

Выполнил: ст.гр.11405121

Рак И. П.

Проверил: старший преподаватель

Будо А. Ю.

### Лабораторная работа №3

#### Начало работы с Git

Создаем папку first-project, где будет два файла todo.txt и readme.txt, в которых содержатся: файл todo.txt, в котором будет список дел, и readme.txt для информации о проекте.

```
$ touch todo.txt
```

```
$ touch readme.txt
```

Инициализируем репозиторий.

Сделать папку репозиторием - git init

```
rip2023@rip2023-VirtualBox:~/lab/first-project$ git init
подсказка: Using 'master' as the name for the initial branch. This default branch name
подсказка: is subject to change. To configure the initial branch name to use in all
подсказка: of your new repositories, which will suppress this warning, call:
подсказка:
подсказка:     git config --global init.defaultBranch <name>
подсказка:
подсказка: Names commonly chosen instead of 'master' are 'main', 'trunk' and
подсказка: 'development'. The just-created branch can be renamed via this command:
подсказка:
подсказка:     git branch -m <name>
Инициализирован пустой репозиторий Git в /home/rip2023/lab/first-project/.git/
```

Чтобы Git начал отслеживать изменения в проекте, папку с файлами можно сделать Git-репозиторием. Для этого нужно переместиться в нее и ввести команду.

«Разгитить» папку, если что-то пошло не так, — `rm -rf .git`.

Если вы случайно сделали Git-репозиторием не ту папку, её можно «разгитить». Для этого нужно удалить скрытую подпапку .git.

```
$ cd <папка с репозиторием> # перешли в папку
```

```
$ rm -rf .git # удалили подпапку .git
```

Проверить состояние репозитория — `git status`

После инициализации репозитория first-project запустите команду `git status` (от англ. status — «статус», «состояние») — она показывает текущее состояние репозитория.

```
rip2023@rip2023-VirtualBox:~/lab/first-project$ touch todo.txt
rip2023@rip2023-VirtualBox:~/lab/first-project$ touch readme.txt
rip2023@rip2023-VirtualBox:~/lab/first-project$ git status
Текущая ветка: master

Еще нет коммитов

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

  readme.txt
  todo.txt

индекс пуст, но есть неотслеживаемые файлы
(используйте «git add», чтобы проиндексировать их)
```

Подготовить файлы к сохранению — `git add` и `git add --all`

```
rip2023@rip2023-VirtualBox:~/lab/first-project$ git add --all
rip2023@rip2023-VirtualBox:~/lab/first-project$ git status
Текущая ветка: master

Еще нет коммитов

Изменения, которые будут включены в коммит:
(используйте «git rm --cached <файл>...», чтобы убрать из индекса)
    новый файл:   readme.txt
    новый файл:   todo.txt
```

Важно помнить, что после каждого исправления файла нужно сохранять их с помощью команды `git add` или `git add` с флагом `*--add*`.

### Делаем первый коммит

Сделать коммит можно командой `git commit` с ключом `-m` (от англ. message — «сообщение»), который присваивает коммиту сообщение.

```
rip2023@rip2023-VirtualBox:~/lab/first-project$ git commit -m 'Мой первый коммит!'
[master (корневой коммит) 6b5a892] Мой первый коммит!
2 files changed, 10 insertions(+)
create mode 100644 readme.txt
create mode 100644 todo.txt
```

### Задание для самостоятельной работы

```
rip2023@rip2023-VirtualBox:~/lab/first-project$ git log
commit 3de3f69f68e43aa3487c0a09156b37ca7958b466 (HEAD -> master)
Author: Ivan <rip20031982@gmail.com>
Date:   Mon Dec 11 02:39:43 2023 +0300

    Мой третий коммит!

commit b277edeace913dfa9d112f63dc69d96705d840b8
Author: Ivan <rip20031982@gmail.com>
Date:   Mon Dec 11 02:38:29 2023 +0300

    Мой второй коммит!

commit 6b5a892667aea1d2f5c874614f5e88fb9669670e
Author: Ivan <rip20031982@gmail.com>
Date:   Mon Dec 11 02:31:10 2023 +0300

    Мой первый коммит!
```

### Знакомство с GitHub

До этого момента вы использовали Git локально: сейчас проект `first-project` хранится только на вашем компьютере. Но одно из ключевых преимуществ Git — удобство командной работы над файлами. Чтобы поделиться репозиторием — например, с коллегами, — нужно завести его удалённую версию.

