

# **Лабораторная работа №2**

**Управление версиями**

Демидова Екатерина Алексеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>5</b>	<b>Контрольные вопросы</b>	<b>14</b>
<b>6</b>	<b>Выводы</b>	<b>16</b>
	<b>Список литературы</b>	<b>17</b>

## Список иллюстраций

4.1	Учетная запись . . . . .	7
4.2	Установка ПО . . . . .	8
4.3	Базовая настройка git . . . . .	8
4.4	Создание ключей ssh . . . . .	9
4.5	Создание ключей pgr . . . . .	9
4.6	Добавление ключа pgr . . . . .	10
4.7	Настройка коммитов . . . . .	10
4.8	Настройка gh . . . . .	11
4.9	Создание репозитория курса . . . . .	11
4.10	Настройка каталога курса . . . . .	12
4.11	Репозиторий git . . . . .	12
4.12	Репозиторий на локальной машине . . . . .	13

# 1 Цель работы

Целью работы является изучение идеологии и применения средств контроля версий, а также освоение умения по работе с git.

## 2 Задание

– Создать базовую конфигурацию для работы с git. – Создать ключ SSH. – Создать ключ PGP. – Настроить подписи git. – Зарегистрироваться на Github. – Создать локальный каталог для выполнения заданий по предмету.

### 3 Теоретическое введение

Система контроля версий (Version Control System, VCS) – это система, записывающая изменения в файл или набор файлов в течение времени и позволяющая вернуться позже к определенной версии [1]. Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд. Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд. | При выполнении работы была использована VCS Github [2]. История Git. В 2005 году компания, разрабатывающая систему контроля версий BitKeeper, порвала отношения с сообществом разработчиков ядра Linux. После этого сообщество приняло решение о разработке своей собственной системы контроля версий. Основными ценностями новой системы стали: полная децентрализация, скорость, простая архитектура, хорошая поддержка нелинейной разработки [1].

## 4 Выполнение лабораторной работы

1. Создаю учетную запись на <https://github.com>. и заполняю основные данные (рис. 4.1)

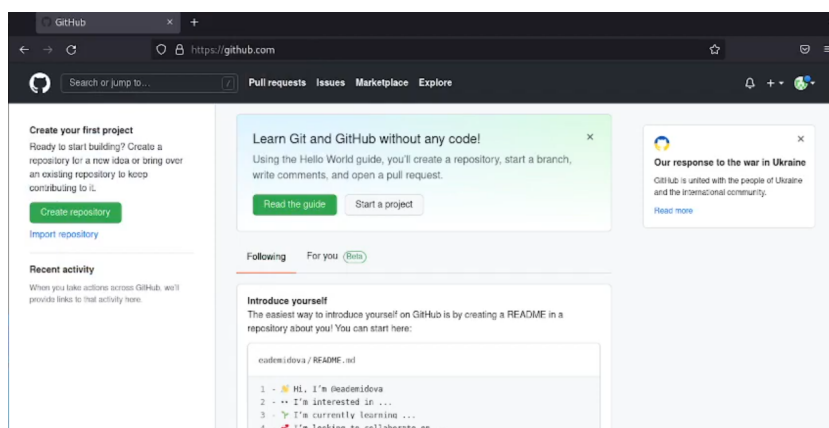
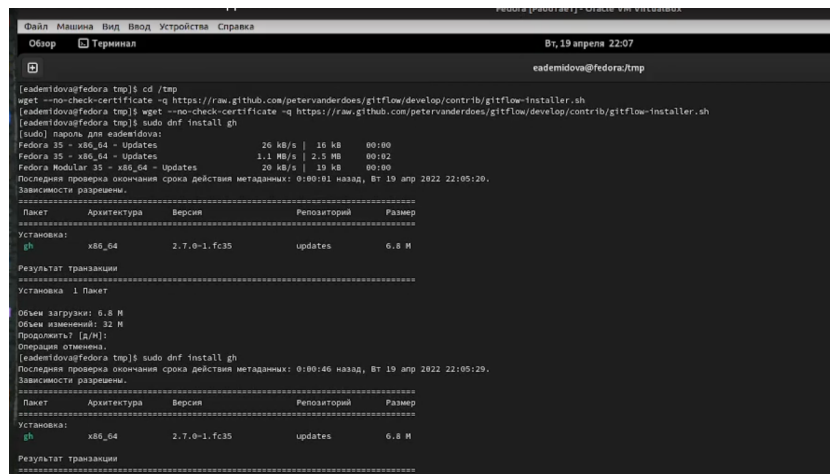


