Шаблон отчёта по лабораторной работе №2

Первоначальна настройка git

Абдуллахи Шугофа

Содержание

# 1 Цель работы

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

# 2 Задание

* Создать базовую конфигурацию для работы с git.
* Создать ключ SSH.
* Создать ключ PGP.
* Настроить подписи git.
* Зарегистрироваться на Github.
* Создать локальный каталог для выполнения заданий по предмету.

# 3 Выполнение лабораторной работы

### 3.0.1 Установка программного обеспечения

* устоновим git

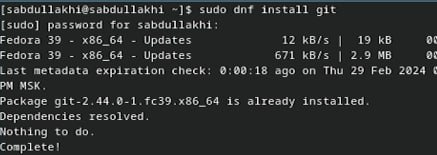


Рис. 1: git устоновки

* Установка gh

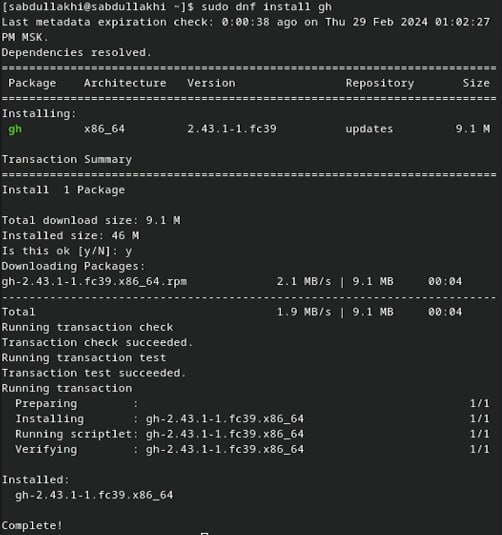


Рис. 2: gh устоновки

### 3.0.2 Базовая настройка git

* зададим имя и email владельца репозитория:
* ностроим utf-8 в выводде сообщений git:
* Настройте верификацию и подписание коммитов git (см. Верификация коммитов git с помощью GPG).
* Зададим имя начальной ветки (будем называть её master):
* Параметр autocrlf:
* Параметр safecrlf:

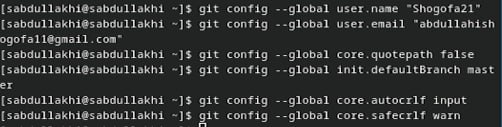


Рис. 3: задала имя, email, ностроила utf-8, имя master, autocrlf и safecrlf

## 3.1 Создайте ключи ssh

* по алгоритму rsa с ключём размером 4096 бит:

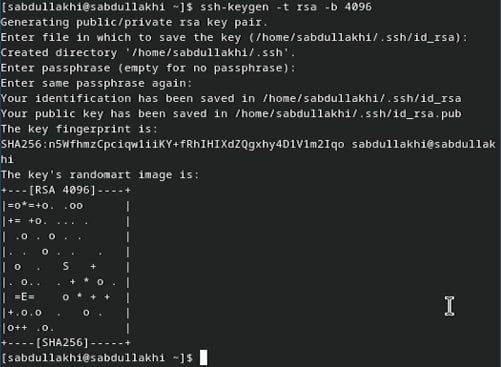


Рис. 4: создала ключ в рвзмере 4096 бит

* по алгоритму ed25519:

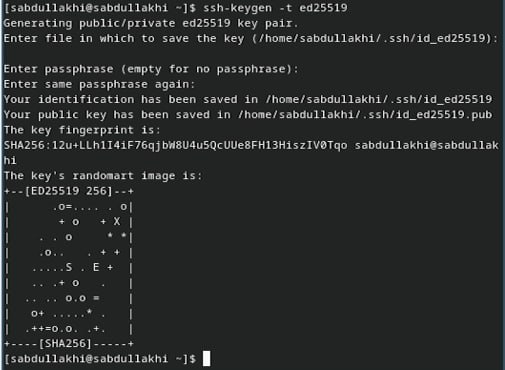


Рис. 5: создала ключ по алгоритму ed25519

## 3.2 Создайте ключи pgp

* Генерируем ключ
* Из предложенных опций выбираем:
  + выбрала тип RSA and RSA
  + размер 4096
  + выберила срок действия; значение по умолчанию — 0
* GPG запросит личную информацию, которая сохранится в ключе:
  + написала имя Shogofa Abdullahi
  + напасала электронной почты abdullahishogofa11@gmail.com
* При вводе email убедитесь, что он соответствует адресу, используемому на GitHub.
  + Комментарий. Можно ввести что угодно или нажать клавишу ввода, чтобы оставить это поле пустым.

