

Отчёта по лабораторной работе №2

Управление версиями

Башкирова Я.Д

Содержание

1	Цель работы	5
2	Задачи	6
3	Ход работы	7
3.1	Настройка git:	7
3.2	Подключение репозитория к github:	8
3.3	Первичная конфигурация:	9
3.4	Конфигурация git-flow:	10
4	Выводы	12

List of Tables

List of Figures

1 Цель работы

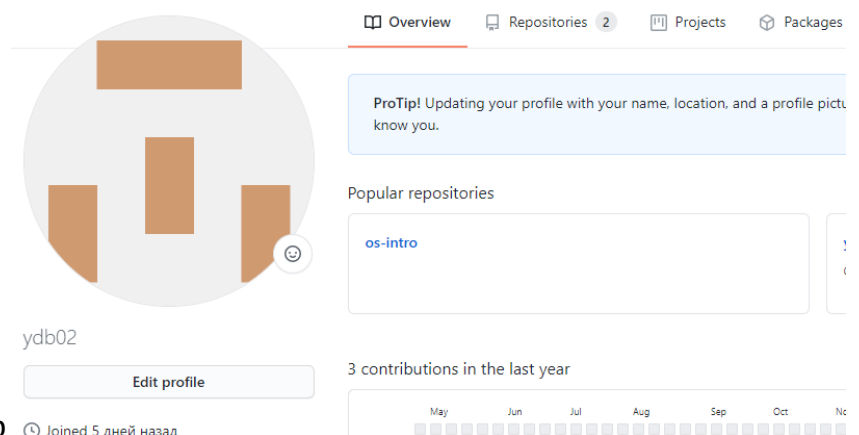
изучить идеологию и применение средств контроля версий.

2 Задачи

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

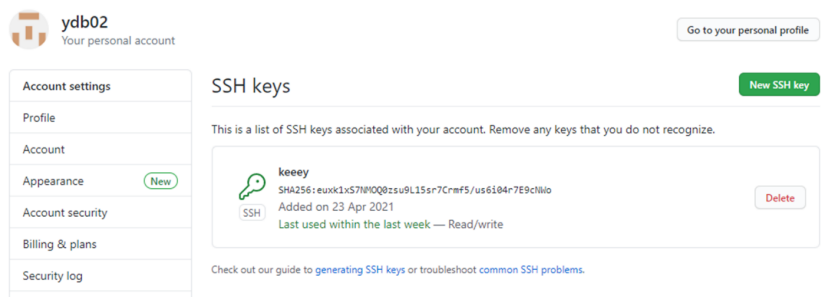
3 Ход работы

3.1 Настройка git:



Создаем учетную запись на github

Настраиваем систему контроля версий git: для этого генерируем ключ с помощью команды `ssh-keygen -C «Имя Фамилия work@mail»`, загружаем сгенеренный 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-



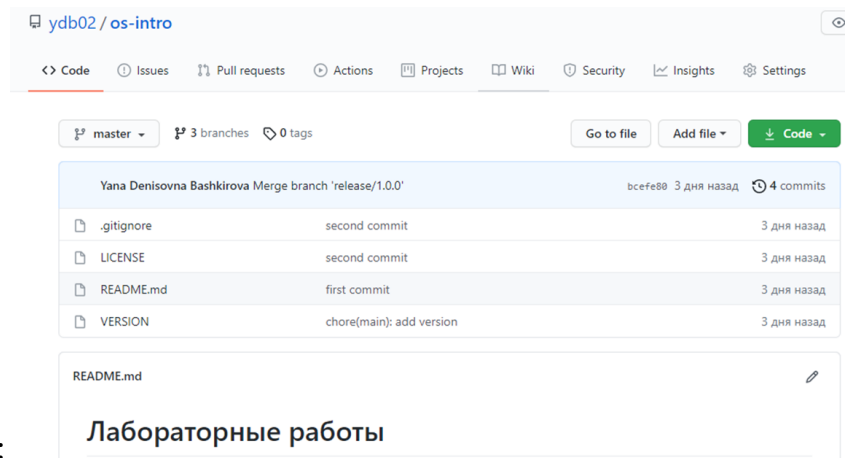
шееся на сайте поле.

```

ydbashkirova@dk4n70 ~ $ cd .ssh
ydbashkirova@dk4n70 ~/.ssh $ ssh-keygen -C "Яна Башкирова <1032200525>"
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/y/d/ydbashkirova/.ssh/id_rsa):
/afs/.dk.sci.pfu.edu.ru/home/y/d/ydbashkirova/.ssh/id_rsa already exists.
Overwrite (y/n)? Y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/y/d/ydbashkirova/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/y/d/ydbashkirova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:euxk1xS7NM0Q0zsu9L15sr7Crmf5/us6i04r7E9cNWо Яна Башкирова <1032200525>
The key's randomart image is:
+----[RSA 3072]-----+
|
|   o
|  + o o
|  + +o
| S . XE
| o ..o*
| . * o=..
| = +o.O..o
| o=X.B&Oo|
+----[SHA256]-----+
ydbashkirova@dk4n70 ~/.ssh $ ls
id_rsa id_rsa.pub known_hosts
ydbashkirova@dk4n70 ~/.ssh $ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC+pDAiKXF0lv0rlnaxknyNR/hvkb20UskBIRE+GaEf
jdRgOg7umFSNPFqBjKJTxApibFgq41eIA4Rw3yNoWCWE/D6qv4Y8Cvt4FTPN50qXt7ha7EK55b8z03N
sZrMXWBVn0oMOMxfIEERTLzCIA+a4C/KswurXALpv9uZZI9Ke08vbfj9BEFhWAsrkPXkh3SxLeBs9ETn
/RP+0FxsGvIB10KN41AUeXyJeP6voSEQbC6byTc8cd+4evwCDG/uCRY5AEFGsoYDUQwnlav07wjdhAa
NupMXM5Fg1wHsZEYQssRJWoiHMuJ/cTCG3zui5gmJ+vz5d+XIItvZxTIMKhpoygIhHBeG8/fk+9J5N5
3rf0bipYrI/BWsgUP8DY2Wqd1NKTW4TGJynurcGHJWkRix/+L+9+NCg92djiLX0sz7uYxIIZEGmje2ld
NbtB7fchFgShKfTy69Muh5zD36ErWY7417xSZkNVcM9mqJpcq1H0a8cYU2dqu9Ynj5Jjh8k= Яна Баш
кирова <1032200525>