Рис. 4.1: Учетная запись

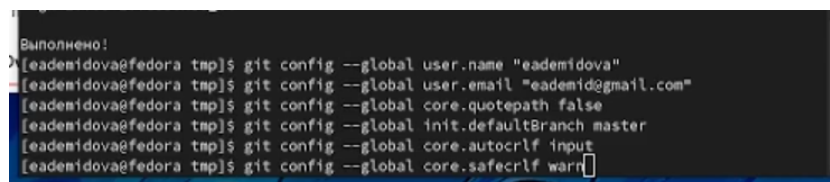
2. Устанавливаю программное обеспечение, а именно git-flow и gh (рис. 4.2)



```
[eademidova@fedora tmp]$ cd /tmp
[eademidova@fedora tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
[eademidova@fedora tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
[eademidova@fedora tmp]$ sudo dnf install git
[sudo] пароль для eademidova:
Fedora 35 - x86_64 - Updates                26 kB/s | 16 kB    00:00
Fedora 35 - x86_64 - Updates                3.1 MB/s | 2.5 MB  00:02
Fedora Modular 35 - x86_64 - Updates        20 kB/s | 19 kB   00:00
Последняя проверка окончания срока действия метаданных: 0:00:01 назад, Вт, 19 апр 2022 22:05:20.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Резпозиторий  Размер
-----
Установка:
gh         x86_64       2.35.1.fc35 updates      6.8 М
=====
Результат транзакции
=====
Установка 1 Пакет
=====
Объем загрузки: 6.8 М
Объем измененный: 32 М
Продолжить? [д/н]:
Операция отменена.
[eademidova@fedora tmp]$ sudo dnf install git
Последняя проверка окончания срока действия метаданных: 0:00:46 назад, Вт, 19 апр 2022 22:05:29.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Резпозиторий  Размер
-----
Установка:
gh         x86_64       2.35.1.fc35 updates      6.8 М
=====
Результат транзакции
=====
```

Рис. 4.2: Усатновка ПО

- Провожу базовую настройку git. Задаю имя и email владельца репозитория, настраиваю utf-8 в выводе сообщений git, верификацию и подписание коммитов git (рис. 4.3)

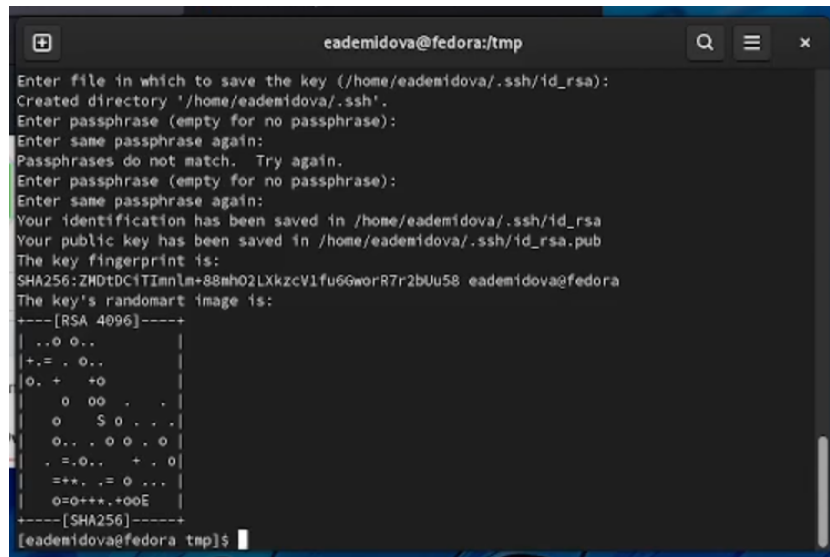


```
Выполнено!
[eademidova@fedora tmp]$ git config --global user.name "eademidova"
[eademidova@fedora tmp]$ git config --global user.email "eademid@gmail.com"
[eademidova@fedora tmp]$ git config --global core.quotepath false
[eademidova@fedora tmp]$ git config --global init.defaultBranch master
[eademidova@fedora tmp]$ git config --global core.autocrlf input
[eademidova@fedora tmp]$ git config --global core.safecrlf warn
```

