

# **Лабораторная работа №3**

**Дисциплина: Операционные системы**

Егорова Александра

# Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Выводы	13

# List of Figures

3.1	Рис.1	6
3.2	Рис.2	6
3.3	Рис.3	7
3.4	Рис.4	7
3.5	Рис.5	7
3.6	Рис.6	7
3.7	Рис.7	8
3.8	Рис.8	8
3.9	Рис.9	8
3.10	Рис.10	8
3.11	Рис.11	8
3.12	Рис.12	9
3.13	Рис.13	9
3.14	Рис.14	9
3.15	Рис.15	10
3.16	Рис.16	10
3.17	Рис.17	10
3.18	Рис.18	10
3.19	Рис.19	11
3.20	Рис.20	11
3.21	Рис.21	11
3.22	Рис.22	11
3.23	Рис.23	12
3.24	Рис.24	12

# **1 Цель работы**

Изучить идеологию и применение средств контроля версий.

## 2 Задание

Сделайте отчёт по предыдущей лабораторной работе в формате Markdown.

### 3 Выполнение лабораторной работы

Создаем учётную запись на <https://github.com>. (рис. -fig. 3.1)

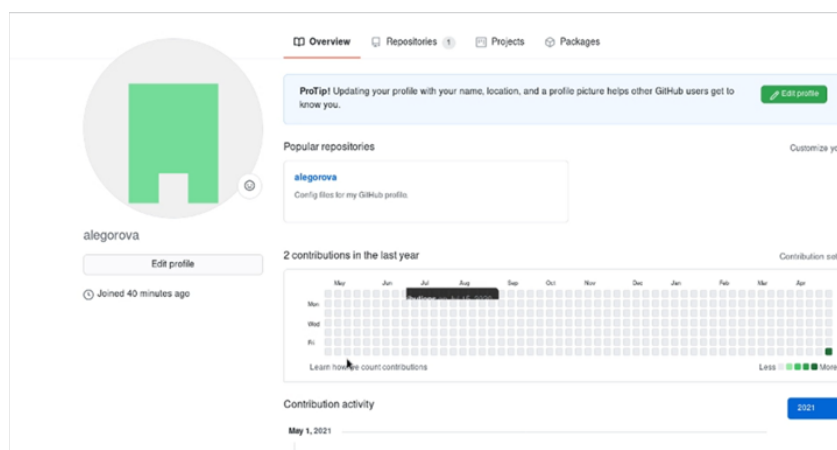


Figure 3.1: Рис.1

Настраиваем систему контроля версий git. Синхронизируем учётную запись github с компьютером: `git config --global user.name "Имя Фамилия"` `git config --global user.email "work@mail"` (рис. -fig. 3.2)



Figure 3.2: Рис.2

Создаем новый ключ на github и привязываю его к компьютеру через консоль. (рис. -fig. 3.3) (рис. -fig. 3.4) (рис. -fig. 3.5) (рис. -fig. 3.6)

```
[aegorova@aegorova ~]$ ssh-keygen -C "aegorova <e.alexandra@gmail.com>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aegorova/.ssh/id_rsa):
/home/aegorova/.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 /home/aegorova/.ssh/id_rsa.
Your public key has been saved in /home/aegorova/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Gyd6p1G0/GM0tvYSIZI/rpfcnSZB1FJp5yr5VGvDmq8 aegorova <e.alexandra@gmail.com>
The key's randomart image is:
+---[RSA 2048]-----+
|
|..+
|.+++.
|.+++.
|.o+o.o.S.
|o+B.,.=
|=B.=B.=
|B.B.o0o
|E0..o=+.
+---[SHA256]-----+
[aegorova@aegorova ~]$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDQY/bfZ0pPMZKJ2hBKuoJn5oc15sb9wUso5S5CqUB5/SzqDMIFe93HDbuMD0I2cEJ4
n1TXJ8OLmwtJGfBIEK7o1F/URNaQhFBpypNGp/1TgQj1f171dDFwTBBPqQV0V0enXE0FcpV1tSc06rXPxanp20yQ57Q01v5Vi4d1UW6
mthh1vAHhncCKa/lALXyew7gzaiYK4U107/Nc8ECnaTLgd9iv/B5fvSgrYG3QRoYYLMVd7en11X6Kq+exRr0rJTAngMZoznWSPft0ce
jx9zkWtWsgz8YzAd/tZ4dJ45DvX1AozRDb2EzYz1GTTdjqz2MkssPRnz5+JlN8TjMG7bSh aegorova <e.alexandra@gmail.com>
```

