

Отчёт по лабораторной работе

Дисциплина: операционные системы

Пермякова Елизавета Евгеньевна

Содержание

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

Список иллюстраций

3.1	Учетная запись	6
3.2	Система контроля git	6
3.3	Новый ключ	7
3.4	Создание репозитория	7
3.5	Копируем ссылку на репозиторий	7
3.6	Репозиторий	8
3.7	Создание файлов	8
3.8	Первый коммит	8
3.9	Сохранение	9
3.10	Файл лицензии	9
3.11	Шаблон игнорируемых файлов	10
3.12	Скачивание шаблона	10
3.13	Отправление на github	11
3.14	Инициализация git-flow	11
3.15	git brunch	11
3.16	Ветка develop	12
3.17	Создание релиза	12
3.18	Версия	12
3.19	Релизная ветка	13
3.20	Отправление данных	13
3.21	Releases	14
3.22	Заполнение полей	14
3.23	Сформированный релиз	15

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,dosximd(вархиве,поскольку он должен содержать скриншоты,Makefile ит.д.)

3 Выполнение лабораторной работы

Создаем учетную запись на <https://github.com>. (рис. @fig:001)

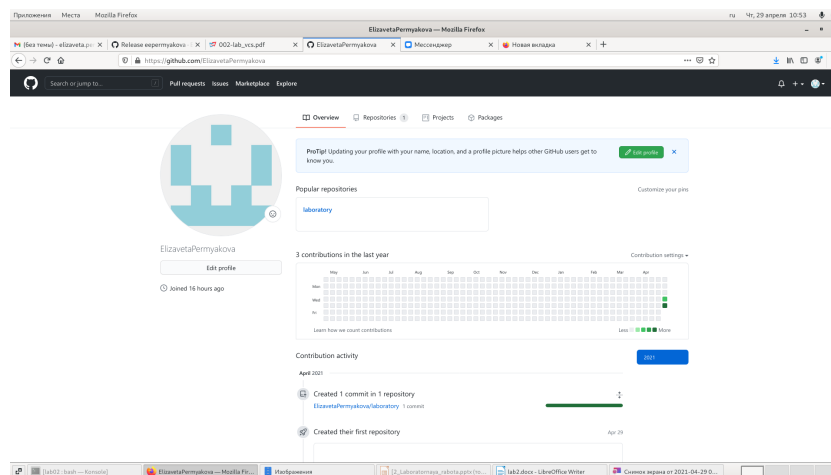


Рис. 3.1: Учетная запись

Настраиваем систему контроля версий git. Синхронизируем учётную запись github с компьютером. (рис. @fig:002)

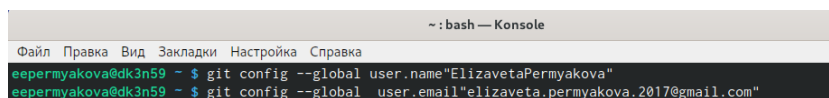


Рис. 3.2: Система контроля git

После этого создаём новый ключ на github и привязываем его к компьютеру через консоль. (рис. @fig:003)

```

eepermyakova@dk3n59 ~ $ ssh-keygen -C"ElizavetaPermyakova <elizaveta.permyakova.2017@gmail.com>"
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/e/e/eepermyakova/.ssh/id_rsa):
Created directory '/afs/.dk.sci.pfu.edu.ru/home/e/e/eepermyakova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/e/e/eepermyakova/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/e/e/eepermyakova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:AtK7fDvSsRawFyyTzmVegHuJ18Q9UDKx08Bp0bSLyo ElizavetaPermyakova <elizaveta.permyakova.2017@gmail.com>
The key's randomart image is:
+-----[RSA 3072]-----+
|
|.o=ooo+
|o=x.o. o
|..*=oo .
|oBoO
|= .X.+S
|E ..X.*..
|. oO.=
|..*.
|o..
+-----[SHA256]-----+
eepermyakova@dk3n59 ~ $

```

Рис. 3.3: Новый ключ

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). Копируем в консоль ссылку на репозиторий (для дальнейшей работы с файлами).(рис. @fig:004)