## Создаём удалённый репозиторий

В этой работе будем использовать уже существующий репозиторий на Github **first-project**. Устанавливаем утилиту xclip и копируем ключ

Устанавливаем утилиту xclip.

```
# sudo apt install xclip
```

```
rip2023@rip2023-VirtualBox: ~$ sudo apt install xclip
[sudo] пароль для rip2023:
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Следующие НОВЫЕ пакеты будут установлены:
  xclip
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 4 пакетов не обновлено.
Необходимо скачать 18,3 kB архивов.
После данной операции объем занятого дискового пространства возрастёт на 60,4 kB.
Пол:1 http://by.archive.ubuntu.com/ubuntu jammy/universe amd64 xclip amd64 0.13-2 [18,3 kB]
Получено 18,3 kB за 0с (134 kB/s)
Выбор ранее не выбранного пакета xclip.
(Чтение базы данных ... на данный момент установлено 182884 файла и каталога.)
Подготовка к распаковке ./xclip_0.13-2_amd64.deb ...
Распаковывается xclip (0.13-2) ...
Настраивается пакет xclip (0.13-2) ...
Обрабатываются триггеры для man-db (2.10.2-1) ...
rip2023@rip2023-VirtualBox: ~$
```

## Генерируем SSH-ключ

Генерируем и проверяем SSH-ключ

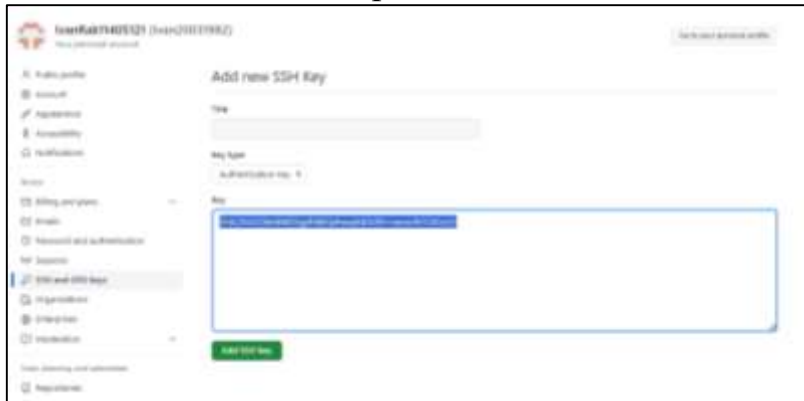
```
rip2023@rip2023-VirtualBox: ~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/rip2023/.ssh/id_ed25519):
Created directory '/home/rip2023/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/rip2023/.ssh/id_ed25519
Your public key has been saved in /home/rip2023/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:SeTRHY4iNmrgEw6C/60vFCHF3msnQT66VtxkUaiMofKI rip2023@rip2023-VirtualBox
The key's randomart image is:
+--[ED25519 256]--+
|  . . o.+...oo  |
|  . = o B.*o+o  |
|  . . X * X+=o.  |
|  . * B + ==.o   |
|  E =   So . +   |
|  +     . o      |
|  o              |
|  . .            |
|  .+.            |
+-----[SHA256]-----+
rip2023@rip2023-VirtualBox: ~$ ls -la .ssh/
итого 16
drwx----- 2 rip2023 rip2023 4096 сие 11 03:40 .
drwxr-x--- 17 rip2023 rip2023 4096 сие 11 03:40 ..
-rw----- 1 rip2023 rip2023 419 сие 11 03:40 id_ed25519
-rw-r--r-- 1 rip2023 rip2023 108 сие 11 03:40 id_ed25519.pub
```

Копируем ключ.

```
rip2023@rip2023-VirtualBox: ~$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIFg7X/S3mk6/oam0IH805nuVxRo02aIso7KQrE0spU3o rip2023@rip2023-VirtualBox
```



## Вставляем в настройках в гитхабе



## Связываем локальный и удалённый репозитории

```
rip2023@rip2023-VirtualBox:~/lab/first-project$ git remote add
использование: git remote add [<опции>] <имя> <url>

-f, --fetch                извлечь внешние ветки
--tags                    импортировать все метки и ассоциированные объекты при извлечении
                           или не извлекать метки вообще (--no-tags)
-t, --track <ветка>      отслеживаемые ветки
-m, --master <ветка>     мастер ветка
--mirror[=(push|fetch)]  настроить внешний репозиторий как зеркало для отправки или извлечени
я изменений

rip2023@rip2023-VirtualBox:~/lab/first-project$ git remote add origin git@github.com:Ivan20031
982/first-project.git
error: внешний репозиторий origin уже существует
rip2023@rip2023-VirtualBox:~/lab/first-project$ git remote add amega git@github.com:Ivan200319
82/first-project.git
```

