Отчет по лабораторной работе №1

Дисциплина: Информационная безопасность

Боровикова Карина Владимировна

Содержание

# 1 Цель работы

Целью данной лабораторной работы является подготовка рабочего пространства для следующих лабораторных работ

# 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 Теоретическое введение

## 3.1 Системы контроля версий

Системы контроля версий (Version Control System, VCS) применяются при работе несколь- ких 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 боль- шинства 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 (авто- матически 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.

Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с нескольки- ми версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.

Среди классических VCS наиболее известны CVS, Subversion, а среди распределён- ных — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

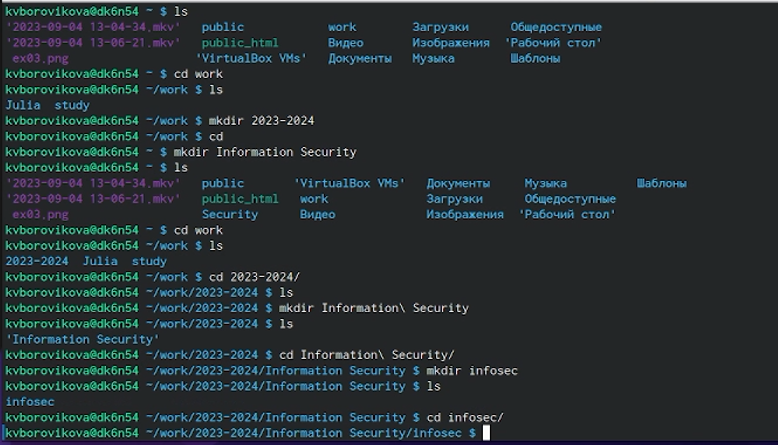
## 3.2 Markdown

Markdown (произносится маркда́ун) — облегчённый язык разметки, созданный с целью обозначения форматирования в простом тексте, с максимальным сохранением его читаемости человеком, и пригодный для машинного преобразования в языки для продвинутых публикаций (HTML, Rich Text и других).

Более подробно о Git и Markdown см. в [1,2].

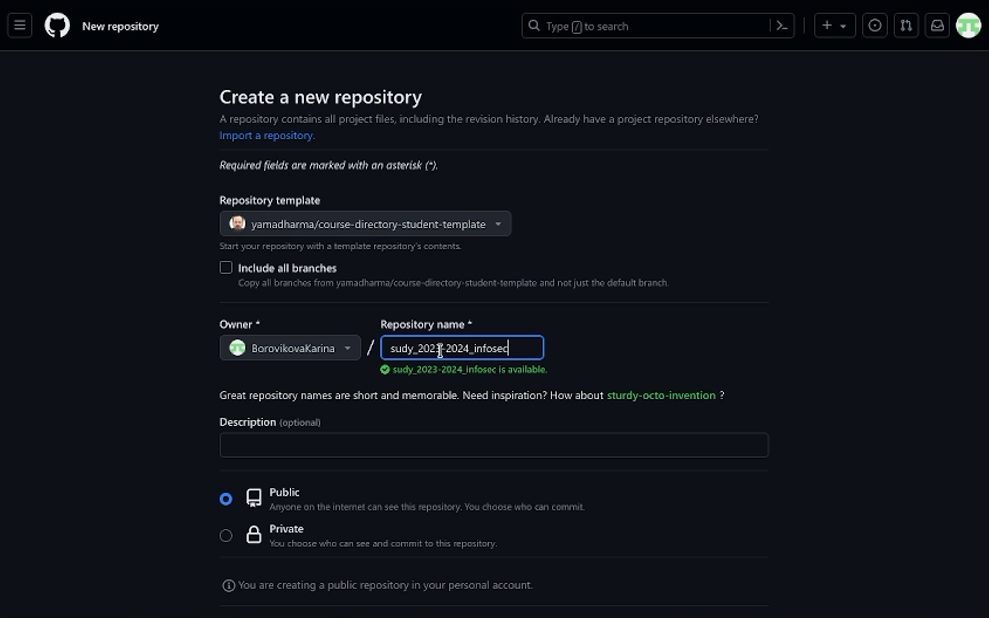
# 4 Выполнение лабораторной работы

1. Работать будем на компьютере в дисплейном классе, поэтому в настройке виртуального окружения нет необходимости, поэтому сразу создаем рабочее пространство на компьютере в соответствии с указаниями к лабораторной работе и переходим в папку infosec: (рис. ??).



