РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Математический институт им. С.М. Никольского

**Лабораторная работа №2**

**на тему**

**«**Первоначальна настройка git**»**

по дисциплине «Операционные системы»

Выполнил:

Студент группы НПМбв-02-21

Студенческий билет№:

1032189468

Синин Евгений Ринатович

Руководитель:

преподаватель Математического института им. С.М. Никольского

« » 2025 г.

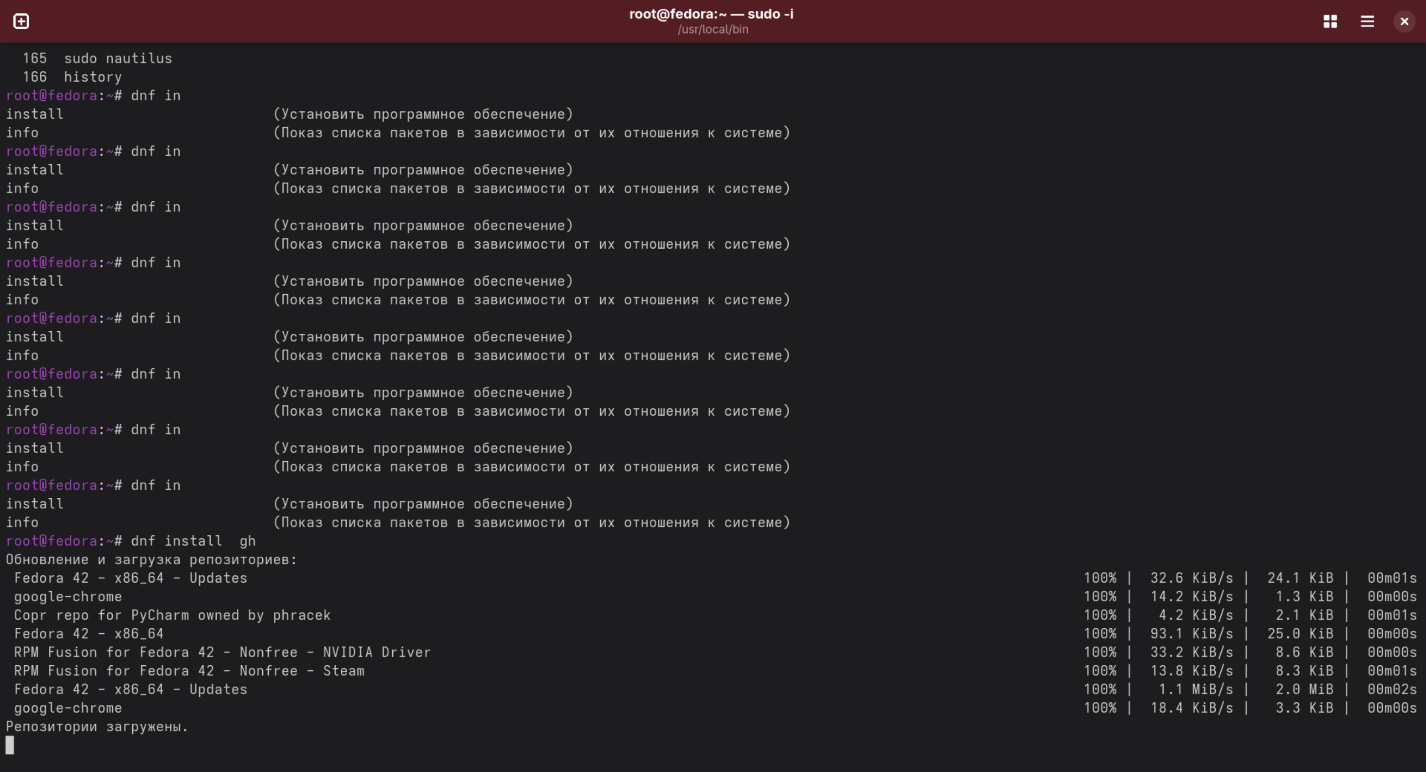
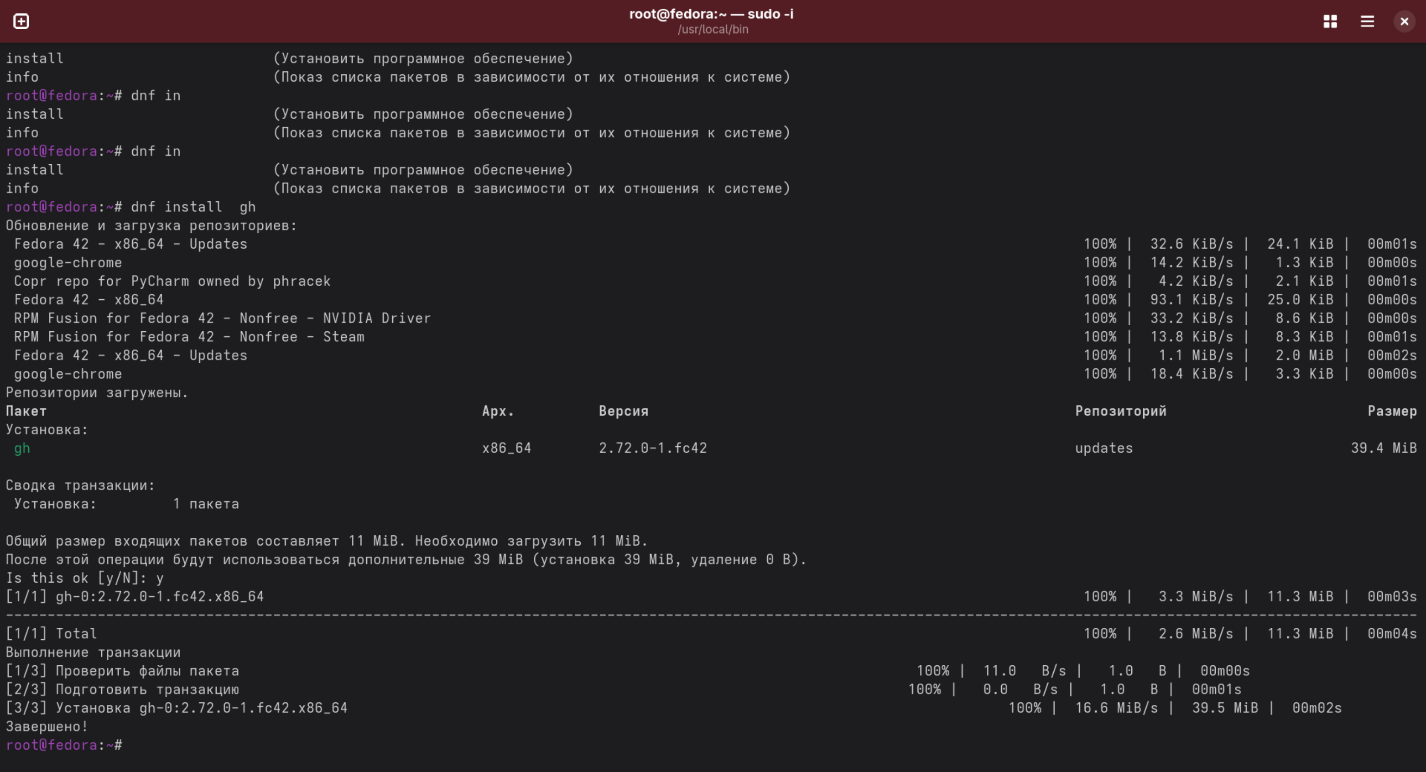
Цель работы

