

Лабораторная работа № 2

Первоначальная настройка git

Мальянц Виктория Кареновна

Содержание

| | | |
|----------|--|-----------|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Выполнение лабораторной работы | 7 |
| 3.1 | Установка программного обеспечения | 7 |
| 3.2 | Базовая настройка git | 7 |
| 3.3 | Создание ключа ssh | 8 |
| 3.4 | Создание ключа pgr | 9 |
| 3.5 | Настройка github | 10 |
| 3.6 | Добавление pgr ключа в GitHub | 11 |
| 3.7 | Настройка автоматических подписей коммитов git | 12 |
| 3.8 | Настройка gh | 13 |
| 3.9 | Создание репозитория курса на основе шаблона | 13 |
| 3.10 | Настройка каталога курса | 14 |
| 4 | Выводы | 17 |
| 5 | Контрольные вопросы | 18 |
| | Список литературы | 20 |

Список иллюстраций

| | | |
|------|--|----|
| 3.1 | Переключение на роль супер-пользователя с помощью <code>sudo -i</code> . . . | 7 |
| 3.2 | Установка <code>git</code> | 7 |
| 3.3 | Установка <code>gh</code> | 7 |
| 3.4 | Имя и email владельца репозитория | 7 |
| 3.5 | Настройка <code>utf-8</code> в выводе сообщений <code>git</code> | 8 |
| 3.6 | Имя начальной ветки | 8 |
| 3.7 | Параметр <code>autocrlf</code> | 8 |
| 3.8 | Параметр <code>safecrlf</code> | 8 |
| 3.9 | Создание ключа <code>ssh</code> по алгоритму <code>rsa</code> с ключом размером 4096 бит | 8 |
| 3.10 | Создание ключа <code>ssh</code> по алгоритму <code>ed25519</code> | 9 |
| 3.11 | Фраза-пароль | 9 |
| 3.12 | Генерация ключа <code>gpg</code> | 10 |
| 3.13 | Профиль <code>GitHub</code> | 10 |
| 3.14 | Список ключей | 11 |
| 3.15 | Копирование сгенерированного ключа <code>gpg</code> | 11 |
| 3.16 | Создание <code>gpg</code> ключа | 12 |
| 3.17 | Ключ <code>gpg</code> создан | 12 |
| 3.18 | Настройка для применения подписей коммитов | 13 |
| 3.19 | Авторизация в <code>gh</code> | 13 |
| 3.20 | Создание каталога и перемещение в него | 13 |
| 3.21 | Создание репозитория на основе шаблона | 14 |
| 3.22 | Переход в каталог курса | 14 |
| 3.23 | Удаление лишних файлов | 14 |
| 3.24 | Создание необходимых каталогов | 14 |
| 3.25 | Фраза-пароль | 15 |
| 3.26 | Создание каталогов и отправление файлов на сервер | 15 |
| 3.27 | Отправление файлов на сервер | 15 |
| 3.28 | Каталоги в <code>GitHub</code> | 16 |

Список таблиц

1 Цель работы

Изучить идеологию и применение средств контроля версий git

2 Задание

1. Установка программного обеспечения
2. Базовая настройка git
3. Создание ключа ssh
4. Создание ключа pgr
5. Настройка github
6. Добавление pgr ключа в GitHub
7. Настройка автоматических подписей коммитов git
8. Настройка gh
9. Создание репозитория курса на основе шаблона
10. Настройка каталога курса
11. Контрольные вопросы

3 Выполнение лабораторной работы

3.1 Установка программного обеспечения

Переключаюсь на роль супер-пользователя с помощью `sudo -i` (рис. 3.1).

```
[vkmajyanc@vkmajyanc ~]$ sudo -i
[sudo] пароль для vkmajyanc:
[root@vkmajyanc ~]#
```

Рис. 3.1: Переключение на роль супер-пользователя с помощью `sudo -i`

Устанавливаю `git` (рис. 3.2).

```
[root@vkmajyanc ~]# dnf install git
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет "git-2.48.1-1.fc41.x86_64" уже установлен.
Нечего делать.
[root@vkmajyanc ~]#
```

Рис. 3.2: Установка `git`

Устанавливаю `gh` (рис. 3.3).

