

# **Лабораторная работа № 2**

**Первоначальная настройка git**

Мальянц Виктория Кареновна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
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
<b>4</b>	<b>Выводы</b>	<b>17</b>
<b>5</b>	<b>Контрольные вопросы</b>	<b>18</b>
	<b>Список литературы</b>	<b>20</b>

# 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 ключем 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).

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

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

Устанавливаю `git` (рис. 3.2).

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

Рис. 3.2: Установка `git`

Устанавливаю `gh` (рис. 3.3).

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

Рис. 3.3: Установка `gh`

### 3.2 Базовая настройка `git`

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

```
[root@vkmaliyanc ~]# git config --global user.name "Victoria Malyants"
[root@vkmaliyanc ~]# 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:W0770zmg1A3N/ENieKqacQ=nlUdK1gW/UCaZlCsAw root@kmaljanc
The key's randomart image is:
+--[ED25519 256]--+
|E               o+o|
|oo...o...oB+|
|oo=.o...+ +B|
|oB...o...+ o o|
|o-o o S...+ o |
|...o =...+...|
|...B...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).

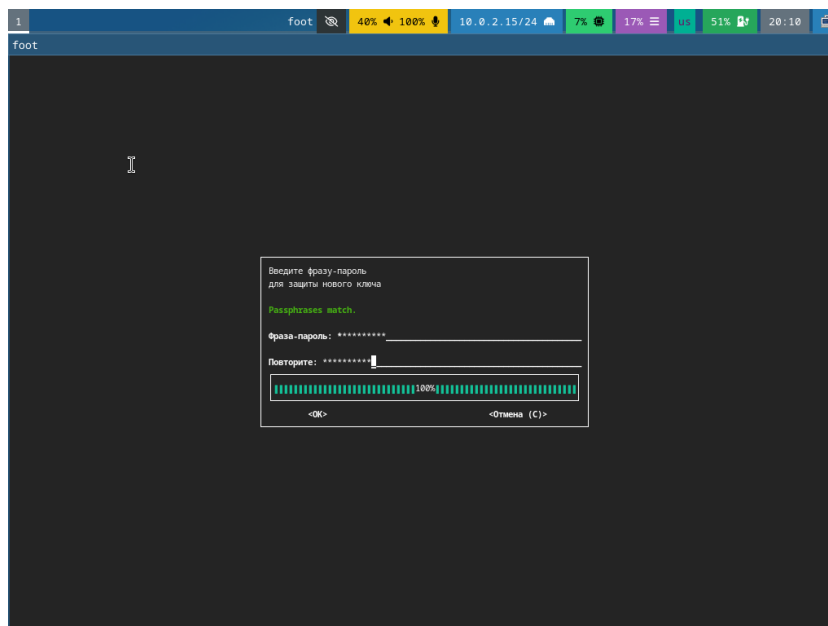


Рис. 3.11: Фраза-пароль

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).

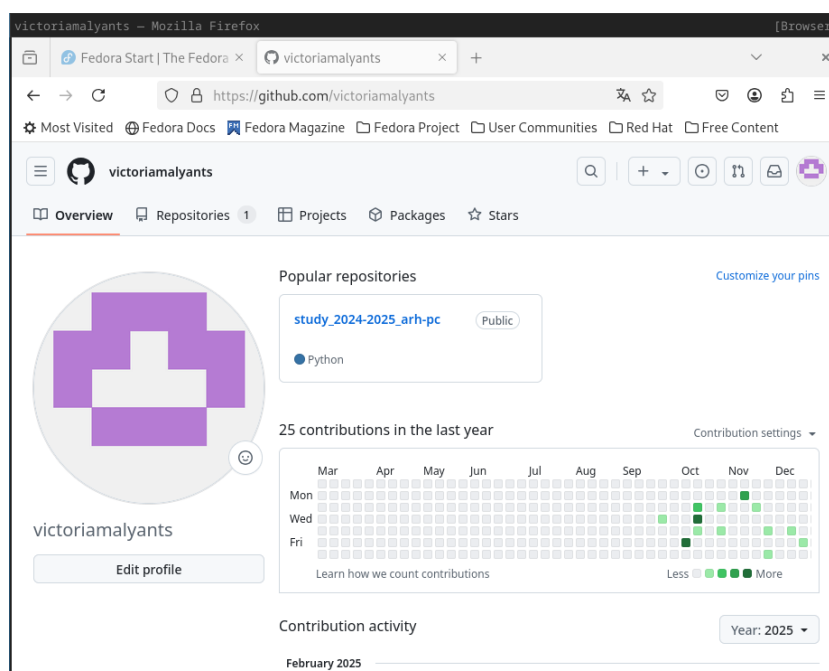


Рис. 3.13: Профиль GitHub

### 3.6 Добавление pgr ключа в GitHub

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

