

Лабораторная работа №2

Операционные системы

Тойчубекова Асель Нурлановна

29 января 2024

Российский университет дружбы народов, Москва, Россия

Информация

- Тойчубекова Асель Нурлановна
- Студент 1 курса
- Российский университет дружбы народов
- 1032235033@pfur.ru

Целью лабораторной работы №2 является изучение идеологии и применение средств контроля версий, а также освоение умений по работе с git.

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется.

Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями. Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

Ниже на фото представлены основные команды `git`.

Теоретическое введение

Git

Создание репозитория

Создать новый локальный репозиторий:

```
$ git init {имя проекта}
```

Скачать репозиторий:

```
$ git clone {url}
```

Состояние репозитория

Список незакоммиченных файлов:

```
$ git status
```

Изменения в ещё не добавленных в индекс файлах:

```
$ git diff
```

Изменения в уже добавленных:

```
$ git diff --cached
```

Все изменения:

```
$ git diff HEAD
```

Изменения между двумя коммитами:

```
$ git diff hash1 hash2
```

Получить даты изменений и их авторов:

```
$ git blame {имя файла}
```

Получить список изменений файла в определённом коммите:

```
$ git show hash:file
```

Список коммитов:

```
$ git log
```

Показать историю изменений, включая диффы:

```
$ git log -p {файл или папка}
```

Работа с ветками

Список локальных веток:

```
$ git branch
```

Список всех веток:

```
$ git branch -av
```

Переключиться на ветку mybranch:

```
$ git checkout mybranch
```

Создать ветку mybranch и переключиться на неё:

```
$ git checkout -b mybranch
```

Удалить ветку mybranch:

```
$ git branch -d mybranch
```

Стереть ветку a в b:

```
$ git checkout b && git merge a
```

Создать тег для текущего коммита:

```
$ git tag {имя тега}
```

Сохраняем изменения

Добавить файл в индекс:

```
$ git add {имя файла}
```

Добавить все файлы в папке:

```
$ git add .
```

Закommaитить добавленные в индекс файлы с сообщением:

```
$ git commit -m "сообщение"
```

Закommaитить все файлы с сообщением:

```
$ git commit -am "сообщение"
```

Удалить изменённый файл из индекса, оставив изменения в нём:

```
$ git reset {имя файла}
```

Отменить все изменения в папке и вернуть всё к последнему коммиту:

```
$ git reset --hard
```

Обновление

Скачать изменения из удалённого репозитория:

```
$ git fetch
```

Скачать изменения из удалённого репозитория и стереть:

```
$ git pull
```

Скачать изменения из удалённого репозитория и наложить незапушенные коммиты поверх скачанных:

```
$ git pull --rebase
```

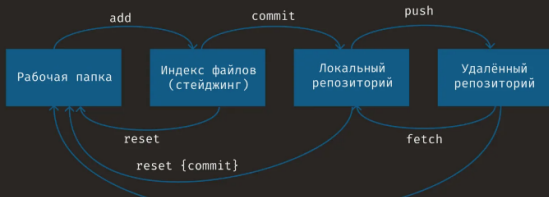
Отправить локальные коммиты в удалённый репозиторий:

```
$ git push
```

И главное!

Не забываем пользоваться подсказкой

```
$ git {имя команды} --help
```



Выполнение лабораторной работы

Установка программного обеспечения

Сперва захожу в терминал и устанавливаю git,используя команду 'dnf install git'.

```
[antoychubekova@antoychubekova ~]$ sudo dnf install git
[sudo] пароль для antoychubekova:
Fedora 39 - x86_64 - Updates
Fedora 39 - x86_64 - Updates
Последняя проверка окончания срока действия метаданных: 0:00:08 назад, Вс 25 фев 2024 16:27:52.
Пакет git-2.43.2-1.fc39.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
```

Рис. 2: Установка git

Устанавливаю gh, используя команду 'dnf install gh'.

```
[antojchubekova@antojchubekova ~]$ sudo dnf install gh
Последняя проверка окончания срока действия метаданных: 0:00:56 назад, Вс 25 фев 2024 16:27:52.
Пакет gh-2.43.1-1.fc39.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
```

Рис. 3: Установка gh

Базовая настройка git.