Figure 3.3: Рис.3

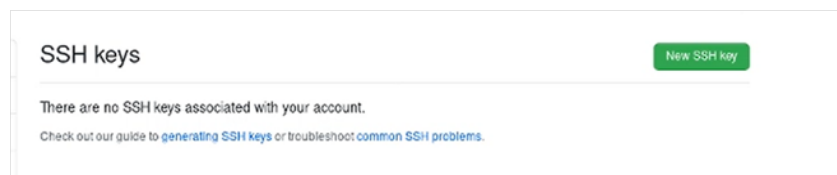


Figure 3.4: Рис.4

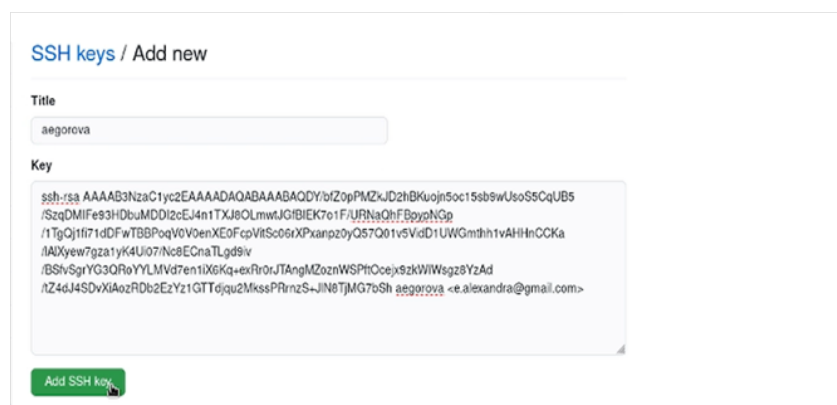


Figure 3.5: Рис.5



Figure 3.6: Рис.6

Подключение репозитория к github. Заходим в «repositories» и создаем новый репозиторий (имя «labor2»). Копируем в консоль ссылку на репозиторий. (рис. -fig. 3.7) (рис. -fig. 3.8)

Figure 3.7: Рис.7

```
[aegorova@aegorova ~]$ git clone https://github.com/aegorova/labor2.git
Cloning into 'labor2'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

Figure 3.8: Рис.8

Работаем с каталогом и папками через консоль. Создаем файлы. (рис. -fig. 3.9) (рис. -fig. 3.10)

```
[aegorova@aegorova ~]$ cd labor2
[aegorova@aegorova labor2]$ mkdir 2020-2021
[aegorova@aegorova labor2]$ cd 2020-2021
[aegorova@aegorova 2020-2021]$ mkdir 05
[aegorova@aegorova 2020-2021]$ cd 05
[aegorova@aegorova 05]$ mkdir laboratory
[aegorova@aegorova 05]$ cd laboratory
[aegorova@aegorova laboratory]$
```

Figure 3.9: Рис.9

```
[aegorova@aegorova laboratory]$ touch h.txt
```

Figure 3.10: Рис.10

Добавляем первый коммит и выкладываем на github. Чтобы правильно разместить первый коммит, необходимо добавить команду `git add .`, далее с помощью команды `git commit -am "first commit"` выкладываем коммит. Сохраняем первый коммит (`git push`). (рис. -fig. 3.11) (рис. -fig. 3.12)

```
[aegorova@aegorova laboratory]$ git add .
[aegorova@aegorova laboratory]$ git commit -am "first commit"
[main 98f3c66] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 2020-2021/05/laboratory/h.txt
```

Figure 3.11: Рис.11



```
[aegorova@aegorova laboratory]$ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

    git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

    git config --global push.default simple

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Username for 'https://github.com': aegorova
Password for 'https://aegorova@github.com':
Counting objects: 7, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 406 bytes | 0 bytes/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/aegorova/labor2.git
5332bfc..98f3c66  main -> main
```

Figure 3.12: Рис.12

Первичная конфигурация. Добавим файл лицензии. (рис. -fig. 3.13)

```
[aegorova@aegorova laboratory]$ wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
--2021-05-01 23:21:33-- https://creativecommons.org/licenses/by/4.0/legalcode.txt
Распознаётся creativecommons.org (creativecommons.org)... 172.67.34.140, 104.20.150.16, 104.20.151.16, ...
Подключение к creativecommons.org (creativecommons.org)[172.67.34.140]:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: нет данных [text/plain]
Сохранение в: «LICENSE»

[ <=> ] 18 657 --.-K/s 3a 0s

2021-05-01 23:21:35 (71,3 MB/s) - «LICENSE» сохранён [18657]
```

Figure 3.13: Рис.13

Добавим шаблон игнорируемых файлов. Получим список имеющихся шаблонов (на скрине представлены не все шаблоны). (рис. -fig. 3.14)

```
[aegorova@aegorova 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,appcelerator titanium,appcode,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
calabash,carthage,certificates,ceylon,cfwheels
chefcookbook,chocolatey,clean,clion,clion+all
clion+iml,closure,cloud9,cmake,cocoapods
cocos2dx,cocoscreator,code,code-java,codeblocks
```

Figure 3.14: Рис.14

Затем скачаем шаблон, например, для C. Также добавим новые файлы и выполним коммит. Отправляем на github (git push). (рис. -fig. 3.15) (рис. -fig. 3.16)