```
[root@vkmaljyanc ~]# gpg --list-secret-keys --keyid-format LONG
gpg: проверка таблицы доверия
gpg: marginals needed: 3 completes needed: 1 trust model: gpg
gpg: глубина: 0 достоверных: 3 подписанных: 0 доверие: 0-, 0q, 0n, 0m, 0f, 3u
[keyboard]
-----
sec rsa4096/5818ECA509826955 2025-02-25 [SC]
03F3A18EBACDAFC361EF698F5818ECA509826955
uid [ абсолютно ] MalyantsVictoria <1132246740@pfur.ru>
ssb rsa4096/9ADE6DE178AA4491 2025-02-25 [E]

sec rsa4096/2DBF2DB279C10DC3 2025-02-26 [SC]
532A0CB8B00891814E926942DBF2DB279C10DC3
uid [ абсолютно ] MalyantsVictoria <1132246740@pfur.ru>
ssb rsa4096/AD2A46E9AD97B6FA 2025-02-26 [E]

sec rsa4096/87F51413323C7EB0 2025-02-26 [SC]
563FA7F79097D0653E430B5B87F51413323C7EB0
uid [ абсолютно ] MalyantsVictoria <1132246740@pfur.ru>
ssb rsa4096/69900692170C1597 2025-02-26 [E]

[root@vkmaljyanc ~]#
```

Рис. 3.14: Список ключей

Копирую сгенерированный ключ `grg` (рис. 3.15).

