

Лабораторная работа №2

Операционные системы

Мустафина А. Ю. НКАбд-03-24

7 марта 2025

Российский университет дружбы народов, Москва, Россия

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

1. Создать базовую конфигурацию для работы с git.
2. Создать ключ SSH.
3. Создать ключ PGP.
4. Настроить подписи git.
5. Зарегистрироваться на Github.
6. Создать локальный каталог для выполнения заданий по предмету.

Выполнение лабораторной работы

Устанавливаю программное обеспечение через терминал с помощью команд `dnf install git` и `dnf install gh` (рис. 1).

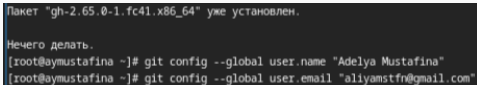
```
[aymustafina@aymustafina ~]$ dnf install git
Для выполнения запрошенной операции требуются привилегии суперпользователя. Пожалуйста, войдите в систему как пользователь с повышенными правами или используйте опции "--assumeno" или "--downloadonly", чтобы выполнить команду без изменения состояния системы.
[aymustafina@aymustafina ~]$ sudo -i
[sudo] пароль для аymustafина:
[root@aymustafina ~]# dnf install git
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет "git-2.48.1-1.fc41.x86_64" уже установлен.

Нечего делать.
[root@aymustafina ~]# dnf install gh
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет "gh-2.65.0-1.fc41.x86_64" уже установлен.

Нечего делать.
[root@aymustafina ~]#
```

Рис. 1: Установка

Задаю имя и email владельца репозитория с помощью команд: `git config --global user.name "Name Surname"` `git config --global user.email "work@mail"` (рис. 2).



```
Пакет "gh-2.65.0-1.fc41.x86_64" уже установлен.  
  
Нечего делать.  
[root@aymustafina ~]# git config --global user.name "Adelya Mustafina"  
[root@aymustafina ~]# git config --global user.email "aliyamstfn@gmail.com"
```

Рис. 2: Имя и email

Настроим utf-8 в выводе сообщений git: `git config --global core.quotepath false`

Зададим имя начальной ветки (будем называть её master): `git config --global init.defaultBranch master`

Параметр autocrlf: `git config --global core.autocrlf input`

Параметр safecrlf: `git config --global core.safecrlf warn` (рис. 3).

```
Нечего делать.  
[root@aymustafina ~]# git config --global user.name "Adelya Mustafina"  
[root@aymustafina ~]# git config --global user.email "aliyamstfr@gmail.com"  
[root@aymustafina ~]# git config --global core.quotepath false  
[root@aymustafina ~]# git config --global init.defaultBranch master  
[root@aymustafina ~]# git config --global core.autocrlf input  
[root@aymustafina ~]# git config --global core.safecrlf warn
```

Рис. 3: Базовая настройка

по алгоритму rsa с ключём размером 4096 бит: `ssh-keygen -t rsa -b 4096` (рис. 4).

```
[root@aymustafina ~]# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase for "/root/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:vMx4DQfaqVHM51EUfifXNMB5ikzh+2C6oj1GzLi2XwE root@aymustafina
The key's randomart image is:
+----[RSA 4096]-----+
|           o*oo..|
|          o  .+ o oo|
|         E= .o = +|
|        =. = +.o + |
|       * S.o+   |
|      . X =+ o   |
|      = =o. .   |
|      o.=. .    |
|     .o=oo.     |
+-----[SHA256]-----+
[root@aymustafina ~]#
```

Рис. 4: Создание ключа ssh по rsa

по алгоритму ed25519: `ssh-keygen -t ed25519` (рис. 5).

```
[root@aymustafina ~]# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase for "/root/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:FnVwLA4VGS7Q0S6Q5ZXAHV9bDzPiR2PHFKewf0vGACc root@aymustafina
The key's randomart image is:
+--[ED25519 256]--+
|      .+*0E=.0+*|
|    oo++*==B X+|
|      .+...+.+ |
|      .oo oo |
|      S.  .+.|
|      .  o..|
|      .  |
|      |
+-----[SHA256]-----+
[root@aymustafina ~]#
```

Рис. 5: Создание ключа ssh по ed25519

Создание ключа pgp

Генерирую ключ `gpg --full-generate-key`

Из предложенных опций выбираю: тип RSA and RSA; размер 4096; выбрала срок действия; значение по умолчанию — 0 (срок действия не истекает никогда). GPG запросит личную информацию, которая сохранится в ключе: Имя. Адрес электронной почты. При вводе email убеждаюсь, что он соответствует адресу, используемому на GitHub. Комментарий. Нажимаю клавишу ввода, чтобы оставить это поле пустым. (рис. 6).

```
[root@aymustafina ~]# gpg --full-generate-key
gpg (GnuPG) 2.4.5; Copyright (C) 2024 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 '/root/.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.
```

Ввожу фразу-пароль для защиты ключа (рис. 7).

