

Отчёт по лабораторной работе №2

Управление версиями

Мурашко Виктория Витальевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Настройка git	7
3.2	Подключение репозитория к github	8
3.3	Первичная конфигурация	8
3.4	Конфигурация git-flow	9
4	Выводы	12

List of Tables

List of Figures

1 Цель работы

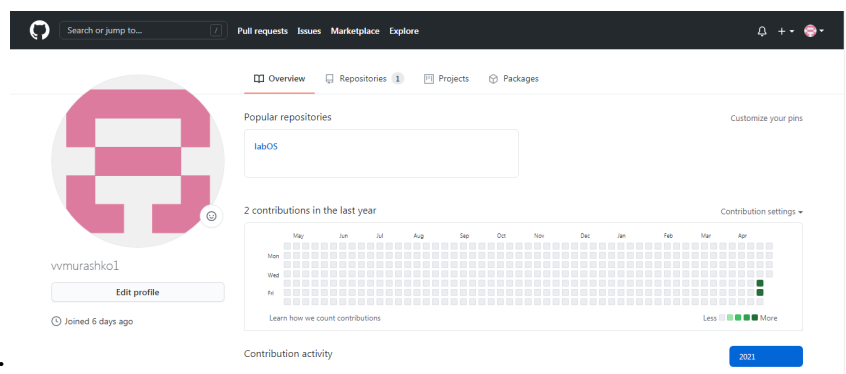
Изучить идеологию и применение средств контроля версий.

2 Задание

1. Настройка git
2. Подключение репозитория к github
3. Первичная конфигурация
4. Конфигурация git-flow

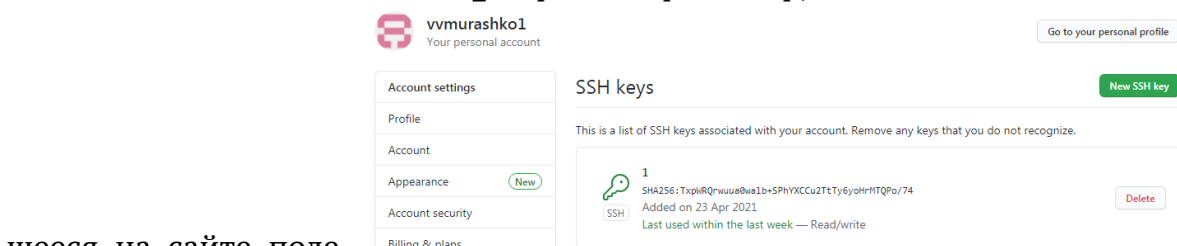
3 Выполнение лабораторной работы

3.1 Настройка git



Создаём учётную запись на github:

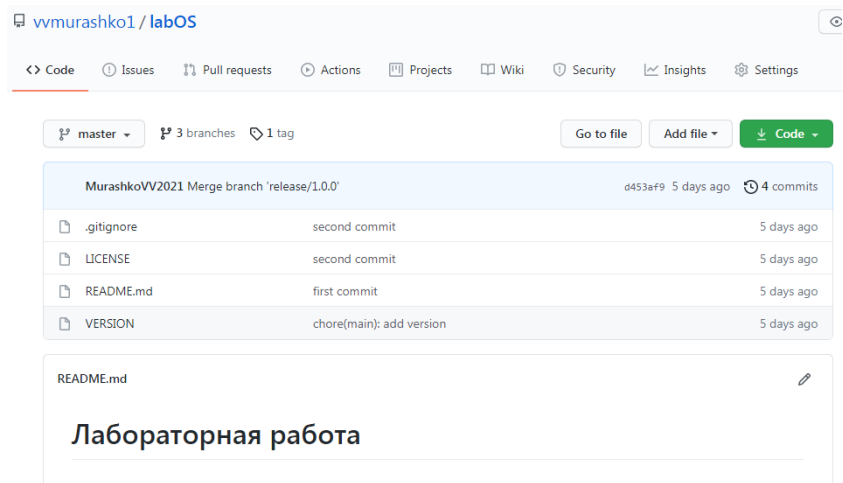
Настраиваем систему контроля версий git: для этого генерируем ключ с помощью команды `ssh-keygen -C «Имя Фамилия work@mail»`, загружаем сгенерённый ключ на github, далее, скопировав из локальной консоли ключ в буфер обмена с помощью команды `cat ~/.ssh/id_rsa.pub | xclip -sel clip`, вставляем ключ в появив-



шающаяся на сайте поле.



