

# lab02

21 April, 2022 Moscow, Russia

## Лабораторная работа № 2. Управление версиями.

### Цель работы

- Изучить идеологию и применение средств контроля версий.
- Освоить умения по работе с git.

### 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.

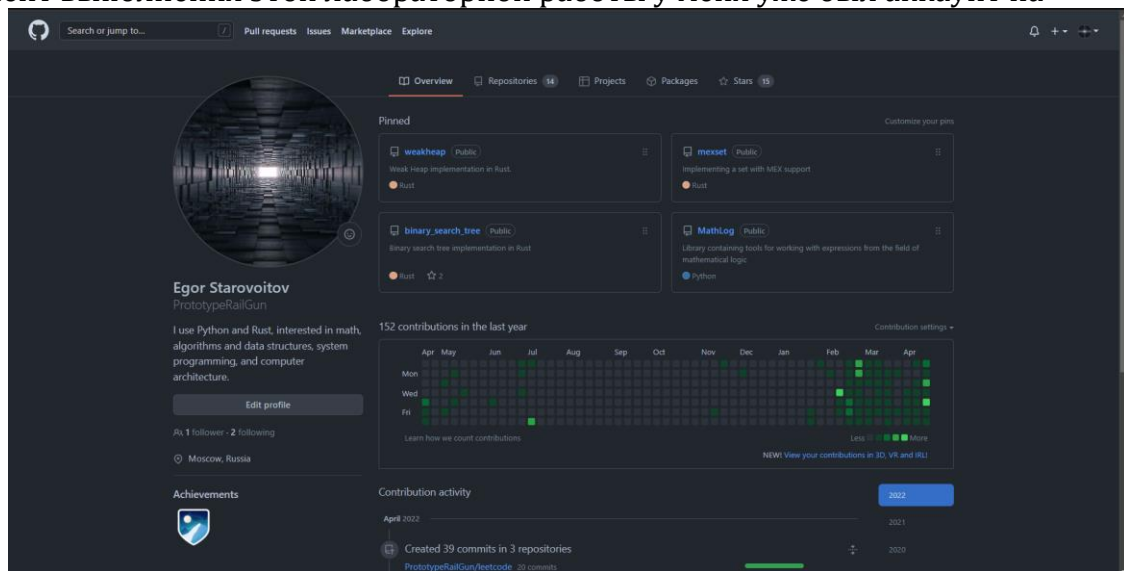
## Выполнение лабораторной работы

### Шаг 1. Настройка github

Инструкция была следующей: 1. Создайте учётную запись на <https://github.com>. 2. Заполните основные данные на <https://github.com>.

На момент выполнения этой лабораторной работы у меня уже был аккаунт на

GitHub.