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

## Установка программного обеспечения

### **Установка git**

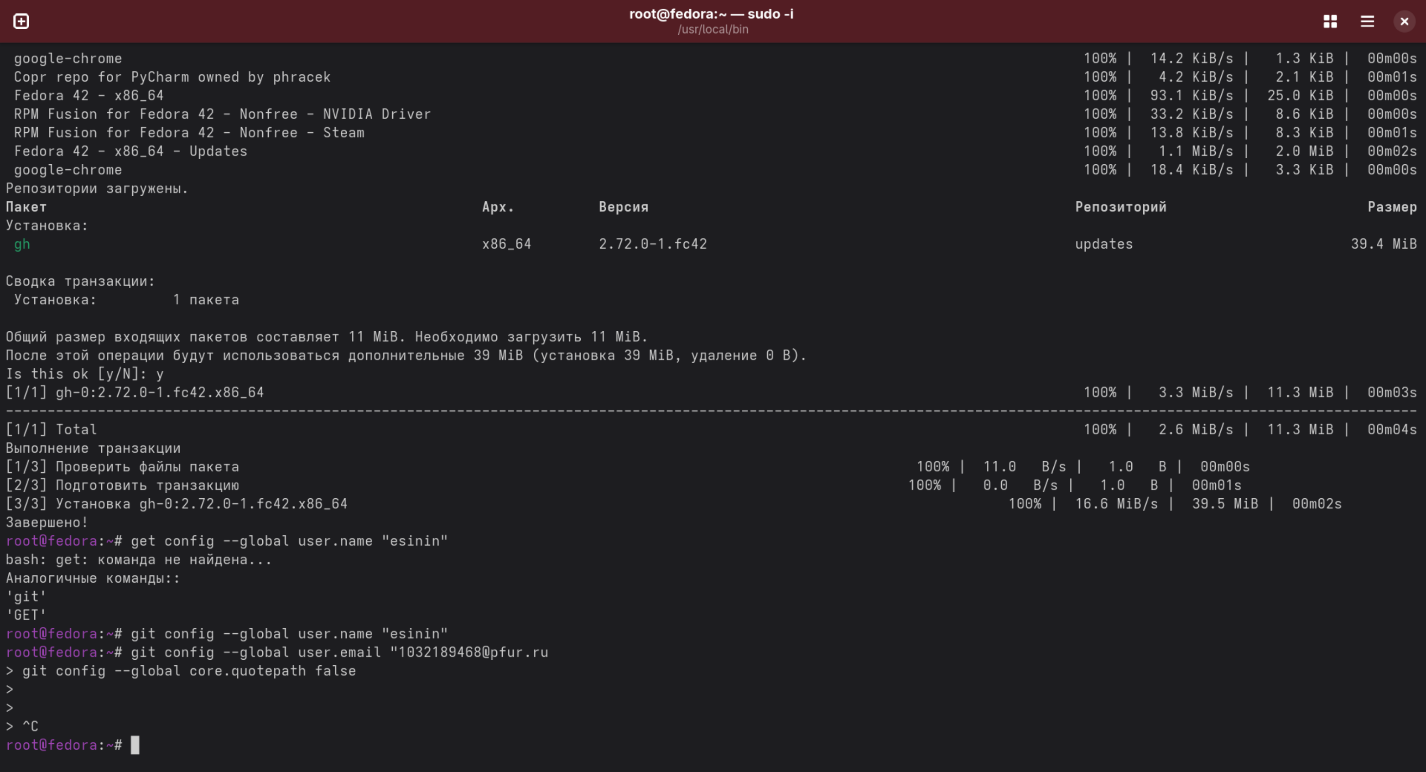
С помощью команды

* Установим *git*:
* dnf install git  
    
    
  **Установка gh**dnf install gh

## Базовая настройка git

* Зададим имя и email владельца репозитория:
* git config --global user.name "Name Surname"
* git config --global user.email "work@mail"
* Настроим utf-8 в выводе сообщений git:

git config --global core.quotepath false

****

Настройте верификацию и подписание коммитов git

Предложено сделать с помощью GPG (ссылка на источник не открывается.)

