

Лабораторная работа №3

Дисциплина: Операционные системы

Коновалова Татьяна Борисовна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	16

List of Tables

List of Figures

3.1	Создание учётной записи	7
3.2	Создание нового ключа	8
3.3	Создание репозитория	9
3.4	Создание файлов	10
3.5	Добавление первого коммита	10
3.6	Сохранение первого коммита	10
3.7	Добавление файла лицензии	11
3.8	Добавление шаблона игнорируемых файлов	11
3.9	Скачивание шаблона на С	12
3.10	Отправляем на GitHub созданные шаблоны	12
3.11	Инициализация git-flow	12
3.12	Создание релиза	13
3.13	Записывание версии и добавление в индекс	13
3.14	Заливаем ветку	14
3.15	Отправляем данные на GitHub	14

1 Цель работы

Цель данной лабораторной работы — Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

2 Задание

1.Сделайте отчёт по предыдущей лабораторной работе в формате Markdown. 2.В качестве отчёта предоставить отчёты в 3 форматах: pdf,docx и md

3 Выполнение лабораторной работы

1). Создаем учетную запись на <https://github.com>. (иллюстрация на рис. 1)

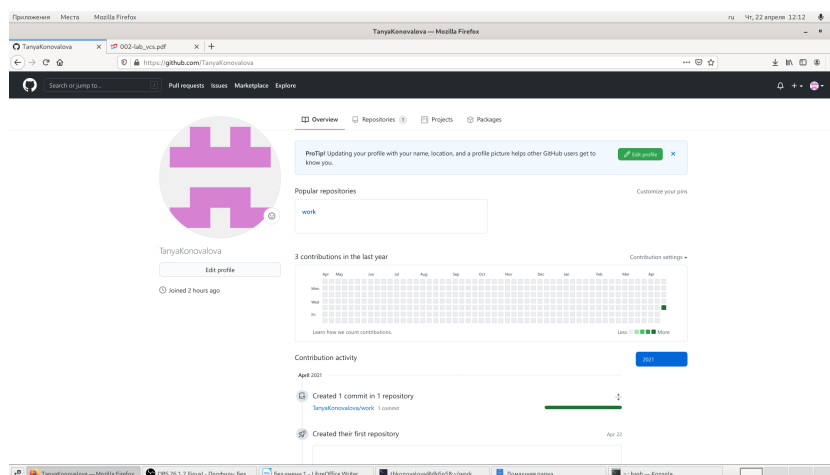


Figure 3.1: Создание учётной записи

2). 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"` После этого создаём новый ключ на github (команда `ssh-keygen -C "TanyaKonovalova tanya.konovalova.02@mail"`) и привязываем его к компьютеру через консоль (алгоритм действий представлен на рис.2).

