

Лабораторная работа №2

Первоначальная настройка git

Мухин Тимофей Владимирович

Содержание

1	Цель работы	3
2	Выполнение лабораторной работы	4
3	Выводы	7

1 Цель работы

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

2 Выполнение лабораторной работы

1. Установим git и gh из репозитория

```
tvmukhin@homepc:~$ sudo dnf install git gh
```

2. Базовая настройка git. Зададим имя и email владельца репозитория

```
tvmukhin@homepc:~$ git config --global user.name "Timofey"
tvmukhin@homepc:~$ git config --global user.email "video2244@gmail.com"
```

3. Настроим верификацию и подписание комитов, зададим имя начальной ветки, параметры autocrlf и safecrlf

```
tvmukhin@homepc:~$ git config --global core.quotepath false
tvmukhin@homepc:~$ git config --global init.defaultBranch master
tvmukhin@homepc:~$ git config --global core.autocrlf input
tvmukhin@homepc:~$ git config --global core.safecrlf warn
tvmukhin@homepc:~$
```

4. Создаем ключи ssh rsa и ed25519

```
tvmukhin@homepc:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/tvmukhin/.ssh/id_ed25519):
```

5. Создаем ключи gpg и добавляем ключ в github

```

tvmukhin@homepc:~$ 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/tvmukhin/.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)

```

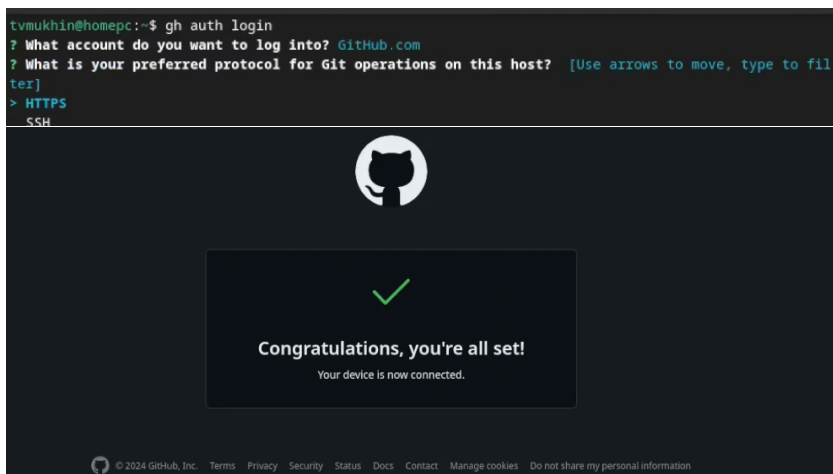
6. Используя введенный email, укажем Git применять его при подписи коммитов

```

tvmukhin@homepc:~$ git config --global user.signingkey
tvmukhin@homepc:~$ git config --global user.signingkey key1
tvmukhin@homepc:~$ git config --global commit.gpgsign true
tvmukhin@homepc:~$

```

7. Настройка gh. Авторизуемся.



8. Создание репозитория курса на основе шаблона. Создаем директорию.

```
tvmukhin@homepc:~$ mkdir -p ~/work/study/2023-2024/Операционные_системы/os-intro
tvmukhin@homepc:~$

tvmukhin@homepc:~$ cd ~/work/study/2023-2024/Операционные_системы/os-intro/
tvmukhin@homepc:~/work/study/2023-2024/Операционные_системы/os-intro$ gh repo create study_2023-2024_
_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository Seychik1/study_2023-2024_os-intro on GitHub
https://github.com/Seychik1/study_2023-2024_os-intro
tvmukhin@homepc:~/work/study/2023-2024/Операционные_системы/os-intro$ git clone
```

9. Клонировуем репозиторий на компьютер

```
и репозиторий существует.
tvmukhin@homepc:~/work/study/2023-2024/Операционные_системы/os-intro$ git clone --recurse https://gi
thub.com/Seychik1/study_2023-2024_os-intro.git os-intro
Клонирование в «os-intro»...
```

10. Настройка каталога курса. Удаляем лишние файлы, создаем необходимые каталоги

```
os-intro
tvmukhin@homepc:~/work/study/2023-2024/Операционные_системы/os-intro$ cd os-intro/
tvmukhin@homepc:~/work/study/2023-2024/Операционные_системы/os-intro/os-intro$ ls
CHANGELOG.md  COURSE  Makefile  README.en.md  README.md
config        LICENSE  package.json  README.git-flow.md  template
tvmukhin@homepc:~/work/study/2023-2024/Операционные_системы/os-intro/os-intro$ rm package.json
tvmukhin@homepc:~/work/study/2023-2024/Операционные_системы/os-intro/os-intro$ echo os-intro > CO
E
tvmukhin@homepc:~/work/study/2023-2024/Операционные_системы/os-intro/os-intro$ make
Usage:
  make <target>

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

11. Делаем коммит и отправляем данные на сервер

```
tvmukhin@homepc:~/work/study/2023-2024/Операционные_системы/os-intro$ git commit -am 'feat(main): make course
structure'

tvmukhin@homepc:~/work/study/2023-2024/Операционные_системы/os-intro$ git push
```

3 Выводы

В ходе выполнения работы было изучено применение и работа системы контроля версий git