

Отчёт по лабораторной работе №3

Дисциплина: Операционные системы

Ездаков Егор Андреевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	15

Список таблиц

Список иллюстраций

3.1	Создание учётной записи	7
3.2	Создание нового ssh ключа	8
3.3	Добавление ssh ключа на github	8
3.4	Создание каталога	9
3.5	Создание репозитория	9
3.6	Инициализация систем git	9
3.7	Создание заготовки для файла README.md	9
3.8	Первый коммит	10
3.9	Первый коммит	10
3.10	Файл лицензии	10
3.11	Список шаблонов игнорируемых файлов	11
3.12	Загрузка шаблона для C	11
3.13	Добавление файлов на GitHub	11
3.14	Добавление файлов на GitHub	11
3.15	Инициализация git-flow	12
3.16	Установка префикса	12
3.17	Создание релиза	12
3.18	Создание релиза	12
3.19	Запись версии	12
3.20	Добавление в индекс	13
3.21	Залив релизной ветки	13
3.22	Отправка данных на GitHub	13
3.23	Создание релиза	14

1 Цель работы

Изучить идеологию и применение средств контроля версий

2 Задание

Сделайте отчёт по предыдущей лабораторной работе в формате Markdown. В качестве отчёта просьба предоставить отчёты в 3 форматах: pdf, docx и md

3 Выполнение лабораторной работы

1. Создал учетную запись на GitHub (рис. -fig. 3.1).

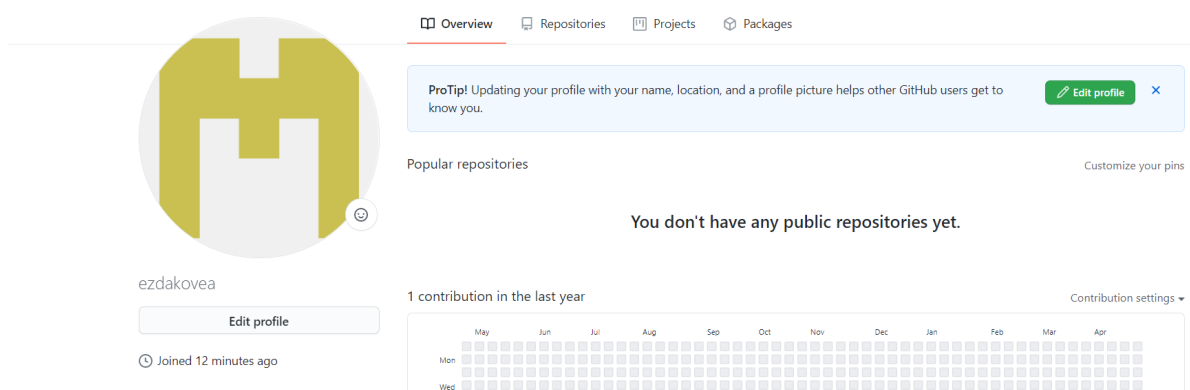


Рис. 3.1: Создание учётной записи

2. Настроил систему контроля версий git (рис. -fig. 3.2, рис. -fig. 3.3)

Сгенерировал ssh-ключ:

```

Файл  Правка  Вид  Поиск  Терминал  Справка
The key's randomart image is:
+----[RSA 2048]-----+
|          ..          |
|         ....         |
|        o  ++         |
|       o  +.o+         |
|      o   .S*+         |
|     .  .o&+          |
|    +  .X=o+          |
|   + B=B..+ .         |
|  o+E++ .+           |
+-----[SHA256]-----+
[eaездakov@eaездakov ~]$ cat/home/eaездakov/.ssh/id_rsa.ppub
bash: cat/home/eaездakov/.ssh/id_rsa.ppub: Нет такого файла или каталога
[eaездakov@eaездakov ~]$ cat/home/eaездakov/.ssh/id_rsa.pub
bash: cat/home/eaездakov/.ssh/id_rsa.pub: Нет такого файла или каталога
[eaездakov@eaездakov ~]$ cat /home/eaездakov/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQ CzDWIqFk01yh8PYztVzE8vcfEhI0fvAaWQiHpuB8B0
qjltzpVhqMKX0ogS70kW0AZhCm/thjoPRsPAd8bZ/mM9UNFAETlHLu+GK+AtbuMqyS+nbFoacfYJubQn
rx8bX/CMFNeN0afftPFqGLGEEFGZ32HHAzzZmQ9anZpwdolBhnNPdouBLp6/9LaF3TNb5ykXFEUCXKoE
eYAJPPeDjCQqKc0bcmgTzDQm1VwSEMPmR+SGzs6/Pj1KlsUf87NyIxTbZgvxKD70+C3C0nMzY+4vnBUE
oUTNBKZKmSAB38+pLvmx23mt3zIXZqaerfJvoWvqSwgk2ZKi/ZnFESVdK1sL  eздakovea <ездakov0
2@mail.ru>
[eaездakov@eaездakov ~]$ █