Рис. 4.3: Базовая настройка git

- Создаю ключи ssh по алгоритму rsa и ed25519 (рис. 4.4)

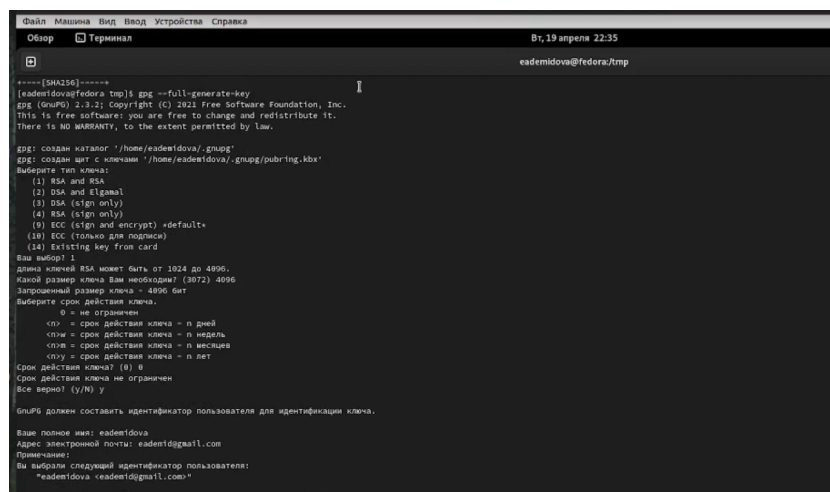




```
eademidova@fedora:/tmp
Enter file in which to save the key (/home/eademidova/.ssh/id_rsa):
Created directory '/home/eademidova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/eademidova/.ssh/id_rsa
Your public key has been saved in /home/eademidova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ZHDtDC1TImnIm+88mh02LXkzcV1fu6GworR7r2bUu58 eademidova@fedora
The key's randomart image is:
+----[RSA 4096]-----+
| .. 0 0.. |
| +. = 0.. |
| 0. + +0 |
|   0 00 . . |
|   o $ 0 . . |
|  o.. . 0 0 . 0 |
| . =.0.. + . 0 |
| =++. . = 0 ... |
| o=0+++.+ooE |
+----[SHA256]-----+
eademidova@fedora /tmp$
```

Рис. 4.4: Создание ключей ssh

5. Создаю ключи pgr, выбирая необходимые опции (рис. 4.5)



```
Файл Машина Вид Ввод устройства Справка
Обзор Терминал 19 апреля 22:35
eademidova@fedora:/tmp

+----[SHA256]-----+
eademidova@fedora /tmp$ gpg --full-generate-key
gpg (GnuPG) 2.3.2; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: создан каталог '/home/eademidova/.gnupg'
gpg: создан шит с ключами '/home/eademidova/.gnupg/pubring.kbx'
Выберите тип ключа:
(1) RSA and RSA
(2) DSA and ElGamal
(3) DSA (sign only)
(4) RSA (sign only)
(5) ECC (sign and encrypt) «default»
(10) ECC (только для подписи)
(14) Existing key from card
Ваш выбор? 1
длина ключей RSA может быть от 1024 до 4096.
Какой размер ключа Вам необходим? (3072) 4096
Запрошенный размер ключа = 4096 бит
Выберите срок действия ключа.
0 = не ограничен
<n> = срок действия ключа - n дней
<nw> = срок действия ключа - n недель
<nm> = срок действия ключа - n месяцев
<ny> = срок действия ключа - n лет
Срок действия ключа? (0) 0
Срок действия ключа не ограничен
Все верно? (y/N) y

GnuPG должен составить идентификатор пользователя для идентификации ключа.
Ваше полное имя: eademidova
Адрес электронной почты: eademid@gmail.com
Примечание:
Вы выбрали следующий идентификатор пользователя:
"eademidova <eademid@gmail.com>"
Сгенерировать GnuPG-ключ (y/N) (y) (G)Примечание: (z)Запросить (O)Отменить (C)Выход
```