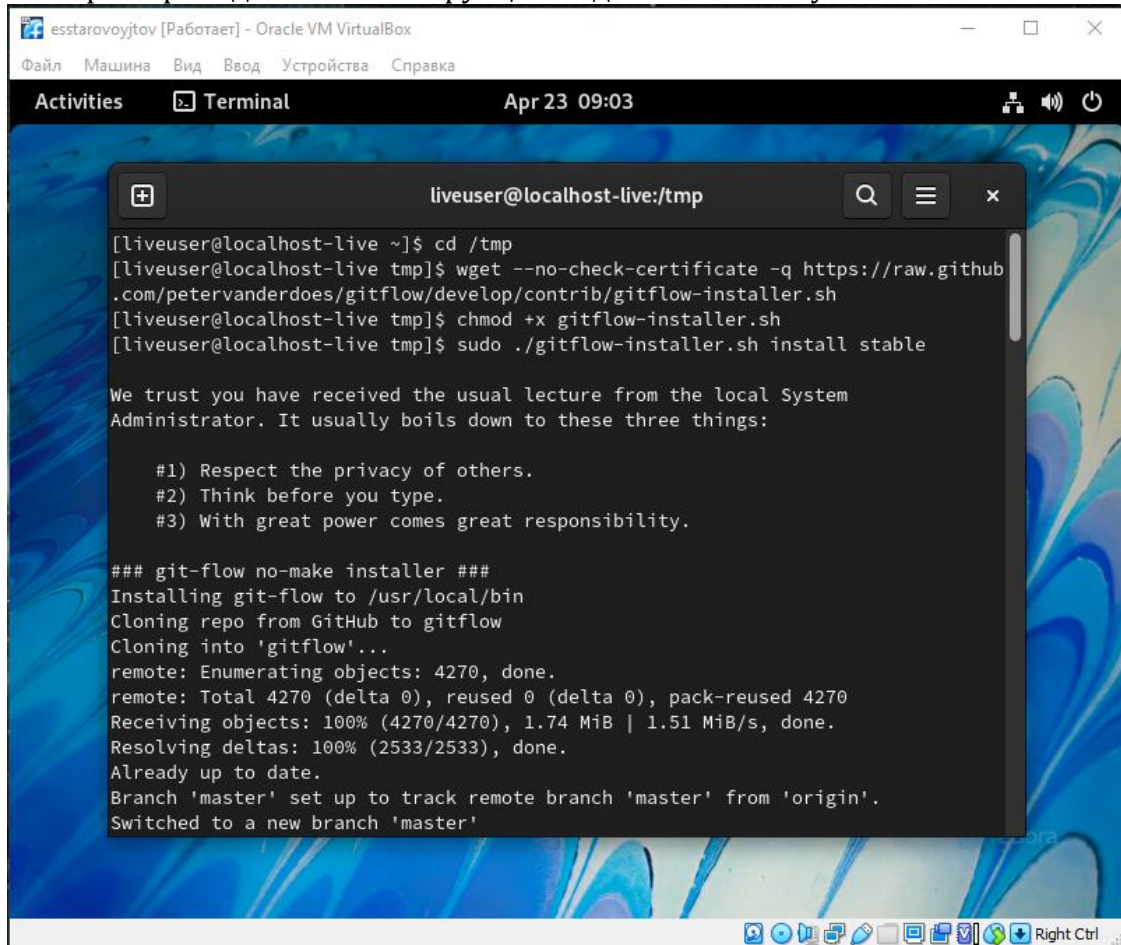


## Шаг 2. Установка программного обеспечения

### Шаг 2.1. Установка git-flow в Fedora Linux

– Это программное обеспечение удалено из репозитория. – Необходимо устанавливать его вручную.

Я набрал приведенный в инструкции код и вот что получилось:



The screenshot shows a terminal window titled "liveuser@localhost-live:/tmp" within an Oracle VM VirtualBox environment. The terminal displays the execution of a script to install git-flow. The script first clones a repository from GitHub and then installs the git-flow tool. The output shows the cloning process, object enumeration, and the successful installation of git-flow to the system's local bin directory. The terminal also displays a message from the local system administrator regarding system security and a list of three principles: respect privacy, think before typing, and responsibility with power.

```
[liveuser@localhost-live ~]$ cd /tmp
[liveuser@localhost-live tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
[liveuser@localhost-live tmp]$ chmod +x gitflow-installer.sh
[liveuser@localhost-live tmp]$ sudo ./gitflow-installer.sh install stable

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

### git-flow no-make installer ###
Installing git-flow to /usr/local/bin
Cloning repo from GitHub to gitflow
Cloning into 'gitflow'...
remote: Enumerating objects: 4270, done.
remote: Total 4270 (delta 0), reused 0 (delta 0), pack-reused 4270
Receiving objects: 100% (4270/4270), 1.74 MiB | 1.51 MiB/s, done.
Resolving deltas: 100% (2533/2533), done.
Already up to date.
Branch 'master' set up to track remote branch 'master' from 'origin'.
Switched to a new branch 'master'
```