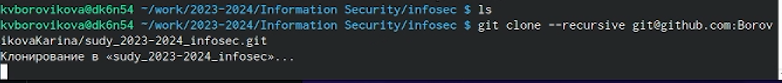
Переходим в infosec

1. Создаем репозиторий по данному в задании шаблону. (рис. ??).



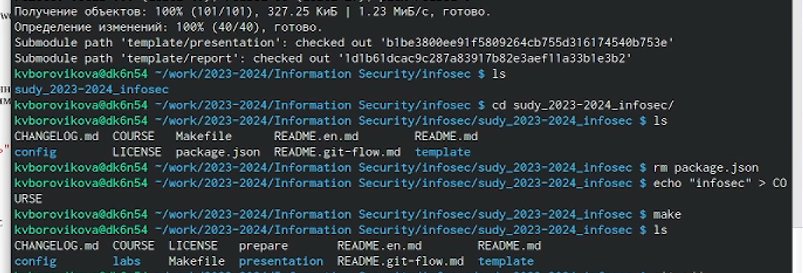
Создаем репозиторий

1. Клонируем созданный нами репозиторий рекурсивно на наш компьютер. (рис. ??)



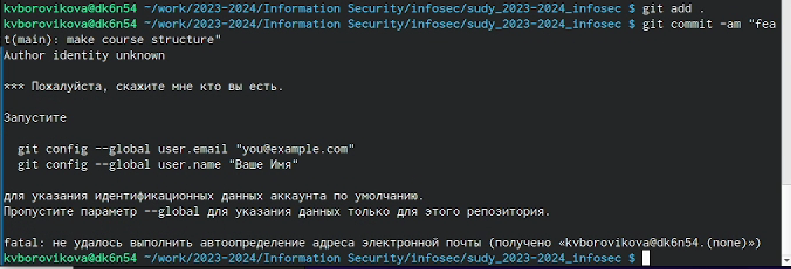
Рекурсивно клонируем наш репозиторий себе на компьютер

1. Далее переходим в склонированный репозиторий и выполняем действия указанные в лабораторной работе. (рис. ??)
   * yдаляем packaje.json;
   * в файл COURSE записываем название курса “infosec”
   * запускаем make и создаем структуру курса



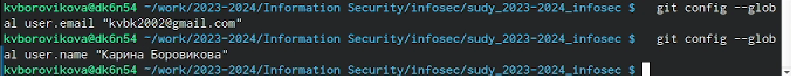
Действия по созданию структуры курса

1. Пробуем добавить изменения в индекс и закоммитить изменения, при попытке сделать коммит видим сообщение о том, что мы - неизвестный автор и о том, что коммит не выполнен. (рис. ??)



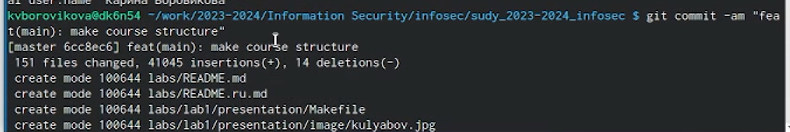
Пробуем сделать коммит

1. Конфигурируем почту и имя для git. (рис. ??)



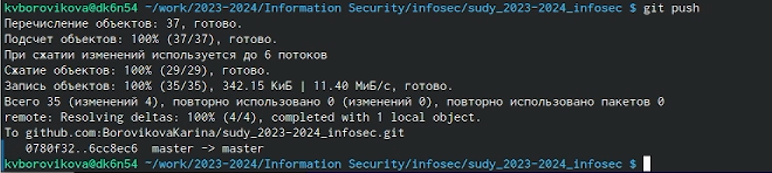
Конфигурируем почту и имя

1. Делаем коммит. (рис. ??)