```


Рис. 3.2: Создание нового ssh ключа

Добавил его в настройках учетной записи на GitHub

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



ssh-rsa AAAAB3NzaC1yc2EAAA

SHA256:p89IJYR619to3j0njFBEOqI+83tRPhDvmh5ywU/qm7M

Added on May 1, 2021

Never used — Read/write

SSH

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

Рис. 3.3: Добавление ssh ключа на github

3. Создал структуру каталога лабораторных работ с помощью mkdir (рис. - fig. 3.4)


```
[eaezdakov@eaezdakov ~]$ mkdir -p ~/work/2020-2021/Operating systems/laboratory  
[eaezdakov@eaezdakov ~]$
```


Рис. 3.4: Создание каталога

4. Создал репозиторий на GitHub. Назвал его os-intro (рис. -fig. 3.5):

Create a new repository

A repository contains all project files, including the revision history. Already have a [Import a repository](#).

Owner * Repository name *

 ezdakovea ▾ / 

Great repository names are The repository **os-intro** already exists on this account. about p:

Description (optional)


☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

Рис. 3.5: Создание репозитория

5. Перешёл в каталог laboratory: cd
6. Инициализировал системы git (рис. -fig. 3.6):

```
[eaezdakov@eaezdakov laboratory]$ git init
```

Рис. 3.6: Инициализация систем git

7. Создал заготовку для файла README.md (рис. -fig. 3.7):

```
[eaezdakov@eaezdakov laboratory]$ echo "# Лабораторные работы" >> README.md  
[eaezdakov@eaezdakov laboratory]$ git add README.md
```

Рис. 3.7: Создание заготовки для файла README.md

8. Создал первый коммит и отправил его на GitHub (рис. -fig. 3.8, рис. -fig. 3.9):

```
[eaezdakov@eaezdakov laboratory]$ echo "# Лабораторные работы" >> README.md
[eaezdakov@eaezdakov laboratory]$ git add README.md
[eaezdakov@eaezdakov laboratory]$ git commit -m "first commit"
[master (root-commit) 80b058e] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

Рис. 3.8: Первый коммит

```
[eaezdakov@eaezdakov laboratory]$ git remote add origin git@github.com:ezdakovea/os-intro.git
[eaezdakov@eaezdakov 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:nThbg6kXUpJWGL7E1IGOCspRomTxdCARLviKw6E5SY8.
RSA key fingerprint is MD5:16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,140.82.121.3' (RSA) to the list of known hosts.
Counting objects: 3, done.
Writing objects: 100% (3/3), 244 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:ezdakovea/os-intro.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
[eaezdakov@eaezdakov laboratory]$ █
```

Рис. 3.9: Первый коммит

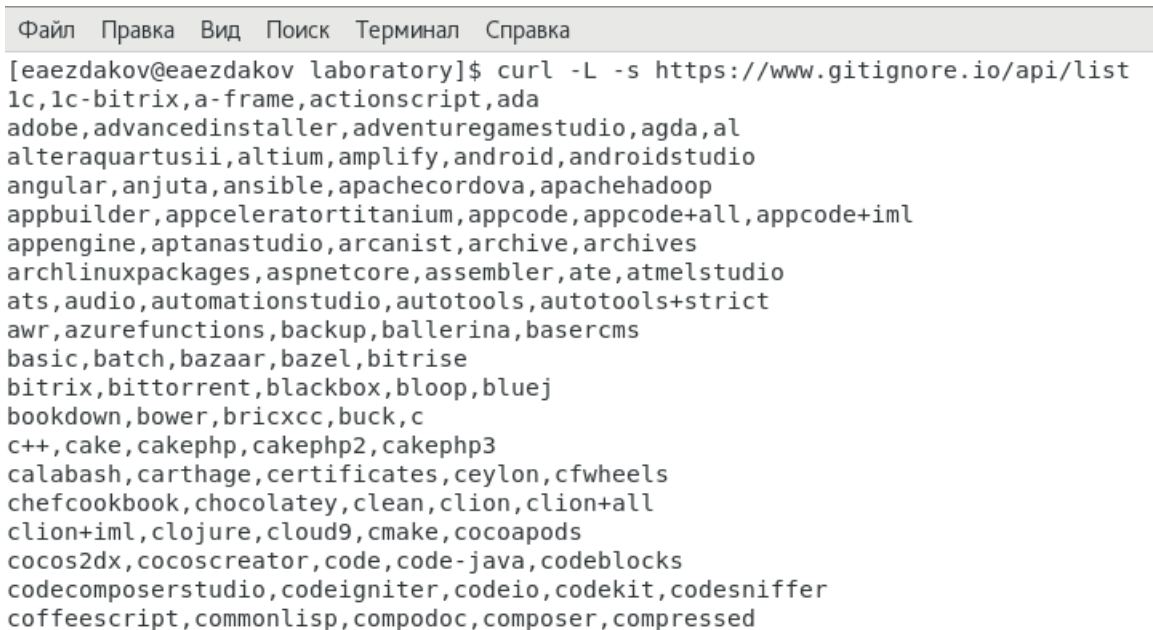
9. Добавил файл лицензии (рис. -fig. 3.10):