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

Рис. 3.3: Установка `gh`

3.2 Базовая настройка `git`

Задаю имя и email владельца репозитория (рис. 3.4).

```
[root@vkmajyanc ~]# git config --global user.name "Victoria Malyants"
[root@vkmajyanc ~]# git config --global user.email "1132245740@pfur.ru"
```

Рис. 3.4: Имя и email владельца репозитория

Настраиваю utf-8 в выводе сообщений git (рис. 3.5).

```
[root@kmaljanc ~]# git config --global core.quotePath false
```

Рис. 3.5: Настройка utf-8 в выводе сообщений git

Задаю имя начальной ветки (назову ее master) (рис. 3.6).

```
[root@kmaljanc ~]# git config --global init.defaultBranch master
```

Рис. 3.6: Имя начальной ветки

Параметр autocrlf (рис. 3.7).

```
[root@kmaljanc ~]# git config --global core.autocrlf input
```

Рис. 3.7: Параметр autocrlf

Параметр safecrlf (рис. 3.8).

```
[root@kmaljanc ~]# git config --global core.safecrlf warn
```

Рис. 3.8: Параметр safecrlf

3.3 Создание ключа ssh

Создание ключа ssh по алгоритму rsa с ключем размером 4096 бит (рис. 3.9).

```
[root@kmaljanc ~]# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)?
[root@kmaljanc ~]# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
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:SPXINrR6cs/aAh5L9TLcuuMqk3jp4FzWjQRhVeZK4 root@kmaljanc
The key's randomart image is:
+----[RSA 4096]-----+
|
| o +
| . . . .
| o . . o * o
| : o E . S * .
| o * o o . + .
| = . + +00
| = + . +0 * .
| oB+o+ .+o...
+-----[SHA256]-----+
[root@kmaljanc ~]#
```

Рис. 3.9: Создание ключа ssh по алгоритму rsa с ключем размером 4096 бит

Создание ключа ssh по алгоритму ed25519 (рис. 3.10).


```
[root@kmaljanc ~]# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
/root/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
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:W07X0zmg1A3N/ENieKqacQ=nlUdK1gW/UCaZlCsAw root@kmaljanc
The key's randomart image is:
+--[ED25519 256]--+
|E
|oo...oB+
|oo=.o..++B|
|oB..o..+o o|
|o-o o S +o |
|...o =. . . |
| .B. o |
| ..* . |
| +o |
+-----[SHA256]-----+
[root@kmaljanc ~]#
```

Рис. 3.10: Создание ключа ssh по алгоритму ed25519

3.4 Создание ключа prg

Генерирую ключ prg, поэтому ввожу фразу-пароль для его защиты (рис. 3.11).