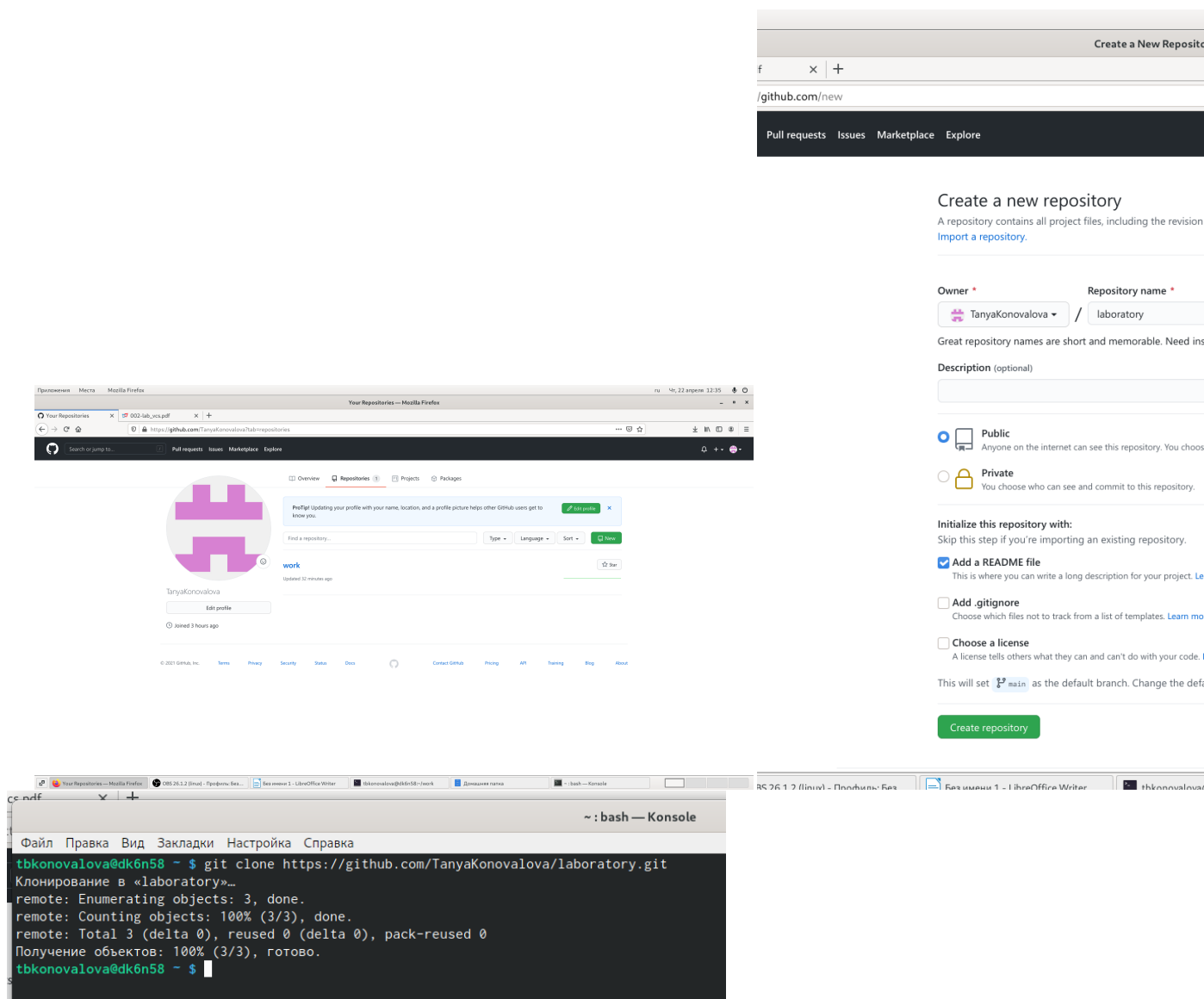
```

Приложения Места Терминал
tbkono
tbkonovalova@dk6n58:~
tbkonovalova@dk6n58 ~$ git user.TanyaKonovalova
git: «user.TanyaKonovalova» не является командой git. Смотрите «git --help».
tbkonovalova@dk6n58 ~$ git config --global user.Tanya.Konovalova
tbkonovalova@dk6n58 ~$ git config --global user.tanya.konovalova.02@mail.ru
tbkonovalova@dk6n58 ~$ git config --global quotepath
error: key does not contain a section: quotepath
tbkonovalova@dk6n58 ~$ sh-keygen -C"Tanya.Konovalova <tanya.konovalova.02@mail>"
bash: sh-keygen: команда не найдена
tbkonovalova@dk6n58 ~$ ssh-keygen -C"Tanya.Konovalova <tanya.konovalova.02@mail>"
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/.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/t/b/tbkonovalova/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/t/b/tbkonovalova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:9efXeuPPh1huuX54pa/TrfuBYr7qYTbHtY++XmSc Tanya.Konovalova <tanya.konovalova.02@mail>
The key's randomart image is:
+---[RSA 3072]-----+
|
|      .
|     . . o
|    S . . + .
|      ++=
|     oB=%
|    .o+E*B
|     .oO^*%
+---[SHA256]-----+
tbkonovalova@dk6n58 ~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
tbkonovalova@dk6n58 ~$

```

Figure 3.2: Создание нового ключа

3). Следующим шагом будет создание и подключение репозитория к github. В gethup заходим в «repository» и создаём новый репозиторий (имя «laboratory», а заголовок для файла README) (иллюстрация представлена на рис. 3.1, 3.2). Копируем в консоль ссылку на репозиторий (для дальнейшей работы с файлами) (алгоритм действий см. на рис.3.3):



4). В лабораторной работе описан алгоритм создания структуры каталога через консоль. Но легче будет создать репозиторий в gethur и после этого работать с каталогом и папками через консоль (перед этим необходимо скопировать ссылку на репозиторий в консоль, в формате https или ssh). Перед тем, как создавать файлы, заходим в наш репозиторий (алгоритм действий представлен на рис. 4.1):

```
tbkonovalova@dk6n58 ~ $ cd laboratory
tbkonovalova@dk6n58 ~/laboratory $ ls
README.md
```