```
[root@vmxali]# gpg --armor --export 563FA7F790970D653E430B5887F51413323C7E0
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGe+1cBEAChxYuBNwSImfVFFOX+XCh7Yp+qmN149al+sohP/V8qULqMJN7a
Cc8wkITBk2R1+msKq7J4deQ9dMlrg1GceZHQv5OpXkHmZs6zicLz2x64focFxa1
oCso/P9pqQFZIZHkhZDthf141psId6wYm11Xdc73gzUZpFwEYcedZhm+2Q2Rq1k
xhE29t8j1MPG+P/LM8tQc6+4x/F/WT8H8qJL6+byS/Wyr7F+eNyAlm1W0tM0Q2KF
qd+Yf35mkYP1+6SDbXLz1ARaEk8vpwyXcHcUGL7w3Gzh6E0hZrnJ91JtClbsaCa
zLqIWA/Ew1kyonNbwnz1zAvhV4+Vmc1BQKQ1SgegLBd9HjTdgwXKQJqubEMGSW3v
pratd6YanI7fwjzeZr60+KzLEnv+PM+Wdm2XB453fzXHX5UzvbvANe4ZQmsG71
ewJkmACz9H0K2ep/VzCtZg019LFouQYXo6P8oiC2Z/zsqXT1Q6dFkmv1ibqz1U4
03VUCuXIeXqD.Jw8QKQKTOtC1QJEZOWoX18b1N5X3JbuGoBgYVXK195+nxCTUK
Kj1MisuAuKfBW+3pP68aHajJ3XxvD0/+2cyoraR1mVdGSLPVi0isC0xwZXLJc
Um7R2UCvg+oBWM+1R+1SV583FY28bWfSeMtmB+1utK8ELZ1854uBV8WmAABoA
tCVNyx5YwS0c12p3Rvcml1JDBwMTMyMjQ1ZmZkQW9HbmdXBMd+Amn+1QJRBWBMCA
7F1EEVj++n95CX8GU+QwtbH/UUEZ18trFAme+1cCGwF-Cwk1Bw1C1gIGFqQJCaS
BBYCaECWgcCF4AAcGkqh/UUEZ18trfDnag//UJ76RdB+PEIKQdtszeyhq/3hY5E
5BVb8c2RnXw104XFcb1+FKUXscPuj1yXfSQ4eqDVEK2tOZTFYmRvG5tzyqR/Whr
r2LSVze175BSDx7Ly6FY8NcAqAxlXnUZUvWNRX5uqCXN4Nj1k8ozes1mh6p6
DocJc+fn139dYL3Ru08Fgfet1M38sT821pJvMelnhz3NQ0kvoHuKfH5Q0wzrGn1Pw
AxNfEYtC1VrrkduEKPC0vpzXbMw1L2k8hx6n94dG67+FGpq/W3nRey3t3ktKu49
jCsJlmtQxHd7GwzduEB9seSE3FmUoDU/9WfHulIMH6se+1z180Gyzo1W1TK
+9H/7f9eZrJoZP26q072AR6GkCwnM272HjceCtjg0s5ntNZV71cwnH9pRzP6y