Рис. 4.5: Создание ключей pgr

6. Добавляю PGR ключ в GitHub. Для этого вводим список ключей и копируем отпечаток приватного ключа, а затем добавляем его в специальное поле ввода в настройках GitHub (рис. 4.6)

## GPG keys / Add new

Key

-----BEGIN PGP PUBLIC KEY BLOCK-----

```
mQINBGJIDe4BEADNkxolBuUzMiGY7uaVx7mGVf5UFqne5D6Nng0tkM9bb/EoZN
ZaPQy+TwGjMRIEpaDGAgn1DeITXJJqngWN9Dt0aeWSu08We70DrZm9noNTaEM
YSkxvzPBSXFFvIqJontTRHcQSAaPLDWNM2YxipKkKauq0Lsa9buzXTtg19QgX+
gQ2TdpzvkdYoHaRpDGT4R9lkZBu9DrKibKz3FhMeUAqmY1RRAddYgQY/bsq8lp
2LRkwe+cmf03/WRS1rtx/qhXlavFe1YbSpd72ZGPyCUinL.DSSm8wcmfPKOZ4e
chwwt/20YGw3haxFu/Bolal.zaCiaMN+/qF4vSglqu6PzniTCor3QAO0cw6u4qFOH
IQYKJAZINaDNxtAqIXCSZQo01vw9zo9x+7ql84KAoTQWVG74yRijVQymUdFPsgIX
```