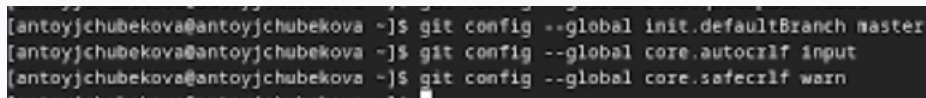
Пользуясь командой 'git config --global user.name "Asel Toychubekova"' задаю свое имя, а командой 'git config --global user.email "aseltoychubekova714@gmail.com"' задаю свой email для репозитория, а также настраиваю utf-8 в выводе сообщений git командой 'git config --global core.quotepath false'.

```
[antoychubekova@antoychubekova ~]$ git config --global user.name "Asel Toichubekova"  
[antoychubekova@antoychubekova ~]$ git config --global user.email "aseltoychubekova714@gmail.com"  
[antoychubekova@antoychubekova ~]$ git config --global core.quotepath false
```

Рис. 4: Установка имени, email пользователя и utf-8

Базовая настройка git.

Задаю имя начальной ветки, название-master, командой 'git config --global init.defaultBranch master', затем настраиваю параметры autocrlf(git config --global core.autocrlf input) и safecrlf(git config --global core.safecrlf warn).

A screenshot of a terminal window showing three git configuration commands being executed. The prompt is [antoyjchubekova@antoyjchubekova ~]. The commands are: git config --global init.defaultBranch master, git config --global core.autocrlf input, and git config --global core.safecrlf warn. The output of each command is not visible.

```
[antoyjchubekova@antoyjchubekova ~]$ git config --global init.defaultBranch master
[antoyjchubekova@antoyjchubekova ~]$ git config --global core.autocrlf input
[antoyjchubekova@antoyjchubekova ~]$ git config --global core.safecrlf warn
```

Рис. 5: Задание начальной ветки и настройка параметров autocrlf, safecrlf

Создание ключей ssh

Создаю ключ по алгоритму rsa с ключём размером 4096 бит, после чего создаю ключ по алгоритму ed25519, используя команду 'ssh keygen -t'.

```
[antoyjchubekova@antoyjchubekova ~]$ ssh keygen -t rsa -b 4096
ssh: Could not resolve hostname keygen: Name or service not known
[antoyjchubekova@antoyjchubekova ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/antoyjchubekova/.ssh/id_rsa):
Created directory '/home/antoyjchubekova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/antoyjchubekova/.ssh/id_rsa
Your public key has been saved in /home/antoyjchubekova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:pxujy0T1aTqPKn4txso0p0RWN3UyPqM4x5vYpxu5nUE antoyjchubekova@antoyjchubekova
The key's randomart image is:
+---[RSA 4096]---+
|      + O      |
|      + *      |
|     = +       |
|    . = .      |
|   o ..o.E5 .  |
|o  .++= o      |
|..o+^..o+      |
|=== o^..o+     |
|0=+o.+++.     |
+---[SHA256]-----+
[antoyjchubekova@antoyjchubekova ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/antoyjchubekova/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/antoyjchubekova/.ssh/id_ed25519
Your public key has been saved in /home/antoyjchubekova/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:6HYDN5VpxTDJUiheV2DdVtYo81FpcoNU45g/2b79NZN antoyjchubekova@antoyjchubekova
The key's randomart image is:
```