## Шаг 2.2. Установка gh в Fedora Linux

Я набрал приведенный в инструкции код и вот что получилось:

```
[liveuser@localhost-live ~]$ sudo dnf install gh
Last metadata expiration check: 0:05:14 ago on Sat 23 Apr 2022 10:06:33 AM EDT.
Dependencies resolved.
=====
Package            Architecture      Version           Repository        Size
=====
Installing:
gh                 x86_64            2.7.0-1.fc35     updates           6.8 M
=====
Transaction Summary
=====
Install 1 Package

Total download size: 6.8 M
Installed size: 32 M
Is this ok [y/N]: y
Downloading Packages:
gh-2.7.0-1.fc35.x86_64.rpm                6.9 MB/s | 6.8 MB    00:00
-----
Total                                       4.1 MB/s | 6.8 MB    00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                :                               1/1
  Installing               : gh-2.7.0-1.fc35.x86_64       1/1
  Running scriptlet: gh-2.7.0-1.fc35.x86_64               1/1
  Verifying                : gh-2.7.0-1.fc35.x86_64       1/1

Installed:
gh-2.7.0-1.fc35.x86_64
```

## Шаг 3. Базовая настройка git

- Настроим utf-8 и зададим имя и email владельца репозитория:

1. `git config --global core.quotepath false`
2. `git config --global user.name "Egor Starovoyjtov"`
3. `git config --global user.email "prototyperrailgun@gmail.com"`

```
[liveuser@localhost-live tmp]$ git config --global core.quotepath false
[liveuser@localhost-live tmp]$ git config --global user.name "Egor Starovoyjtov"
[liveuser@localhost-live tmp]$ git config --global user.email "prototyperrailgun@gmail.com"
[liveuser@localhost-live tmp]$
```

- Настройте верификацию и подписание коммитов git.
- Зададим имя начальной ветки (будем называть её master):

```
[liveuser@localhost-live tmp]$ git config --global init.defaultBranch master
[liveuser@localhost-live tmp]$ git config --global core.autocrlf input
[liveuser@localhost-live tmp]$ git config --global core.safecrlf warn
[liveuser@localhost-live tmp]$
```

## Шаг 4. Создаем ключи ssh

- По алгоритму rsa с ключём размером 4096 бит: `ssh-keygen -t rsa -b 4096`
- По алгоритму ed25519: `ssh-keygen -t ed25519`

Я ввел перечисленные выше команды и получил следующий вывод. Ключи успешно

```
[liveuser@localhost-live tmp]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/liveuser/.ssh/id_rsa):
Created directory '/home/liveuser/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/liveuser/.ssh/id_rsa
Your public key has been saved in /home/liveuser/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:D0CF4AgNhF0NeIV1rum6ZmL3ioDAxBjCe3KwFyZaCw liveuser@localhost-live
The key's randomart image is:
+---[RSA 4096]-----+
|O=.o=B*oo          |
|=E+*o=oo          |
|.o*.o . .         |
|oo =   +          |
|..B . o S         |
|.o o . . o        |
|. . . . .         |
|   .o. .          |
|   ++o.o          |
+-----[SHA256]-----+
```

```
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/liveuser/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/liveuser/.ssh/id_ed25519
Your public key has been saved in /home/liveuser/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:v76qWHqZW5DsFaiqXab0KXLQKXlyjCdClqly+4bXBG4 liveuser@localhost-live
The key's randomart image is:
+---[ED25519 256]--+
|
| .
| . .
| .o . .
| ..oo.+ S
|=o++E..o .
|+0ooooooo. .
|B=B=o+=. .
|=BB=+.oo.o+.
+-----[SHA256]-----+
[liveuser@localhost-live tmp]$
```

Ключ ssh (ed25519)

## Шаг 5. Создаем ключ PGP

- Генерируем ключ: `gpg --full-generate-key`
- Из предложенных опций выбираем:
- тип RSA and RSA;

- размер 4096;
- выбираем срок действия; значение по умолчанию— 0 (срок действия не истекает никогда).
- GPG запросит личную информацию, которая сохранится в ключе:
- Имя (не менее 5 символов).
- Адрес электронной почты.
- При вводе email убедитесь, что он соответствует адресу, используемому на GitHub.
- Комментарий. Можно ввести что угодно или нажать клавишу ввода, чтобы оставить это поле пустым.