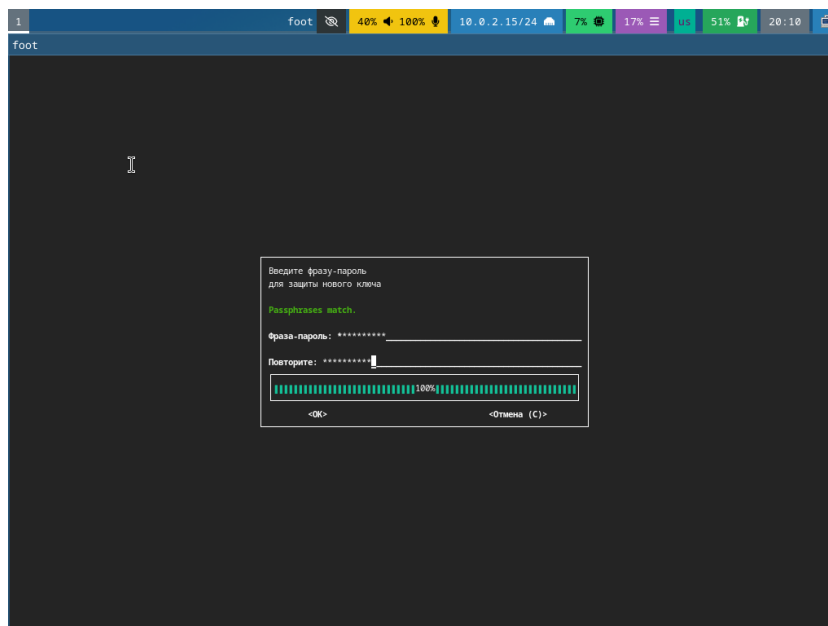


Рис. 3.11: Фраза-пароль

Генерирую ключ prg, из предложенных опций выбираю: тип RSA and RSA, размер 4096, 0 (срок действия не истекает никогда) (рис. 3.12).

```
[root@vmalyants ~]# 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.

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
<nw> = срок действия ключа - n недель
<nm> = срок действия ключа - n месяцев
<ny> = срок действия ключа - n лет
Срок действия ключа? (0) 0
Срок действия ключа не ограничен
Все верно? (y/N) y

GnuPG должен составить идентификатор пользователя для идентификации ключа.
Ваше полное имя: MalyantsVictoria
Адрес электронной почты: 1132246740@pfur.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
"Victoria" <1132246740@pfur.ru>

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? 0
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печатать
на клавиатуре, движение мыши, обращение к диску); 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 (печатать
на клавиатуре, движение мыши, обращение к диску); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
```

Рис. 3.12: Генерация ключа gpg

3.5 Настройка github

Профиль на GitHub был создан и настроен ранее (рис. 3.13).

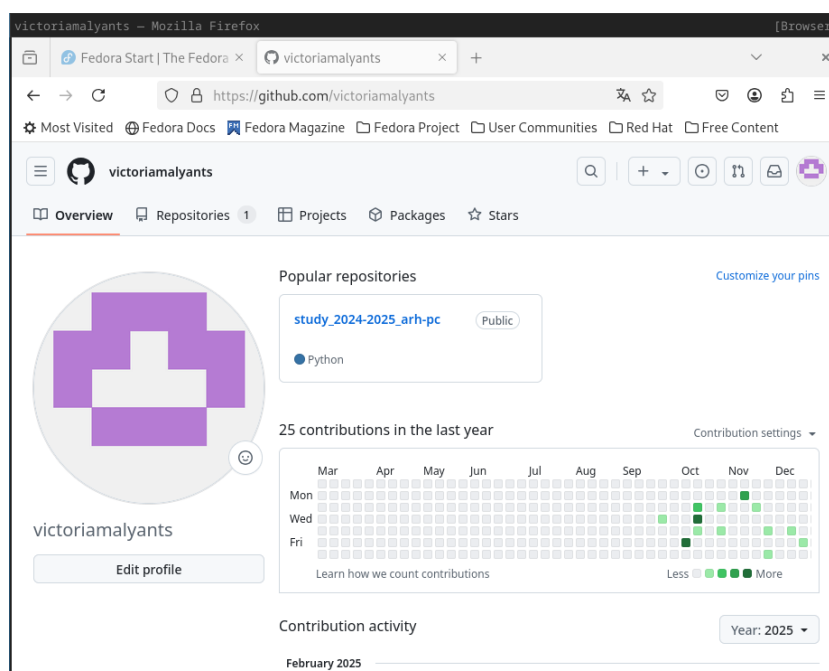


Рис. 3.13: Профиль GitHub

3.6 Добавление pgp ключа в GitHub

Вывожу список ключей и копирую отпечаток приватного ключа (рис. 3.14).

```
[root@vkmaljjanc ~]# 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: 3 подписанных: 0 доверие: 0-, 0q, 0n, 0f, 3u
[keyboard]
-----
sec   rsa4096/5818ECA509026955 2025-02-25 [SC]
      03F3A18EBCADACF361EF698F5818ECA509826955
uid   [ абсолютно ] MalyantsVictoria <1132246740@pfur.ru>
ssb   rsa4096/9ADE6DE178AA4491 2025-02-25 [E]

sec   rsa4096/2DBF2DB279C10DC3 2025-02-26 [SC]
      532A0CB880DB891814E9269A2DBF2DB279C10DC3
uid   [ абсолютно ] MalyantsVictoria <1132246740@pfur.ru>
ssb   rsa4096/AD2A46E9AD97B6FA 2025-02-26 [E]

sec   rsa4096/87F51413323C7EB0 2025-02-26 [SC]
      563FA7F7909700653E43085B87F51413323C7EB0
uid   [ абсолютно ] MalyantsVictoria <1132246740@pfur.ru>
ssb   rsa4096/6990692170C1597 2025-02-26 [E]

[root@vkmaljjanc ~]#
```