e903K10sroSn3V0qgTjBkPp2Jx1wnzyVNAUwPAPYmBvGaGfYhsq1516hokSL1Z
+RrEZd147Umkrqynah0nz47fokkxMnScRkn0sP/e82Huk/3FMWCGV25SqA1
oq1ptChc+hp2HueUdm+wo66DLVBJHkSZOTdrfcQw59rTkWkXhN4GzG70tZQeRQ
U03DwiJg17luY1DTc0M4tEYs/Ef8vSu4Mb+7cBMJ/bjJenL2wC71D09F1XvYmr/Z
984mcAsyQBuYhmM5Ag0EZ7763VwEQAmmHSK18D18c1W1Z9JabUzIC2Iqo9FXnAJ
XXtcFYyxfmC0tZcFwL4YTAvRT1ZG3XhrTzQawP/9h8dm2DLKcXzRW7xhVbVHRJ
jig7R0WU1BX17Re0EBdKlXJMLj4Tyc+11v1xQ54C8H8Y4S1Vqubs9JY Cqyr16c
KddZeyT4ndkXtm1+TnrhRTd1WjhgdknZdkCTwG6651BNLEqoB1L18bang1Ihe5P
jPnfduoJ211f211YwPL3dP5th9rQ1S5Y6R0KHrpjarpBha3Sa490iEPvgAvDv1
uqh18/XDgX8JhBADmE+GwLdOHWsc1ky0bux2BTkT4GaIoknDU0p451Vr66u1ym
Wf88A6n3LjyrfvUuW09gDSF3wChWmeYdz13B85D1W0ipXba+fA7Gnwh1E1RL
LEUgKke9FYfx6mnxZECJ+hcBg3E2ZRoLRLXmYpZbWleq/P6T3Hu13HRE9PS0+
+NGUHT9Z1c7fwyBt1f1w89umYUjZj/yYfMM/9dkSwdeG3x15JmJcXvMqENYKF
```

Рис. 3.15: Копирование сгенерированного ключа `grg`

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

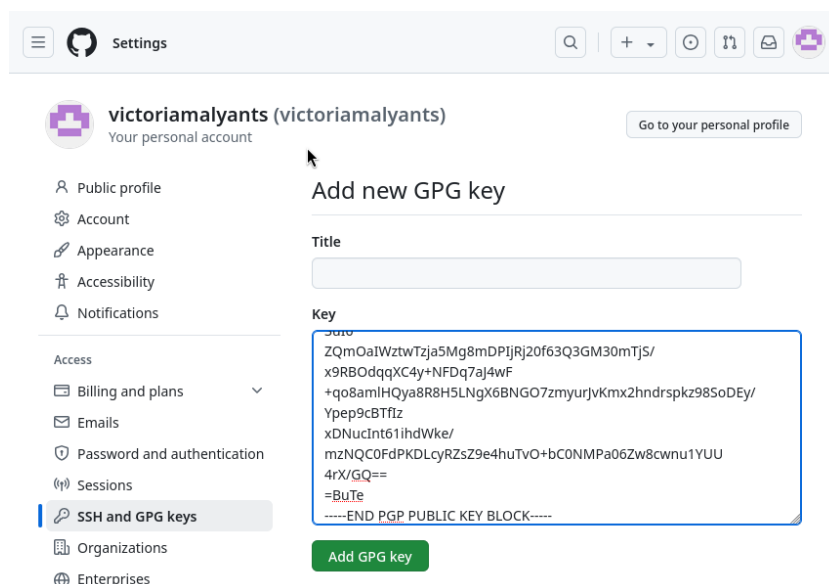


Рис. 3.16: Создание gpg ключа

Ключ gpg создан (рис. 3.17).

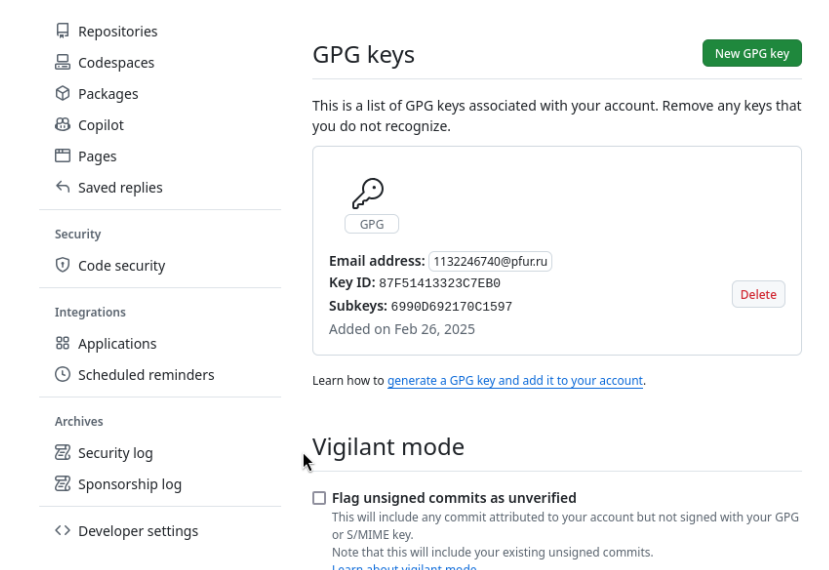


Рис. 3.17: Ключ gpg создан

## 3.7 Настройка автоматических подписей коммитов git

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

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

Рис. 3.18: Настройка для применения подписей коммитов

## 3.8 Настройка gh

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