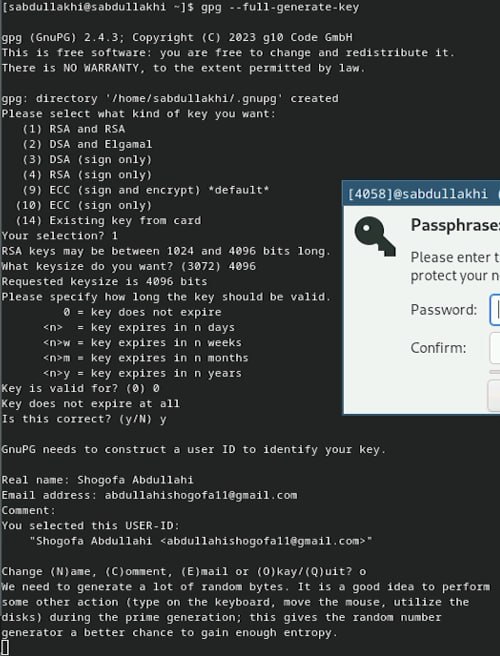


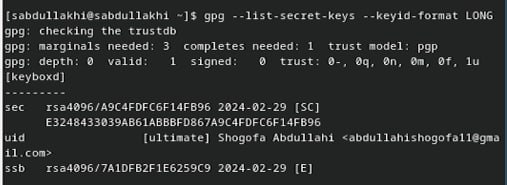
Рис. 6: генерируем ключ gpg

### 3.2.1 Настройка github

у меня уже был github, по этому не была надо опыят настройка github.

### 3.2.2 Добавление PGP ключа в GitHub

* Выводим список ключей и копируем отпечаток приватного ключа:



* Выводим список ключей и копируем отпечаток приватного ключа:



Рис. 7: скопировалв gpg ключ

* Перешла в настройки GitHub (https://github.com/settings/keys), нажмила на кнопку New GPG key и вставьла полученный ключ в поле ввода.

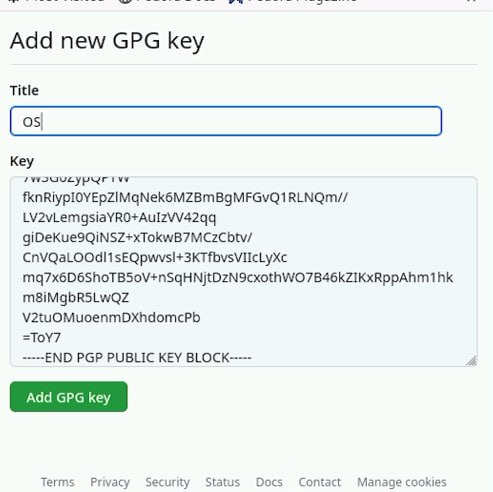


Рис. 8: создала gpg ключ

## 3.3 Настройка автоматических подписей коммитов git

* Используя введёный email, укажите Git применять его при подписи коммитов:

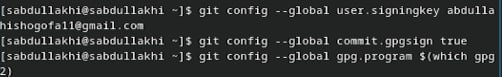


Рис. 9: написала email и остольные команды

### 3.3.1 Настройка gh

* Для начала необходимо авторизоваться



Рис. 10: ностроика gh, и ответла на несколько вопросов

* Авторизоваться можно через броузер.

# 4 Сознание репозитория курса на основе шаблона

* Создаём репозиторий курса на основе шаблона. Все нужные команды для создания были в указаниях к лабораторной работе. В 4 команде, вместо , указываем своё имя профиля на github.

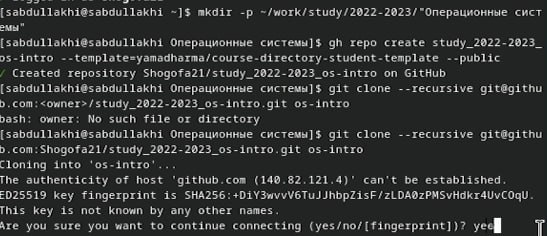


Рис. 11: срздала репозитории

## 4.1 Настройка каталога курса

* Настраиваем каталог курса. Для этого переходим в него командой: cd ~/work/study/2021-2022/“Операционные системы”/os-intro Далее командой ls проверяем, что мы в него перешли. В каталоге “os-intro” нам потребуется удалить файл “package.json”. Выполняем данную задачу командой: rm package.json

настроила каталога курса

Рис. 12: настроила каталога курса

* Создаём необходимые каталоги и отправляем наши файлы на сервер.

