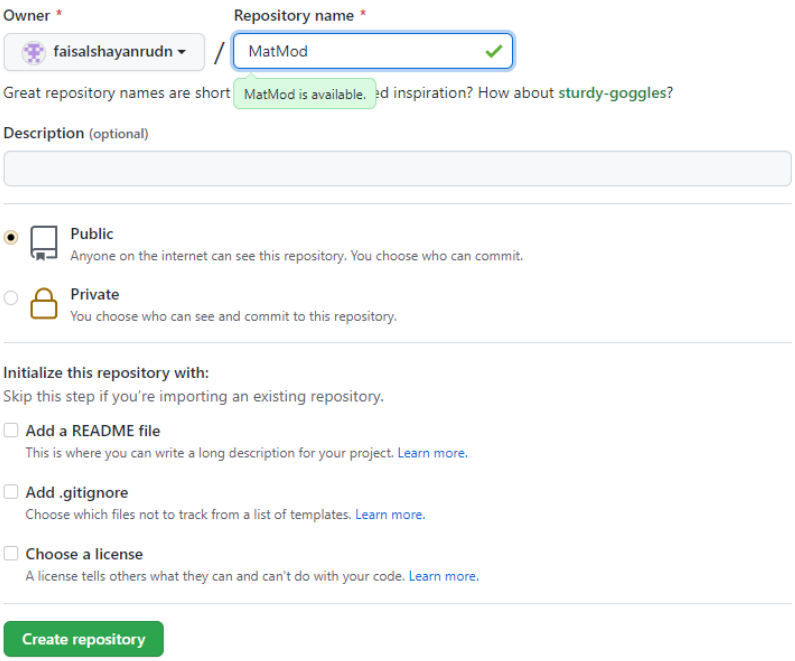

Figure 2.1: Создание учетной записи и репозитория

Инициализируем созданный нами локальный репозиторий, после создаем в нем файл README.md, с которым будем проводить все дальнейшие действия