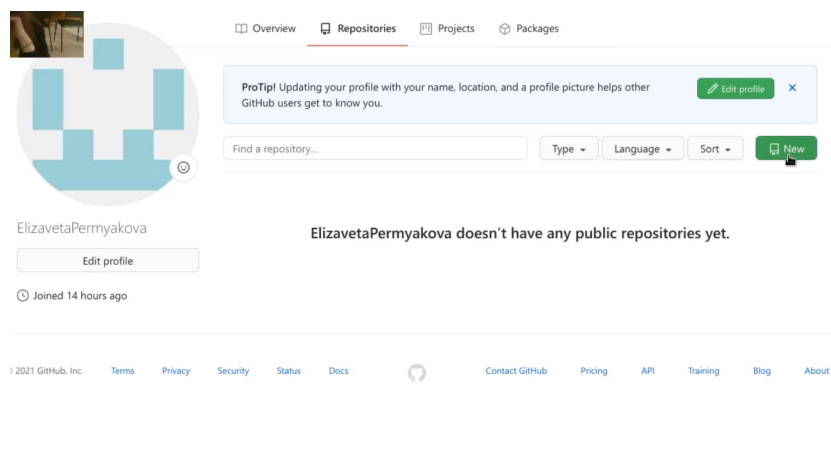


Рис. 3.4: Создание репозитория

(рис. @fig:005)

```

eepermyakova@dk3n59 ~ $ git clone https://github.com/ElizavetaPermyakova/laboratory.git
Клонирование в «laboratory»...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Получение объектов: 100% (3/3), готово.
eepermyakova@dk3n59 ~ $ cd laboratory

```

Рис. 3.5: Копируем ссылку на репозиторий

В лабораторной работе описан алгоритм создания структуры каталога через консоль. Но легче будет создать репозиторий в gethup 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:(рис. @fig:006)

```
eepermyakova@dk3n59 ~ $ cd laboratory
eepermyakova@dk3n59 ~/laboratory $ ls
README.md
```

Рис. 3.6: Репозиторий

После этого можем уже создавать наши файлы.(рис. @fig:007)

```
eepermyakova@dk3n59 ~/laboratory $ mkdir 2020-2021
eepermyakova@dk3n59 ~/laboratory $ cd 2020-2021
eepermyakova@dk3n59 ~/laboratory/2020-2021 $ mkdir 05
eepermyakova@dk3n59 ~/laboratory/2020-2021 $ cd 05
eepermyakova@dk3n59 ~/laboratory/2020-2021/05 $ mkdir lab02
eepermyakova@dk3n59 ~/laboratory/2020-2021/05 $ cd lab02
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $
```

Рис. 3.7: Создание файлов

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 -m “first commit” выкладываем коммит.(рис. @fig:007)

```
touch: невозможно выполнить touch для 'lab02/2': Нет такого файла или каталога
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $ touch b.txt
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $ git add .
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $ git commit -m "first commit"
[main ecd0c0f] first commit
Committer: Elizaveta Evgenjevna Permyakova <eepermyakova@dk3n59.dk.sci.pfu.edu.ru>
Ваше имя или электронная почта настроены автоматически на основании вашего имени пользователя и имени машины. Пожалуйста, проверьте, что они определены правильно.
Вы можете отключить это уведомление установив их напрямую. Запустите следующую команду и следуйте инструкциям вашего текстового редактора, для редактирования вашего файла конфигурации:

git config --global --edit

После этого, изменить авторство этой коммита можно будет с помощью команды:

git commit --amend --reset-author

2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 2020-2021/05/lab02/.txt
create mode 100644 2020-2021/05/lab02/b.txt
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $
```

Рис. 3.8: Первый коммит

Сохраняем первый коммит, используя команду git push.(рис. @fig:009)

```
fatal: Authentication failed for 'https://github.com:ElizavetaPermyakova/laboratory.git'
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $ git push
Username for 'https://github.com': ElizavetaPermyakova
Password for 'https://ElizavetaPermyakova@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), 437 bytes | 437.00 KiB/s, готово.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com:ElizavetaPermyakova/laboratory.git
   cb290ff..ecd0c0f  main -> main
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $
```