Создание ключа gpg

Генерирую ключ, командой 'gpg --full-generate-key', выбирая подходящие из предложенных опций.

```
[antoyjchubekova@antoyjchubekova ~]$ gpg --full-generate-key
gpg (GnuPG) 2.4.3; Copyright (C) 2023 g10 Code GmbH
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/antoyjchubekova/.gnupg"
Выберите тип ключа:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (9) 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
  <n>w = срок действия ключа - n недель
  <n>m = срок действия ключа - n месяцев
  <n>y = срок действия ключа - n лет
Срок действия ключа? (0)
Срок действия ключа не ограничен
Все верно? (y/N) y

GnuPG должен составить идентификатор пользователя для идентификации ключа.
```

Создание ключа pgp

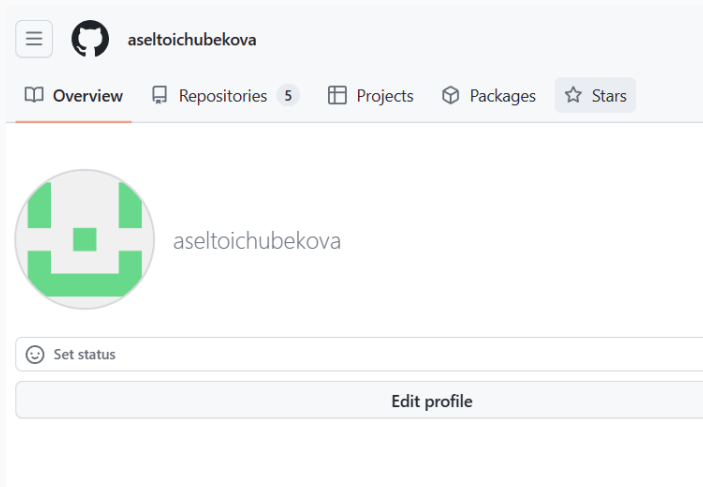
```
Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? o
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
gpg: /home/antoyjchubekova/.gnupg/trustdb.gpg: создана таблица доверия
gpg: создан каталог '/home/antoyjchubekova/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/home/antoyjchubekova/.gnupg/openpgp-revocs.d/0A5B0496C6B2D2895CC0B4D72E36AB2269233CC8.rev'.
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2024-02-25 [SC]
       0A5B0496C6B2D2895CC0B4D72E36AB2269233CC8
uid           Asel Toichubekova (asel) <aseltoychubekova714@gmail.com>
sub   rsa4096 2024-02-25 [E]
```

Рис. 8: Создание pgp ключа

Настройка github

У меня уже был настроен github, я создала учетную запись заполнила основные данные на прошлом семестре.



Добавление PGP ключа в GitHub

Вывожу список ключей, командой 'gpg --list-secret-keys --keyid-format LONG'.

```
[antoychubekova@antoychubekova ~]$ gpg --list-secret-key --keyid-format LONG
gpg: проверка таблицы доверия
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: глубина: 0  достоверных:   1  подписанных:   0  доверие: 0-, 0q, 0n, 0m, 0f, 1u
[keyboard]
-----
sec   rsa4096/2E36AB2269233CC8 2024-02-25 [SC]
      0A5B0496C682D2895CC0B4D72E36AB2269233CC8
uid           [ абсолютно ] Asel Toichubekova (asel) <aseltoychubekova714@gmail.com>
ssb   rsa4096/4EA7573ECDE05D8E 2024-02-25 [E]
```

