

Лабораторная работа 1

Стаменкович Огнен НПИбд-02-19

11 февраля, 2022, Москва, Россия

Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

Целью данной работы является изучение идеологии и применения средств контроля версий.



Задачи лабораторной работы

1. Создать учетную запись на github.com
2. Настроить репозиторий
3. Изучить механизм управления версиями

Процесс выполнения лабораторной работы


Создаем учетную запись на github.com и репозиторий


Owner * Repository name *

 Ognenrudn / MatMod 

Great repository names are short and memorable. Need inspiration? How about [crispy-spork?](#)

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 репозитория

Инициализируем локальный репозиторий

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

Figure 2: Инициализация репозитория

Создаем SSH-ключ

```
PS C:\MatMod> git init
Initialized empty Git repository in C:\MatMod\.git\
PS C:\MatMod> echo "# matmod\мне работы" >> README.md
PS C:\MatMod> git add README.md
PS C:\MatMod> git config --global user.name Ognenrudn
PS C:\MatMod> git config --global user.email "1032195215@pfur.ru"
PS C:\MatMod> git commit -m "first commit"
[master (root-commit) 5486b56] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
PS C:\MatMod> ssh-keygen -C "ognenrudn 1032195215@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):
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:jgs3YgkH6jouAWLUGX1fzi5/joqP88/v41281Yzps ognenrudn 1032195215@pfur.ru
The key's randomart image is:
+----[RSA 2048]-----+
|
|++0+
|00+
|+0+
|..+..+
|.. 0 . S
|.. + + 0
|.. + . 0
|.. + 0 . + B .
|+0+++++. 0 L
+----[SHA256]-----+
PS C:\MatMod> cat .ssh\id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAWd/bfuo/4YNfCK3prxjjozgaXumv/inQw1x8HeB2o4VDNlTynGobxf1av2eMqHjtkoH43w3rtUVT0
klvbyB6+IRmwnJtjpl6Klwe/tbKvKv+jeghs9ym0/jKs6cFup001enPQ9A/yfArbosc19188HfTJCxoAwMwQTVIMLTw10gtYd/WHHShzghYEddVK14
ACdJ7/zsdcjvZ/kyML007DrFXcj/wK10Kt53wPHSue9/HosQ+AzfyLAj25BenvMhQWk4nR23n9V1Cq4HmsnRLXsSuswGn+oZCtyn63sJ8MPkdqSe
9htKcJad1qH1skEgHx ognenrudn 1032195215@pfur.ru
PS C:\MatMod>
```

Figure 3: Создание SSH-ключа

Создаем SSH-ключ

SSH keys / Add new

Title

MatMod

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDAWD/bfbuo/4YNfCK5prjioxzgaXumV/inQMlxE8HeB2O4VDN1TYngQbXF1av
2eMgHjtKoH+3w/RtUVTOKUvbY86+IRrwKNjhjpi6LKIwe7etbKYkV+jeghS9ynO7JKs6cpfuPO0ienPQ9A/Y7Arbosec19188Hf
TJCXoAwMwqTVTMLTwCl0qtydOWmHShZgHyEDdVK14MCDj7z7zsdcjv2/KyYML0O7DrFXCj/vjKlbK9t53wUPHSuE9/Ho5Q
+AzFyLAJZ56envMhQWHK4mR23nN9VICq4hmsnRLox5SuswGN+oZCtyn63sj8MPKdQSe9HtHCja6lQhLSqk6g4Lx
Oggenrudn 1032195215@pfur.ru
```

Add SSH key

Figure 4: Добавление ключа на github.com

Загружаем служебные файлы

```
PS C:\MatMod> git remote add origin git@github.com:Ognenrudn/MatMod.git
PS C:\MatMod> wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
PS C:\MatMod> wget https://www.toptal.com/developers/gitignore/api/python -O .gitignore
PS C:\MatMod> 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:\MatMod> git commit -m add license
[master 0d2cc38] add license
 2 files changed, 355 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 LICENSE
PS C:\MatMod> git push -u origin master
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DIY3wvvV6TUJhbpZ1sF/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% (3/3), done.
Writing objects: 100% (7/7), 7.71 KiB | 1.54 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Ognenrudn/MatMod.git
 * [new branch] master -> master
branch 'master' set up to track 'origin/master'.
PS C:\MatMod> git push
Everything up-to-date
PS C:\MatMod>
```

Figure 5: Загрузка файлов лицензии и gitignore

Использование системы управления версиями

```
PS C:\MatMod> 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': [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:/MatMod/.git/hooks]
PS C:\MatMod> git branch
* develop
  master
PS C:\MatMod> 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:\MatMod> echo "1.0.0" >> version
PS C:\MatMod> git add .
PS C:\MatMod> git commit -am "main: add version"
[release/1.0.0 4681280] main: add version
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 version
PS C:\MatMod> 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.
version | Bin x -> 1 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 version
Already on 'master'
Your branch is ahead of 'origin/master' by 2 commits.
```

Figure 6: Инициализация git-flow и создание релиза

Использование системы управления версиями

```
PS C:\MatMod> 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), 468 bytes | 234.00 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:Ognenrudn/MatMod.git
 * 0d2cc38..2e049c4 master -> master
 * [new branch] develop -> develop
PS C:\MatMod> git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 161 bytes | 161.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Ognenrudn/MatMod.git
 * [new tag] v1.0.0 -> v1.0.0
PS C:\MatMod>
```

Figure 7: Отправка изменений в сетевой репозиторий

Выводы по проделанной работе

Мы приобрели практические навыки работы с системой контроля версий git и создали свой репозиторий