```
[eaezdakov@eaezdakov laboratory]$ wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
--2021-05-01 12:37:48-- https://creativecommons.org/licenses/by/4.0/legalcode.txt
Распознаётся creativecommons.org (creativecommons.org)... 104.20.151.16, 172.67.34.140, 104.20.150.16, ...
Подключение к creativecommons.org (creativecommons.org)|104.20.151.16|:443... со
единение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: нет данных [text/plain]
Сохранение в: «LICENSE»

[ <=> ] 18 657 --.-K/s за 0,003s
2021-05-01 12:37:48 (6,28 MB/s) - «LICENSE» сохранён [18657]
[eaezdakov@eaezdakov laboratory]$ █
```

Рис. 3.10: Файл лицензии

10. Просмотрел список имеющихся шаблонов игнорируемых файлов (рис. - fig. 3.11):



```
Файл  Правка  Вид  Поиск  Терминал  Справка
[eaездakov@eaездakov 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
calabash,carthage,certificates,ceylon,cfwheels
chefcookbook,chocolatey,clean,clion,clion+all
clion+iml,clojure,cloud9,cmake,cocoapods
cocos2dx,cocoscreator,code,code-java,codeblocks
codecomposerstudio,codeigniter,codeio,codekit,codesniffer
coffeescript,commonlisp,compodoc,composer,compressed
```

Рис. 3.11: Список шаблонов игнорируемых файлов

11. Загрузил шаблон для C (рис. -fig. 3.12):



```
[eaездakov@eaездakov laboratory]$ curl -L -s https://www.gitignore.io/api/c >> .gitignore
[eaездakov@eaездakov laboratory]$
```

Рис. 3.12: Загрузка шаблона для C

12. Добавил новые файлы, отправил на GitHub (рис. -fig. 3.13, рис. -fig. 3.14):

```
[eaездakov@eaездakov laboratory]$ git add .
[eaездakov@eaездakov laboratory]$ git commit -am 'Создали шаблон для C'
[master 7684635] Создали шаблон для C
2 files changed, 514 insertions(+)
create mode 100644 .gitignore
create mode 100644 LICENSE
```

Рис. 3.13: Добавление файлов на GitHub

```
[eaездakov@eaездakov laboratory]$ git push
```

Рис. 3.14: Добавление файлов на GitHub

13. Инициализировал git-flow (рис. -fig. 3.15):

```
[eaezdakov@eaezdakov laboratory]$ git flow init
```

Рис. 3.15: Инициализация git-flow

14. Префикс для ярлыков установил в v (рис. -fig. 3.16):

```
Version tag prefix? [] v
```

Рис. 3.16: Установка префикса

15. Находясь на ветке develop, создал релиз с версией 1.0.0 (рис. -fig. 3.17, рис. -fig. 3.18):

```
[eaezdakov@eaezdakov laboratory]$ git branch
* develop
  master
```

Рис. 3.17: Создание релиза

```
[eaezdakov@eaezdakov 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'
```

Рис. 3.18: Создание релиза

16. Записал версию (рис. -fig. 3.19):

```
[eaezdakov@eaezdakov laboratory]$ echo "1.0.0" >> VERSION
[eaezdakov@eaezdakov laboratory]$ █
```

Рис. 3.19: Запись версии

17. Добавил в индекс (рис. -fig. 3.20):