```
[aegorova@aegorova laboratory]$ curl -L -s https://www.gitignore.io/api/c >> .gitignore
[aegorova@aegorova laboratory]$ git add .
[aegorova@aegorova laboratory]$ git commit -am "first commit"
[main 46ed2b4] first commit
2 files changed, 455 insertions(+)
create mode 100644 2020-2021/05/laboratory/.gitignore
create mode 100644 2020-2021/05/laboratory/LICENSE
```

Figure 3.15: Рис.15

```
[aegorova@aegorova laboratory]$ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

    git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

    git config --global push.default simple

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Username for 'https://github.com': aegorova
Password for 'https://aegorova@github.com':
Counting objects: 11, done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 6.59 KiB | 0 bytes/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To https://github.com/aegorova/labor2.git
 98f3c66..46ed2b4 main -> main
```

Figure 3.16: Рис.16

Конфигурация git-flow. Инициализируем git-flow с помощью команды `git flow init -f`. Префикс для ярлыков установим в `v`. (рис. -fig. 3.17)

```
[aegorova@aegorova laboratory]$ git flow init -f
Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/] v
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? [] v
```

Figure 3.17: Рис.17

Проверяем, что находимся на ветке `develop` (`git branch`). (рис. -fig. 3.18)

```
[aegorova@aegorova laboratory]$ git branch
* develop
main
```

Figure 3.18: Рис.18

Создадим релиз с версией 1.0.0. Запишем версию и добавим в индекс. (рис. -fig. 3.19) (рис. -fig. 3.20)

```
[aegorova@aegorova laboratory]$ 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'
```

Figure 3.19: Рис.19

```
[aegorova@aegorova laboratory]$ echo "1.0.0">> VERSION
[aegorova@aegorova laboratory]$ git add .
[aegorova@aegorova laboratory]$ git commit -am 'chore(main): add version'
[release/1.0.0 6fbbf60] chore(main): add version
1 file changed, 2 insertions(+)
create mode 100644 2020-2021/05/laboratory/VERSION
```

Figure 3.20: Рис.20

Зальём релизную ветку в основную ветку. Отправим данные на github. (рис. -fig. 3.21) (рис. -fig. 3.22)

```
[aegorova@aegorova laboratory]$ git flow release finish 1.0.0
Switched to branch 'main'
Merge made by the 'recursive' strategy.
2020-2021/05/laboratory/VERSION | 2 ++
1 file changed, 2 insertions(+)
create mode 100644 2020-2021/05/laboratory/VERSION
fatal: no tag message?
tagging failed. Please run finish again to retry.
```

Figure 3.21: Рис.21

```
[aegorova@aegorova laboratory]$ git push --all
Username for 'https://github.com': aegorova
Password for 'https://aegorova@github.com':
Counting objects: 11, done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (7/7), 623 bytes | 0 bytes/s, done.
Total 7 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/aegorova/labor2.git
 46ed2b4..787aa68 main -> main
 * [new branch]      develop -> develop
 * [new branch]      release/1.0.0 -> release/1.0.0
[aegorova@aegorova laboratory]$ git push --tags
Username for 'https://github.com': aegorova
Password for 'https://aegorova@github.com':
Everything up-to-date
```

Figure 3.22: Рис.22

Создаем релиз на github. Заходим в «Releases», нажимаем «Создать новый релиз». Заходим в теги и заполняем все поля. После создания тега, автоматически сформируется релиз. (рис. -fig. 3.23) (рис. -fig. 3.24)

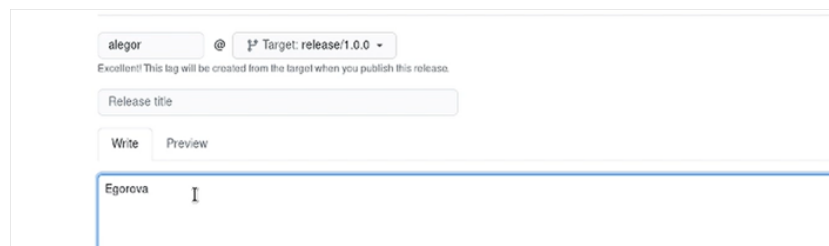


Figure 3.23: Рис.23

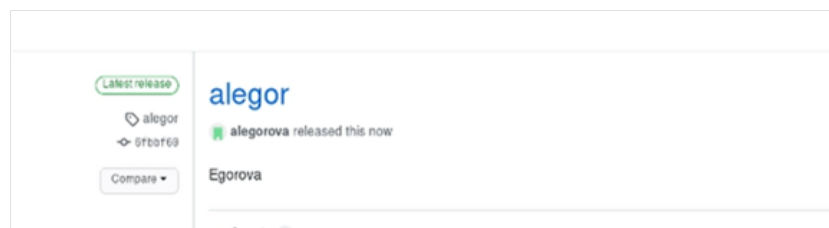


Figure 3.24: Рис.24

## **4 Выводы**

В ходе выполнения лабораторной работы я изучила идеологию и применение средств контроля версий.