Figure 3.3: Создание репозитория

После этого можем уже создавать наши файлы:

```
tbkonovalova@dk6n58 ~/laboratory $ mkdir 2020-2021
tbkonovalova@dk6n58 ~/laboratory $ cd 2020-2021
tbkonovalova@dk6n58 ~/laboratory/2020-2021 $ mkdir OS
tbkonovalova@dk6n58 ~/laboratory/2020-2021 $ cd OS
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ mkdir lab02
Can't open /dev/fd0: Нет такого файла или каталога
Cannot initialize 'A:'
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ mkdir lab02
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ cd lab02
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS/lab02 $
```

Figure 3.4: Создание файлов

5). Добавляем первый коммит и выкладываем на gethup. Для того, чтобы правильно разместить первый коммит, необходимо добавить команду `git add .`, после этого с помощью команды `git commit -am "first commit"` выкладываем коммит:

```
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ touch lab02/2.txt
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ git commit -am "first commit"
На ветке main
Ваша ветка обновлена в соответствии с «origin/main».

Неотслеживаемые файлы:
(используйте «git add <файл>...», чтобы добавить в то, что будет включено в коммит)
./

ничего не добавлено в коммит, но есть неотслеживаемые файлы (используйте «git add», чтобы отслеживать их)
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ git add .
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ git commit -am "first commit"
[main 3a96461] first commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 2020-2021/OS/.txt
 create mode 100644 2020-2021/OS/lab02/2.txt
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $
```

Figure 3.5: Добавление первого коммита

6). Сохраняем первый коммит, используя команду `git push`:

```
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ git push
Username for 'https://github.com': TanyaKonovalova
Password for 'https://TanyaKonovalova@github.com':
Перечисление объектов: 7, готово.
Подсчет объектов: 100% (7/7), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (6/6), 435 bytes | 435.00 KiB/s, готово.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/TanyaKonovalova/laboratory.git
 561af50..3a96461  main -> main
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $
```

Figure 3.6: Сохранение первого коммита

7). Первичная конфигурация (Пояснения к каждому действию алгоритма представлены на рис. 71-74): а. Добавляем файл лицензии;

```
wget: не удается разрешить адрес «LICENSE»
tbkonovalova@tk6n58 ~/laboratory/2020-2021/OS $ wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
--2021-04-22 13:14:33-- https://creativecommons.org/licenses/by/4.0/legalcode.txt
Подключение к creativecommons.org (creativecommons.org)... 104.20.151.16, 104.20.158.16, 172.67.34.140, ...
Подключение к creativecommons.org (creativecommons.org)[104.20.151.16]:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 ОК
Длина: нет данных [text/plain]
Сохранение в: «LICENSE»

LICENSE [ <> ] 10,22K --.-KB/s за 0,001s

2021-04-22 13:14:34 (13,3 MB/s) - «LICENSE» сохранён [18657]
tbkonovalova@tk6n58 ~/laboratory/2020-2021/OS $
```

Figure 3.7: Добавление файла лицензии

б. Добавим шаблон игнорируемых файлов.Просмотрим список имеющихся шаблонов (на скриншоте список шаблонов представлен не в целом виде):

```
Файл Правка Вид Закладки Настройка Справка
tbkonovalova@tk6n58 ~/laboratory/2020-2021/OS $ 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,briccc,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
compressedarchive,compression,conan,concrete5,coq
cordova,craftcms,crashlytics,crbasic,crossbar
crystal,cs-cart,csharp,cuda,cvs
cypressio,d,dart,darteditor,data
database,datarecovery,dbeaver,defold,delphi
dframe,diff,direnv,diskimage,django
dm,docfx,docpress,docz,dotenv
dotfiles,dotnetcore,dotsettings,dreamweaver,dropbox
drupal,drupal7,drupal8,e2studio,eagle
easybook,eclipse,eiffelstudio,elasticbeanstalk,elisp
elixir,elm,emacs,ember,ensime
episerver,erlang,espresso,executable,exercism
expressionengine,extjs,fancy,fastlane,finale
firebase,flashbuilder,flask,flatpak,flex
flexbuilder,floobits,flutter,font,fontforge
forcedotcom,forgegradle,fortran,freepascal,fsharp
fuelphp,fusetools,games,gcov,genero4gl
geth,ggts,gis,glt,gitbook
go,godot,goodsync,gpg,gradle
grails,greenfoot,groovy,grunt,gwt
```

Figure 3.8: Добавление шаблона игнорируемых файлов

с. Скачиваем шаблон, например, для С. Также добавляем новые файлы и выполняем КОММИТ:

```

yil,yil2,zendframework,zephir,zig
zsh,zukencr8000@tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ curl -L -s https://www.gitignore.io/api/c >> .gitignore
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ git add .
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ git commit -am 'Создали шаблон для C'
[main f8247c2] Создали шаблон для C
 2 files changed, 455 insertions(+)
 create mode 100644 2020-2021/OS/.gitignore
 create mode 100644 2020-2021/OS/LICENSE
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $

```

Figure 3.9: Скачивание шаблона на C

- d. Отправим на github (для этого сохраним все созданные шаблоны и файлы, используя команду git push):

```

tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ git push
Username for 'https://github.com': TanyaKonovalova
Password for 'https://TanyaKonovalova@github.com':
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (6/6), 6.62 KiB | 6.62 MiB/s, готово.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/TanyaKonovalova/laboratory.git
 3a96461..f8247c2  main -> main
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $

```

Figure 3.10: Отправляем на GitHub созданные шаблоны

- 8). Работаем с конфигурацией git-flow (алгоритм к каждому действию представлен на рис. 81-87): а. Инициализируем git-flow, используя команду git flow init -f (префикс для ярлыков установлен в v):

```

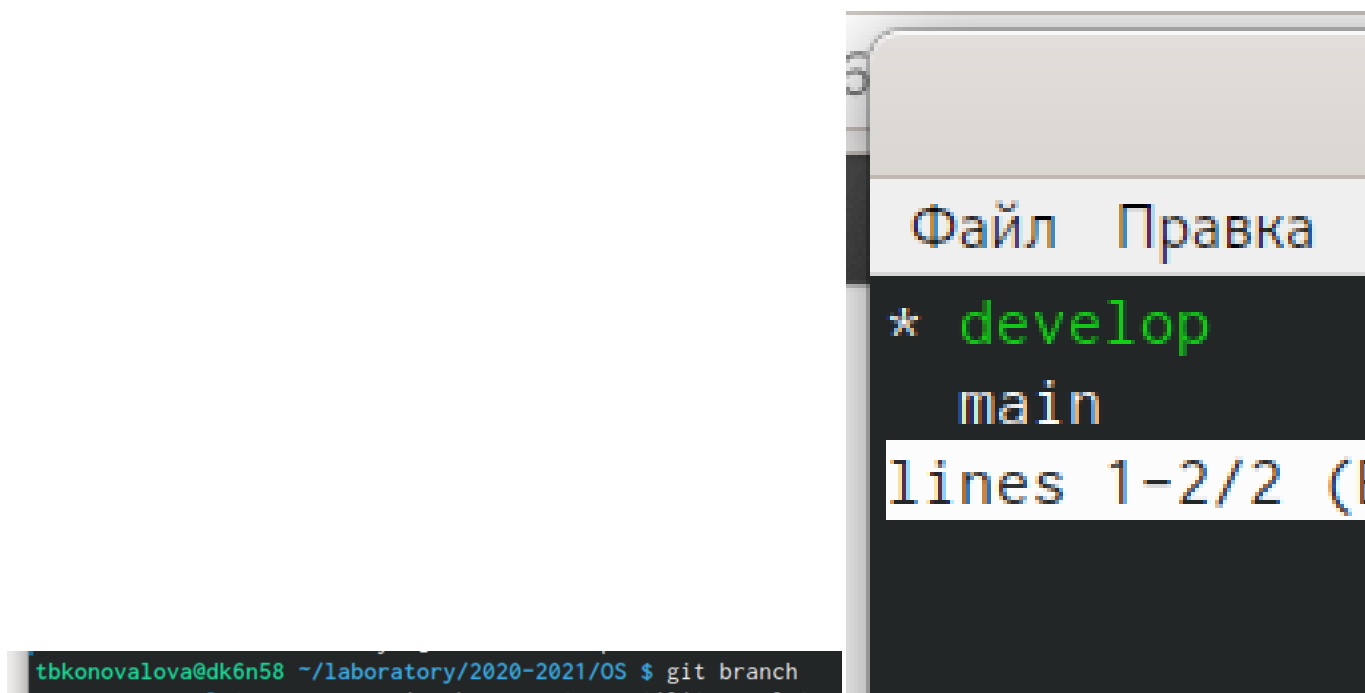
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ git flow init -f
Which branch should be used for bringing forth production releases?
  - main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/] v
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/t/b/tbkonovalova/laboratory/.git/hooks]
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $

```

Figure 3.11: Инициализация git-flow

- б. Проверяем, что мы находимся на ветке develop (используем команду git branch):



с. Создаём релиз с версией 1.0.0:

```
tbkonovalova@dk6n58 ~/.laboratory/2020-2021/OS $ 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'
```