Рис. 3.14: Список ключей

Копирую сгенерированный ключ gpg (рис. 3.15).

```
[root@vkmaljjanc ~]# gpg --armor --export 563FA7F7909700653E43085B87F51413323C7EB0
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGe+t1cBEACHXfuBNMSImFVfFOX+XcH7Yp+qmN149a1+sohP/V8quLqMJN7a
Cc8WckITBk2R1+mrKgtJ74deQ9dM1rG1ceZHQvs0pxMkbZs61c1yZx64focYFax1
oCso/P9pQZ1ZHKH2DthF14Ips6duYw11xDc73gzUZXpFw0EYcdeZhm+2Q2Rq1iik
xhE292t8pjMGP+/LM8tQsG+4xF/WCT8HG6J16+bY5/WyzF7eNyAWh1wNOTM0Q2KF
qd+Yf35mKYp1+650BxL241RaEk8vpwyXcHCUGL7W3o2gh6E0h2znj91jTc1bsaCa
zLQIWA/Ew1rkyoNb0wtr1ZaVhv4wMIqBKQ1SGegLJBd9HjTdgw8XQJqu0EMGsW3v
pratD6YaInf7uJeZr60+KzLEnv+PMc+Wdm2XB453fzh0H5U2bvnAyME4ZQM5G71
ewJkmACZ9H2Ep/V2cTuZg019LFouQYXo6P8oIC22/zsqXT1Q6Dfmmv1ibq1NA4
O3VUCbIXEnqDjWBQnKQTT0C10JEZ0wOX18b1N53JbuGoBgYXV1K19s5+aNXctUK
Kj1M1suAuKFBW+p3P60aHABjWf3Xvv0/+2coyaR1mYvDgSLPV1oIsC0xoZXHLJc
Um7R2UCvq/obNMF+1R/s1V5B3FYz8bNFSeMTm0+1utK8ELZ1854uBVBMWwARAQAB
tCVNYWxSYW50c1ZpY3RvcmlhIDwvMTYyMjQyMjQyMjQyMjQyMjQyMjQyMjQyMjQy
FjEEVj+n95CX0G0+QwtbH/UUEz18frAFame+t1cGwMFcWk1Bw1C1g1GFQpJCA5C
BBYCAwEChgcCF4AAGCqQh/UUEz18frDNaq//UJ76RdBy+PEKQeDtszeyHq/3hYe5
5BVbVc2RnW1o4Xf0cb1+FKUXsqPuj1yXf5q4eqDVEK2t0ZfTwpRVegSty6JRWh
z2LSVze17sB5D7L7y6CYf0XnCAqAxi1xU2UvWNUH5uQcXN4NjK80zes1Lmhd6p
DocJe+fn139dYLP3uQ8FfgetiM38sT821pjVmeNhcZ9Nq0vohukN5QqzrGnTrpw
AxNfeyTciVrrkduEKPC0vzpXbMviLZk8hx6n94dG67+FGpq/W3nREY3tQtXku49c
jcSj1mnTqYhD7GwrdUeb9eSE3FhmUoDU/9wFhuUmiH6se/zVj18OGyzo1WT1K
+9h/7f9eZrJoZPZ6q0f2ArR6kbCwnMZ7t2HjeCtjg0DsSntNZV71cwiNH9pRZp6y
e903K10sroSn3V0qGtjBkP2jxiWnzyWNAUwPaTPYmVbGaGYHsgX516hokSL1Z+
RrEZd147UKImrqrwnahg0n47fokkkdMhScyRkn0Sy/e8H2uk/SHMVG6V25rSgA1
oq1tpChc+HP2heuaUdm+wx66DLVBjHx520TirdfcQw5q9TWKvRN04vG20TK2QeQR
U03dw1gjl7Uy1DTcCw4txEYS/ef8vsu4Wb+7cBM/JbJGenL2wc71D09F1XvYmz/Z
984mcAsyqBuyHhW5Ag0E2763VwEQAMH5IK0I80ciW1Zy9gJabUZIc21qo9fXnAj
XXtcFyyxfmC0tZKfCwL4YTaVRTIZG3XhzTrQawP/9h0dm2DLxkZRBwPjxHbvHJR
jig7R0WU1Bx17Re0Eb0kUxJM1j4Tyc+11Vv1xQ54CoH8h4SV1Vqubs5RjCYqz16c
KddZYEYt4nkdxTm1+TnhRTdIwJhgdknZdkCTNg665JBNLEgoB1YL8banqIhteSP
jPnfduoj211f211YxUPL3dP5Tn9pQR15y6RQKnRjjaizpH8a35An4901EpVgAvDvi
uqh18/XDGx8JhBADmE/+CwL0DzhW5c1Ky0bux28TtQJGaToKNDU0p4sIVr66u1ym
Wf88A6n3LjyfuUuUQ9gGDSF3wCHMeY2dzL3B8SD1W0ipXba+FA7Gnwwh1E1RL
LReUgKke9FYx6mnxZECj+hcgBe3E2gRo1LRXyWP19z3Wmq/P6TXBu19HRE9P5o/+
NNGUHT9Z1rfwdybB1f1W89umWYUjZ/yYFMW/9dkSwdeGs5IMrJcXVMsqENYKF
```