Рис. 3.9: Сохранение

Первичная конфигурация:

1. Добавляем файл лицензии.(рис. @fig:010)

```
cb290ff..ecd0c0f  main -> main
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $ wget https://creativecommons.org/licenses/by/4.0/legalcode.txt
--2021-04-29 10:11:17-- https://creativecommons.org/licenses/by/4.0/legalcode.txt
Распознаётся creativecommons.org (creativecommons.org). 104.20.150.16, 104.20.151.16, 172.67.34.140, ...
Подключение к creativecommons.org (creativecommons.org)[104.20.150.16]:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: нет данных [text/plain]
Сохранение в: «LICENSE»

LICENSE [ <=> ] 18,22K --.-KB/s за 0,001s

2021-04-29 10:11:17 (13,6 MB/s) - «LICENSE» сохранён [18657]
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $
```

Рис. 3.10: Файл лицензии

2. Добавим шаблон игнорируемых файлов. Просмотрим список имеющихся шаблонов (на скриншоте список шаблонов представлен не в целом виде).(рис. @fig:011)

```

eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $ curl -L -s https://www.gitignore.io/api/list
lc,lc-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+all,appcode+iml
appengine,aptanastudio,arcanist,archive,archives
archlinuxpackages,aspnetcore,assembler,ate,atmelstudio
ats,audio,automationstudio,autotools,autotools+strict
avr,azurefunctions,backup,ballerina,basercms
basic,batch,bazaar,bazel,bitrise
bitrix,bittorrent,blackbox,bloop,bluej
bookdown,bower,brickcc,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
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
epi-server,erlang,espresso,executable,exercism
expressionengine,extjs,fancy,fastlane,finale
firebase,flashbuilder,flask,flatpak,flex

```

Рис. 3.11: Шаблон игнорируемых файлов

3. Скачиваем шаблон, например, для C. Также добавляем новые файлы и выполняем коммит.(рис. @fig:012)

```

xojo,xtext,y86,yarn,yeoman
yii,yii2,zendframework,zephir,zig
zsh,zukencr8000eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $ curl -L -s https://www.gitignore.io/api/c >> .
gitignore
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $ git add .
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $ git commit -am "Создали шаблон для C"
[main 8964dd7] Создали шаблон для C
Committer: Elizaveta Evgenjevna Permyakova <eepermyakova@dk3n59.dk.sci.pfu.edu.ru>
Ваше имя или электронная почта настроены автоматически на основании вашего
имени пользователя и имени машины. Пожалуйста, проверьте, что они
определены правильно.
Вы можете отключить это уведомление установив их напрямую. Запустите следующую
команду и следуйте инструкциям вашего текстового редактора, для
редактирования вашего файла конфигурации:

    git config --global --edit

После этого, изменить авторство этой коммита можно будет с помощью команды:

    git commit --amend --reset-author

2 files changed, 455 insertions(+)
create mode 100644 2020-2021/05/lab02/.gitignore
create mode 100644 2020-2021/05/lab02/LICENSE
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $

```

Рис. 3.12: Скачивание шаблона

4. Отправим на github (для этого сохраним все созданные шаблоны и файлы, используя команду git push).(рис. @fig:013)

```

create mode 100644 2020-2021/05/lab02/LICENSE
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $ git push
Username for 'https://github.com': ElizavetaPermyakova
Password for 'https://ElizavetaPermyakova@github.com':
Перечисление объектов: 11, готово.
Подсчет объектов: 100% (11/11), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (7/7), 6.65 KiB | 6.65 MiB/s, готово.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ElizavetaPermyakova/laboratory.git
ecd0c0f..8964dd7 main -> main
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $

```

Рис. 3.13: Отправление на github

Работаем с конфигурацией git-flow.