Я проследовал инструкции и создал PGP ключ.

```
[liveuser@localhost-live ~]$ gpg --list-secret-keys --keyid-format LONG
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
/home/liveuser/.gnupg/pubring.kbx
-----
sec  ed25519/8168E8CF1E2DDB31 2022-04-23 [SC]
     A7B8E7569EF18C4A3C5D98BA8168E8CF1E2DDB31
uid                          [ultimate] Egor Starovoyjtov <prototypemailgun@gmail.com>
ssb  cv25519/C79733092D8BFF52 2022-04-23 [E]
```

## Шаг 6. Добавляем PGP ключ в GitHub

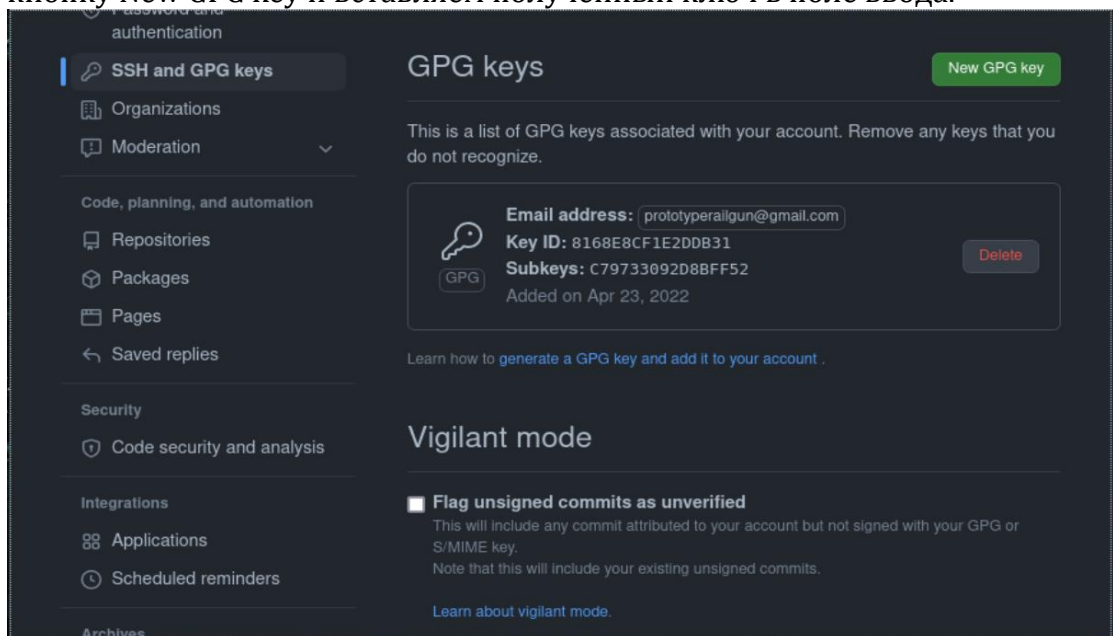
Скачиваем утилиту xclip:

```
liveuser@localhost-live:~
[liveuser@localhost-live ~]$ sudo dnf install xclip
Fedora 35 - x86_64 6.0 MB/s | 79 MB 00:13
Fedora 35 openh264 (From Cisco) - x86_64 3.7 kB/s | 2.5 kB 00:00
Fedora Modular 35 - x86_64 3.4 MB/s | 3.3 MB 00:00
Fedora 35 - x86_64 - Updates 5.2 MB/s | 29 MB 00:05
Fedora Modular 35 - x86_64 - Updates 2.3 MB/s | 2.9 MB 00:01
Dependencies resolved.
=====
Package Architecture Version Repository Size
=====
Installing:
xclip x86_64 0.13-15.git11cba61.fc35 fedora 36 k
Transaction Summary
=====
Install 1 Package

Total download size: 36 k
Installed size: 62 k
Is this ok [y/N]: y
Downloading Packages:
xclip-0.13-15.git11cba61.fc35.x86_64.rpm 757 kB/s | 36 kB 00:00
-----
Total 53 kB/s | 36 kB 00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
```

С помощью xclip копируем сгенерированный PGP ключ в буфер обмена: `gpg --armor --export <PGP Fingerprint> | xclip -sel clip`

Переходим в настройки GitHub (<https://github.com/settings/keys>), нажимаем на кнопку New GPG key и вставляем полученный ключ в поле ввода.



