

# **Отчет по лабораторной работе №2**

**Операционные системы**

Ермакова Анастасия Алексеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
4.1	Установка программного обеспечения . . . . .	8
4.2	Базовая настройка git . . . . .	8
4.3	Создание ключа ssh . . . . .	9
4.4	Создание ключа pgr . . . . .	10
4.5	Настройка github . . . . .	11
4.6	Добавление PGP ключа в GitHub . . . . .	11
4.7	Настройка автоматических подписей коммитов git . . . . .	13
4.8	Настройка gh . . . . .	13
4.9	Шаблон для рабочего пространства . . . . .	14
<b>5</b>	<b>Выводы</b>	<b>16</b>
<b>6</b>	<b>Список литературы</b>	<b>17</b>

# Список иллюстраций

4.1	Установка git . . . . .	8
4.2	Установка gh . . . . .	8
4.3	Задание имени и почты . . . . .	9
4.4	Настройка git . . . . .	9
4.5	Настройка git . . . . .	9
4.6	Создание ключей . . . . .	9
4.7	Создание ключей . . . . .	10
4.8	Генерирование ключей . . . . .	11
4.9	Вывод списка ключей . . . . .	12
4.10	Вывод ключа на экран . . . . .	12
4.11	Добавление ключа . . . . .	12
4.12	Добавленный ключ . . . . .	13
4.13	Настройки подписей коммитов . . . . .	13
4.14	Авторизация . . . . .	13
4.15	Успешная авторизация . . . . .	14
4.16	Успешная авторизация . . . . .	14
4.17	Создание репозитория курса . . . . .	14
4.18	Настройки каталога курса . . . . .	15
4.19	Отправка файлов на сервер . . . . .	15
4.20	Отправка файлов на сервер . . . . .	15

## **Список таблиц**

# 1 Цель работы

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

## 2 Задание

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

## 3 Теоретическое введение

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется.

# 4 Выполнение лабораторной работы

## 4.1 Установка программного обеспечения

Первым делом открываю терминал, переключаюсь на роль суперпользователя и прописываю команду для установки git (рис. 4.1). Он уже оказался установлен.

```
[aezetakova@fedora ~]$ sudo -i
[sudo] пароль для aezetakova:
[root@fedora ~]# dnf install git
Обновление и загрузка репозитория:
Fedora 41 - x86_64 - Updates 100% | 31.5 KiB/s | 2
Fedora 41 - x86_64 - Updates 100% | 1.1 MiB/s | 2
Репозитории загружены.
Пакет "git-2.48.1-1.fc41.x86_64" уже установлен.
Нечего делать.
```

Рис. 4.1: Установка git

Далее устанавливаю gh (рис. 4.2).

```
[root@fedora ~]# dnf install gh
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет
Установка:
gh
Арх.
x86_64
Версия
2.65.0-1.fc41
Репозиторий
updates
Сводка транзакции:
Установка: 1 пакета
Общий размер входящих пакетов составляет 10 MiB. Необходимо загрузить 10 MiB.
После этой операции будут использоваться дополнительные 43 MiB (установка 43 MiB, удаление 0 B).
Is this ok [y/N]: y
[1/1] gh-2.65.0-1.fc41.x86_64 100% | 6.9 MiB/s | 1
-----
[1/1] Total 100% | 4.3 MiB/s | 1
Выполнение транзакции
[1/3] Проверить файлы пакета 100% | 8.0 B/s | 1.0 B | 00:00s
[2/3] Подготовить транзакцию 100% | 1.0 B/s | 1.0 B | 00:01s
[3/3] Установка gh-2.65.0-1.fc41.x86_64 100% | 11.9 MiB/s | 42.7 MiB
Завершено!
```

Рис. 4.2: Установка gh

## 4.2 Базовая настройка git

Задаю имя и email владельца репозитория (меня) (рис. 4.3).



```
[root@fedora ~]# git config --global user.name "Nastya Yermakova"
[root@fedora ~]# git config --global user.email "knopka6363@yandex.ru"
```

Рис. 4.3: Задание имени и почты

Настраиваю utf-8 в выводе сообщений git (рис. 4.4).

```
[root@fedora ~]# git config --global core.quotepath false
```

Рис. 4.4: Настройка git

Задаю имя начальной ветки master, параметр autocrlf и параметр safecrlf (рис. 4.5).

```
[root@fedora ~]# git config --global init.defaultBranch master
[root@fedora ~]# git config --global core.autocrlf input
[root@fedora ~]# git config --global core.safecrlf warn
```

Рис. 4.5: Настройка git

## 4.3 Создание ключа ssh

Сначала создаю ключ по алгоритму ed25519 (рис. 4.6).

```
[root@fedora ~]# 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
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:G5fjvtvGb/SNjYjJ9Hjo2/pM7zBSgN7wnoFQxFXThZU root@fedora
The key's randomart image is:
+--[ED25519 256]--+
|      oo...o. ++|
|      ...  .oE |
|      . o .   |
|      o = o    |
|      S B .    |
|      *,= .    |
|      .o=B+o =. |
|      .+++++ + |
|      .BO=++   |
+----[SHA256]-----+
```