Пинструкции сперва необходимо создать ключ

gpg --full-generate-key

тип: rsa and rsa

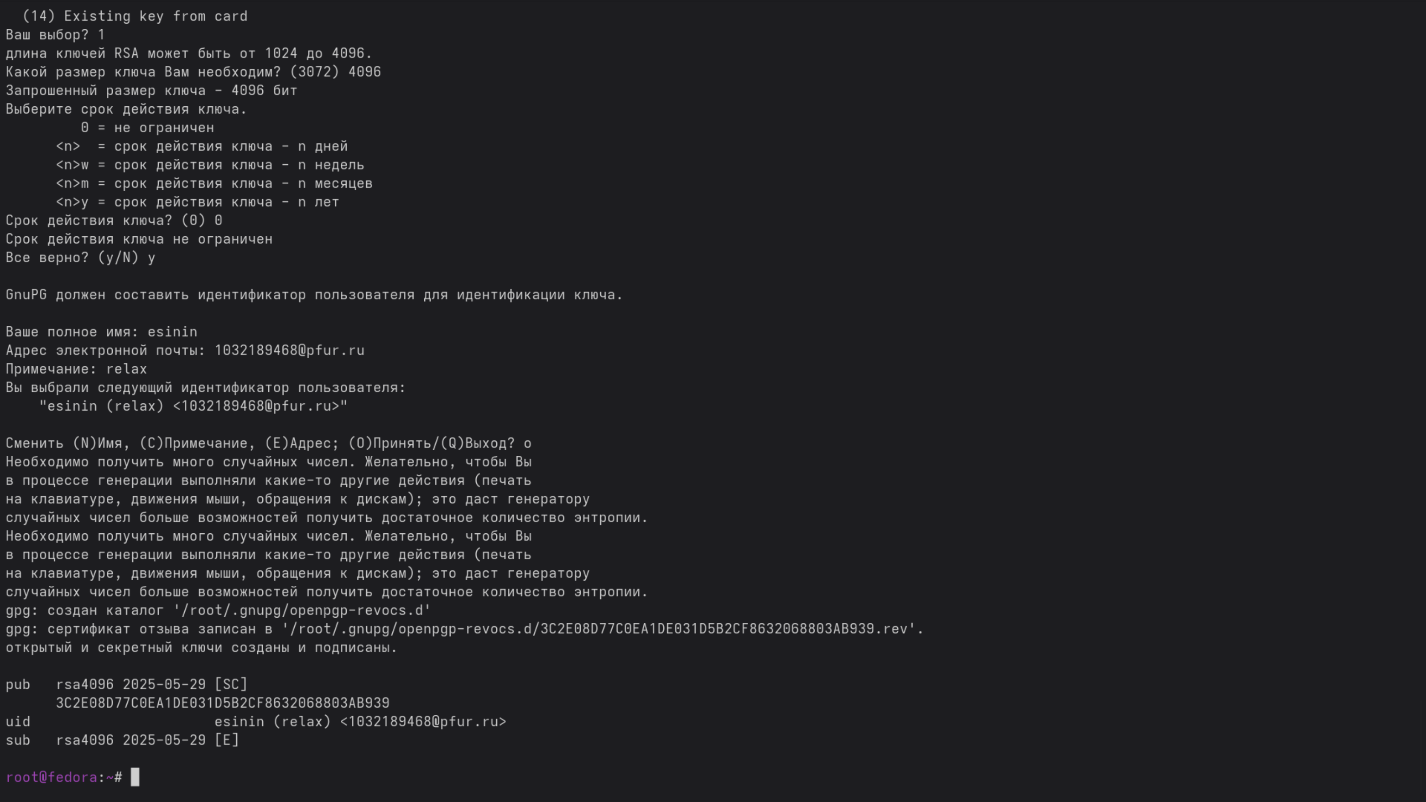
длинна 4096

срок 0 (навсегда)