## Шаг 7. Настраиваем автоматическую подпись коммитов в git

Используя введенный email указываем git использовать его при подписи коммитов:

```
[liveuser@localhost-live ~]$ gpg --armor --export 8168E8CF1E2DDB31 | xclip -sel clip
[liveuser@localhost-live ~]$ git config --global user.signingkey 8168E8CF1E2DDB31
[liveuser@localhost-live ~]$ git config --global commit.gpgsign true
[liveuser@localhost-live ~]$ git config --global gpg.program $(which gpg2)
[liveuser@localhost-live ~]$
```

## Шаг 8. Настройка gh

1. Для начала авторизируемся: `gh auto login`.
2. Утилита задает наводящие вопросы, код подтверждения ввожу через браузер

```
[liveuser@localhost-live ~]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

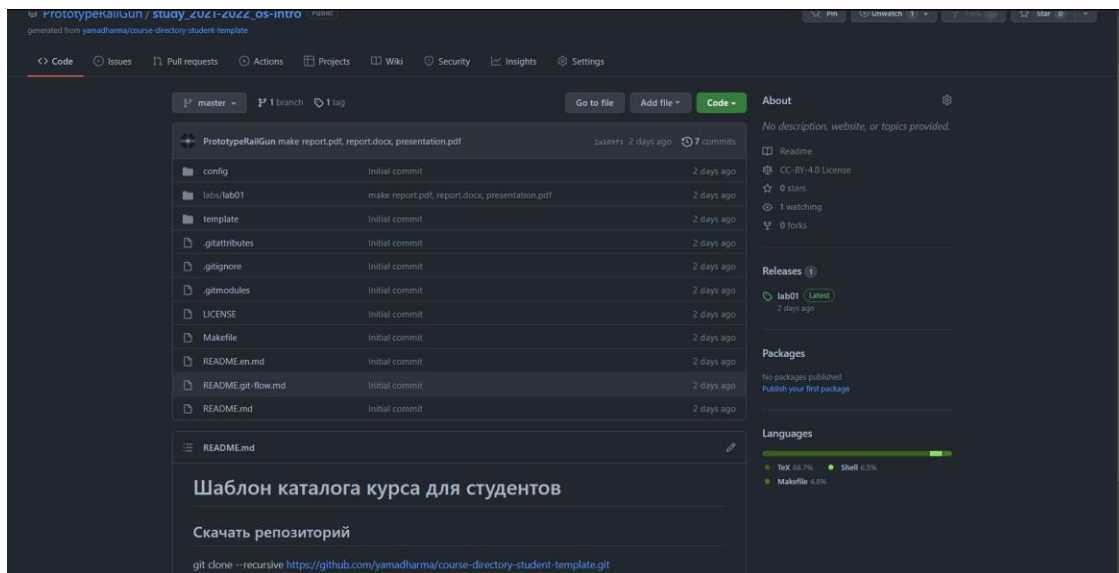
! First copy your one-time code: 4A6D-A6F4
Press Enter to open github.com in your browser...
✓ Authentication complete.
- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as PrototypeRailGun
[liveuser@localhost-live ~]$
```



## Шаг 9.

На этом этапе мне осталось только клонировать свой репозиторий, так как он был создан еще до начала выполнения первой лабораторной работы.

```
[liveuser@localhost-live ~]$ git clone https://github.com/PrototypeRailGun/study_2021-2022_os-intro
Cloning into 'study_2021-2022_os-intro'...
remote: Enumerating objects: 88, done.
remote: Counting objects: 100% (88/88), done.
remote: Compressing objects: 100% (71/71), done.
remote: Total 88 (delta 15), reused 75 (delta 7), pack-reused 0
Receiving objects: 100% (88/88), 35.71 MiB | 1.60 MiB/s, done.
Resolving deltas: 100% (15/15), done.
[liveuser@localhost-live ~]$
```



*Репозиторий*

## Вывод

Я изучил идеологию и применение средств контроля версий, а также освоил базовые навыки по работе с git и github. А именно: научился создавать ssh и pgr ключи, изучил базовые команды утилит git и gh, такие как: checkout, pull, status, diff, commit, push, config, clone и т.д.