Рис. 10: Список ключей

Добавление PGP ключа в GitHub

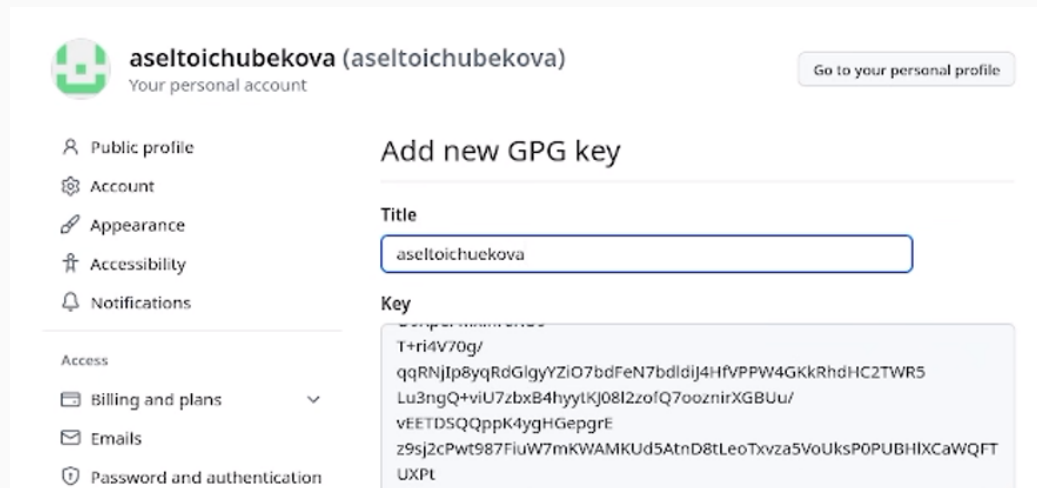
Копирую сгенерированный PGP ключ в буфер обмена, командой 'gpg --armor --export ключ | xclip -sel clip', перед этим установив команду xclip.

```
[antoyjchubekova@antoyjchubekova ~]$ sudo dnf install xclip
Последняя проверка окончания срока действия метаданных: 0:09:09 назад, Вс 25 фев 2024 17:01:20.
Нет соответствия аргументу: xclip
Ошибка: Совпадений не найдено: xclip
[antoyjchubekova@antoyjchubekova ~]$ sudo dnf install xclip
Последняя проверка окончания срока действия метаданных: 0:09:49 назад, Вс 25 фев 2024 17:01:20.
Зависимости разрешены.
=====
Пакет                Архитектура          Версия                Репозиторий           Размер
=====
Установка:
xclip                x86_64                0.13-20.git11cba61.fc39  Fedora                37 к
=====
Результат транзакции
=====
Установка 1 Пакет


Объем загрузки: 37 к
Объем изменений: 62 к
Продолжить? [д/н]: д
Загрузка пакетов:
xclip-0.13-20.git11cba61.fc39.x86_64.rpm                                222 kB/s | 37 kB    00:00
=====
Общий размер                45 kB/s | 37 kB    00:00
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка                  :
Установка                   : xclip-0.13-20.git11cba61.fc39.x86_64
Загрузка пакетов           : xclip-0.13-20.git11cba61.fc39.x86_64
=====
```






Добавление PGP ключа в GitHub

Перехожу в настройки GitHub, нажимаю на кнопку New GPG key и вставляю полученный ключ в поле ввода.






The screenshot shows the GitHub interface for a user named 'aseltoichubekova'. On the left is a sidebar with navigation links: 'Public profile', 'Account', 'Appearance', 'Accessibility', 'Notifications', 'Access', 'Billing and plans', 'Emails', and 'Password and authentication'. The main content area is titled 'Add new GPG key'. It contains a 'Title' field with the value 'aseltoichuekova' and a 'Key' field containing a long PGP key string. A button 'Go to your personal profile' is located in the top right corner.

 **aseltoichubekova** (aseltoichubekova)
Your personal account [Go to your personal profile](#)

-  Public profile
-  Account
-  Appearance
-  Accessibility
-  Notifications

Access

-  Billing and plans
-  Emails
-  Password and authentication

Add new GPG key

Title

Key

```
T+ri4V70g/  
qqRNjIp8yqRdGlgYyZiO7bdFeN7bdldij4HfVPPW4GKkRhdHC2TWR5  
Lu3ngQ+viU7zbxB4hytLKJ08I2zofQ7ooznirXGBUu/  
vEETDSQQppK4yHGepgrE  
z9sj2cPwt987FluW7mKWAMKUd5AtnD8tLeoTxvza5VoUksP0PUBHIXCaWQFT  
UXPt
```