## Проверка.

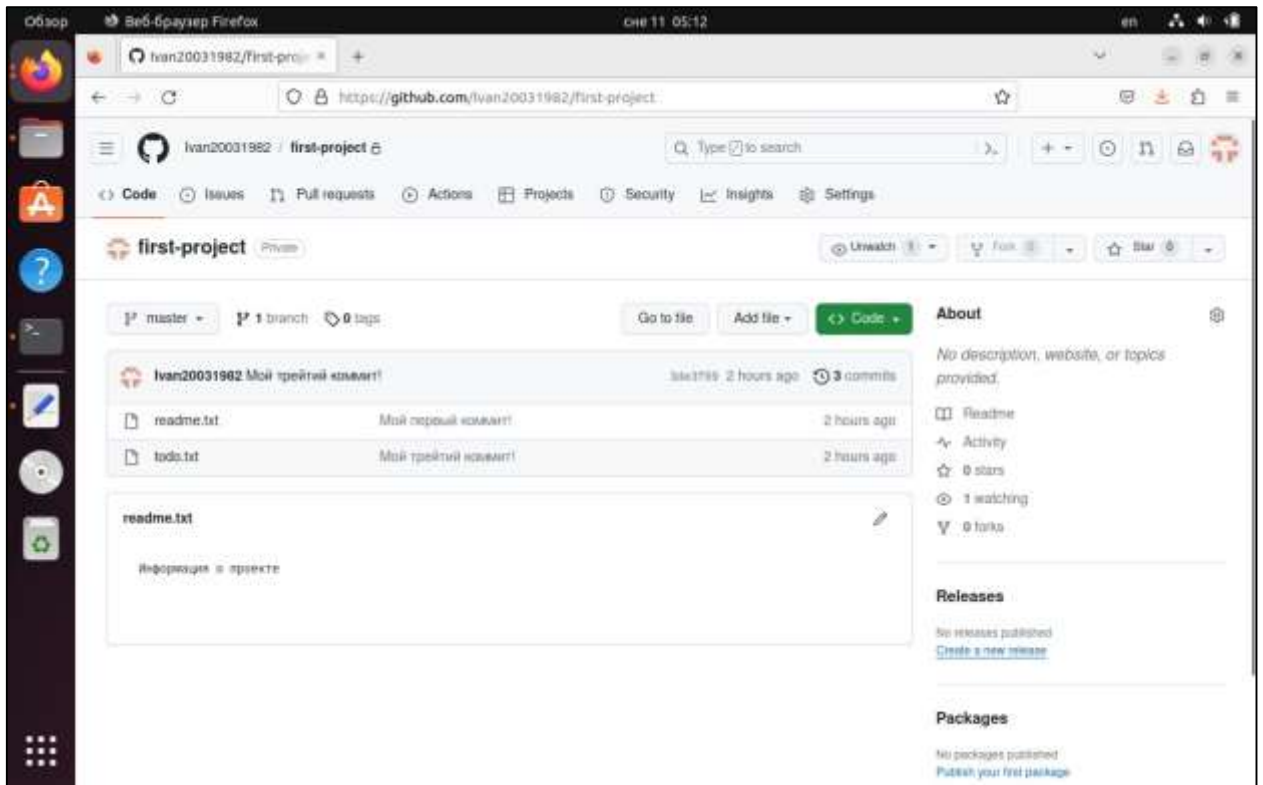
```
rip2023@rip2023-VirtualBox:~/lab/first-project$ git remote -v
amega  git@github.com:Ivan20031982/first-project.git (fetch)
amega  git@github.com:Ivan20031982/first-project.git (push)
origin  https://github.com/Ivan20031982/first-project.git (fetch)
origin  https://github.com/Ivan20031982/first-project.git (push)
```

## Синхронизируем локальный и удалённый репозитории

Если 2 раза нажать на Tab, то можно узнать, что надо писать в своей версии linux: main или master.

```
rip2023@rip2023-VirtualBox:~/lab/first-project$ git branch
HEAD    master
rip2023@rip2023-VirtualBox:~/lab/first-project$ git push -u amega master
Перечисление объектов: 10, готово.
Подсчет объектов: 100% (10/10), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (9/9), готово.
Запись объектов: 100% (10/10), 958 байтов | 479.00 КиБ/с, готово.
Всего 10 (изменений 2), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (2/2), done.
To github.com:Ivan20031982/first-project.git
 * [new branch]    master -> master
Ветка «master» отслеживает внешнюю ветку «master» из «amega».
```

Заходим на гитхаб.