Рис. 4.6: Создание ключей

Затем по алгоритму rsa с ключем размером 4096 бит (рис. 4.7).

```
[root@fedora ~]# 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
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:PPa6e3sTBi3GPVvPYQW2A1KyHYFRzpl0xpmmENy0kYs root@fedora
The key's randomart image is:
+---[RSA 4096]-----+
|  ..oo.+=*.oo. |
|  ..o+.@ *+. |
|  ...oX+*oo+ |
|  Eo.== = ... |
|  S. o . |
|  . o o |
|  .. . |
|  .. o |
|  ++.o . |
+-----[SHA256]-----+
```

Рис. 4.7: Создание ключей

## 4.4 Создание ключа pgr

Генерирую ключи (рис. 4.8). Из предложенных опций выбираю те, что указаны по заданию.

```

[root@fedora ~]# 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
    <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) 0
Срок действия ключа не ограничен
Все верно? (y/N) y

GnuPG должен составить идентификатор пользователя для идентификации ключа.
Ваше полное имя: ааеаакова
Адрес электронной почты: knopka6363@yandex.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
    "ааеаакова <knopka6363@yandex.ru>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? o

```

Рис. 4.8: Генерирование ключей

## 4.5 Настройка github

У меня уже был создан аккаунт, поэтому этот пункт я пропускаю.

## 4.6 Добавление PGP ключа в GitHub

Вывожу список ключей (рис. 4.9). Мне нужен первый.

```
[root@fedora ~]# 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: 1 подписанных: 0 доверие: 0-, 0q, 0n
[keyboard]
-----
sec   rsa4096/9F79AC46934AFFA0 2025-03-05 [SC]
      549504F1ECEC1215E23E29EF9F79AC46934AFFA0
uid   [ абсолютно ] ааегамакова <кнопка6363@yandex.ru>
ssb   rsa4096/E7FCAD28D3C14667 2025-03-05 [E]
```

Рис. 4.9: Вывод списка ключей

Вместо неустановленной команды хслір вывожу ключ на экран с помощью команды `gpg --armor --export` ключ. (рис. 4.10).

```
[root@fedora ~]# gpg --armor --export 9F79AC46934AFFA0
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGfIcIABEAD5WqarYUPzZVcCi701yt4DCt6X1jzaLhqhFGaz6nGLiWwd61gJ
WQV0t7H6c8E+7cGMGn3SLu6cCy7yftMkhDnImHBmPDtkG1NbvZCm3mfsSSHrf+M3
```

Рис. 4.10: Вывод ключа на экран

Перехожу в настройки GitHub, нажимаю кнопку New GPG key и вставляю полученный ключ в поле ввода. Добавляю его. (рис. 4.11).

Title

fjhvks

Key

```

yKumCESm34KKZ8KKZqKMEtGZ7N3x103g7TtTE30yCF3HmZnWwYF07
Rbbs3KJJG
D+0PwEas0EFYNODEDKFntqal6gjWRjWRTazqdIDh5W4f+DcFXi0ahdc
glh8
WP1uTxsx7GjiXWts0LYXYGxv1L0lvEdlSEK8sgCaVCQuA7rdXylFd9l+8V
hj
/UjwcZxY9btFqoQYZA+Q/
t9si497VTQf8j0wVXMxsKZzTG+tC5zDt1yb1v9bpWNA
=yHIM
-----END PGP PUBLIC KEY BLOCK-----

```

Add GPG key

Рис. 4.11: Добавление ключа

Ключ добавлен (рис. 4.12).

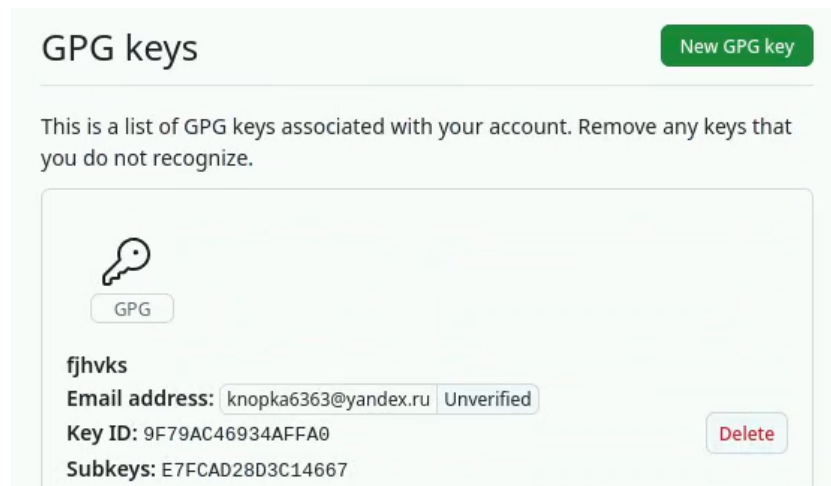


Рис. 4.12: Добавленный ключ

## 4.7 Настройка автоматических подписей коммитов git

Используя введенный email, указываю Git применять его при подписи коммитов (рис. 4.13).

```
[root@fedora ~]# git config --global user.signingkey 9F79AC46934AFFA0
[root@fedora ~]# git config --global commit.gpgsign true
[root@fedora ~]# git config --global gpg.program $(which gpg2)
```

Рис. 4.13: Настройки подписей коммитов

## 4.8 Настройка gh

Авторизуюсь через браузер (рис. 4.14).

```
[root@fedora ~]# gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? 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: DA4B-9A39
Press Enter to open https://github.com/login/device in your browser...
```

Рис. 4.14: Авторизация

Авторизация прошла успешно (рис. 4.15-4.16).