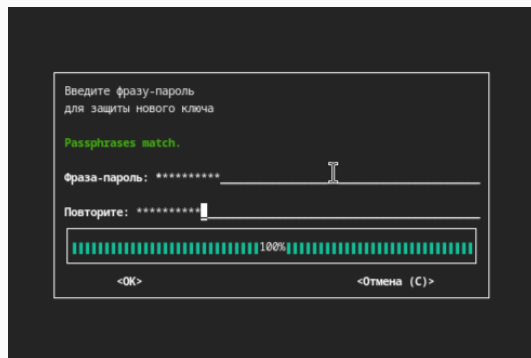


Рис. 7: Фраза-пароль

Открытый и секретный ключ созданы (рис. 8).

```
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: /root/.gnupg/trustdb.gpg: создана таблица доверия
gpg: создан каталог '/root/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/root/.gnupg/openpgp-revocs.d/D751C72AB78BA87B691E
44ED7D8ACD846906604E.rev'.
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2025-03-02 [SC]
       D751C72AB78BA87B691E44ED7D8ACD846906604E
uid           Adelya Mustafina <aliyamstfn@gmail.com>
sub    rsa4096 2025-03-02 [E]

[root@aymustafina ~]#
```

Рис. 8: Создание ключей

У меня уже был создан аккаунт на github, основные данные аккаунта заполнены (рис. 9).

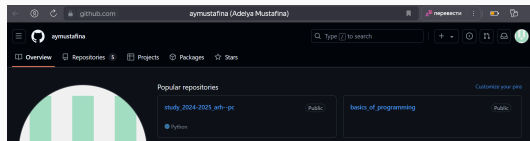


Рис. 9: Настройка github

Добавление PGP ключа в GitHub

Вывожу список ключей и копирую отпечаток приватного ключа: `gpg --list-secret-keys --keyid-format LONG`

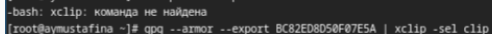
Отпечаток ключа — это последовательность байтов, используемая для идентификации более длинного, по сравнению с самим отпечатком (рис. 10).

```
[root@aymustafina ~]# gpg --list-secret-keys --keyid-format LONG
gpg: проверка таблицы доверия
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: глубина: 0 достоверных: 2 подписанных: 0 доверие: 0-,
  0q, 0n, 0m, 0f, 2u
[keyboxd]
-----
sec   rsa4096/BC82ED8D50F07E5A 2025-03-02 [SC]
      ACDA5D9B96A8FC41F8D33B28BC82ED8D50F07E5A
uid           [ абсолютно ] Adelya Mustafina <aliyamstfn@gmail.com>
ssb   rsa4096/1251AB79DF88E459 2025-03-02 [E]
```

Рис. 10: Вывод ключей

Формат строки:

сес Алгоритм/Отпечаток_ключа Дата_создания [Флаги] [Годен_до] ID_ключа Копирую
сгенерированный PGP ключ в буфер обмена: `gpg --armor --export | xclip -sel cli` (рис. 11).



```
-bash: xclip: команда не найдена  
[root@aymustafina ~]# gpg --armor --export BC82ED8D50F07E5A | xclip -sel clip
```

Рис. 11: Копирование ключей

Добавление PGP ключа в GitHub

Перехожу в настройки Github, нажимаю кнопку New GPG key и вставляю полученный ключ в поле ввода (рис. 12).

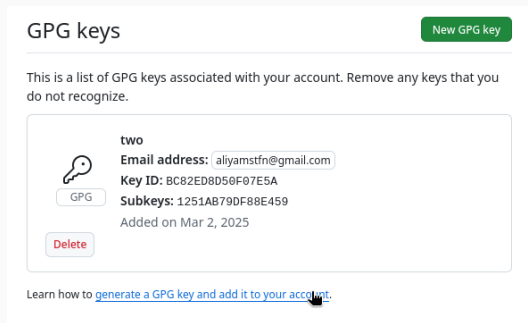


Рис. 12: New GPG key

Настройка автоматических подписей коммитов git

Используя введённый email, укажите Git применять его при подписи коммитов:

`git config --global user.signingkey git config --global commit.gpgsign true git config --global
gpg.program $(which gpg2)` (рис. 13).

```
[root@aymustafina ~]# git config --global user.signingkey BC82ED8D50F07E5A  
[root@aymustafina ~]# git config --global commit.gpgsign true  
[root@aymustafina ~]# git config --global gpg.program $(which gpg2)  
[root@aymustafina ~]# gh auth login
```