Рис. 3.15: Копирование сгенерированного ключа gpg

Перехожу в настройки GitHub, нажимаю кнопку New GPG key и вставляю полу-
ченный ключ в поле ввода (рис. 3.16).

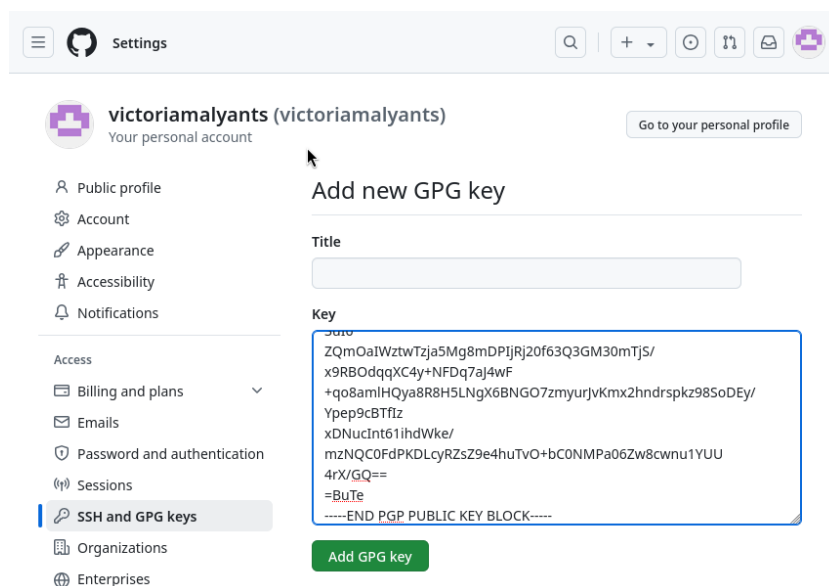


Рис. 3.16: Создание gpg ключа

Ключ gpg создан (рис. 3.17).