```
[eaezdakov@eaezdakov laboratory]$ git add .
[eaezdakov@eaezdakov laboratory]$ git commit -am 'chore(main): add version'
[release/1.0.0 b6644e1] chore(main): add version
1 file changed, 1 insertion(+)
create mode 100644 VERSION
```

Рис. 3.20: Добавление в индекс

18. Залил релизную ветку в основную ветку (рис. -fig. 3.21):

```
[eaezdakov@eaezdakov laboratory]$ git flow release finish 1.0.0
```

Рис. 3.21: Залив релизной ветки

19. Отправил данные на GitHub (рис. -fig. 3.22):

```
[eaezdakov@eaezdakov laboratory]$ git push --all
Counting objects: 5, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 386 bytes | 0 bytes/s, done.
Total 4 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To git@github.com:ezdakovea/os-intro.git
 7684635..68fc0a8 master -> master
 * [new branch]      develop -> develop
 * [new branch]      release/1.0.0 -> release/1.0.0
[eaezdakov@eaezdakov laboratory]$ git push --tags
Everything up-to-date
[eaezdakov@eaezdakov laboratory]$
```

Рис. 3.22: Отправка данных на GitHub

20. Создаем релиз на github. Для этого заходим в «Releases», нажимаем «Создать новый релиз». Заходим в теги и заполняем все поля (создаём теги для версии 1.0.0). После создания тега, автоматически сформируется релиз (рис. -fig. 3.23).

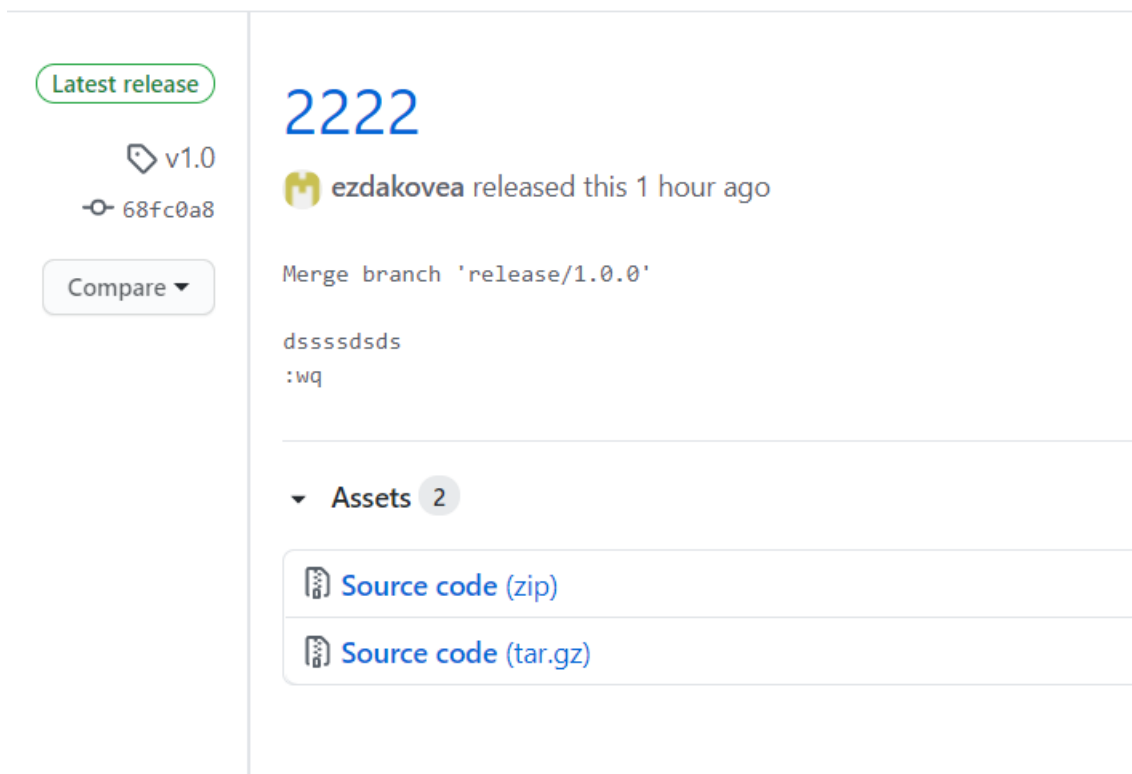


Рис. 3.23: Создание релиза

4 Выводы

Я изучил идеологию и применение контроля версий.