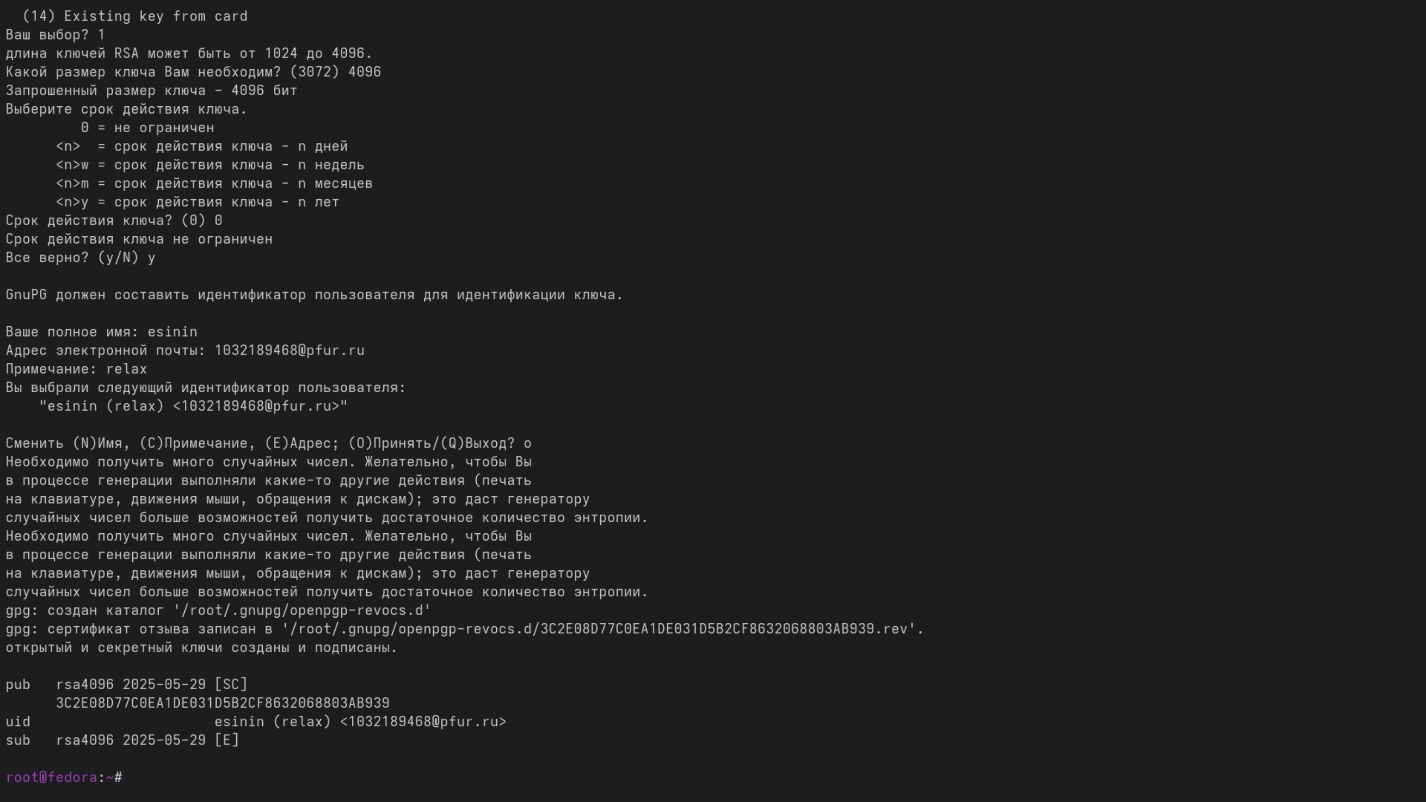
личная информация и ученическая почта

для защиты ключа пароль – lolk

ключ создан



Тепрь необходимо установить gpg ключь для Git

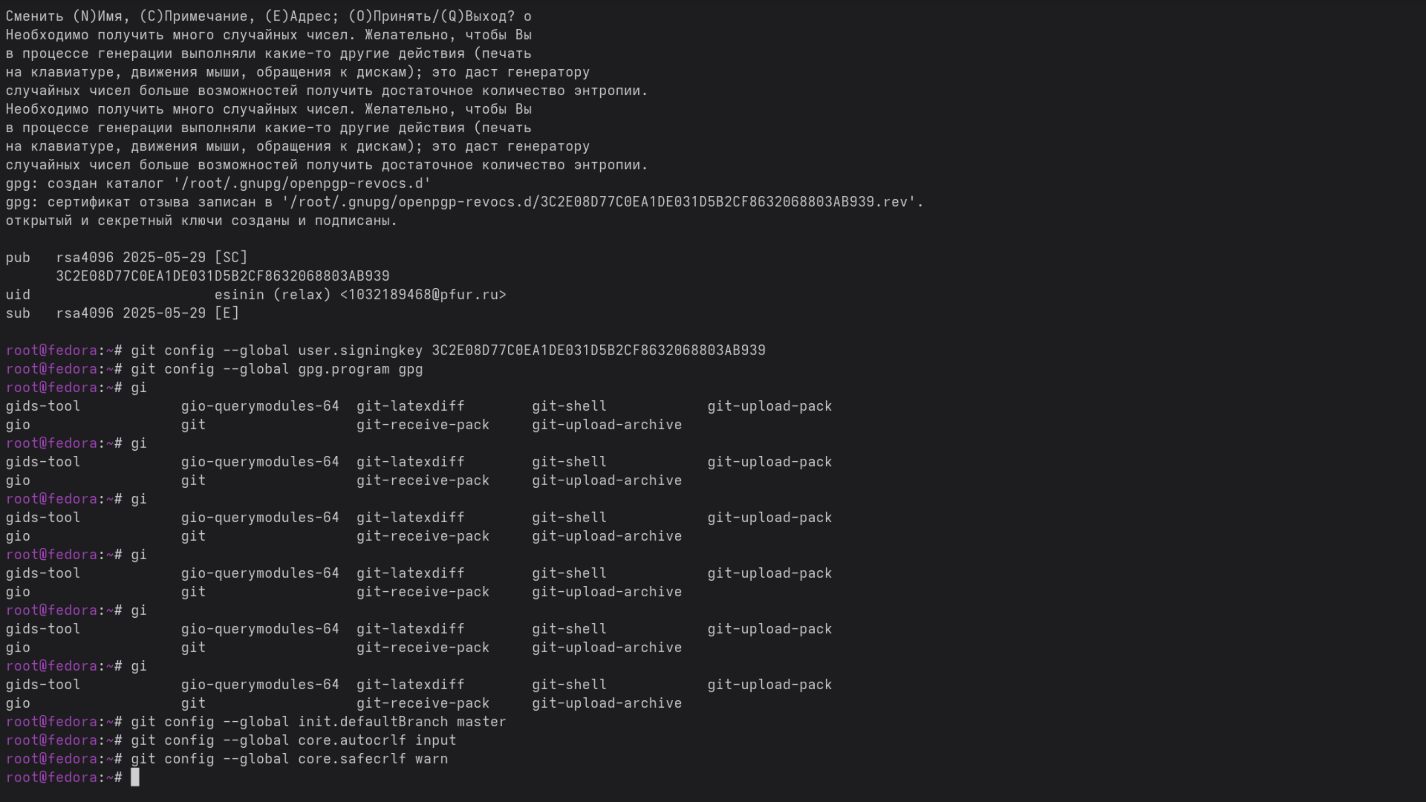
git config --global user.signingkey YOUR\_KEY\_ID

Зададим имя начальной ветки (будем называть её master):