3.2 Подключение репозитория к github



Создаём репозиторий на github:

```
vvmurashko1@dk6n53 ~ $ cd
vvmurashko1@dk6n53 ~ $ mkdir laboratory
vvmurashko1@dk6n53 ~ $ cd laboratory
```

Переходим в рабочий каталог laboratory:

Инициализируем системы git:

```
vvmurashko1@dk6n53 ~/laboratory $ git init
Инициализирован пустой репозиторий Git в /afs/.dk.sci.pfu.edu.ru/home/v/v/vvmurashko1/laboratory/.git/
```

Создаём заготовку для файла README.md:

```
vvmurashko1@dk6n53 ~/laboratory $ echo "# Лабораторная работа #" >> README.md
vvmurashko1@dk6n53 ~/laboratory $ git add README.md
```

```
vvmurashko1@dk6n53 ~/laboratory $ git commit -m "first commit"
[master (корневой коммит) 5e2419a] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
vvmurashko1@dk6n53 ~/laboratory $ git remote add origin git@github.com:
vvmurashko1@dk6n53 ~/laboratory $ git push -u origin master
The authenticity of host 'github.com (140.82.121.3)' can't be established
RSA key fingerprint is SHA256:nThbg6kXUpJWGL17E1IGOCspRomTxdCARLviKw6E5S
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com,140.82.121.3' (RSA) to the list
of known hosts.
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 248 bytes | 248.00 KiB/s, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com: vvmurashko1/labOS.git
 * [new branch] master -> master
Ветка «master» отслеживает внешнюю ветку «master» из «origin».
```

Делаем первый коммит и выкладываем на github:

3.3 Первичная конфигурация

```
vvmurashko1@dk6n53 ~/laboratory $ wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
--2021-04-23 14:29:13-- https://creativecommons.org/licenses/by/4.0/legalcode.txt
Результат: creativecommons.org (creativecommons.org): 104.20.151.16, 172.07.34.140, 104.20.150.16, ...
Подключение к creativecommons.org (creativecommons.org)[104.20.151.16]:443... соединение установлено.
HTTP-запрос: отправлен. Ожидание ответа... 200 OK
Длина: нет данных [text/plain]
Скачивание в: «LICENSE»
[ 10,22K --:--KB/s за 0,001s ]
```

Добавим файл лицензии:

Добавим шаблон игнорируемых файлов и посмотрим список имеющихся


```
vvmurashko1@dk6n53 ~/laboratory $ curl -L -s https://www.gitignore.io/api/list
1c,1c-bitrix,a-frame,actionscript,ada
adobe,advancedinstaller,adventuregamestudio,agda,al
alteraquartusii,altium,amplify,android,androidstudio
angular,anjuta,ansible,apachecordova,apachehadoop
appbuilder,appcelerator titanium,appcode,appcode+all,appcode+iml
appengine,aptanastudio,arcanist,archive,archives
archlinuxpackages,aspnetcore,assembler,ate,atmelstudio
ats,audio,automationstudio,autotools,autotools+strict
awr,azurefunctions,backup,ballerina,basercms
basic,batch,bazaar,bazel,bitrise
bitrix,bittorrent,blackbox,bloop,bluej
bookdown,bower,bricxcc,buck,c
c++,cake,cakephp,cakephp2,cakephp3
```

шаблонов:

Затем скачаем шаблон: `zsh,zukencr8000 vvmurashko1@dk6n53 ~/laboratory $ curl -L -s https://www.gitignore.io/api/c >> .gitignore`

Добавим новые файлы: `vvmurashko1@dk6n53 ~/laboratory $ git add .`

```
vvmurashko1@dk6n53 ~/laboratory $ git commit -am "second commit"
[master 99e6ff1] second commit
2 files changed, 455 insertions(+)
create mode 100644 .gitignore
create mode 100644 LICENSE
```