Add GPG key

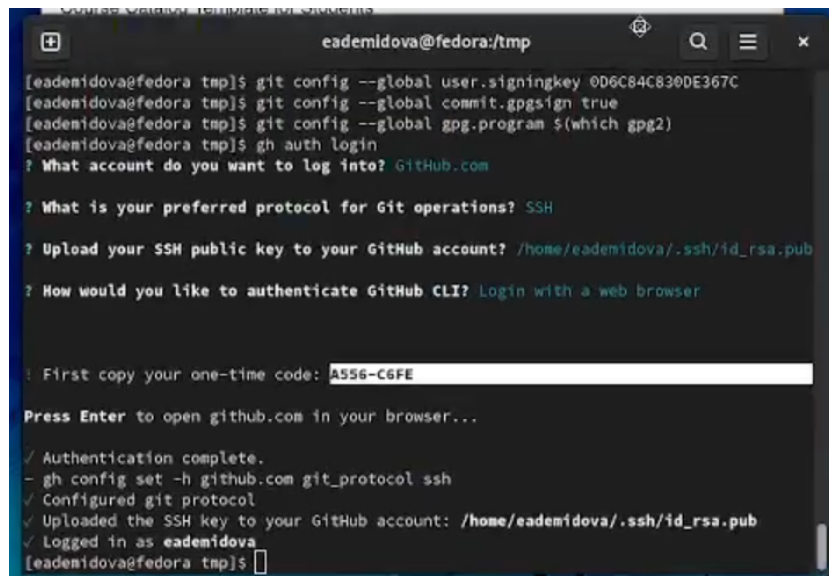
Рис. 4.6: Добавление ключа pgr

- Настраиваем автоматические подписи коммитов git. Используя введённый email, укажем Git применять его при подписи коммитов (рис. 4.7)

```
eademidova@fedora:/tmp
iA0nS1glLdjKc179Tp6KUUC5A3k+7E0owgEa4lv4onIs2FoQm6s9mkFANullygiF
eFqco58jTijjnvYTFM1sPMHYVfYlPC0EghcrH3HQVHoFyQAIAHh2HppKb70dpR0L
/iAnQjA8m4fPV3hcGXsXjT6LRahv62Xjd3DpZ8UZWtUyV1dJ9zJC000mo6zcpH+T
03Na0jvFz9/8DI2tRKb6dCHXTjFFr4tvIUEQFcxnrdjCp2Vg6XRxtjPngew74I0L
KnjV1ktoFQE3+Z5Jb96JwlbX1zJ+Uin+04i8k3nAoprtIwMY2dv4K0AmjGekqdj
ABEBAA6jA3jYEGAEIACAWIQQGLWGsUYXgwvhlirNMbITIMN42FAUCV180zgIbDAAK
CRANbITIMN42f05UEACbrJzDCdu5+bvRxx9jrdZpm7zLz3wbPQgj8CDLquNG3HxI
sc8VmK2RUDEfQncQzT+e/ScgiqxPaymXG3jB2nnszft3mX8BWPvj9dZa8dQil04U
Jox6B8EwLTFDD16x0ELKlqnBu8n9Rsi4nLss1X0m2ks5QUTdIMKA9PkHik/Icr9zw
BC9L00ddqtRD0tScw2kfm5/wEFEdiZzh2Ngf+mLc0rARmL/kRR+IRpc+cVvSHeDM
RTxWz52ax2sITGGiv8gPS3+gmEhz27kSXk10t8n0sg07L+/iA23TRk7xHzRfLpUc
TE5nLz9qgpoIg8B1hJuGedzQVW23GyQZ6wPyajx06IcHVtqOp62i7Q10JHijI07k
CCN7N80XQY3TdDC+VeNxRFsuw7tE0jyWoi3nZZVwe1LD7fzJNzy9Fi9p8EDdWR+x
0uM43GYSH1X+yWJpSczrPW3FR/g8e7W2ML9QuxfbNp3EYghWpw9B87goiqE65Gc
kAvbhaI3ALbsI30j6PsfJ2H3R+yMI7AXM22hM6lnabhw+jpL0pdjrbMH0Jlu2XT
npoPmdkjv0Ih+LjCxDaq7LdXyPpS2Iu6QsqsgdadHwVPi/GTcwCa0LR4Q28SuhGt
/txP0LsNAC7Ko3oQfH5DoP7yr4C8pebCFDzJr+bAII0QZKwj2psZ/PmZh8I2NQ==
=MWR/
-----END PGP PUBLIC KEY BLOCK-----
[eademidova@fedora tmp]$ gpg --armor --export 0D6C84C830DE367C | xclip -sel clip
[eademidova@fedora tmp]$ git config --global user.signingkey 0D6C84C830DE367C
[eademidova@fedora tmp]$ git config --global commit.gpgsign true
[eademidova@fedora tmp]$ git config --global gpg.program $(which gpg2)
```

Рис. 4.7: Настройка коммитов

- Настроим gh. Авторизуемся и ответм на несколько наводящих вопросов (рис. 4.8)



```
eademidova@fedora:tmp
[eademidova@fedora tmp]$ git config --global user.signingkey 0D6C84C830DE367C
[eademidova@fedora tmp]$ git config --global commit.gpgsign true
[eademidova@fedora tmp]$ git config --global gpg.program $(which gpg2)
[eademidova@fedora tmp]$ gh auth login
? What account do you want to log into? GitHub.com

? What is your preferred protocol for Git operations? SSH

? Upload your SSH public key to your GitHub account? /home/eademidova/.ssh/id_rsa.pub

? How would you like to authenticate GitHub CLI? Login with a web browser

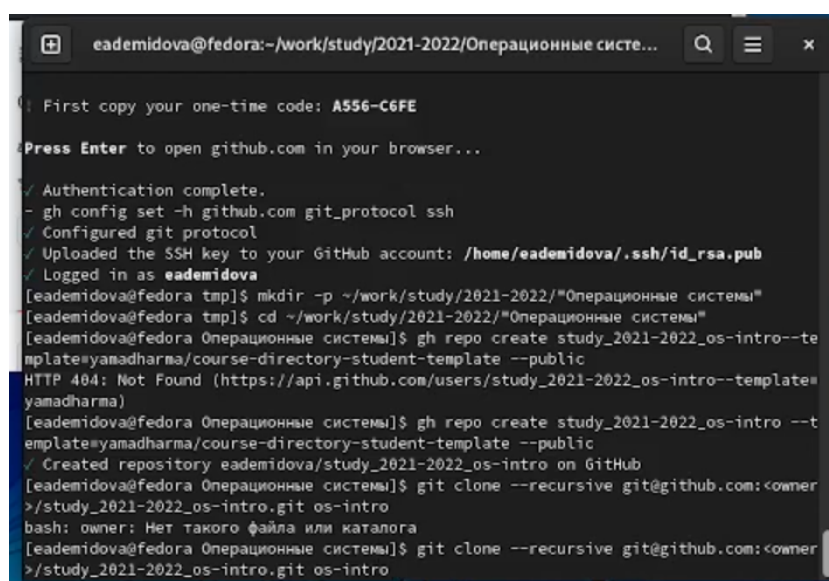
: First copy your one-time code: A556-C6FE

Press Enter to open github.com in your browser...

✓ Authentication complete.
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Uploaded the SSH key to your GitHub account: /home/eademidova/.ssh/id_rsa.pub
✓ Logged in as eademidova
[eademidova@fedora tmp]$
```