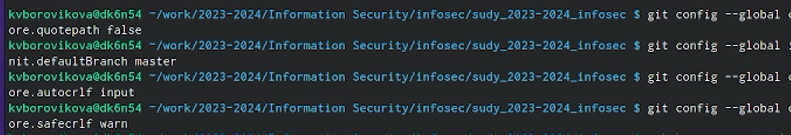
Делаем коммит, он прошел успешно

1. Пушим изменения на git. (рис. ??)



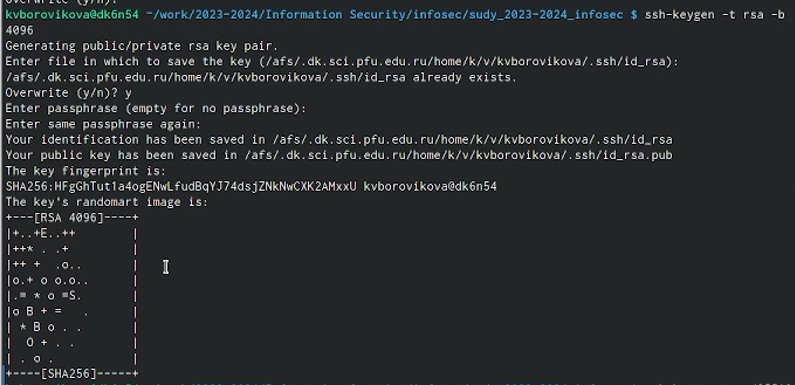
Пушим изменения на git

1. Продолжаем конфигурировать git. (рис. ??)



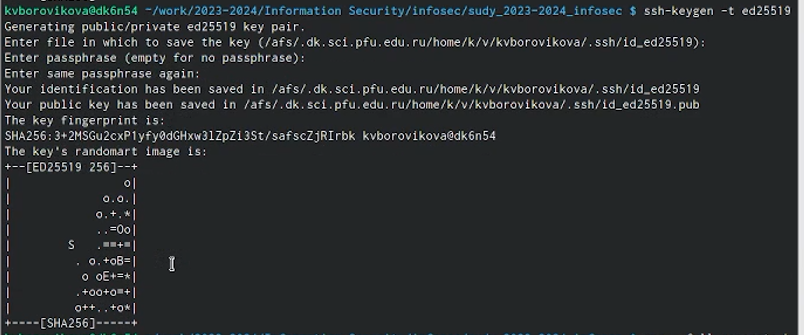
Конфигурируем Git

1. Генерируем ключ rsa, видим, что он уже существует, перезапишем и добавим этот ключ на GitHub. (рис. ??)



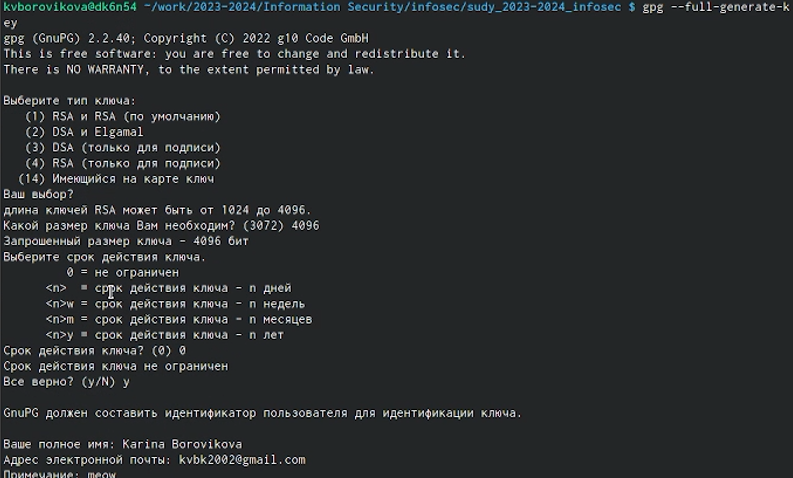
Генерация ключа rsa

1. Генерируем ключ ed25519, добавляем этот ключ на GitHub. (рис. ??)



