

Лабораторная работа №1. Математическое моделирование.

Система контроля версий Git

Миша Нкого Хосе Адольфо Мба НФИбд-02-19

Содержание

| | | |
|----------|--|-----------|
| 1 | Цель работы | 4 |
| 2 | Выполнение лабораторной работы | 5 |
| 2.1 | Создаем учетную запись на github.com. Создаем публичный репозиторий на github.com. | 5 |
| 2.2 | Создаем локальный репозиторий. Инициализируем его. Создаем в нем файл README.md. | 6 |
| 2.3 | Создаем SSH-ключ. Прописываем его в настройках на сайте. . . . | 6 |
| 2.4 | Загружаем и отправляем файлы лицензионного соглашения и gitignore в сетевой репозиторий. | 7 |
| 2.5 | Изучаем систему управления версиями файлов. Создаем ветку, начинаем и завершаем в ней релиз. | 8 |
| 2.6 | Выполним объединение веток | 9 |
| 3 | Вывод | 10 |
| | Список литературы | 11 |

List of Figures

| | | |
|-----|--|---|
| 2.1 | Создание репозитория на github.com. | 5 |
| 2.2 | Инициализация репозитория и файла README.md. | 6 |
| 2.3 | Создание ключа и прописывание его на github. | 6 |
| 2.4 | Добавление ключа на github.com | 7 |
| 2.5 | Загрузка файлов | 7 |
| 2.6 | Инициализация начала и завершения релиза ветки | 8 |
| 2.7 | Завершаем релиз. Отправляем список изменений в сетевой репозиторий | 8 |
| 2.8 | Объединение веток в сетевом репозитории | 9 |


1 Цель работы

Целью данной работы мы ставим изучение и применения средств контроля версий git, а также работу с различными командами в консоли.

2 Выполнение лабораторной работы


2.1 Создаем учетную запись на github.com. Создаем публичный репозиторий на github.com.


Owner * Repository name *

 timbow64 ▾ / MatMod ✓

Great repository names are short and memorable. Need inspiration? How about [bug-free-octo-system?](#)

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 2.1: Создание репозитория на github.com.

2.2 Создаем локальный репозиторий. Инициализируем его. Создаем в нем файл README.md.

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

Figure 2.2: Инициализация репозитория и файла README.md.

2.3 Создаем SSH-ключ. Прописываем его в настройках на сайте.

```
PS C:\work> git config --global user.name timbow64
PS C:\work> git config --global user.email "1032189237@pfur.ru"
PS C:\work> git commit -m "first commit"
[master (root-commit) 41c0f33] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
PS C:\work> ssh-keygen -C "timbow64 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:
SHA256:3/vLCIQ51ke1cTXwIyw2oZOANod65m84Nu/Jz7PpfD4 timbow64 1032189063@pfur.
ru
The key's randomart image is:
+---[RSA 2048]---+
  oo.o ooo
  ==.. B . .
  o*oo X o o
  ..o+ . = . .
  + S
  . o o
  o o +
  =.o+.oEo
  . *oB=.o.
+---[SHA256]-----+
PS C:\work> cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCEjEXrZusmXkz1VqFMg087oHwYwynJR5IHgbzw
XAGVvGN9xhIZmuAcIX+4+Kpw2XsDYsI18SLPPzVgVgKinDe+AYDfrC1sxMvEKd1rCm7fbdc0FkQU
TpmAv9PyA2Gpdq+d3r7te91sgAmL6g08TyKx/zjXYTHdfJJzwSwLAU1ZmFq8t8bE+fnfBHVgvUN
HLH0nuZ9wsHqj1l+WEd3sE7w/v/S1Cb/hN5VhN/8P36GchbNKvKyeXZ1hyy/IX0BTwRg2XYw7kmfV
bht7cnKDI479c5FZV0o+XBFY1l+A3CZicsIEGLThpuSOT4WANTaa3K/kmo/fd+SBXV8jS1HaqiFZ
timbow64 1032189063@pfur.ru
PS C:\work>
```

Figure 2.3: Создание ключа и прописывание его на github.

