Отчёт по лабораторной работе №2

Дисциплина:Архитектура компьютеров и операционных систем

Румянцев Артём Олегович

Содержание

# 1 Цель работы

Цель данной лабораторной работы – изучение идеологии и применения средств контроля версий, освоение умения по работе с git.

# 2 Задание

1.Создать базовую конфигурацию для работы c 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 # Теоретическое введение

Основные команды git

Перечислим наиболее часто используемые команды git.

Создание основного дерева репозитория:

git init

Получение обновлений (изменений) текущего дерева из центрального репозитория:

git pull

Отправка всех произведённых изменений локального дерева в центральный репозиторий:

git push

Просмотр списка изменённых файлов в текущей директории:

git status

Просмотр текущих изменений:

git diff

Сохранение текущих изменений:

добавить все изменённые и/или созданные файлы и/или каталоги:

git add .

добавить конкретные изменённые и/или созданные файлы и/или каталоги:

git add имена\_файлов

удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории):

git rm имена\_файлов

Сохранение добавленных изменений:

сохранить все добавленные изменения и все изменённые файлы:

git commit -am ‘Описание коммита’

сохранить добавленные изменения с внесением комментария через встроенный редактор:

git commit новой ветки, базирующейся на текущей:

git checkout -b имя\_ветки

переключение на некоторую ветку:

git checkout имя\_ветки

(при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой)

отправка изменений конкретной ветки в центральный репозиторий:

git push origin имя\_ветки

слияние ветки с текущим деревом:

git merge –no-ff имя\_ветки

Удаление ветки:

удаление локальной уже слитой с основным деревом ветки:

git branch -d имя\_ветки

принудительное удаление локальной ветки:

git branch -D имя\_ветки

удаление ветки с центрального репозитория:

git push origin :имя\_ветки

# 3 Выполнение лабораторной работы

## 3.1 Установка программного обеспечения

Установка необходимого программного обеспечения git и gh через терминал с помощью команд: dnf install git и dnf install gh (рис.1).

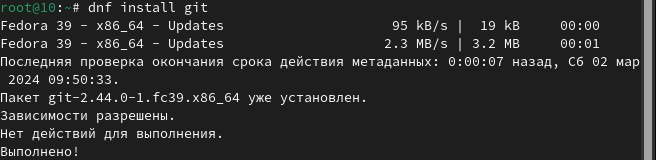


Рис 1

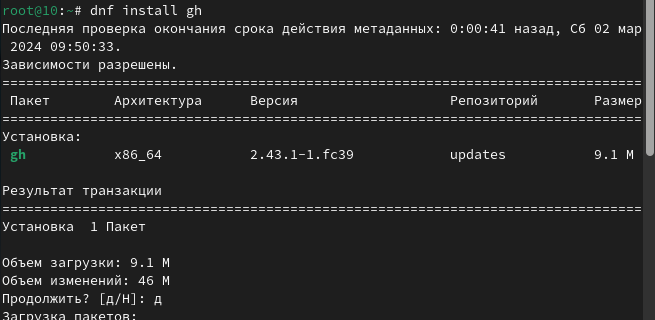


Рис 1

## 3.2 Базовая настройка git

Задаю в качестве имени и email владельца репозитория свои имя,фамилию и электронную почту(рис 2)

