

Лабораторная работа 1

Деакиссим Манн Орнела НФИбд-01-19

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 *

Ornelt / MatMod ✓

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

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

```
C:\MatMod> git init
Initialized empty Git repository in C:/MatMod/.git/
PS C:\MatMod> echo ? nlsoparomw paborn >> README.md
PS C:\MatMod> git add README.md
PS C:\MatMod> git config --global user.name Ornelll
PS C:\MatMod> git config --global user.email "1032189185@pfur.ru"
PS C:\MatMod> git commit -m "first commit"
[master (root-commit) fa7067d] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
PS C:\MatMod> ssh-keygen -C "Ornelll 1032189185@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:T6kXxAervofRQ5zW4SzmDIPGNVaxss1ljBoORfKAA Ornelll 1032189185@pfur.ru
The key's randomart image is:
+--[RSA 2048]--+
|      .o=O    ..+
|     +XE.....+
|    =Bod....+.
|   +..+...+..+
|  +.+...+..+
| +.o.o..+
| S        |
+-----[SHA256]-----
C:\MatMod> cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaGlnLnZlcAEAAAADAQAAG8ReKnRPFXt9UplZNMHnFokkYxSFNTtTPK6Id180wvudwdIpc2cmwGa9uoStTacDlYBVozMkLYL
wHPPUgukOCYNcskk4BEjOFderdsyrcn/CCLIGSP1v590ms/OqphryZE3P8v3M/xzhRuVA4lycGGPFNL7gcN/FUVUGovycOB0soJ57JAiotb
r/qN3TTE/AKhPTdyAkwGEPIeSLAUvlEBldfyS28EfFdqzdtaagLUzd/LFL6SMHRZURDeFRkkZV05SToEqRGv/jkWbkGwlItyv+qxTdSpPRvdYxtqd
dpJPATnmDzEjUmyv5 Ornelll 1032189185@pfur.ru
```

Figure 3: Создание SSH-ключа

Создаем SSH-ключ

SSH keys / Add new

Title

MatMod

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQD8G8mKNwiPFtx9UpL9ZMNnf0kKf5XfNitiPKk61dlI8oWYuOdw4ipZcowvG+a9
wo+5tTmcDIYBVoZMk1ywwhPPUGwOkYNcsKXc5qbE/0FderdSyrcln/CCIIIGSPiV59Omns/O+pH9yZFE3p8v3M/x2hRvA44lyG
sGKPfNL72gcN/XN340/EFUVOqxYcoD8QSOJ5j7aioToB/gN3TT6/ARr9TAUWGZePHfslAaUvIEbdfiysz8eEffdoq+tdaqpLUzd
71rL65SMHrZQRDefrikk2V05SToJeqrGXv//JkwKBEGWIIYv+y+qTXdSppfRVdYYxtq0o2jP7Nim8rzhOj3yxAv5 Ornel1
1032189185@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:Ornell/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 3be32d1] 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 (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:0dV3vvv61u3Jhbpz1sF/zLDA0zPMsvHdkr40vcOgu.
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: 2, done.
Counting objects: 100% (2/2), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 7.70 KiB | 1.92 MiB/s, done.
Total (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Ornell/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" 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:/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 df15369] 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 0 -> 16 bytes
```

Figure 6: Инициализация git-flow и создание релиза

Использование системы управления версиями

```
PS C:\MatMod>
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), 467 bytes | 233.00 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:Ornell/MatMod.git
   3be37a1..e3aeeb3  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), 159 bytes | 159.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Ornell/MatMod.git
   * [new tag]       v1.0.0 -> v1.0.0
PS C:\MatMod>
```

Figure 7: Отправка изменений в сетевой репозиторий

Выполним объединение веток

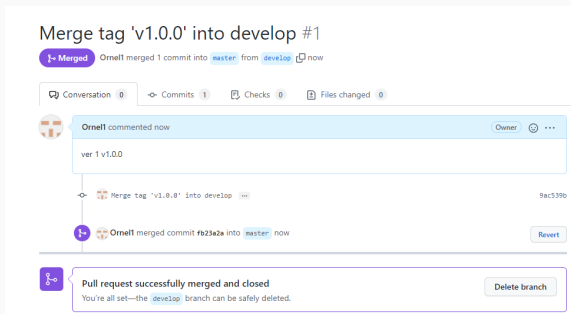


Figure 8: Объединение веток в сетевом репозитории

Выводы по проделанной работе

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