

Лабораторная работа №2

Дисциплина: Архитектура компьютера

Учаева Алёна Сергеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	13
	Список литературы	14

Список иллюстраций

4.1	Настройка git	9
4.2	Создание ключей	10
4.3	Экспорт ключей	11
4.4	Конфигурация подписей	11
4.5	Авторизация	12
4.6	Создание директории	12
4.7	Настройка директории	12

Список таблиц

1 Цель работы

Целью данной лабораторной работы является изучение идеологии и применение средств контроля версий и усвоение по работе с git.

2 Задание

1. Создать базовую конфигурацию для работы с git.
2. Создать ключ SSH.
3. Создать ключ PGP.
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.

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию,

отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

4 Выполнение лабораторной работы

Устанавливаю git и произвожу базовую настройку(рис. 4.1).

```
[aluchaeva@asuchaeva ~]$ git config --global user.name "Alena Uchaeva"  
[aluchaeva@asuchaeva ~]$ git config --global user.email "alenauchaeva06@gmail.com"  
[aluchaeva@asuchaeva ~]$ git config --global core.autocrlf input  
[aluchaeva@asuchaeva ~]$ git config --global core.safecrlf warn
```

Рис. 4.1: Настройка git

Создаю ssh и gpg ключи(рис. 4.2).

```

The key's randomart image is:
+---[RSA 4096]-----+
|  .o+O*+  |
|  o BB+   |
|  .++o.   |
|  oooo+   |
|  ..+So o  |
|  . o..0+. |
|  . .@o+ E  |
|  . Xo..   |
|  .o+o     |
+---[SHA256]-----+
[aluchaeva@asuchaeva ~]$ 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.

gpg: создан каталог '/home/aluchaeva/.gnupg'
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)

```

Рис. 4.2: Создание ключей

Экспортирую gpg ключ для авторизации на github(рис. 4.3).

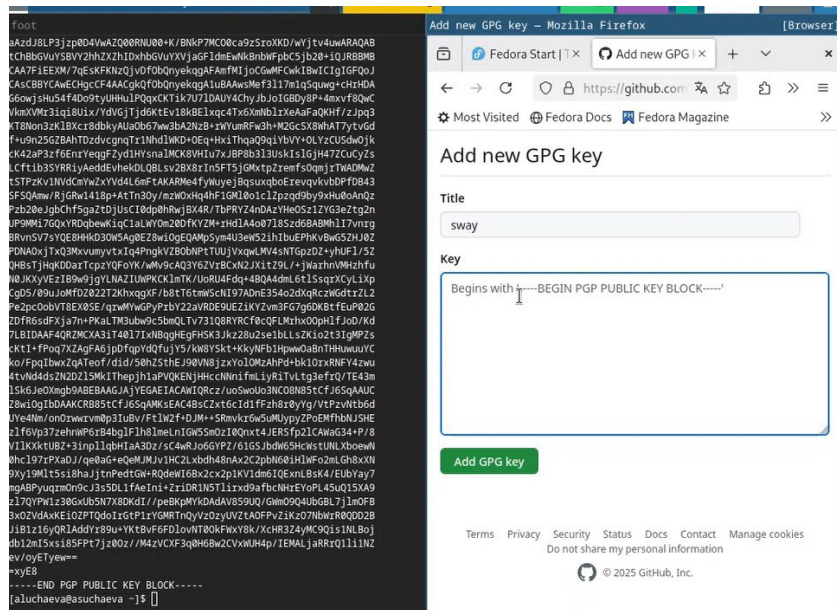


Рис. 4.3: Экспорт ключей

Настраиваю автоматические подписи(рис. 4.4).

```
aluchaeva@asuchaeva ~]$ git config --global user.signingkey 7CE6D09F27A4AA00
aluchaeva@asuchaeva ~]$ git config --global commit.gpgsign true
aluchaeva@asuchaeva ~]$ git config --global gpg.program $(which gpg2)
```

Рис. 4.4: Конфигурация подписей

Авторизуюсь через терминал(рис. 4.5).

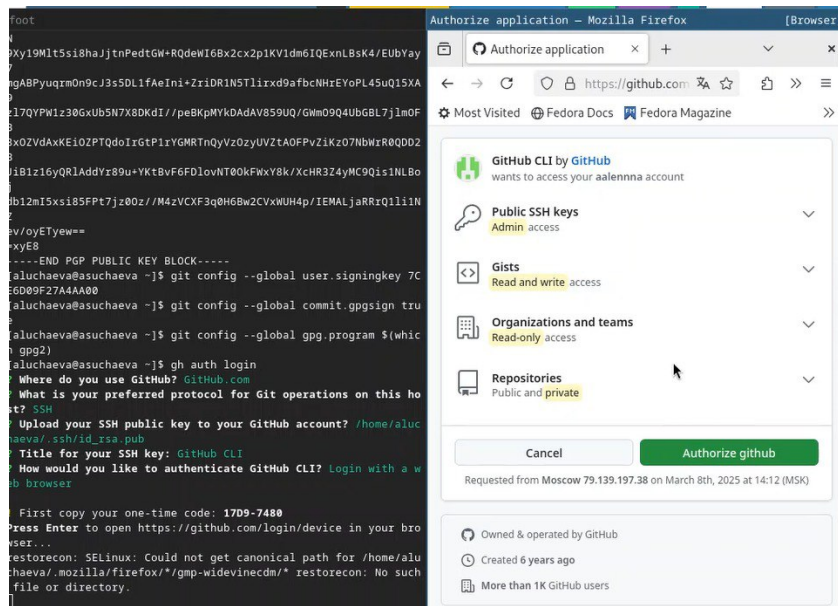


Рис. 4.5: Авторизация

Создаю директорию по шаблону(рис. 4.6).

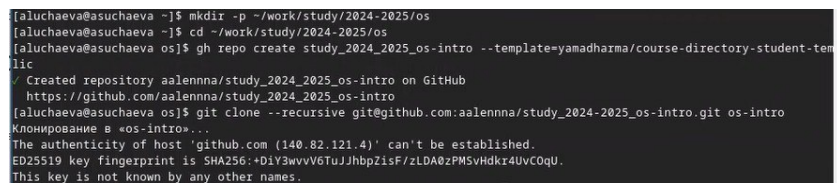


Рис. 4.6: Создание директории

Настраиваю рабочую директорию(рис. 4.7).

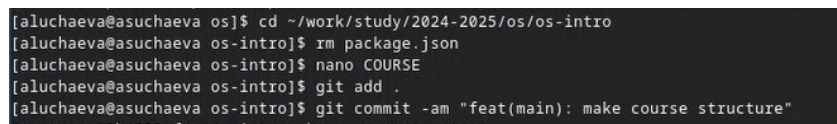


Рис. 4.7: Настройка директории

5 Выводы

В результате выполнения данной лабораторной работы я изучила идеологию и применение средств контроля версий и усвоение по работе с git.

Список литературы