## Практическая работа №1

```
rip2023@rip2023-VirtualBox:~/lab/first-project$ touch README.md
rip2023@rip2023-VirtualBox:~/lab/first-project$ git add README.md
rip2023@rip2023-VirtualBox:~/lab/first-project$ git commit -m 'New file'
[master a2d6a0b] New file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
rip2023@rip2023-VirtualBox:~/lab/first-project$ git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 302 байта | 302.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To github.com:Ivan20031982/first-project.git
   3de3f69..a2d6a0b  master -> master
rip2023@rip2023-VirtualBox:~/lab/first-project$ git add README.md
rip2023@rip2023-VirtualBox:~/lab/first-project$ git commit -m 'Go!'
[master aad2d82] Go!
 1 file changed, 2 insertions(+)
rip2023@rip2023-VirtualBox:~/lab/first-project$ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 312 байтов | 312.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To github.com:Ivan20031982/first-project.git
   a2d6a0b..aad2d82  master -> master
rip2023@rip2023-VirtualBox:~/lab/first-project$
```

## Практическая работа №2

```mermaid

graph TD;

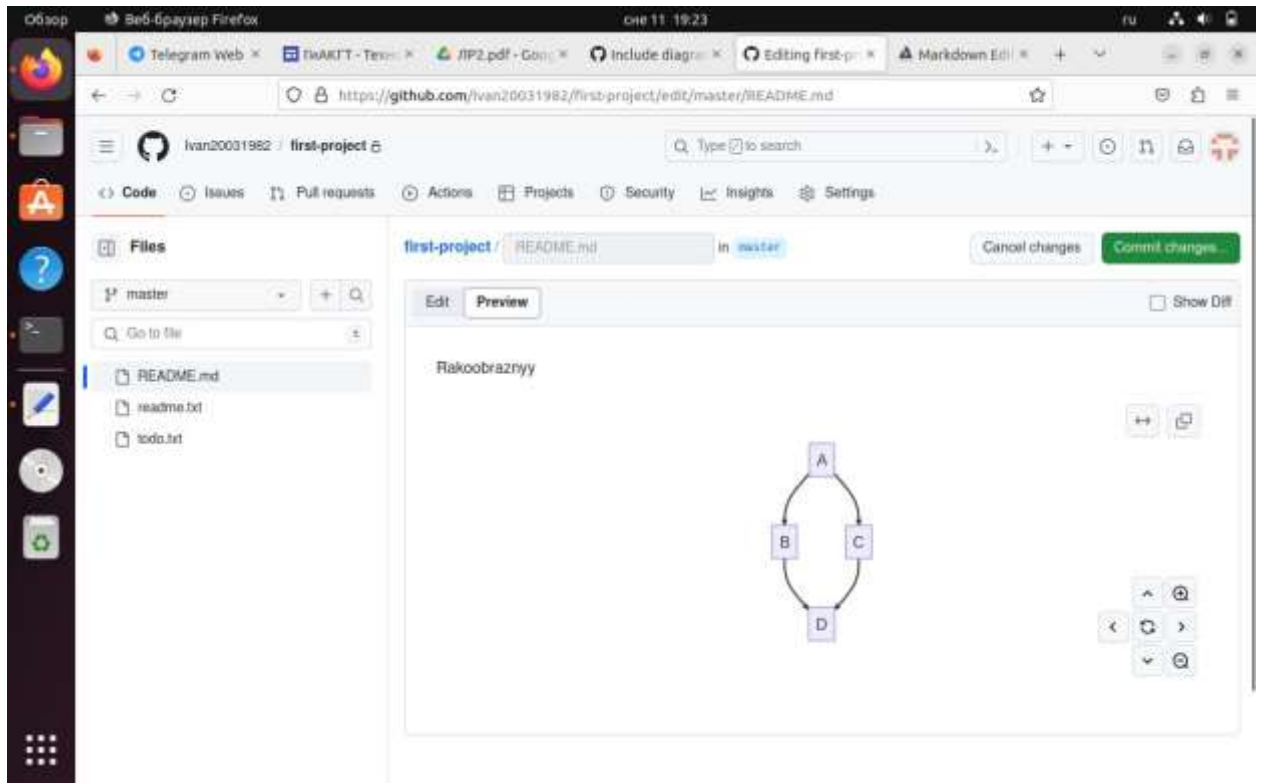
A-->B;

A-->C;

B-->D;

C-->D;

```



## Практическая работа №3

Когда написал ненужное: ```git restore <file>```

Когда уже накомитил: ```git reset --hard <time travel!>```

## Практическая работа №4

Нам понадобятся команды `git log` и `git diff`.