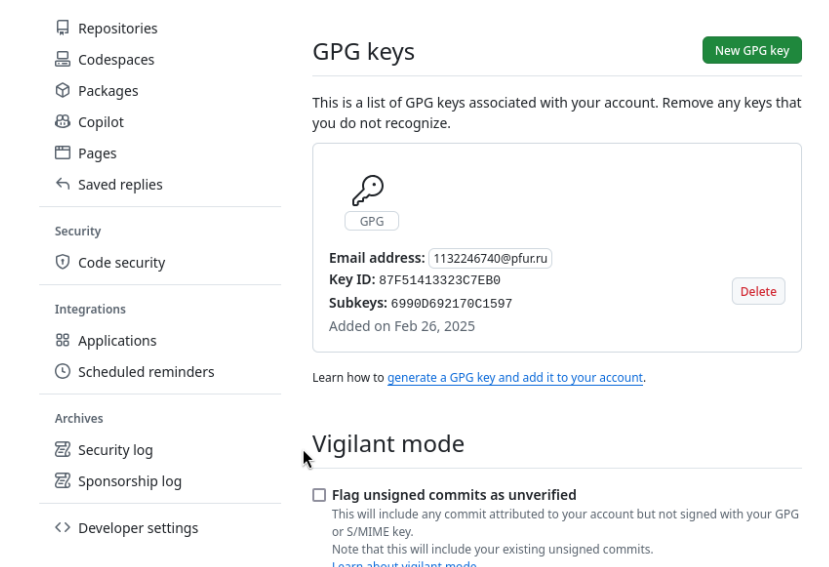


Рис. 3.17: Ключ gpg создан

3.7 Настройка автоматических подписей коммитов git

Используя введенный email, указываю Git применять его при подписи коммитов (рис. 3.18).

```
[root@vkmaljanec ~]# git config --global user.signingkey 87F51413323C7EB0
[root@vkmaljanec ~]# git config --global commit.gpgsign true
[root@vkmaljanec ~]# git config --global gpg.program $(which gpg2)
[root@vkmaljanec ~]#
```

Рис. 3.18: Настройка для применения подписей коммитов

3.8 Настройка gh

Авторизуюсь в gh, отвечаю на наводящие вопросы утилиты (рис. 3.19).

```
[root@vkmaljanec ~]# gh auth login
? Where do you use GitHub? GitHub.com
? 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: GitHub CLI
? How would you like to authenticate GitHub CLI? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'admin:public_key'.
? Paste your authentication token: *****
- gh config set -h github.com git_protocol ssh
- Configured git protocol
- Authentication credentials saved in plain text
- SSH key already existed on your GitHub account: /root/.ssh/id_ed25519.pub
- Logged in as victoriama1yants
[root@vkmaljanec ~]# gh config set -h github.com git_protocol ssh
```

Рис. 3.19: Авторизация в gh

3.9 Создание репозитория курса на основе шаблона

Создаю каталог ~/work/study/2024-2025/“Операционные системы” и перемещаюсь в него (рис. 3.20).

```
[root@vkmaljanec ~]# mkdir -p ~/work/study/2024-2025/“Операционные системы”
[root@vkmaljanec ~]# cd ~/work/study/2024-2025/“Операционные системы”
```

Рис. 3.20: Создание каталога и перемещение в него

Создаю репозиторий на основе шаблона (рис. 3.21).

```
[root@vkmal]# gh repo create study_2024-2025-os-intro --template=yamadharma/course-directory-student-template --public
[root@vkmal]# git clone --recursive git@github.com:victoriamalyants/study_2024-2025-os-intro.git os-intro
Клонирование в «os-intro»...
The authenticity of host 'github.com (148.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+D1Y3wvV6TujJhbpZisF/zLD48zPM5vHokr4UvCoQU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? no/C
[root@vkmal]# git clone --recursive git@github.com:victoriamalyants/study_2024-2025-os-intro.git os-intro
Клонирование в «os-intro»...
The authenticity of host 'github.com (148.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+D1Y3wvV6TujJhbpZisF/zLD48zPM5vHokr4UvCoQU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 36, done.
remote: Counting objects: 100% (36/36), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 36 (delta 1), reused 21 (delta 0), pack-reused 0 (from 0)
Получение объектов: 100% (36/36), 19.38 Киб | 4.84 Мб/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/root/work/study/2024-2025/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 111, done.
remote: Counting objects: 100% (111/111), done.
remote: Compressing objects: 100% (77/77), done.
remote: Total 111 (delta 42), reused 100 (delta 31), pack-reused 0 (from 0)
Получение объектов: 100% (111/111), 102.17 Киб | 138.00 Киб/с, готово.
Определение изменений: 100% (42/42), готово.
Клонирование в «/root/work/study/2024-2025/Операционные системы/os-intro/template/report»...
remote: Enumerating objects: 142, done.
remote: Counting objects: 100% (142/142), done.
remote: Compressing objects: 100% (97/97), done.
remote: Total 142 (delta 60), reused 121 (delta 39), pack-reused 0 (from 0)
Получение объектов: 100% (142/142), 341.09 Киб | 250.00 Киб/с, готово.
Определение изменений: 100% (60/60), готово.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fcd4d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
[root@vkmal]#
```