Далее получаю сгенерированный ключ

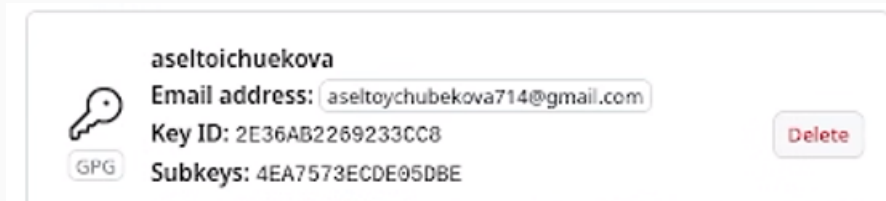


Рис. 13: Сгенерированный ключ

Настройка автоматических подписей коммитов git

Используя введённый email, укажем Git применять его при подписи коммитов (`git config --global user.signingkey email`)

```
[antoyjchubekova@antoyjchubekova ~]$ git config --global user.signingkey aseltoychubekova714@gmail.com
[antoyjchubekova@antoyjchubekova ~]$ git config --global commit.gpgsign true
[antoyjchubekova@antoyjchubekova ~]$ git config --global gpg.program $(which gpg2)
```

Рис. 14: Настройка автоматических подписей коммитов git

Используя команду 'gh auth login', авторизуюсь через браузер

```
[antoyjchubekova@antoyjchubekova ~]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations on this host? SSH
? Upload your SSH public key to your GitHub account? /home/antoyjchubekova/.ssh/id_rsa.pub
? Title for your SSH key: aseltoichubekova
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: DC67-16D9
Press Enter to open github.com in your browser...
█
```

Рис. 15: Авторизация через браузер



Device Activation



Signed in as aseltoichubekova

Enter the code displayed on your device

D C 6 7 - 1 6 D 9

Continue

GitHub staff will never ask you to enter your code on this page.

Создание репозитория курса на основе шаблона

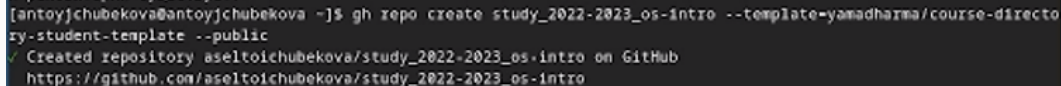
Создаю каталог, в котором мы будем дальше работать, перехожу в него, используя команды `mkdir` и `cd`

```
[antoyjchubekova@antoyjchubekova ~]$ mkdir -p ~/work/study/2023-2024/"Операционные системы"  
[antoyjchubekova@antoyjchubekova ~]$ cd ~/work/study/2023-2024/"Операционные системы"  
[antoyjchubekova@antoyjchubekova Операционные системы]$
```

Рис. 17: Создание каталога

Создание репозитория курса на основе шаблона

Далее создаю репозиторий на основе шаблона, с помощью команды 'gh repo create study_2022-2023_os-intro --template=yamadharma/course-directory-student-template --public'

A terminal window with a dark background and light-colored text. The command entered is 'gh repo create study_2022-2023_os-intro --template=yamadharma/course-directory-student-template --public'. The output shows a green checkmark, the message 'Created repository aseltoichubekova/study_2022-2023_os-intro on GitHub', and the repository URL 'https://github.com/aseltoichubekova/study_2022-2023_os-intro'.

```
[antoj]chubekova@antoj]chubekova ~]$ gh repo create study_2022-2023_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository aseltoichubekova/study_2022-2023_os-intro on GitHub
https://github.com/aseltoichubekova/study_2022-2023_os-intro
```