git config --global init.defaultBranch master

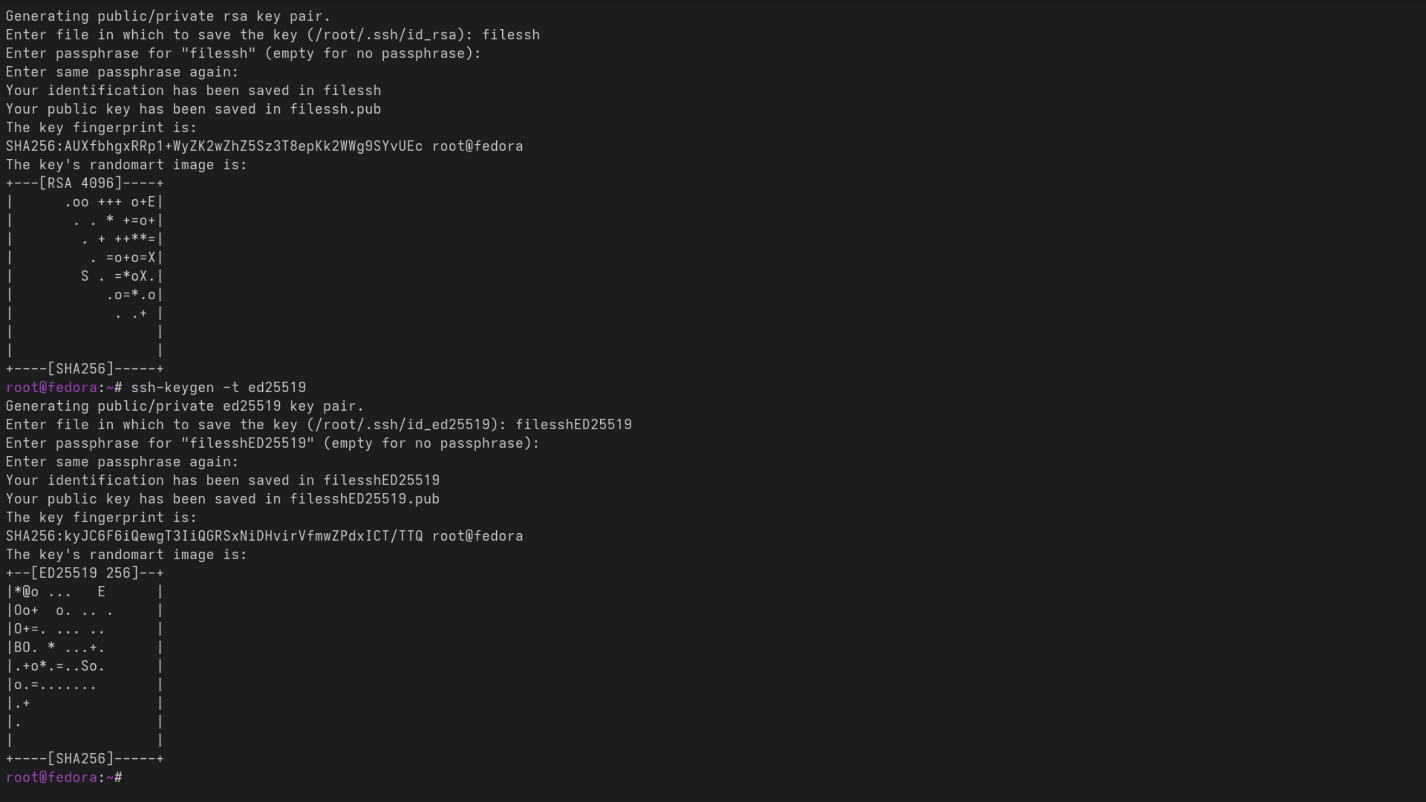
* Параметр autocrlf:
* git config --global core.autocrlf input
* Параметр safecrlf:

git config --global core.safecrlf warn



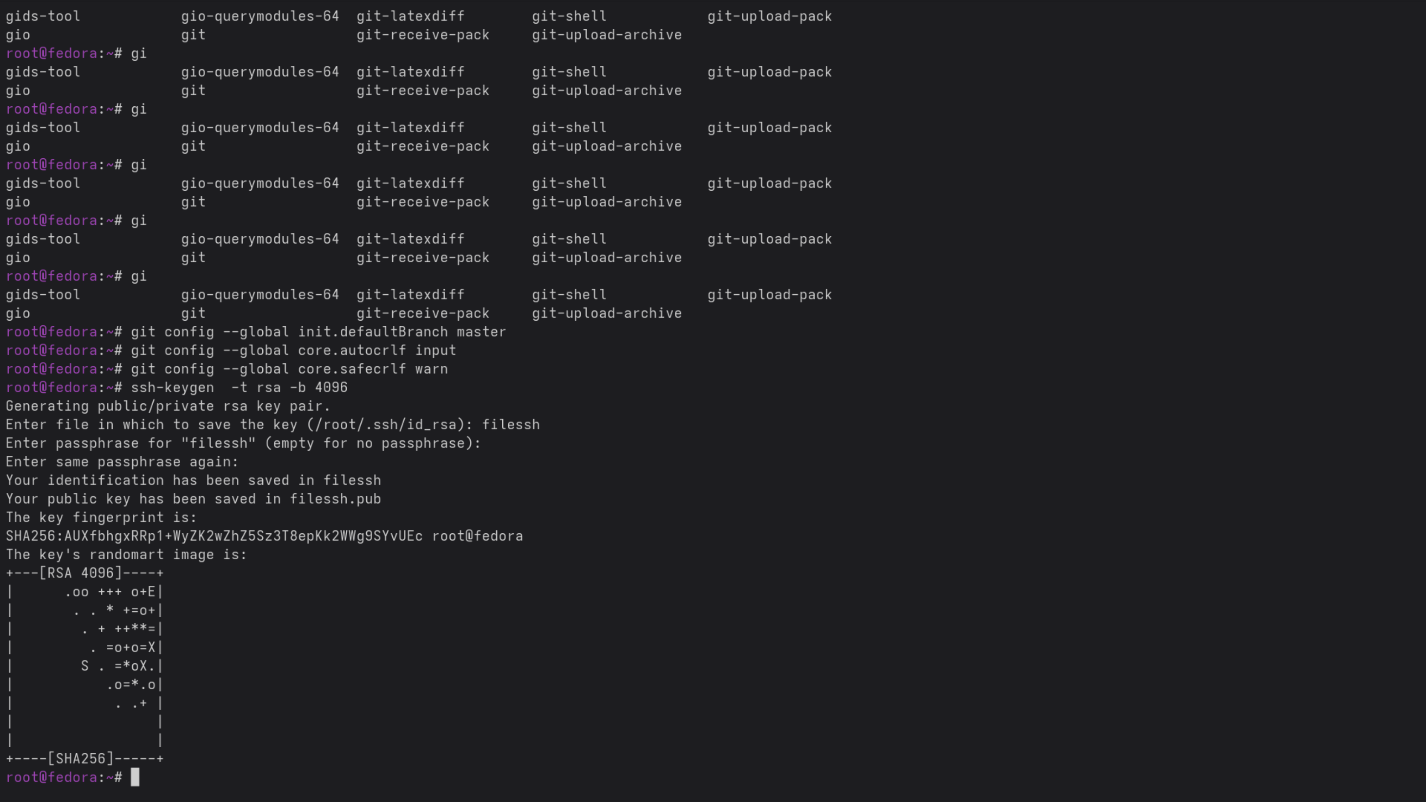
## Создайте ключи ssh

* по алгоритму rsa с ключём размером 4096 бит:
* ssh-keygen -t rsa -b 4096



* по алгоритму ed25519:

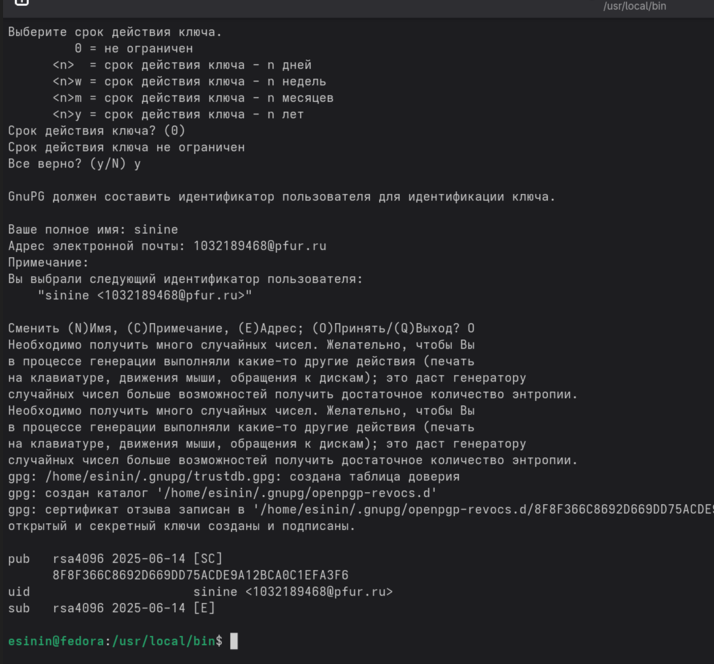
ssh-keygen -t ed25519



## Создайте ключи pgp

Имя sinine

для защиты ключа пароль – lolk

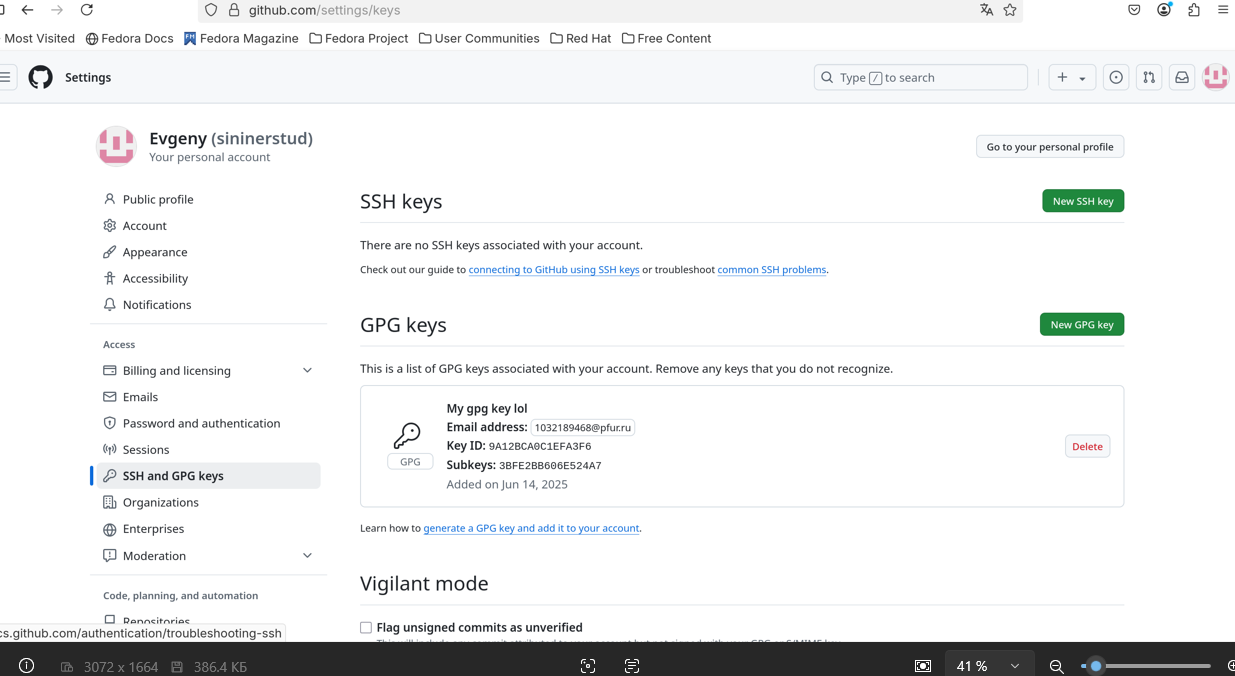


## Настройка github

Git hub

Post- 1032189468@pfur.ru

Username- sininerstud

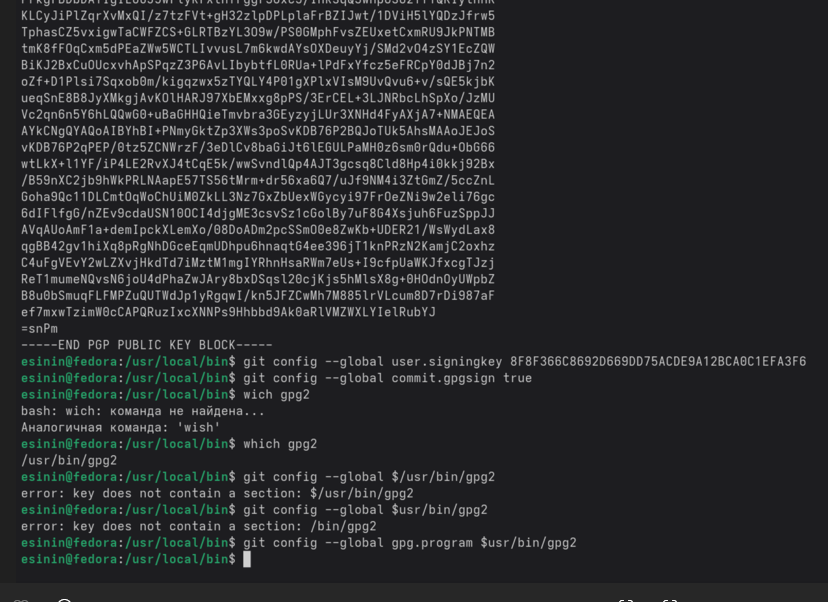


## Добавление PGP ключа в GitHub

* Выводим список ключей и копируем отпечаток приватного ключа:

gpg --list-secret-keys --keyid-format LONG

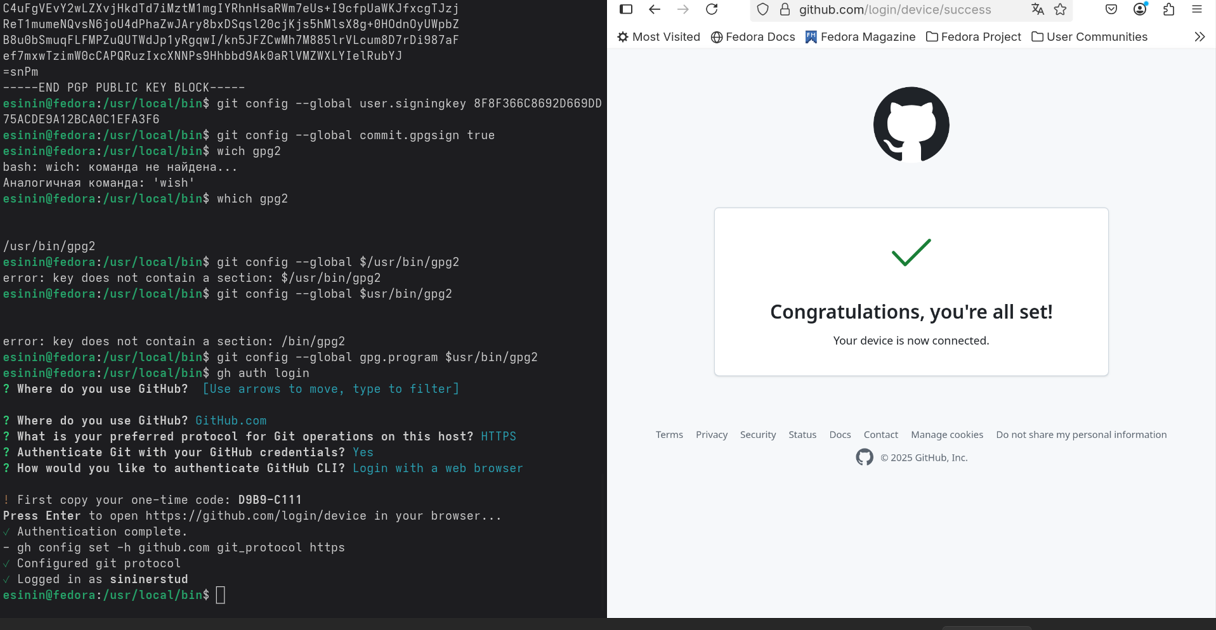
Установил ключ Название- my gpg key



## Настройка автоматических подписей коммитов git

* git config --global user.signingkey <PGP Fingerprint>