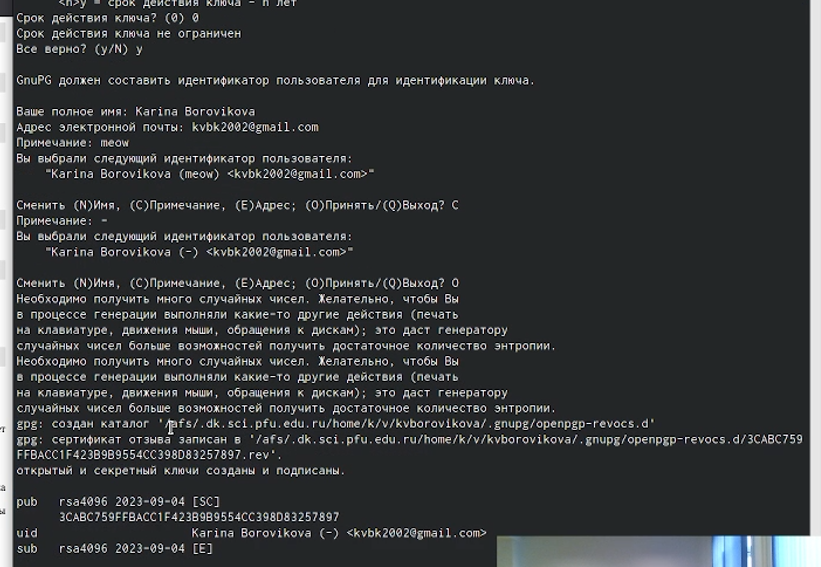
Генерация ключа ed25519

1. Начинаем генерацию ключа gpg. (рис. ??)



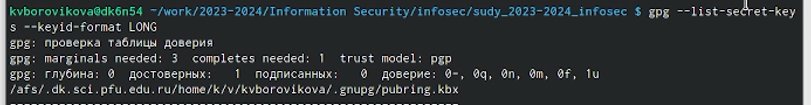
Генерация ключа gpg

1. Генерация завершена. (рис. ??)



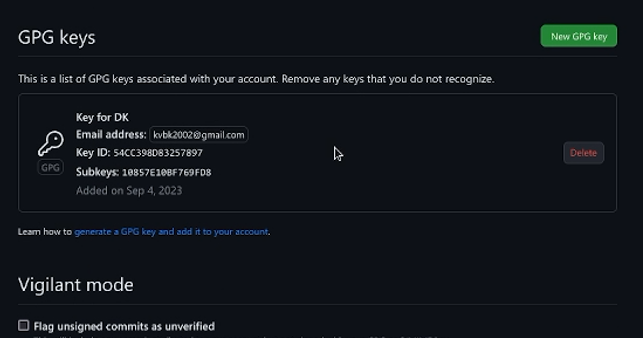
Генерация ключа gpg

1. Выводим список ключей и копируем отпечаток приватного ключа. Далее копируем ключ в буфер обмена и добавляем на GitHub (рис. ??)



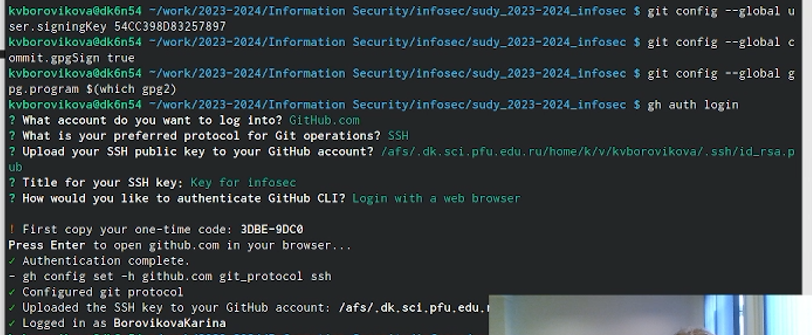
Вывод списка ключей

1. Видим, что ключ добавлен на GitHub (рис. ??)