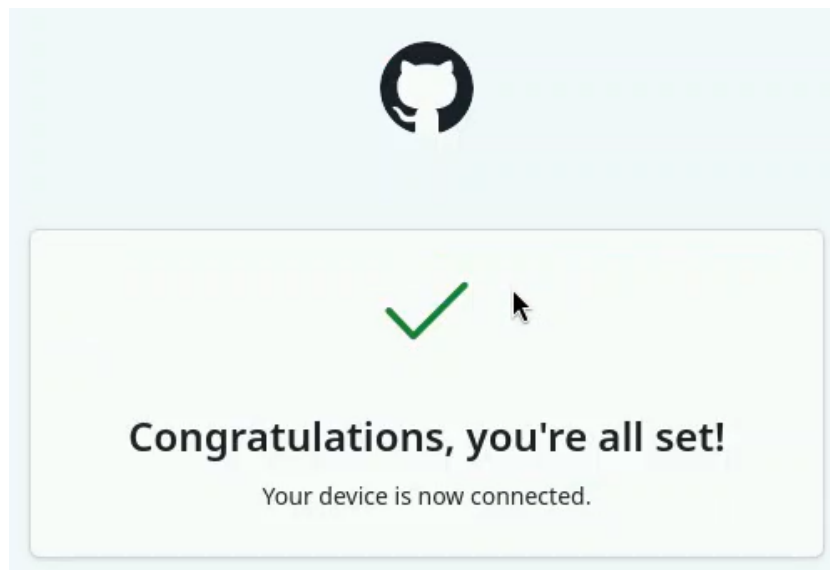


Рис. 4.15: Успешная авторизация

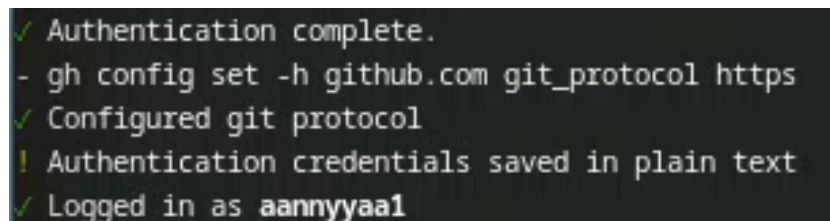


Рис. 4.16: Успешная авторизация

## 4.9 Шаблон для рабочего пространства

Создаю репозиторий курса на основе шаблона рабочего пространства. Создаю сам каталог, перехожу в него и прописываю команду для создания репозитория, указывая шаблон. Клонировать репозиторий. (рис. 4.17).

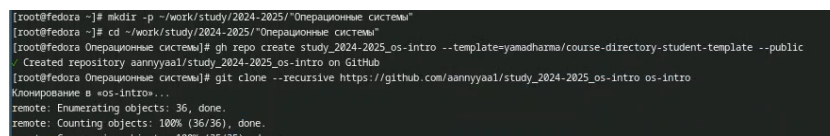


Рис. 4.17: Создание репозитория курса

Далее настраиваю каталог курса. Для этого перехожу в этот каталог, удаляю лишние файлы, создаю необходимые каталоги (рис. 4.18).

```
[root@fedora Операционные системы]# cd ~/work/study/2024-2025/"Операционные системы"/os-intro
[root@fedora os-intro]# rm package.json
rm: удалить обычный файл 'package.json'? y
[root@fedora os-intro]# echo os-intro > COURSE
[root@fedora os-intro]# make
Usage:
  make <target>

Targets:
  list          List of courses
  prepare       Generate directories structure
  submodule     Update submules
[root@fedora os-intro]# make prepare
```

Рис. 4.18: Настройки каталога курса

Отправляю файлы на сервер (рис. 4.19-4.20 ).

```
[root@fedora os-intro]# git add .
[root@fedora os-intro]# git commit -am 'feat(main): make course structur
```

Рис. 4.19: Отправка файлов на сервер

```
create mode 100044 project-personal/stage6/report/pandoc/filters/pandoc-filters.md
create mode 100644 project-personal/stage6/report/report.md
[root@fedora os-intro]# git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 342.32 КиБ | 15.56 МиБ/с, готово.
Total 38 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/aannyaa1/study_2024-2025_os-intro
7c38d54..2f268b7 master -> master
```

Рис. 4.20: Отправка файлов на сервер

## **5 Выводы**

В ходе выполнения данной лабораторной работы я изучила идеологию и применение средств контроля версий и освоила умения по работе с git.



## **6 Список литературы**