* git config --global commit.gpgsign true

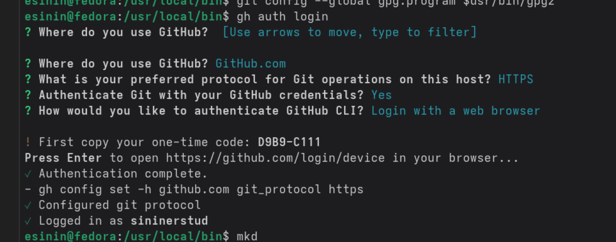
git config --global gpg.program $usr/bin/gpg2



## Настройка gh

Авторизуемся

gh auth login



Репозиторий

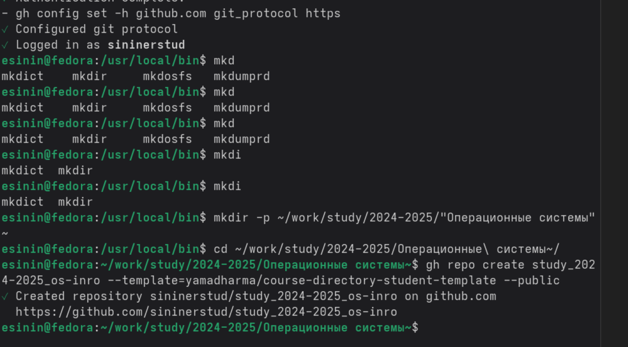
<https://github.com/yamadharma/course-directory-student-template>

Создал папку для репозитория

* mkdir -p ~/work/study/2024-2025/"Операционные системы"

Перешёл в папку в консоли

* cd ~/work/study/2024-2025/"Операционные системы"
* gh repo create study\_2024-2025\_os-intro --template=yamadharma/course-directory-student-template –public