Рис 2

Рис 2

Настраиваю utf-8 в выводе сообщений git дли их корректного отображения

Рис 3

Рис 3

Начальной ветке задаю имя master

Рис 4

Рис 4

Задаю параметры autocrlf и safecrlf для корректного отображения конца строки

Рис 5

Рис 5

## 3.3 Создание ключа SSH

Создаю ключ ssh размером 4096 бит по алгоритму rsa

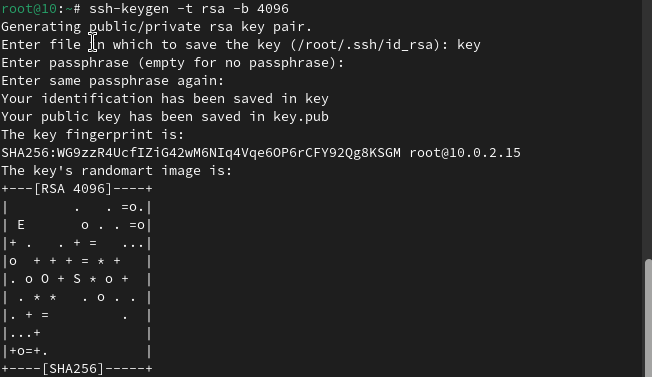


Рис 6

Создаю ключ ssh по алгоритму ed 25519

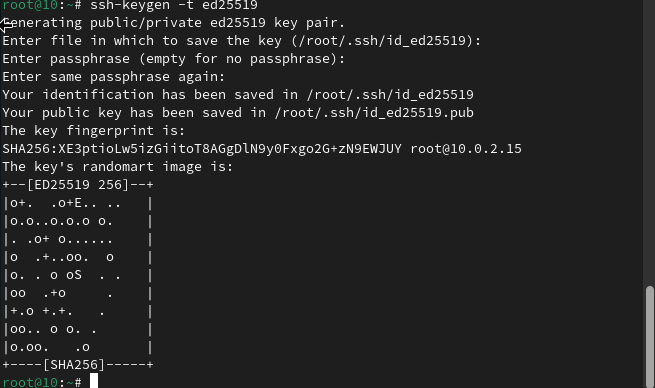


Рис 7

## 3.4 Создание Ключа GPG

Генерирую ключ GPG,затем выбираю тип ключа RSA and RSA, задаю макс. длину ключа;4096,оставляю неограниченный срок действия ключа. Далее отвечаю на вопросы программы о личной информации

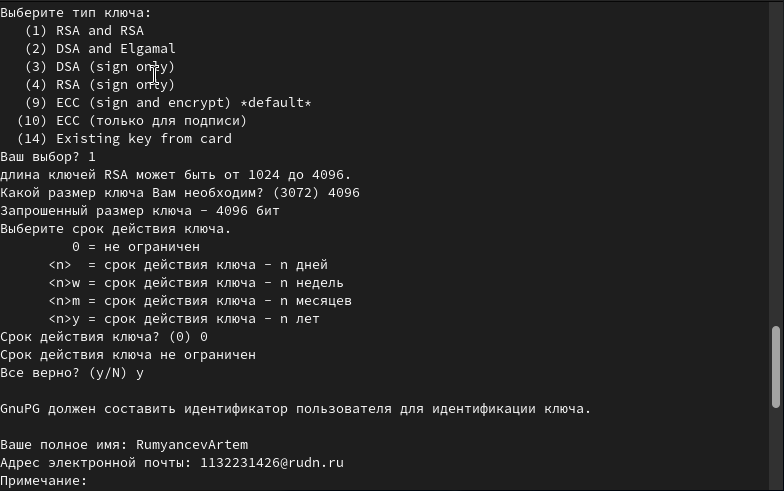


Рис 8

## 3.5 Регистрация на GitHub

У меня уже был создан аккаунт на Github, соответственно, основные данные аккаунта я так же заполнил и проводил его настройку, поэтому просто вхожу в свой аккаунт