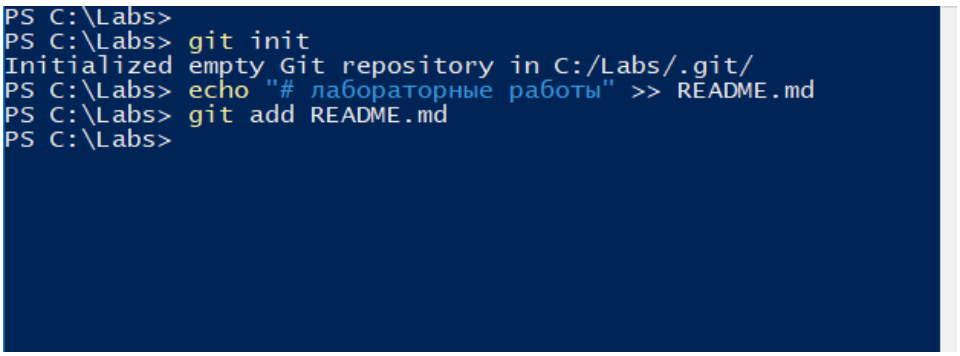

Figure 2.2: Инициализация репозитория и создание файла

Создаем SSH-ключ, который пропишем в настройках на github.com

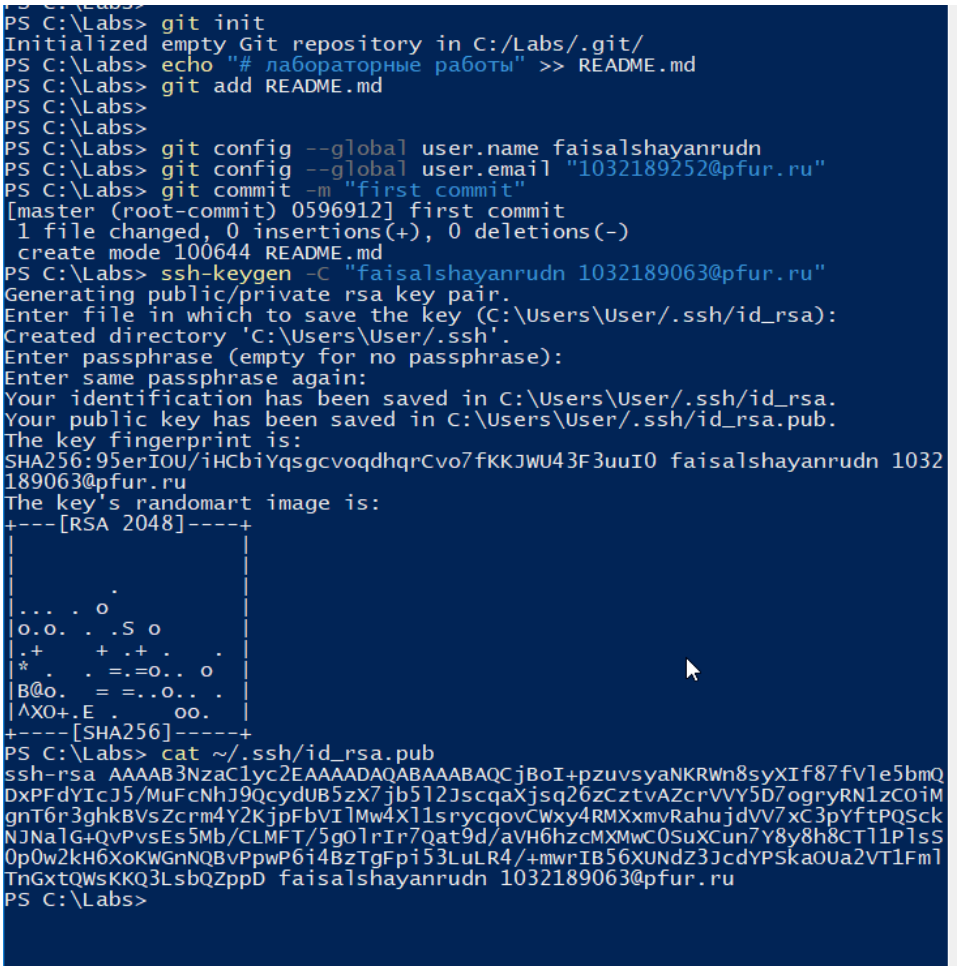

Figure 2.3: Создание ключа

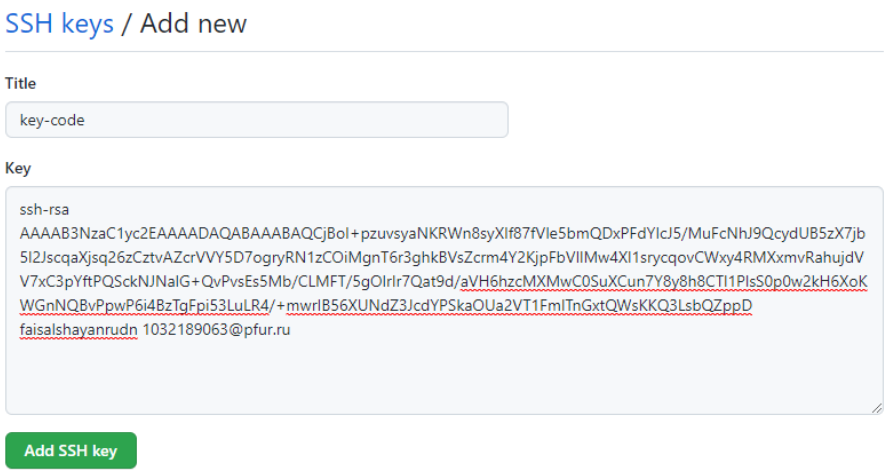

Figure 2.4: Добавляем ключ на сайт

Загружаем файлы лицензионного соглашения и gitignore и отправляем эти файлы в сетевой репозиторий.

```
PS C:\Labs> git remote add origin git@github.com:faisalshayanrudn/MatMod.git
PS C:\Labs> wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
PS C:\Labs> wget https://www.toptal.com/developers/gitignore/api/python -O .gitignore
PS C:\Labs> git add .
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in LICENSE.
The file will have its original line endings in your working directory
PS C:\Labs> git commit -am "add license"
[master 8f3f012] add license
 2 files changed, 555 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 LICENSE
PS C:\Labs> git push -u origin master
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 7.72 KiB | 1.93 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:faisalshayanrudn/MatMod.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS C:\Labs> git push
Everything up-to-date
PS C:\Labs>
```