Figure 3.12: Создание релиза

д. Запишем версию и добавим в индекс: `echo 'hello world' > hello.txt` `git add hello.txt` `git commit -am 'Новый файл'`

```
tbkonovalova@dk6n58 ~/.laboratory/2020-2021/OS $ echo "1.0.0">> VERSION
tbkonovalova@dk6n58 ~/.laboratory/2020-2021/OS $ git add .
tbkonovalova@dk6n58 ~/.laboratory/2020-2021/OS $ git commit -am 'chore(main): add version'
[release/1.0.0 5b322e4] chore(main): add version
1 file changed, 1 insertion(+)
create mode 100644 2020-2021/OS/VERSION
tbkonovalova@dk6n58 ~/.laboratory/2020-2021/OS $
```

Figure 3.13: Записывание версии и добавление в индекс

е. Заливаем релизную ветку в основную ветку (используем команду `git flow release finish 1.0.0`):

```

create mode 100644 2020-2021/OS/VERSION
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ git flow release finish 1.0.0
Переключено на ветку «main»
Ваша ветка обновлена в соответствии с «origin/main».
Merge made by the 'recursive' strategy.
 2020-2021/OS/VERSION | 1 +
 1 file changed, 1 insertion(+)
create mode 100644 2020-2021/OS/VERSION
Уже на «main»
Ваша ветка опережает «origin/main» на 2 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
fatal: нет описания метки?
fatal: Tagging failed. Please run finish again to retry.
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $

```

Figure 3.14: Заливаем ветку

f. Отправляем данные на github: `git push --all` `git push --tags`

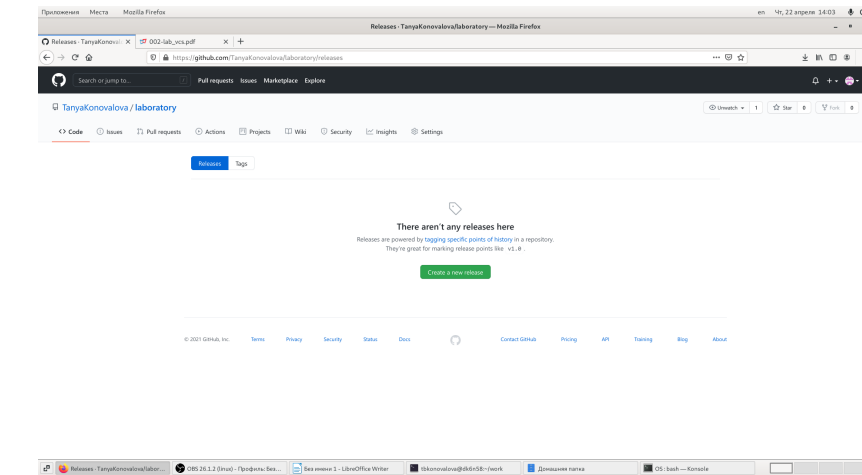
```

tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ git push --all
Username for 'https://github.com': TanyaKonovalova
Password for 'https://TanyaKonovalova@github.com':
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (6/6), 518 bytes | 518.00 KiB/s, готово.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/TanyaKonovalova/laboratory.git
 f8247c2..1b52ff8  main -> main
* [new branch]      develop -> develop
* [new branch]      release/1.0.0 -> release/1.0.0
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ git push --tags
Username for 'https://github.com': TanyaKonovalova
Password for 'https://TanyaKonovalova@github.com':
Everything up-to-date
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $

```

Figure 3.15: Отправляем данные на GitHub

9). 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 (подробное изложение действий представлено на рис. 91-93).



Releases Tags

v1.1 @ Target: release/1.0.0

Excellent! This tag will be created from the target when you publish this release.

2204

Write Preview

Hello!

Attach files by dragging & dropping, selecting or pasting them.

↓ Attach binaries by dropping them here or selecting them.

☐ This is a pre-release
We'll point out that this release is identified as non-production ready.

Publish release Save draft

Releases Tags

v1.1 @ Target: release/1.0.0

Excellent! This tag will be created from the target when you publish this release.

2204

Write Preview

Hello!

Attach files by dragging & dropping, selecting or pasting them.

↓ Attach binaries by dropping them here or selecting them.

☐ This is a pre-release
We'll point out that this release is identified as non-production ready.

Publish release Save draft

Tagging suggestions
It's common practice to prefix your version names with the letter v. Some good tag names might be v1.0 or v2.3.4.
If the tag isn't meant for production use, add a pre-release version after the version name. Some good pre-release versions might be v0.2-alpha or v5.9-beta.3.

Semantic versioning
If you're new to releasing software, we highly recommend reading about [semantic versioning](#).

4 Выводы

Изучила идеологию и научилась применять средства контроля версий.