```

3.2 Подключение репозитория к github:



Создаём репозиторий на github:

Переходим в рабочий каталог laboratory:

Инициализируем системы git:

Создаём заготовку для файла README.md:

Делаем первый коммит и выкладываем на github:


```

перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 257 bytes | 257.00 KiB/s, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:ydb02/os-intro.git
* [new branch] master -> master
Ветка «master» отслеживает внешнюю ветку «master» из «origin».

```

3.3 Первичная конфигурация:

```

ydbashkirova@dk4n70 ~/laboratory $ wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
--2021-04-23 14:22:03-- https://creativecommons.org/licenses/by/4.0/legalcode.txt
Распознаётся creativecommons.org (creativecommons.org)... 104.20.150.16, 172.67.34.140, 104.20.151.16, ...
Подключение к creativecommons.org (creativecommons.org)[104.20.150.16]:443... соединено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: нет данных [text/plain]
Сохранение в: «LICENSE»

LICENSE [ <=> ] 18,22K --.-KB/s за 0,001s
2021-04-23 14:22:03 (15,6 MB/s) - «LICENSE» сохранён [18657]

```

Добавим файл лицензии:

Добавим шаблон игнорируемых файлов и посмотрим список имеющихся

```

ydbashkirova@dk4n70 ~/laboratory $ curl -L -s https://www.gitignore.io/api/list
1c,1c-bitrix,a-frame,actionsript,ada
adobe,advancedinstaller,adventuregamestudio,agda,al
alteraquartusii,altium,amplify,android,androidstudio
angular,anjuta,ansible,apachecordova,apachehadoop
appbuilder,appcleratore titanium,appcode,appcode+iml

```

шаблонов:

```

ydbashkirova@dk4n70 ~/laboratory $ curl -L -s https://www.gitignore.io/api/c >> .gitignore

```

Затем скачаем шаблон:

```

ydbashkirova@dk4n70 ~/laboratory $ git add .

```

Добавим новые файлы:

```

ydbashkirova@dk4n70 ~/laboratory $ git commit -am "second commit"
[master 8629a9f] second commit
2 files changed, 455 insertions(+)
create mode 100644 .gitignore
create mode 100644 LICENSE

```

Выполним коммит:

```

ydbashkirova@dk4n70 ~/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), 6.44 KiB | 6.44 MiB/s, готово.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:ydb02/os-intro.git
891c924..8629a9f master -> master

```

Отправим на github:

3.4 Конфигурация git-flow:

```
ydbashkirova@dk4n70 ~/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/y/d/ydbashkirova/labo
ratory/.git/hooks]
```

Инициализируем git-flow:

Проверим, что мы на ветке develop:

```
ydbashkirova@dk4n70 ~/laboratory $ git branch
ydbashkirova@dk4n70 ~/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:
```

Создадим релиз с версией 1.0.0:

Запишем версию:

```
ydbashkirova@dk4n70 ~/laboratory $ echo "1.0.0" >> VERSION
ydbashkirova@dk4n70 ~/laboratory $ git add .
ydbashkirova@dk4n70 ~/laboratory $ git commit -am 'chore(main): add version'
[release/1.0.0 a4fa877] chore(main): add version
1 file changed, 1 insertion(+)
create mode 100644 VERSION
```

Добавим в индекс:

```
ydbashkirova@dk4n70 ~/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:

```
ydbashkirova@dk4n70 ~/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), 390 bytes | 390.00 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:ydb02/os-intro.git
 8629a9f..bcefe80 master -> master
 * [new branch]      develop -> develop
 * [new branch]      release/1.0.0 -> release/1.0.0
ydbashkirova@dk4n70 ~/laboratory $ git --tags
unknown option: --tags
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
```

Отправим данные на github:


os-intro


 ydb02 выпустил это через 18 секунд

1.0.0

Слияние ветви 'release/1.0.0'

▼ Активы 2

 [Исходный код \(zip\)](#)

 [Исходный код \(tar.gz\)](#)

Создадим релиз на github:

4 Выводы

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