1. Инициализируем git-flow, используя команду `git flow init -f` (префикс для ярлыков установлен в v).(рис. @fig:014)

```

ecd0c0f..8964dd7 main -> main
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $ 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/e/e/eepermyakova/laboratory/.git/hooks]
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $

```

Рис. 3.14: Инициализация git-flow

2. Проверяем, что мы находимся на ветке develop (используем команду `git branch`).(рис. @fig:015)

```

eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $ git branch
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $

```

Рис. 3.15: git branch

(рис. @fig:016)

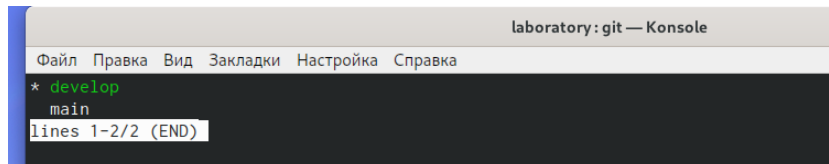


Рис. 3.16: Ветка develop

3. Создаём релиз с версией 1.0.0.(рис. @fig:017)

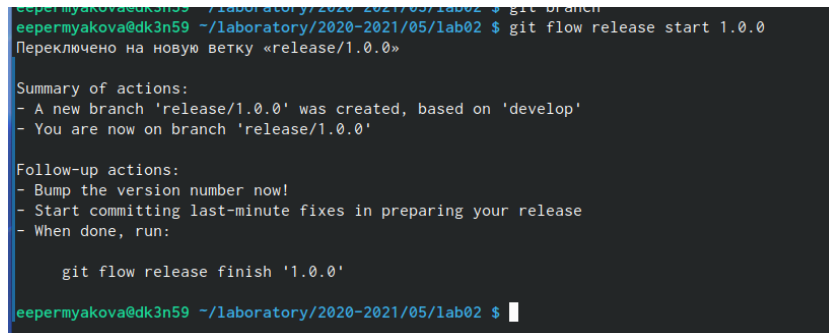


Рис. 3.17: Создание релиза

4. Запишем версию и добавим в индекс.(рис. @fig:018)

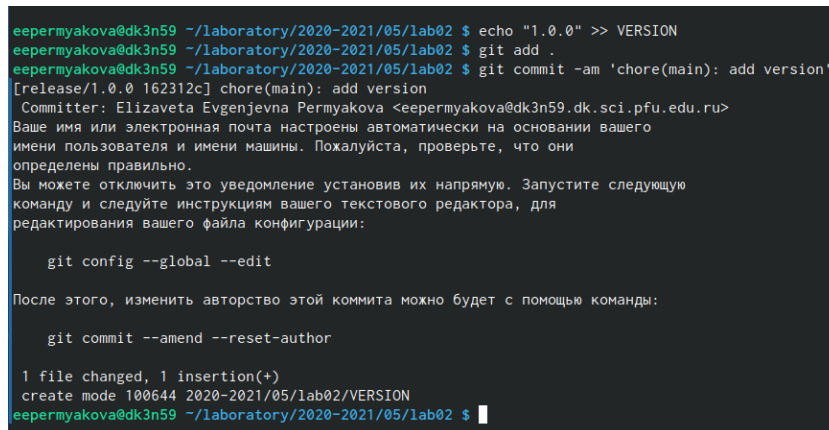


Рис. 3.18: Версия

5. Заливаем релизную ветку в основную ветку (используем команду git flow release finish1.0.0).(рис. @fig:019)

```

create mode 100644 2020-2021/05/lab02/VERSION
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $ 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.
 2020-2021/05/lab02/VERSION | 1 +
 1 file changed, 1 insertion(+)
create mode 100644 2020-2021/05/lab02/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.
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $

```

Рис. 3.19: Релизная ветка

6. Отправляем данные на github.(рис. @fig:020)

```

eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $ git push --all
Username for 'https://github.com': ElizavetaPermyakova
Password for 'https://ElizavetaPermyakova@github.com':
Перечисление объектов: 11, готово.
Подсчет объектов: 100% (11/11), готово.
При скатии изменений используется до 6 потоков
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (7/7), 570 bytes | 570.00 KiB/s, готово.
Total 7 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/ElizavetaPermyakova/laboratory.git
   8964dd7..7c5a82b  main -> main
* [new branch]      develop -> develop
* [new branch]      release/1.0.0 -> release/1.0.0
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $ git push --tags
Username for 'https://github.com': ElizavetaPermyakova
Password for 'https://ElizavetaPermyakova@github.com':
Everything up-to-date
eepermyakova@dk3n59 ~/laboratory/2020-2021/05/lab02 $