Рис. 18: Создание репозитория

Создание репозитория курса на основе шаблона

После клонирую репозиторий, командой 'git clone --recursive git@github.com:aseltoichubekova/study_2022-2023_os-intro.git os-intro'

```
[antojchubekova@antojchubekova Операционные системы]$ git clone --recursive git@github.com:aseltoichubekova/study_2023-2024_os-intro.git os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 32 (delta 1), reused 18 (delta 0), pack-reused 0
Получение объектов: 100% (32/32), 18.60 КиБ | 4.65 МБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yanadharna/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yanadharna/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/antojchubekova/work/study/2023-2024/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 95, done.
remote: Counting objects: 100% (95/95), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 95 (delta 34), reused 87 (delta 26), pack-reused 0
Получение объектов: 100% (95/95), 96.99 КиБ | 902.00 КиБ/с, готово.
Определение изменений: 100% (34/34), готово.
Клонирование в «/home/antojchubekova/work/study/2023-2024/Операционные системы/os-intro/template/report»...
remote: Enumerating objects: 126, done.
remote: Counting objects: 100% (126/126), done.
```

Перехожу в каталог курса -> `cd ~/work/study/2022-2023/“Операционные системы”/os-intro` и удаляю лишний файл `rm package.json`

```
[antoyjchubekova@antoyjchubekova Операционные системы]$ cd os-intro
[antoyjchubekova@antoyjchubekova os-intro]$ rm package.json
[antoyjchubekova@antoyjchubekova os-intro]$ ls
CHANGELOG.md  config  COURSE  LICENSE  Makefile  README.en.md  README.git-flow.md  README.md  template
```

Рис. 20: Удаление лишнего файла

Настройка каталога курса

Создаю необходимые каталоги, используя команды : `echo os-intro > COURSE`,
затем `make`

```
[antoyjchubekova@antoyjchubekova os-intro]$ echo os-intro > COURSE
[antoyjchubekova@antoyjchubekova os-intro]$ make
Usage:
  make <target>

Targets:
  list           List of courses
  prepare        Generate directories structure
  submodule      Update subnules

[antoyjchubekova@antoyjchubekova os-intro]$ make prepare
```

Рис. 21: Создание каталогов

Настройка каталога курса

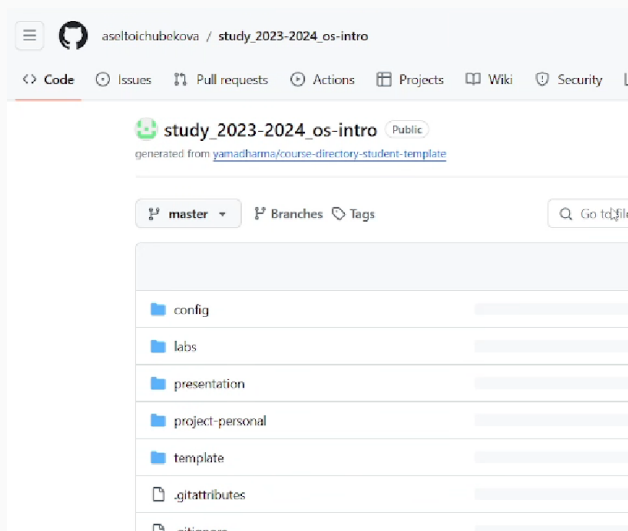
- Отправляю файлы на сервер: - git add .
- git commit -am 'feat(main): make course structure'
 - git push.

```
[antoyjchubekova@antoyjchubekova os-intro]$ git add .
[antoyjchubekova@antoyjchubekova os-intro]$ git commit -am 'feat(main): make course structure'
[master 167046b] feat(main): make course structure
361 files changed, 98413 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0.5-2008-numeric.csl
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/pandocattributes.py
```



Настройка каталога курса

Зайдем в github и видим репозиторий созданный по шаблону



В ходе выполнения лабораторной работы №2 я изучила идеологию и примечание средств контроля версии, а также освоила умения пл работе с git

- <https://esystem.rudn.ru/mod/page/view.php?id=1098933#org2151722>.