* Создал репозиторий под назвинем –

study\_2024-2025\_os-inro.git

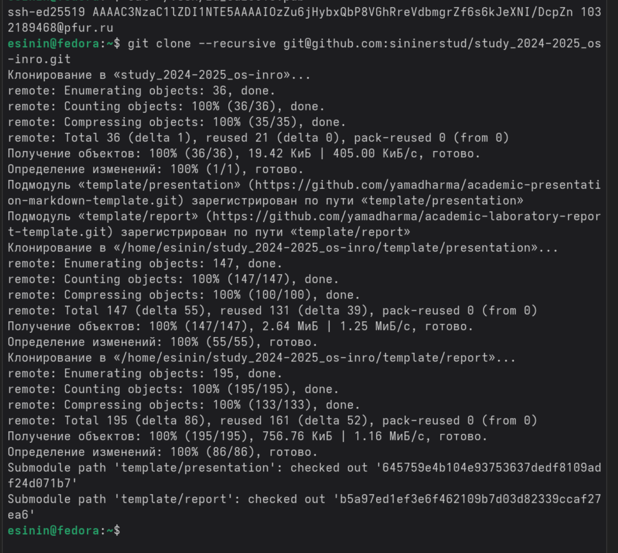
сделать клон не удалось

Отсутствует ssh ключ, пересоздаю

* git clone --recursive git@github.com:<owner>/study\_2024-2025\_os-inro.git os-inro

ФОТО

Клонировал в home/esinin/study



### **Настроим каталог курса**

Перейдём в каталог

cd ~/work/study/2022-2023/"Операционные системы"/os-intro

Удалим лишний файл

rm package.json

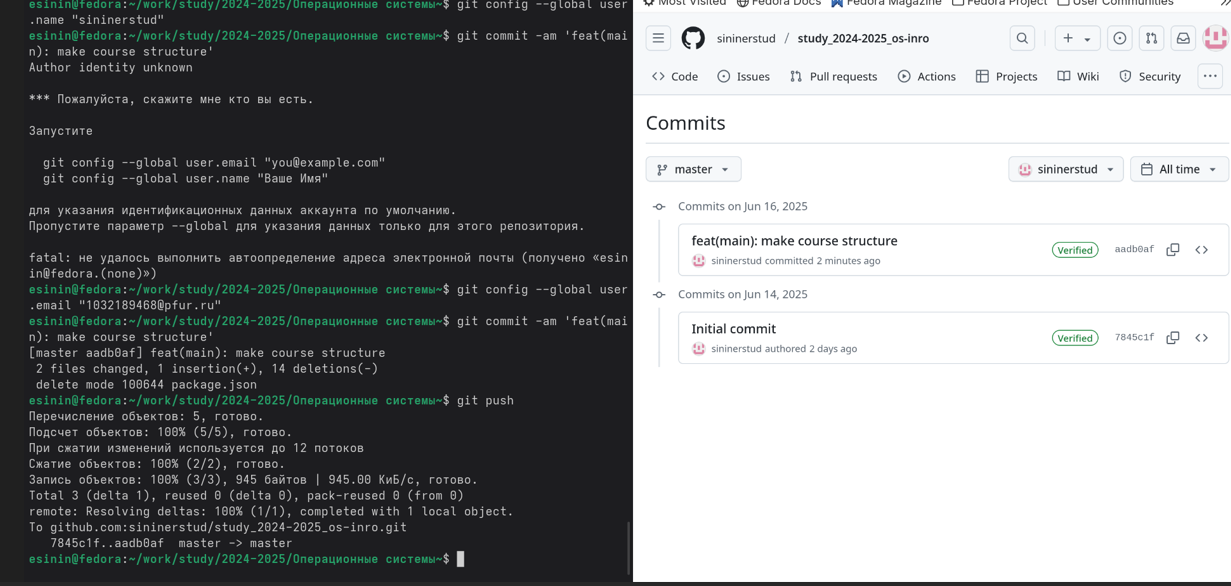
создадим необходимый каталог:

echo os-intro > COURSE

make

* Отправим файлы на сервер:
* git add .
* git commit -am 'feat(main): make course structure'

git push



**Выводы**:

Освоили основы работы со средством контроля версий

Применили навыки и умения по работе с git

# Контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?

Система, применяемая при работе нескольких человек над одним проектом.

1. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

Хранилище (репозиторий)- центральная база данных , хранит всю историю и ветви проекта.

Коммит (commit) фиксация изменений,  
каждый коммит имеет уникальный ИД(ХЭШ), хранит измениния, автора и дату.

История- последовательность комитов , цепочка изменений проека, . Вся история.  
  
Рабочая копия, локальная копия, извлеченная из хранилища , (привязана к определённому коммиту в истории.

1. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

Централизованные- одно хранилище , на сервере, Примеры-  CVS, Subversion,

Децентрализованные – Каждая копия- резервная копия со всей историей, Примеры- Git, Bazaar, Mercurial.

1. Опишите действия с VCS при единоличной работе с хранилищем.

Иницмализая – создать

Рабочая копия,  
Редактирование,  
Подготовка к коммиту, git add <####>

Создание коммита git commit -m “BLA BLA”

Просмотр истории git log

Откаты …ветвления …

1. Опишите порядок работы с общим хранилищем VCS.

1Клонирование

2Внесение изменений

3Подготовка и коммит git add, git commit (сохряняется в локальный)

4Получение изменений Fetch(проверить изменилось ли что ни будь, пока вы работали/Pull(слияние)  
5разрешение конфликтов, если возникли при слиянии

6 публикация изменений Push

1. Каковы основные задачи, решаемые инструментальным средством git?

Отслеживание изменений, фиксация истории

Распределение работы,

1. Назовите и дайте краткую характеристику командам git.

init Создание репозитория

clone создание копии

add Добавление изменения

commit -m Фиксирует изменения

status состояние рабочей копии

log история в текущей ветке

diff Показывает различия между Рабочей копией и послед коммитом

branch Показывает список веток

push Отправляет коммиты из локальной ветки

restore восстанавливает файл до состояния последнего коммита

reset Сброс текущей ветки к указанному коммиту

revert < новый коммит отменяя изменения

1. Приведите примеры использования при работе с локальным и удалённым репозиториями.

(проект см. выше)

1. Что такое и зачем могут быть нужны ветви (branches)?

Ветвь — это Указатель на определённый коммит в истории. Независимая линия разработки.  
нужно для изоляции каких-либо функций, параллельных работ.

1. Как и зачем можно игнорировать некоторые файлы при commit?

Для безопасности (пароли)  
для корректной работы (временные файлы, могут вызвать ошибки)

Как- создать файл. gitignore в корне репозитория