```

Рис. 3.20: Отправление данных

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:021)

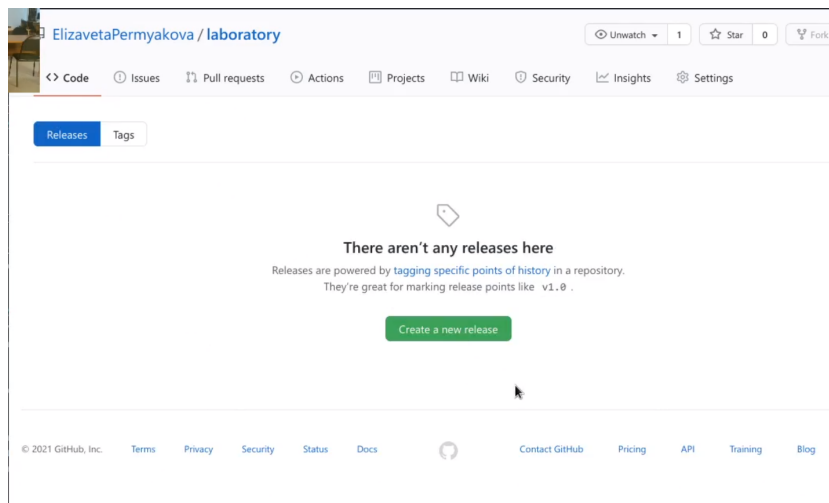


Рис. 3.21: Releases

(рис. @fig:022)

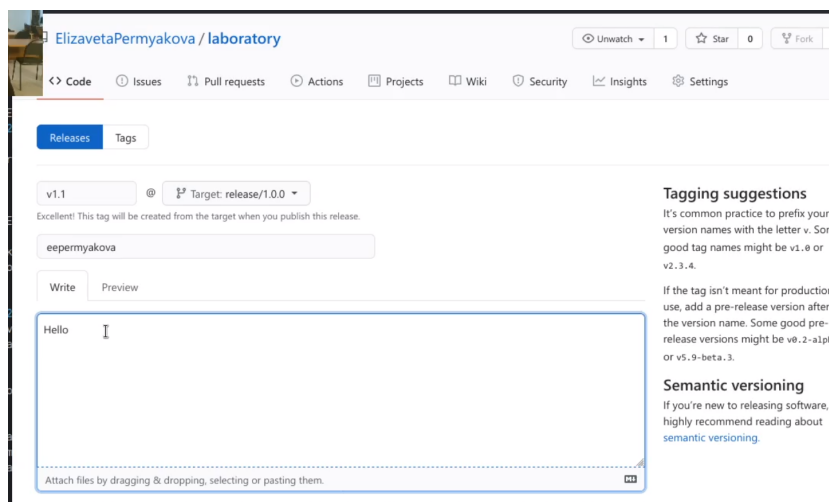


Рис. 3.22: Заполнение полей

(рис. @fig:023)

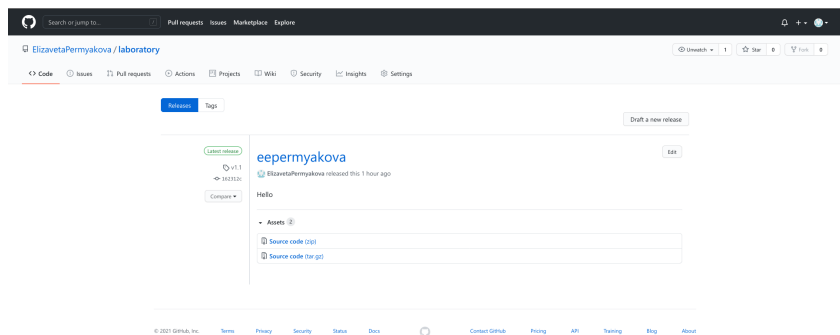


Рис. 3.23: Сформированный релиз

4 Выводы

Я изучила идеологию и научилась применять средства контроля версий.