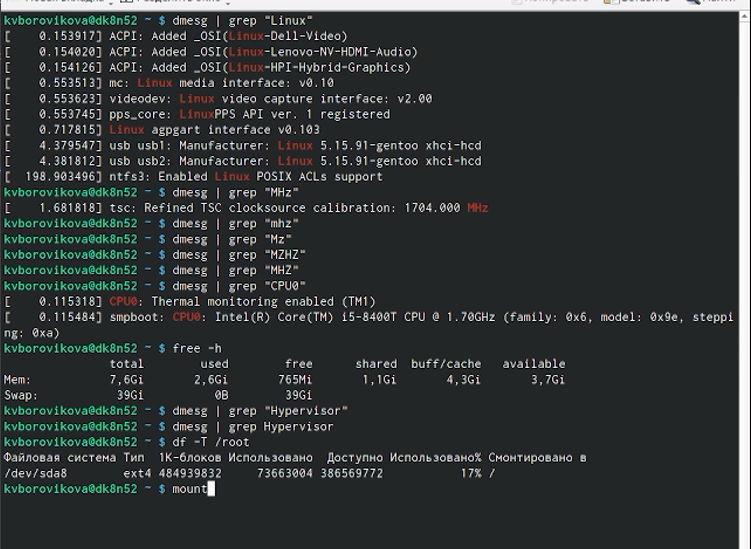
Ключ добавлен на GitHub

1. Заканчиваем конфигурацию git c помощью gh. (рис. ??)



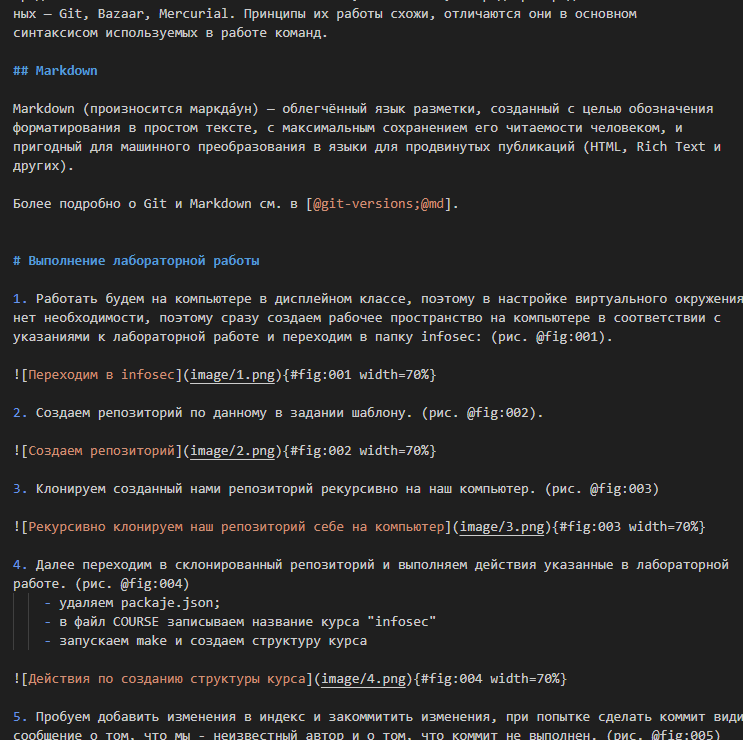
Конфигурируем gh

1. Также сделаем некоторые действия из первого файла с настройкой виртуального окружения. (рис. ??)



Действия из первого файла, последовательность монтирования также были показана, на записи есть данный момент

1. Пишем отчет к лабораторной работе на облегченном языке разметки Markdown. (рис. ??)



Отчет на Markdown

# 5 Выводы

В ходе лабораторной работы нам удалось подготовить пространство для следующих лабораторных работ.

# Список литературы

1. Управление версиями. Git. Введение в систему контроля версий git. [Электронный ресурс]. 2016. URL: <https://asrumiantceva.github.io/post/writing-technical-content/>.

2. Markdown [Электронный ресурс]. 2016. URL: <https://ru.wikipedia.org/wiki/Markdown>.