```
[root@vkmajyanc ~]# 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@vkmajyanc ~]# 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@vkmajyanc ~]# mkdir -p ~/work/study/2024-2025/“Операционные системы”
[root@vkmajyanc ~]# 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@vkmaljanc Операционные системы]# gh repo create study_2024-2025_os-intro --template=yamadharma/course-directory-student-template --public
[root@vkmaljanc Операционные системы]# 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/zLD48zPMSvHokr4UvCoqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? no/C
[root@vkmaljanc Операционные системы]# 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/zLD48zPMSvHokr4UvCoqU.
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@vkmaljanc Операционные системы]#
```

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

Рис. 3.22: Переход в каталог курса

Удаляю лишние файлы (рис. 3.23).

```
[root@vkmaljanc os-intro]# rm package.json
rm: удалить обычный файл 'package.json'?
[root@vkmaljanc os-intro]#
```

Рис. 3.23: Удаление лишних файлов

Создаю необходимые каталоги (рис. 3.24).

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

Рис. 3.24: Создание необходимых каталогов

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

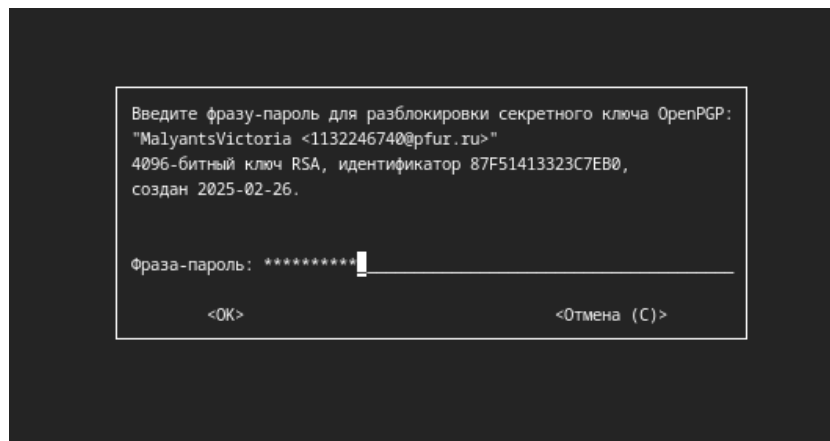


Рис. 3.25: Фраза-пароль

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

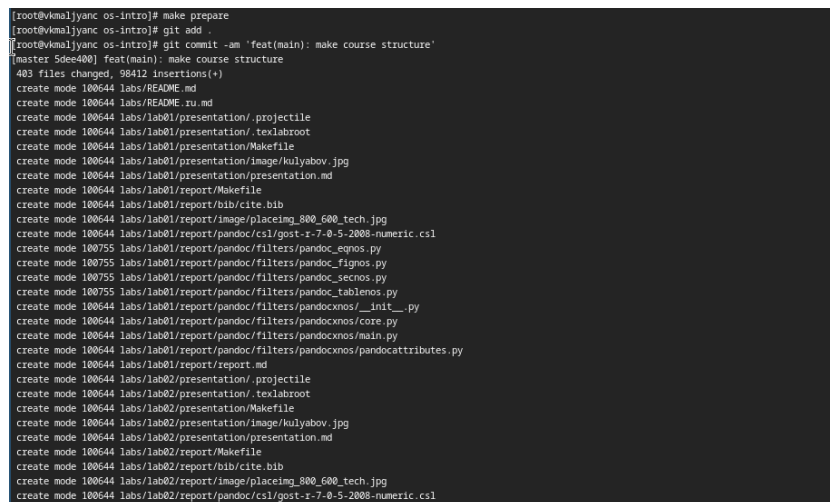


Рис. 3.26: Создание каталогов и отправление файлов на сервер

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

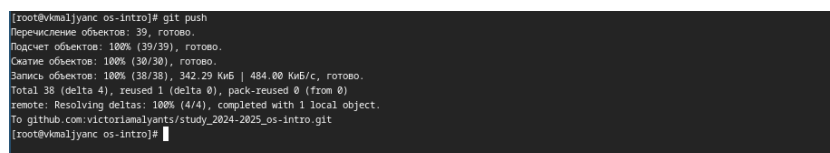


Рис. 3.27: Отправление файлов на сервер

Каталоги и файлы создались успешно (рис. 3.28).

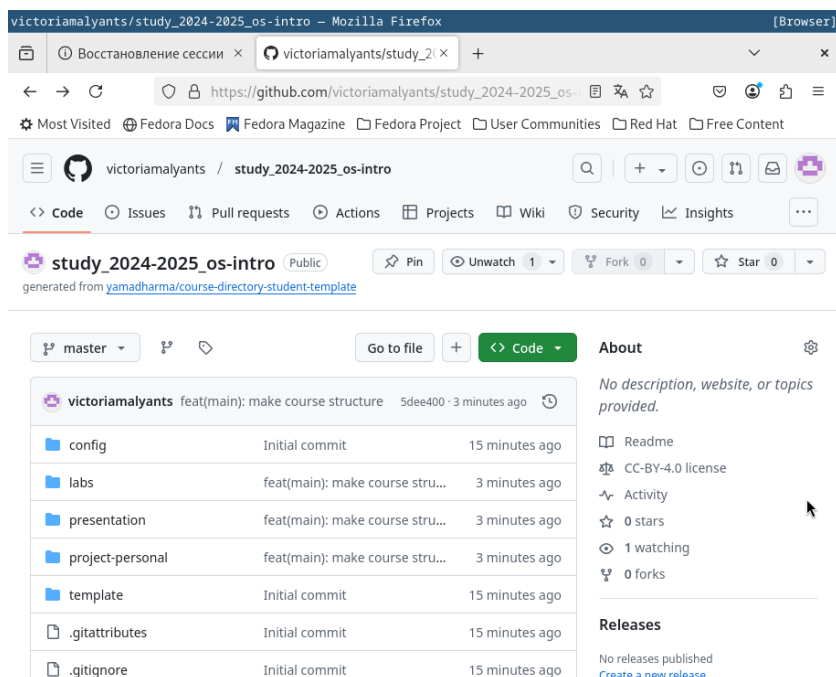


Рис. 3.28: Каталоги в GitHub



## 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.