Рис. 4.8: Настройка gh

9. Создадим репозиторий курса на основе данного шаблона (рис. 4.9)



```
eademidova@fedora:~/work/study/2021-2022/Операционные систе...
: First copy your one-time code: A556-C6FE

Press Enter to open github.com in your browser...

✓ Authentication complete.
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Uploaded the SSH key to your GitHub account: /home/eademidova/.ssh/id_rsa.pub
✓ Logged in as eademidova
[eademidova@fedora tmp]$ mkdir -p ~/work/study/2021-2022/"Операционные системы"
[eademidova@fedora tmp]$ cd ~/work/study/2021-2022/"Операционные системы"
[eademidova@fedora Операционные системы]$ gh repo create study_2021-2022_os-intro --template=yamadharma/course-directory-student-template --public
HTTP 404: Not Found (https://api.github.com/users/study_2021-2022_os-intro--template=yamadharma)
[eademidova@fedora Операционные системы]$ gh repo create study_2021-2022_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository eademidova/study_2021-2022_os-intro on GitHub
[eademidova@fedora Операционные системы]$ git clone --recursive git@github.com:owner>/study_2021-2022_os-intro.git os-intro
bash: owner: Нет такого файла или каталога
[eademidova@fedora Операционные системы]$ git clone --recursive git@github.com:owner>/study_2021-2022_os-intro.git os-intro
```

Рис. 4.9: Создание репозитория курса

10. Перейдем в каталог курса, удалим лишние файлы, создадим необходимые каталоги и отправим файлы на сервер (рис. 4.10)

```
Подмодуль по пути «template/report»: забрано состояние «df7b2ef80f8def3b9a496f8695277469a1a7842a»
[eademidova@fedora Операционные системы]$ cd ~/work/study/2021-2022/"Операционные системы"/os-intro
[eademidova@fedora os-intro]$ rm package.json
[eademidova@fedora os-intro]$ make COURSE=os-intro
[eademidova@fedora os-intro]$ git add .
[eademidova@fedora os-intro]$ git commit -am 'feat(main): make course structure'
```

Рис. 4.10: Настройка каталога курса

Репозиторий git был успешно создан (рис. 4.11)