SSH keys / Add new

Title

key

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCEjEXrZusmXkz1VqFMgO87oHwYwynJR5IHgbzwXAGVvgN9xhIZmuAcIX+4+Kp
W2XsDysl18SLPPzVgVgKinDe+AYDfrClxMvEKd1rCm7fbdcoFkQUtpimAv9PyA2Gpdq+d3r7te91sgAmL6g08TyKx/zjXYTH
dfllzwSwLAU1ZmFq8t8bE+fnfBHVgVUNHLHQnuz9wsHqJl+WEEd3sE7W/v7SlCb/hN5VhN/8P36GchbNkvKyeXZ1hy/IxOB
TwRg2XYW7kmfVbht7cnKDI479C5FZVOo+XBfYl1+A3CZicsIEGLThpuSOT4WANTaa3K/kmo/fd+SBXV8jS1HaqIFZ
timbow64 1032189063@pfur.ru
```

Add SSH key

Figure 2.4: Добавление ключа на github.com

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

```
PS C:\work> git remote add origin git@github.com:timbow64/MatMod.git
PS C:\work> wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
PS C:\work> wget https://www.toptal.com/developers/gitignore/api/python -O .gitignore
PS C:\work> 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:\work> git commit -am "add license"
[master 1d92a72] add license
2 files changed, 555 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 LICENSE
PS C:\work> 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), 7.70 KiB | 2.57 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:timbow64/MatMod.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS C:\work> git push
Everything up-to-date
```

Figure 2.5: Загрузка файлов

2.5 Изучаем систему управления версиями файлов.

Создаем ветку, начинаем и завершаем в ней релиз.

```
PS C:\work> 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:/work/.git/hooks]
PS C:\work> git branch
* develop
master
PS C:\work> git flow release start 1.0.0
Switched to 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:\work> echo "1.0.0" >> version
PS C:\work> git add .
PS C:\work> git commit -am "chore(main): add version"
[release/1.0.0 bab9c8b] chore(main): add version
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 version
PS C:\work> git flow release finish -m "ver 1" 1.0.0
```

Figure 2.6: Инициализация начала и завершения релиза ветки

```
PS C:\work> 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), 474 bytes | 237.00 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:timbow64/MatMod.git
 * [new branch] develop -> develop
PS C:\work> git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 158 bytes | 52.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:timbow64/MatMod.git
 * [new tag] v1.0.0 -> v1.0.0
PS C:\work>
```

Figure 2.7: Завершаем релиз. Отправляем список изменений в сетевой репозиторий

2.6 Выполним объединение веток

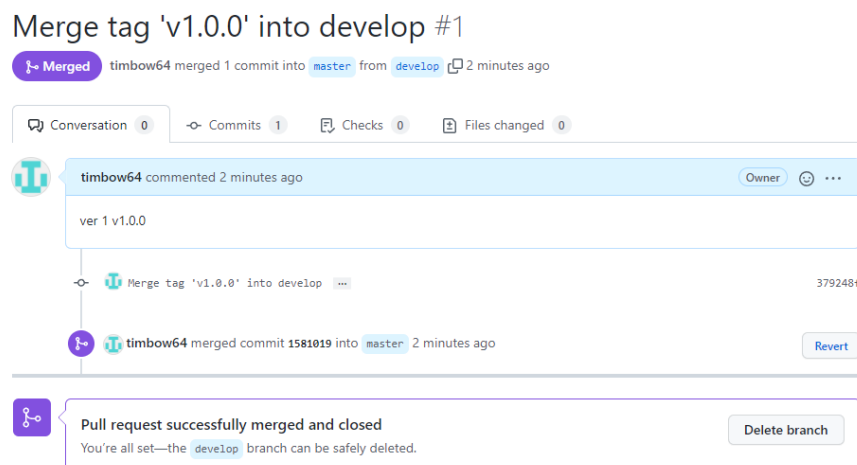


Figure 2.8: Объединение веток в сетевом репозитории

3 Вывод

Мы изучили и применили средства контроля версий git, а также научились работать с различными командами в консоли.

Список литературы

1. Git для чайников
2. Руководство по Markdown