Рис. 3.21: Создание репозитория на основе шаблона

3.10 Настройка каталога курса

Перехожу в каталог курса `~/work/study/2024-2025/“Операционные системы”/os-intro` (рис. 3.22).

```
[root@vkmal]# cd ~/work/study/2024-2025/“Операционные системы”/os-intro
[root@vkmal]#
```

Рис. 3.22: Переход в каталог курса

Удаляю лишние файлы (рис. 3.23).

```
[root@vkmal]# rm package.json
rm: удалить обычный файл 'package.json'?
[root@vkmal]#
```

Рис. 3.23: Удаление лишних файлов

Создаю необходимые каталоги (рис. 3.24).

```
[root@vkmal]# echo os-intro > COURSE
```

Рис. 3.24: Создание необходимых каталогов

Ввожу фразу-пароль для разблокировки ключа gpg (рис. 3.25).

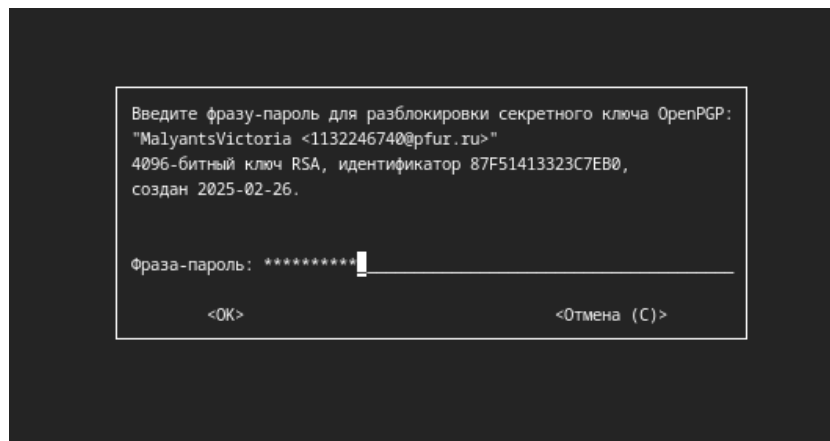


Рис. 3.25: Фраза-пароль

Создаю каталоги и отправляю файлы на сервер (рис. 3.26).

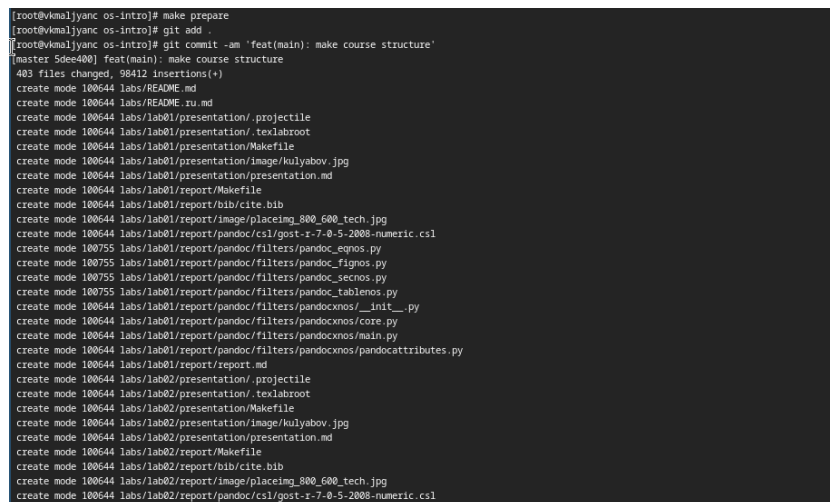


Рис. 3.26: Создание каталогов и отправление файлов на сервер

Отправляю файлы на сервер (рис. 3.27) [1].

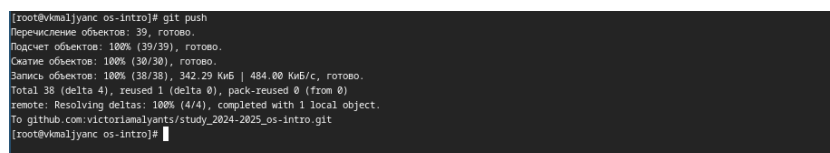


Рис. 3.27: Отправление файлов на сервер

Каталоги и файлы создались успешно (рис. 3.28).

