

Laboratory №3

Krupennikova V.

MAY-2021

RUDN University, Moscow, Russian Federation

Изучить идеологию и применение средств контроля версий

Сделать отчёт по предыдущей лабораторной работе в формате Markdown.

1) Создаю учетную запись на <https://github.com>.

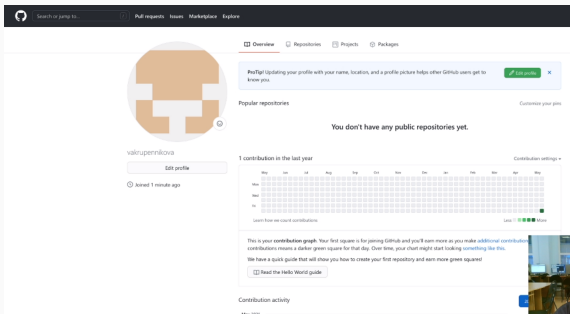


Figure 1: Учетная запись

- 2) Настраиваю систему контроля версий git.
- 3) Синхронизирую учётную запись github с компьютером: `git config --global user.name "Имя Фамилия"` `git config --global user.email "work@mail"`

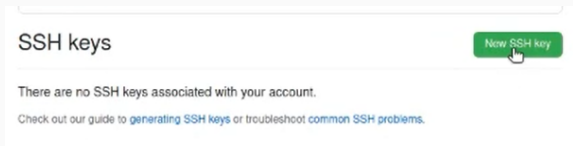
```
vakrupennikova@dk4n58 ~ $ git config --global user.name"vakrupennikova"  
vakrupennikova@dk4n58 ~ $ git config --global user.email"krupka2002@yandex.ru"
```

Figure 2: Настройка системы контроля версий

- 4) Создаю новый ключ на github (команда `ssh-keygen -C "vakrupennikova.krupka2002@yandex.ru"`) и привязываю его к компьютеру через консоль.

```
vakrupennikova@dk4n58 ~ $ ssh-keygen -C "vakrupennikova.krupka2002@yandex.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/v/a/vakrupennikova/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/v/a/vakrupennikova/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/v/a/vakrupennikova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ZK6aLSWHC1tCn+gXhgFS+KGdfmzS7Cn1XCc1KEuyWV4 vakrupennikova.krupka2002@yandex.ru
The key's randomart image is:
+----[RSA 3072]-----+
|oo.                |
|= .                 |
|o*.o               |
|ooO o +.E o         |
|+.O Bo+S. .         |
|. +@.o%o .          |
|. B.++.oo           |
| o o o.             |
| .                  |
+----[SHA256]-----+
```

Figure 3: Создаю новый ключ





- 5) В github захожу в «repositories» и создаю новый репозиторий. Копируем в консоль ссылку на репозиторий.

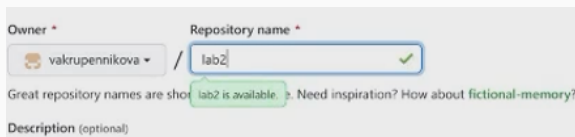


Figure 4: Создаю репозиторий

- 6) Работаю с каталогом и папками через консоль. Перед тем, как создавать файлы, захожу в репозиторий и создаю файлы:

```
vakrupennikova@dk4n58 ~ $ cd lab2
vakrupennikova@dk4n58 ~/lab2 $ mkdir 2020-2021
vakrupennikova@dk4n58 ~/lab2 $ cd 2020-2021
vakrupennikova@dk4n58 ~/lab2/2020-2021 $ mkdir os
vakrupennikova@dk4n58 ~/lab2/2020-2021 $ cd os
vakrupennikova@dk4n58 ~/lab2/2020-2021/os $ mkdir laboratory
vakrupennikova@dk4n58 ~/lab2/2020-2021/os $ cd laboratory
```

Figure 5: Работа с каталогами и папками

```
vakrupennikova@dk4n58 ~/lab2/2020-2021/os/laboratory $ touch h.txt
```


- 7) Добавляю первый коммит и выкладываю на github. Для того, чтобы правильно разместить первый коммит, необходимо добавить команду `git add .`, далее с помощью команды `git commit -m "first commit"` выкладываем коммит:

```
vakrupennikova@dk4n58 ~/lab2/2020-2021/os/laboratory $ git add .
vakrupennikova@dk4n58 ~/lab2/2020-2021/os/laboratory $ git commit -am "first commit"
[main 63e839b] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 2020-2021/os/laboratory/h.txt
```

Figure 6: Выкладываю первый коммит

8) Сохраняю первый коммит (git push):

```
vakrupennikova@dk4n58 ~/lab2/2020-2021/os/laboratory $ git push
Username for 'https://github.com': vakrupennikova
Password for 'https://vakrupennikova@github.com':
Перечисление объектов: 11, готово.
Подсчет объектов: 100% (11/11), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (7/7), 6.59 KiB | 6.59 MiB/s, готово.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/vakrupennikova/lab2.git
  63e839b..4e66693  main -> main
```

Figure 7: Сохраняю первый коммит

- 9) Добавляю файл лицензии (скринкаст оборвался, скриншоты тоже).
- 10) Добавляю шаблон игнорируемых файлов. Получаю список имеющихся шаблонов.
- 11) Скачиваю шаблон, например, для C. Также добавляю новые файлы и выполняю коммит.

12)Отправляю на github (git push):

```
vakrupennikova@dk4n58 ~/lab2/2020-2021/os/laboratory $ git push
Username for 'https://github.com': vakrupennikova
Password for 'https://vakrupennikova@github.com':
Перечисление объектов: 11, готово.
Подсчет объектов: 100% (11/11), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (7/7), 6.59 KiB | 6.59 MiB/s, готово.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/vakrupennikova/lab2.git
  63e839b..4e66693  main -> main
```

Figure 8: Отправляю файлы на github

13) Инициализирую git-flow, используя команду `git flow init -f` (префикс для ярлыков установлен в `v`). Проверяю, что нахожусь на ветке `develop` (git branch). Создаю релиз с версией 1.0.0. Записываю версию и добавляю в индекс.

```
echo "1.0.0" » VERSION
```

```
git add .
```

```
git commit -am 'chore(main): add version'
```

14) Заливаю релизную ветку в основную ветку (команда `git flow release finish 1.0.0`).

15) Отправляю данные на github: git push - -all git push - -tags

```
vakrupennikova@dk4n58 ~/lab2/2020-2021/os/laboratory $ git push --all
Username for 'https://github.com': vakrupennikova
Password for 'https://vakrupennikova@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/vakrupennikova/lab2.git
 * [new branch]      develop -> develop
 * [new branch]      release/1.0.0 -> release/1.0.0
vakrupennikova@dk4n58 ~/lab2/2020-2021/os/laboratory $ git push --tags
Username for 'https://github.com': vakrupennikova
Password for 'https://vakrupennikova@github.com':
Everything up-to-date
```

Figure 9: Отправляю данные

16) Создаю релиз на github. Заходим в «Releases», нажимаю «Создать новый релиз». Захожу в теги и заполняю все поля (теги для версии 1.0.0). После создания тега, автоматически сформируется релиз.

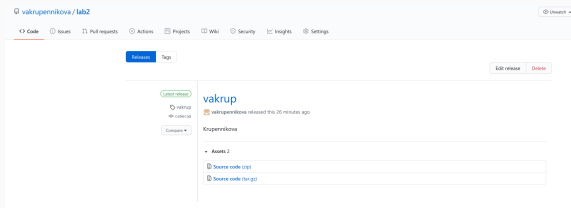


Figure 10: Релиз

The end.