

Лабораторная работа №1

Шаян Фаисал НФИбд-02-19

10 февраля, 2022, Москва, Россия

Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

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

Задачи лабораторной работы

1. Создать учетную запись на github и репозиторий
2. Настроить репозиторий и создать файл для работы дальнейшей
3. Изучить механизм управления версиями и применить простые команды

Процесс выполнения лабораторной работы

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

Owner * / Repository name *

faisalshayanrudn / MatMod ✓

Great repository names are short. **MatMod is available.** Need inspiration? How about [sturdy-goggles?](#)

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Figure 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

```
PS C:\Labs>
PS C:\Labs> git init
Initialized empty Git repository in C:/Labs/.git/
PS C:\Labs> echo "# лабораторные работы" >> README.md
PS C:\Labs> git add README.md
PS C:\Labs>
```

Figure 2: Инициализация репозитория и создание файла

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

```
PS C:\Labs> git init
Initialized empty Git repository in C:/Labs/.git/
PS C:\Labs> echo "# лабораторные работы" >> README.md
PS C:\Labs> git add README.md
PS C:\Labs>
PS C:\Labs> git config --global user.name faisalshayanrudn
PS C:\Labs> git config --global user.email "1032189252@pfur.ru"
PS C:\Labs> git commit -m "first commit"
[master (root-commit) 0596912] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
PS C:\Labs> ssh-keygen -C "faisalshayanrudn 1032189063@pfur.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\User/.ssh/id_rsa):
Created directory 'C:\Users\User/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\User/.ssh/id_rsa.
Your public key has been saved in C:\Users\User/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:95erIOU/iHCbiYqsgcvodhqrCvo7FKKJWu43F3uuI0 faisalshayanrudn 1032189063@pfur.ru
The key's randomart image is:
+---[RSA 2048]-----+
|
|... . O
|o.o. . .S o
|+ . +.+ . O
|+ . =.=o.. O
|B@o. =.=.o..
|/XO+.E . oo.
+---[SHA256]-----+
PS C:\Labs> cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACjBoI+puvsvyaNKRwn8syXif87fv1e5bmQ
DXPfdYIcJ5/MufcNhJ9QcydUB5zX7jb512JscqaXjsq26ZcztvAZcrVVY5D/ogryRN1zc0iM
gnT6r3ghk8BvsZcsm4Y2KjpfbVI1mW4X1lsrycgoVcWxy4RMXmvrRahujdV77x3pYfFP05ck
N2NA16dQvPsf93Mh/GCUPF77Sp01rT7oat9d/avm6hzcX0XwC0SuxCun7V8/8h8CT1P1sS
0p0w2KH6XokW6nNQBvPpwP614BztGfpi53LulR4/umwR1B56XUdZ3JcdyPskao0a2VT1Fm1
TmGxtQWSKQ3LsbQzppd faisalshayanrudn 1032189063@pfur.ru
PS C:\Labs>
```

Figure 3: Создание SSH-ключа

SSH keys / Add new

Title

key-code

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQACj8ol+pzuvsyaNKRWn8syXiI87fVle5bmQDxPFdYicJ5/MuFcNh9QcydUB5zX7jb
5l2JscqaXjsq26zCztvAZcrVVY5D7ogryRN1zCOiMgnT6r3ghkBVszCrm4Y2KjpFbVIIMw4Xi1srycqovCWxy4RMXxmvRahujdV
V7xC3pYfRPQScKNJNaIG+QvPvsEs5Mb/CLUMFT/5gOlrlr7Qat9d/aVH6hzcMXMwC0SuXCun7Y8y8h8CTl1PlsS0p0w2kH6XoK
WGNnQBvPpwP6i4BzTgFpi53LuLR4/+mwrIB56XUNdZ3JcdYP5kaOUa2VT1FmlTnGxtQWskKKQ3LsbQZppD
faisalshayanrudn 1032189063@pfur.ru
```

Add SSH key

Figure 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:+DiY3wvV6TuJJhbpzISF/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 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 6: Инициализируем git-flow и создание релиза

Используем систему управления версиями

```
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 7: Отправляем изменения в сетевой репозиторий

Выполняем объединение веток

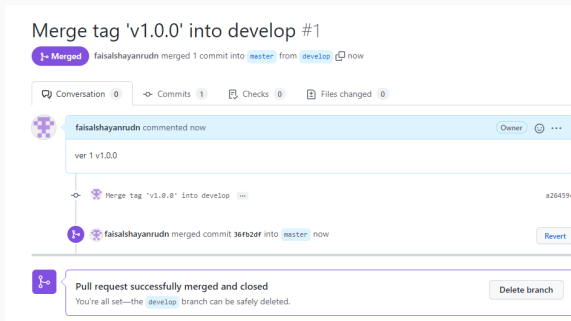


Figure 8: Объединяем ветки в сетевом репозитории

**Сделаем выводы по проделанной
работе в лабораторной №1:**

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