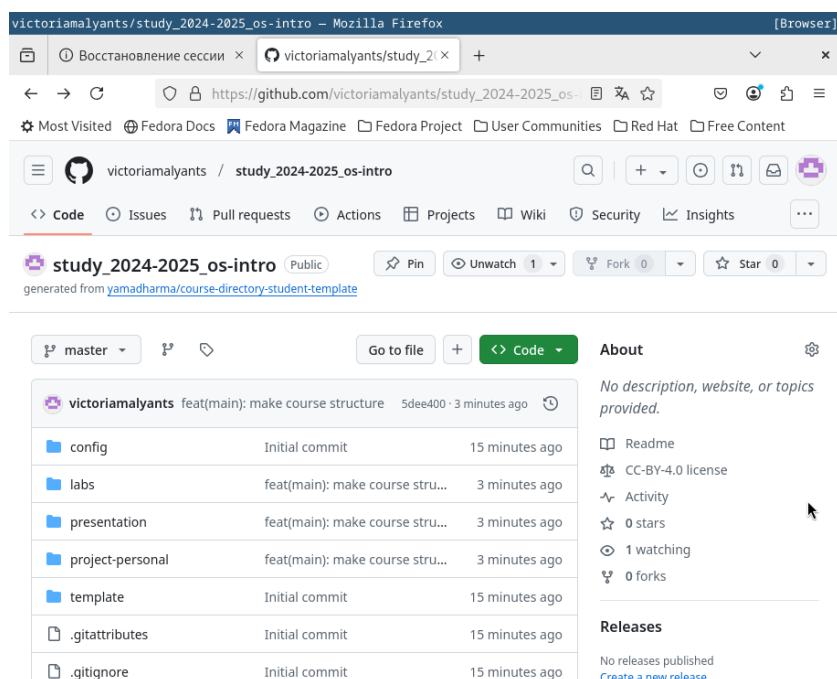


Рис. 3.28: Каталоги в GitHub

4 Выводы

Я изучила идеологию и применение средств контроля версий и освоила умения по работе с git.

5 Контрольные вопросы

1. Системы контроля версий (VCS) предназначены для совместной работы над проектами, а также для отслеживания изменений в файлах.
2. Хранилище - место, где хранятся все версии файлов проекта. Commit - для сохранения изменений в хранилище. История - последовательность всех коммитов. Рабочая копия - локальная версия файлов.
3. Централизованные VCS - все изменения хранятся на центральном сервере, например: CVS, Subversion. Децентрализованные VCS - каждый разработчик проекта имеет полную копию хранилища, включая всю историю изменений, например: Git, Mercurial.
4. Создание локального хранилища, внесение изменений в файлы, использование commit, просмотр изменений и возможность восстановления предыдущих версий
5. Клонирования репозитория, создание новой ветки, внесение изменений в commit, обновление локальной версии, слияние с основной веткой, отправка в хранилище.
6. Отслеживание изменений в файлах, управление версиями и ветвлением, слияние изменений от разных разработчиков, восстановление предыдущих версий файлов и работа с удаленными репозиториями.
7. git init (создание основного дерева репозитория), git pull (получение обновлений текущего дерева из центрального репозитория), git push (отправка всех произведенных изменений локального дерева в центральный репозиторий), git status (просмотр списка измененных файлов в текущей директории)

рии), `git diff` (просмотр текущих изменений), `git add` (сохранение текущих изменений), `git commit` (сохранение добавленных изменений), `git push origin` (отправка изменений конкретной ветки в центральный репозиторий), `git merge --no-ff` (слияние ветки с текущим деревом), `git branch` (управление ветками).

8. Локальный репозиторий: `git add .`, `git commit` Удаленный репозиторий: `git clone`, `git pull`
9. Ветви - нужны для работы с отдельными частями проекта, не мешая основной части.
10. С помощью файла `.gitignore`, чтобы игнорировать добавление файлов, которые не требуются в репозитории.

Список литературы

1. Лабораторная работа № 2.