Figure 2.5: Загрузка файлов с дальнейшей отправкой в сетевой репозиторий

Используем системы управления версиями и тегами. Создаем ветку, начинаем и завершаем в ней релиз.

```
PS C:\Labs> git flow init

Which branch should be used for bringing forth production releases?
   - master
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? [] v
Hooks and filters directory? [C:/Labs/.git/hooks]
PS C:\Labs> git branch
* develop
  master
PS C:\Labs> git flow release start 1.0.0
Switched to a new branch 'release/1.0.0'

Summary of actions:
- A new branch 'release/1.0.0' was created, based on 'develop'
- You are now on branch 'release/1.0.0'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

     git flow release finish '1.0.0'

PS C:\Labs> echo "1.0.0" >> version
PS C:\Labs> git add .
PS C:\Labs> git commit -am "main: add version"
[release/1.0.0 60108bc] main: add version
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 version
PS C:\Labs> git flow release finish -m "ver 1" 1.0.0
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
Merge made by the 'ort' strategy.
```

Figure 2.6: Инициализация git-flow и начало релиза, пробуем простые команды для работы с ветками

```
PS C:\Labs> git push --all
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 495 bytes | 165.00 KiB/s, done.
Total 5 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To github.com:faisalshayanrudn/MatMod.git
   8f3f012..556cf67  master -> master
 * [new branch]      develop -> develop
PS C:\Labs> git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 165 bytes | 82.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:faisalshayanrudn/MatMod.git
 * [new tag]         v1.0.0 -> v1.0.0
PS C:\Labs>
```

Figure 2.7: Завершаем релиз и отправляем изменения в сетевой репозиторий

Выполняем объединение веток с помощью простых команд.

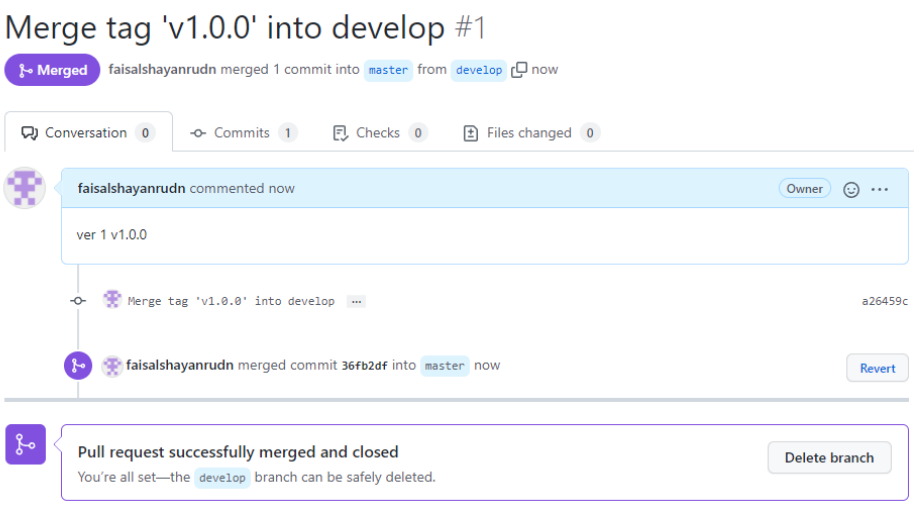

Figure 2.8: Объединяем ветки в сетевом репозитории

3 Вывод:

Мы приобрели практические навыки работы с системой контроля версий git и создали свой репозиторий. Изучили пространства git, применили различные команды и отработали эллементарные действий с системой.

Список литературы

1. Git Шпаргалка
2. Основы Git
3. Руководство по оформлению Markdown файлов