Рис. 13: Настройка подписей git

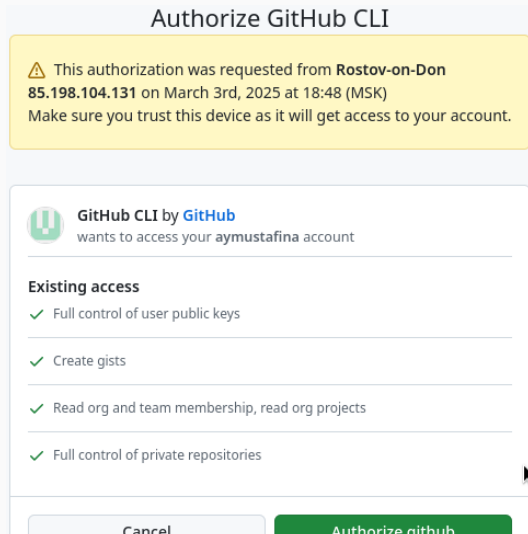
Сначала авторизуюсь в gh, отвечаю на наводящие вопросы, в конце выбираю авторизацию через браузер (рис. 14).

```
[aymustafina@aymustafina ~]$ gh auth status
You are not logged into any GitHub hosts. To log in, run: gh auth login
[aymustafina@aymustafina ~]$ sudo -i
[sudo] пароль для аymustafina:
[root@aymustafina ~]# ls ~/.ssh/id_ed25519.pub
/root/.ssh/id_ed25519.pub
[root@aymustafina ~]# cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIIP6MSzUHKW7vFXhuz5sS2QVjybWxA8AB+B1zZa9+1SY root@aymustafina
[root@aymustafina ~]# ssh -T git@github.com
Hi aymustafina! You've successfully authenticated, but GitHub does not provide shell access.
[root@aymustafina ~]# gh auth login --web
? What is your preferred protocol for Git operations on this host? SSH
? Upload your SSH public key to your GitHub account? /root/.ssh/id_ed25519.pub
? Title for your SSH key: new

! First copy your one-time code: 3A34-99D3
Press Enter to open https://github.com/login/device in your browser... []
```

Рис. 14: Авторизация в gh

Завершаю авторизацию на сайте (рис. 15).



Вижу сообщение о завершении авторизации на сайте (рис. 16).

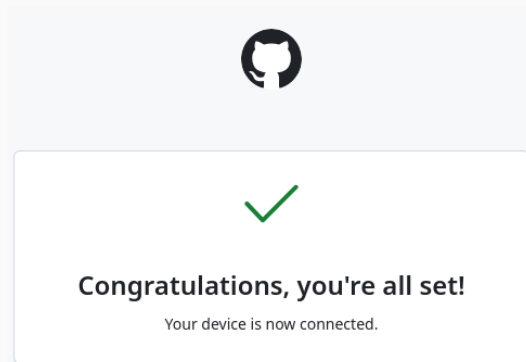


Рис. 16: Завершение авторизации

Создание репозитория курса на основе шаблона

Сначала создаю директорию с помощью утилиты `mkdir` и с помощью `cd` перехожу в созданную директорию:

```
mkdir -p ~/work/study/2024-2025/"Операционные системы" cd  
~/work/study/2024-2025/"Операционные системы" gh repo create study_2024-2025_os-intro  
-template=yamadharm/course-directory-student-template --public git clone --recursive  
git@github.com:/study_2024-2025_os-intro.git os-intro (рис. 18).
```

```
[root@ymustafina ~]# mkdir -p ~/work/study/2024-2025/"Операционные системы"  
[root@ymustafina ~]# cd ~/work/study/2024-2025/"Операционные системы"  
[root@ymustafina Операционные системы]# gh repo create study_2024-2025_os-intro --template=yamadharm/course-directory-student-template --public  
Created repository ymustafina/study_2024-2025_os-intro on GitHub  
https://github.com/ymustafina/study_2024-2025_os-intro  
[root@ymustafina Операционные системы]# gh repo create study_2024-2025_os-intro --template=yamadharm/course-directory-student-template --public  
Created repository ymustafina/study_2024-2025_os-intro on GitHub  
https://github.com/ymustafina/study_2024-2025_os-intro
```

Рис. 17: Создание репозитория курса

Клонирование репозитория (рис. 19).

```
[root@ymustafina Операционные системы]# git clone --recursive git@github.com:ymustafina/study_2024-2025_os-intro.git os-intro  
Клонирование в os-intro  
remote: Enumerating objects: 36, done.
```

Выводы

При выполнении данной лабораторной работы я изучила идеологию и применение средств контроля версий, изучила команды для работы с git.

1. Лабораторная работа №2