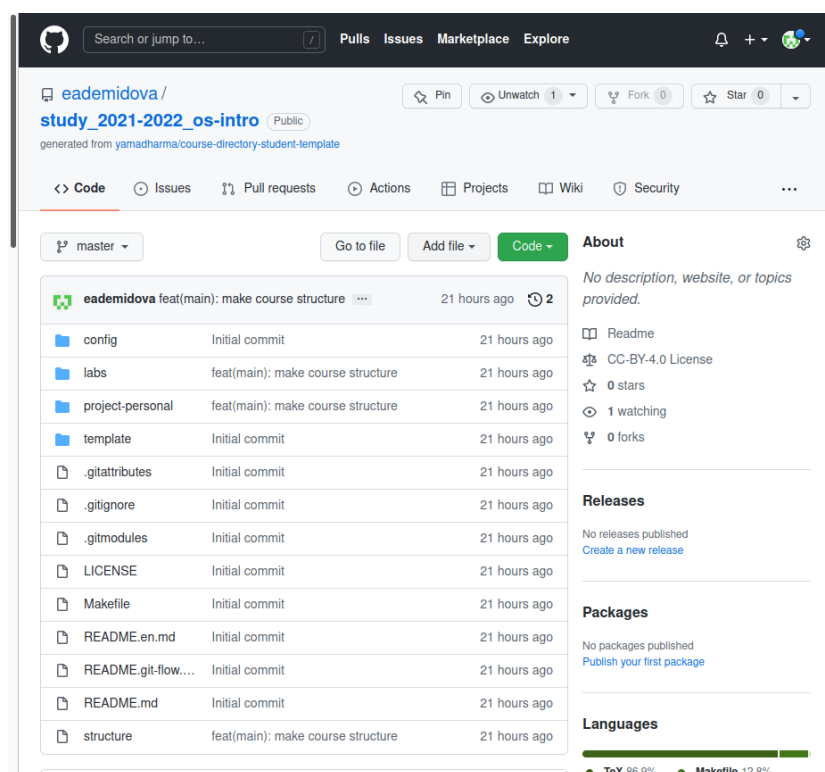


Рис. 4.11: Репозиторий git

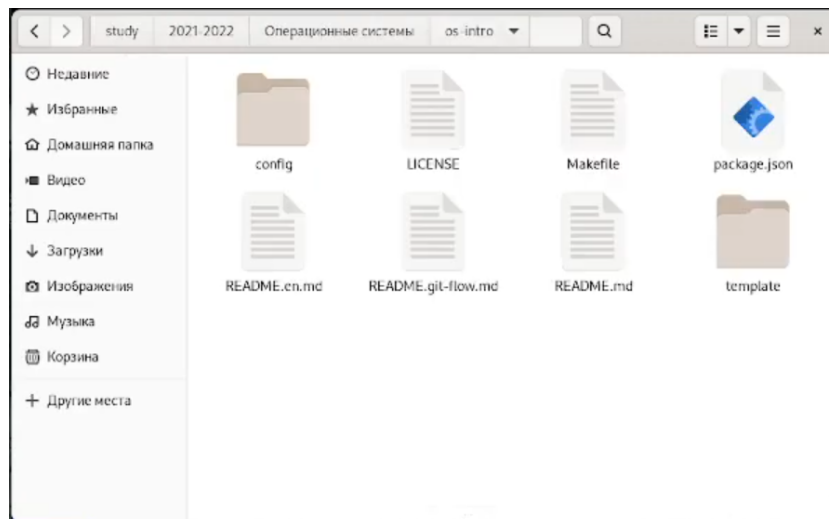


Рис. 4.12: Репозиторий на локальной машине

## 5 Контрольные вопросы

1. Системы контроля версий -VCS- это программное обеспечение, которое используется для облегчения работы с изменяющейся информацией, обычно - в проектах. Чаще всего используется при разработке, когда над одним проектом работает большое количество людей.
2. Репозиторий в системе контроля версий - Это удаленный репозиторий, в котором хранятся все файлы проекта `commit` - фиксирует изменения перед загрузкой файлов в систему контроля версий история хранит все изменения в проекте, и при необходимости позволяет откатиться в желаемое место рабочая копия - - это копия проекта на компьютере разработчика. Если другой член команды изменил проект, вам необходимо скачать новую версию проекта на свой компьютер.
3. В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным. Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд. В децентрализованных системах у каждого из участников проекта есть полная копия проекта на своем компьютере, что делает его менее зависимым от сервера (Git).
4. Для начала необходимо создать и подключить удаленный репозиторий. Затем, поскольку никто, кроме вас, не изменяет проект, по мере изменения проекта отправляйте изменения на сервер, и нет необходимости загружать

изменения.

5. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент.

6. Упрощение обмена информацией, ускорение разработки, устранение ошибок и недочётов при разработке.

git init - инициализирует локальный репозиторий  
git add \* или add. - добавляет файлы в репозиторий  
git commit - версия фиксации  
git pull - загружает текущую версию проекта  
git push - отправляет изменённый проект на сервер  
git checkout - позволяет переключаться между ветками  
git status - текущий статус проекта  
git branch - просмотреть доступные ветки  
git remote add - добавить удалённый репозиторий

git push -all (push origin master/любой branch)

7. Ветви функций, также иногда называемые ветвями тем, используются для разработки новых функций, которые должны появиться в текущих или будущих выпусках.
8. Существуют временные и системные файлы, которые загромождают проект и не нужны. путь к ним можно добавить в файл .gitignore, тогда они не будут добавлены в проект

## 6 Выводы

В результате выполнения лабораторной работы была изучена идеология и применение средств контроля версий. А также освоены умения по работе с git, были обретены навыки создания ключей, репозитория, добавления и удаления файлов из репозитория.



## Список литературы

1. Статья о системах контроля версий [Электронный ресурс]. 2020. URL: <https://habr.com/ru/company/otus/blog/521290/>.
2. GitHub [Электронный ресурс]. 2021. URL: <https://github.com/>.