Выполним коммит:

```
vvmurashko1@dk6n53 ~/laboratory $ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (4/4), 6.43 KiB | 6.43 MiB/s, готово.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:vvmurashko1/lab0S.git
5e2419a..99e6ff1 master -> master
```

Отправим на github:

3.4 Конфигурация git-flow

```
vvmurashko1@dk6n53 ~/laboratory $ 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? [/afs/.dk.sci.pfu.edu.ru/home/v/v/vvmurashko1/laboratory/.git/hooks]
```

Инициализируем git-flow:

Проверим, что мы на ветке develop: `vvmurashko1@dk6n53 ~/laboratory $ git branch`

```
vvmurashko1@dk6n53 ~/laboratory $ git flow release start 1.0.0
Переключено на новую ветку «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'
```

Создадим релиз с версией 1.0.0:

Запишем версию:

```
vvmurashko1@dk6n53 ~/laboratory $ echo "1.0.0" >> VERSION
vvmurashko1@dk6n53 ~/laboratory $ git add .
vvmurashko1@dk6n53 ~/laboratory $ git commit -am'chore(main): add version'
> git flow release start 1.0.0
>
>
>
> hj
> \BDFHZ
>
>
[release/1.0.0 90495f0] chore(main): add version git flow release start 1.0.0
1 file changed, 1 insertion(+)
create mode 100644 VERSION
```

Добавим в индекс:

```
vvmurashko1@dk6n53 ~/laboratory $ git flow release finish 1.0.0
Переключено на ветку «master»
Ваша ветка обновлена в соответствии с «origin/master».
Merge made by the 'recursive' strategy.
VERSION | 1 +
1 file changed, 1 insertion(+)
create mode 100644 VERSION
Уже на «master»
Ваша ветка опережает «origin/master» на 2 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
fatal: нет описания метки?
Fatal: Tagging failed. Please run finish again to retry.
```

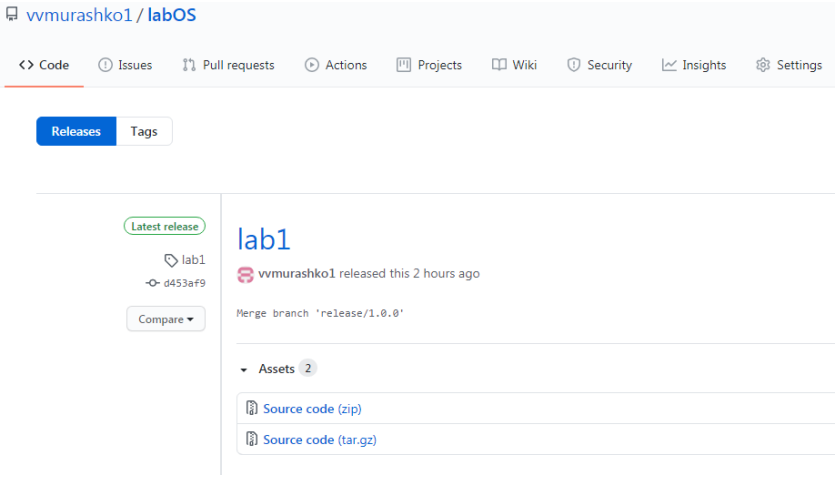
Зальём релизную ветку в основную ветку:

```
vvmurashko1@dk6n53 ~/laboratory $ git push --all
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (4/4), 415 bytes | 415.00 KiB/s, готово.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:vvmurashko1/lab0S.git
  99e6ff1..d453af9 master -> master
  * [new branch]      develop -> develop
  * [new branch]      release/1.0.0 -> release/1.0.0
vvmurashko1@dk6n53 ~/laboratory $ git push --tags
Everything up-to-date
```

Отправим данные на github:

```
vvmurashko1@dk6n53 ~/laboratory $
```

Создадим релиз на github:



4 Выводы

Я изучила идеологию и применение средств контроля версий, научилась генерировать ключи и подключать репозиторий.