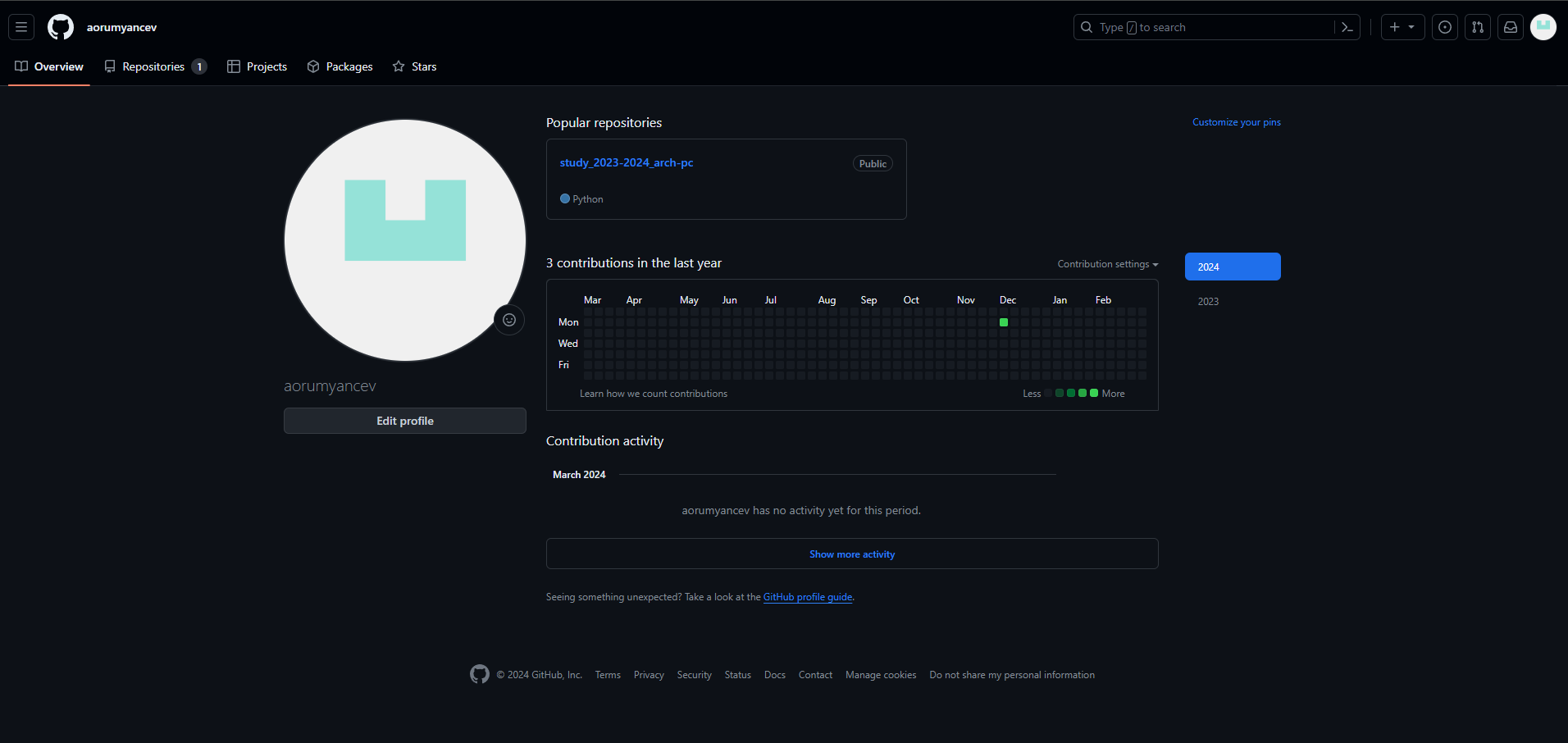


Рис 9

## 3.6 Добавление ключа GPG в GitHub

Вывожу список созданных ключей в терминал, ищу в результате запроса отпечаток ключа (последовательность байтов для идентификации более длинного, по сравнению с самим отпечатком, ключа), он стоит после знака слеша, копирую его в буфер обмена

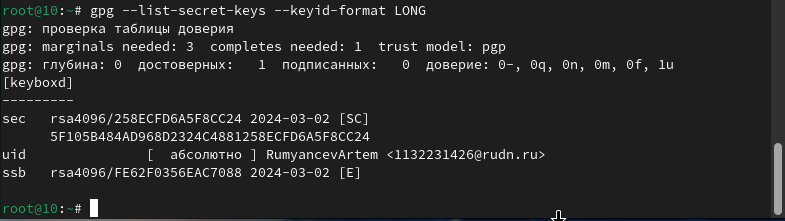


Рис 10

Ввожу в терминале команду,с помощью которой копирую сам ключ GPG в буфер обмена, за это отвечает утилита xclip.

Рис 11

Рис 11

Открываю настройки GirHub, ищу среди них добавление GPG ключа. Нажимаю на New GPG key и вставляю в поле ключ из буфера обмена. Я добавил ключ GPG на GitHub

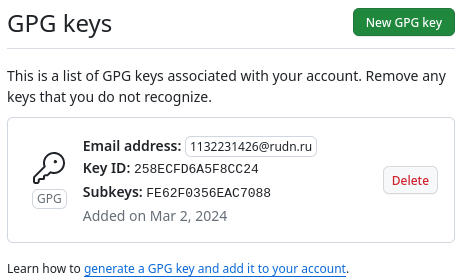


Рис 12

## 3.7 Настроить подписи Git

Настраиваю автоматические подписи коммитов git: используя введенный ранее email, указываю git использовать его при создании подписей коммитов.