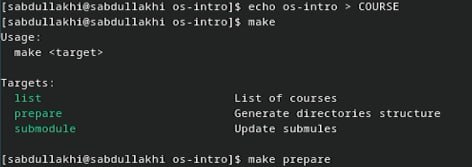


Рис. 13: Создайте необходимые каталоги

Отправьте файлы на сервер

Рис. 14: Отправьте файлы на сервер

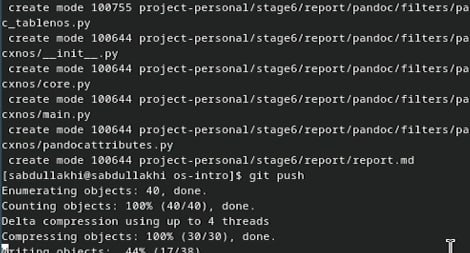


Рис. 15: Отправьте файлы на сервер

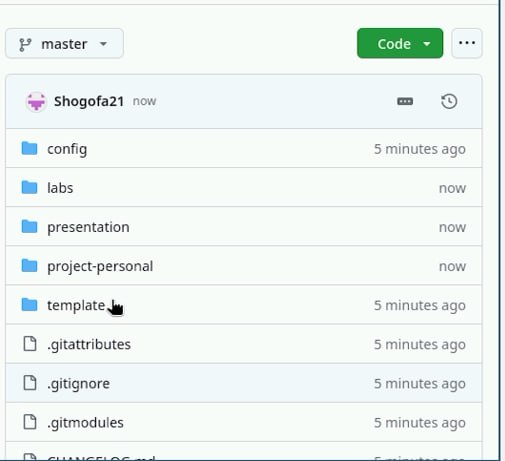


Рис. 16: все готов и провильно

# 5 **Контрольные вопросы:**

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

* Это программное обеспечение для облегчения работы с изменяющейся информацией. VCS позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

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

* Хранилище (repository), или репозитарий, — место хранения всех версий и служебной информации. Commit («[трудовой] вклад», не переводится) — синоним версии; процесс создания новой версии. История – место, где сохраняются все коммиты, по которым можно посмотреть данные о коммитах. Рабочая копия – текущее состояние файлов проекта, основанное на версии, загруженной из хранилища.

1. Что представляют собой и чем отличаются централизованные и децентрализованные VCS?

* Приведите примеры VCS каждого вида.
* Централизованные VCS: одно основное хранилище всего проекта и каждый пользователь копирует себе необходимые ему файлы из этого репозитория, изменяет и, затем, добавляет свои изменения обратно. Децентрализованные VCS: у каждого пользователя свой вариант (возможно не один) репозитория.

1. Опишите действия с VCS при единоличной работе с хранилищем.
2. Опишите порядок работы с общим хранилищем VCS.
3. Каковы основные задачи, решаемые инструментальным средством git?

* Git — это система управления версиями. У Git две основных задачи: первая — хранить информацию о всех изменениях в вашем коде, начиная с самой первой строчки, а вторая — обеспечение удобства командной работы над кодом.

1. Назовите и дайте краткую характеристику командам git.

* git –version (Проверка версии Git)
* git init (Инициализировать ваш текущий рабочий каталог как Git-репозиторий)
* git clone https://www.github.com/username/repo-name (Скопировать существующий удаленный Git-репозиторий)
* git remote (Просмотреть список текущих удалённых репозиториев Git)
* git remote -v (Для более подробного вывода)
* git add my\_script.py (Можете указать в команде конкретный файл).
* git add . (Позволяет охватить все файлы в текущем каталоге, включая файлы, чье имя начинается с точки)
* git commit -am “Commit message” (Вы можете сжать все индексированные файлы и отправить коммит).
* git branch (Просмотреть список текущих веток можно с помощью команды branch)
* git –help (Чтобы узнать больше обо всех доступных параметрах и командах)
* git push origin master (Передать локальные коммиты в ветку удаленного репозитория).

1. Приведите примеры использования при работе с локальным и удалённым репозиториями.
2. Что такое и зачем могут быть нужны ветви (branches)?

* Ветки нужны, чтобы несколько программистов могли вести работу над одним и тем же проектом или даже файлом одновременно, при этом не мешая друг другу. Кроме того, ветки используются для тестирования экспериментальных функций: чтобы не повредить основному проекту, создается новая ветка специально для экспериментов.

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

* Игнорируемые файлы — это, как правило, артефакты сборки и файлы, генерируемые машиной из исходных файлов в вашем репозитории, либо файлы, которые по какой-либо иной причине не должны попадать в коммиты.

# 6 Выводы

В ходе выполнения лабораторной работы изучили идеологию и применение средств контроля версий, а также освоили умения по работе с git.