Рис 13

Рис 13

## 3.8 Настройка gh

Начинаю авторизацию в gh, отвечаю на наводящие вопросы от утилиты, в конце выбираю авторизоваться через браузер

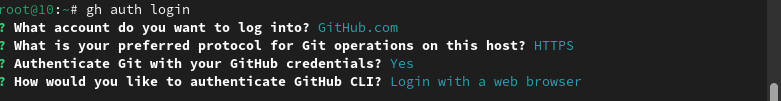


Рис 14

Завершаю авторизацию на сайте

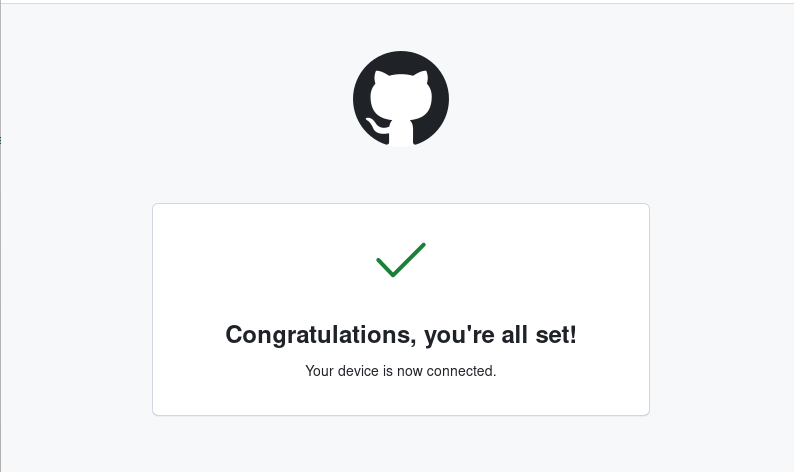


Рис 15

Вижу сообщение о завершении авторизации под именем aorumyancev

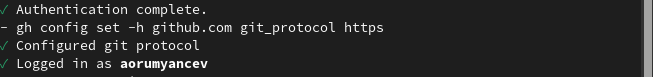


Рис 16

## 3.9 Создание репозитория курса на основе шаблона

Сначала создаю директорию с помощью утилиты mkdir и флага -p, кторый позволяет установить каталоги на всем указанном пути. После этого с помощью утилиты cd перехожу в только что созданную директорию “Операционные системы”. Далее в терминале ввожу команду gh repo create study\_2023-2024\_os-intro –template yamadharma/course-derctory-student-trmplate–public, чтобы создать репозиторий на основе шаблона репозитория. После этого клонирую репозиторий к себе в директорию, я указываю ссылку с протоколом https, а не ssh, потому что при авторизации в gh выбрала протокол https

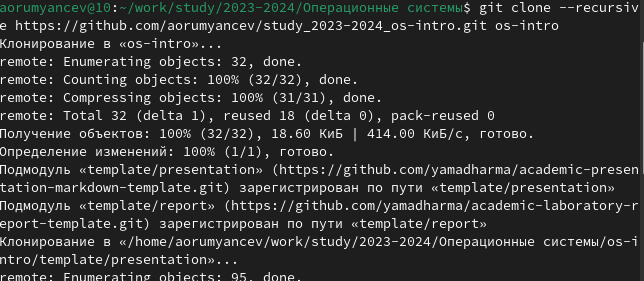


Рис 17

Перехожу в каталог курса с помощью утилиты cd, проверяю содеражание катлога с помощью утилиты ls

Рис 18

Рис 18

Удаляю лишние файлы с помощью утилиты rm, далее создаю неободимые каталоги используя makefile

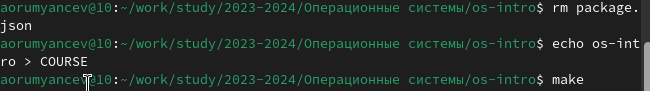


Рис 19

Добавляю все новые файлы для отправки на сервер(сохраняю добавленные изменения с помощью команды git add и комментирую их с помощью git commit

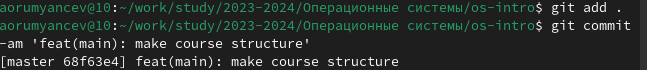


Рис 20

Отправляю файлы на сервер с помощью git push.

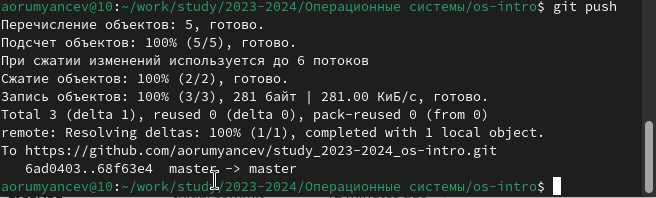


Рис 21

## 3.10 Выводы

При выполнении данной лабораторной работы я изучил идеологию и применение средств контроля версий, освоил умение по работе с git.