Отчёт по лабораторной работе

Дисциплина: Операционные системы

Елена Михайловна Горпинич

Содержание

# Цель работы

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

# 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

# Теоретическое введение

Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды git с различными опциями. Благодаря тому, что Git является рапределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией. ## Основные команды git git add Команда добавляет содержимое рабочего каталога в индекс для последующего коммита. git status Команда git status показывает состояния файлов в рабочем каталоге и индексе git diff Команда git diff используется для вычисления разницы между любыми двумя Git деревьями.

git difftool Команда git difftool просто запускает внешнюю утилиту сравнения для показа различий в двух деревьях, на случай если вы хотите использовать что-либо отличное от встроенного просмотрщика git diff.

git commit Команда git commit берёт все данные, добавленные в индекс с помощью git add, и сохраняет их слепок во внутренней базе данных, а затем сдвигает указатель текущей ветки на этот слепок.

git reset Команда используется в основном для отмены изменений.

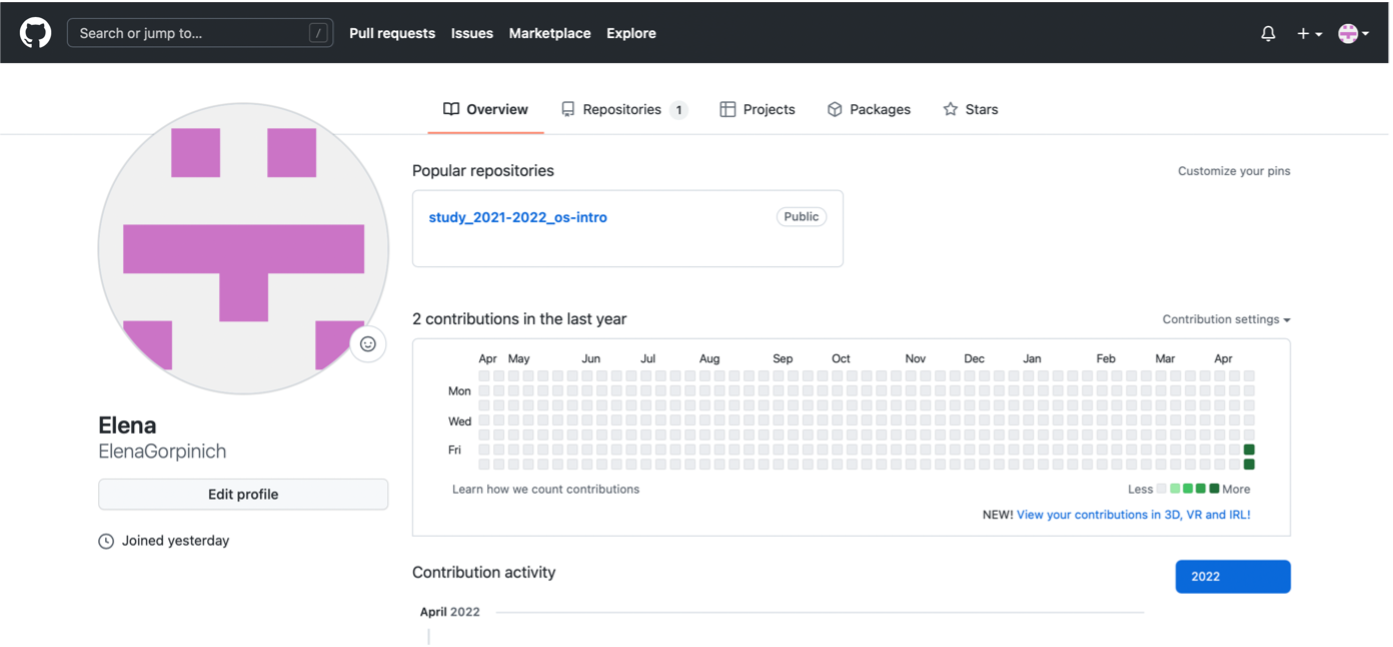
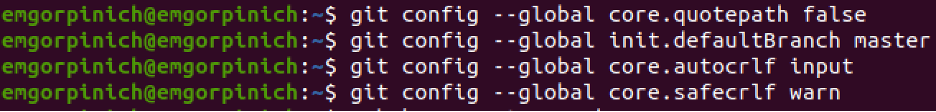
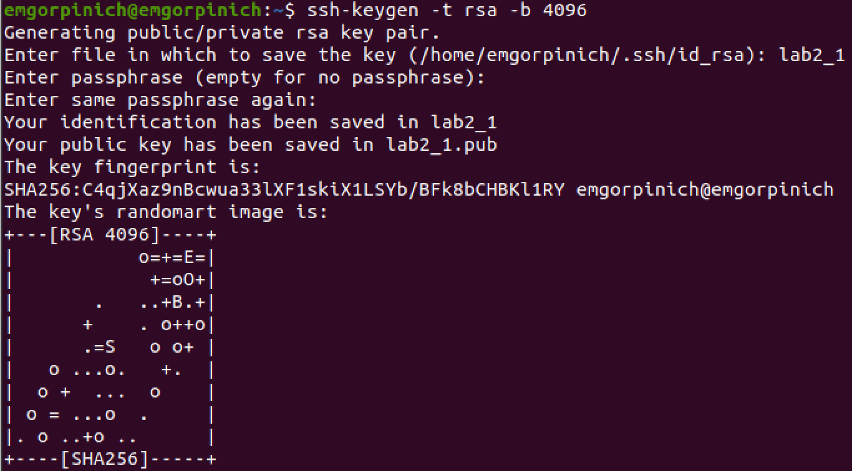
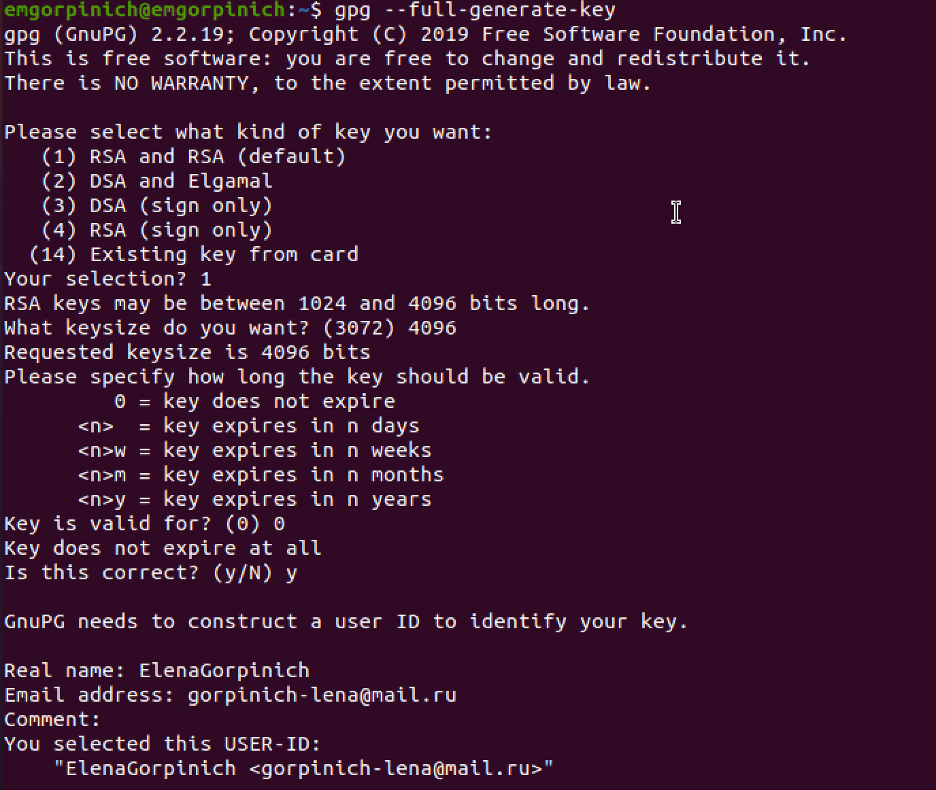
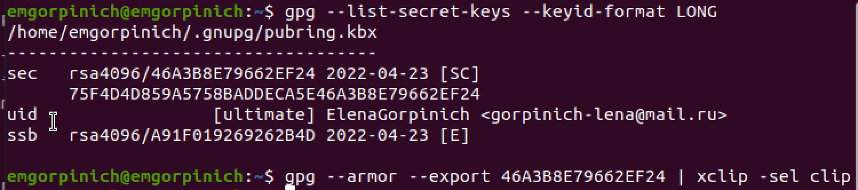
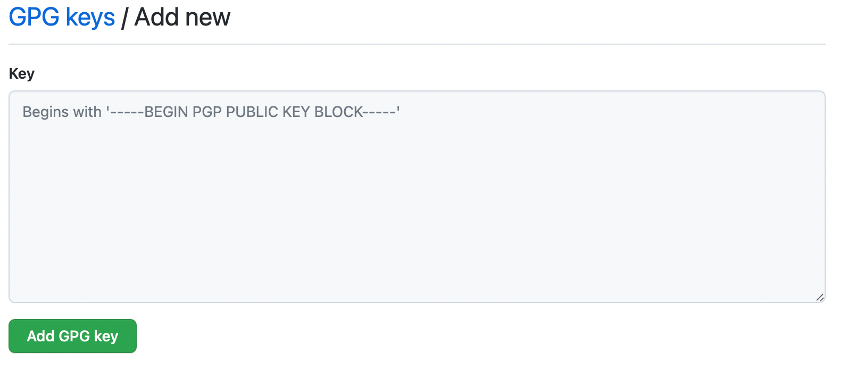
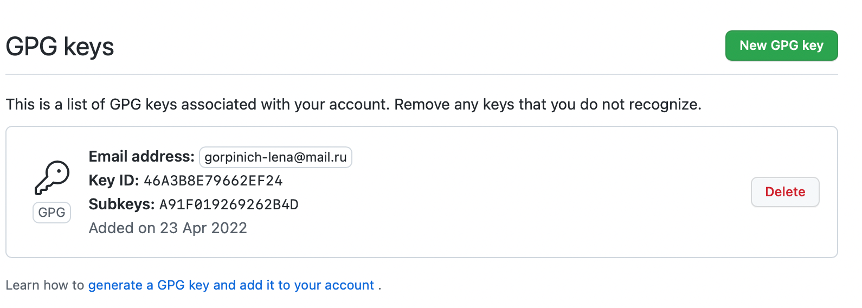
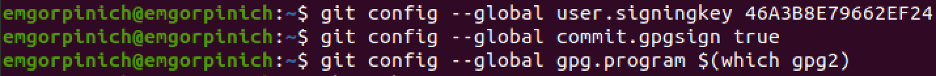
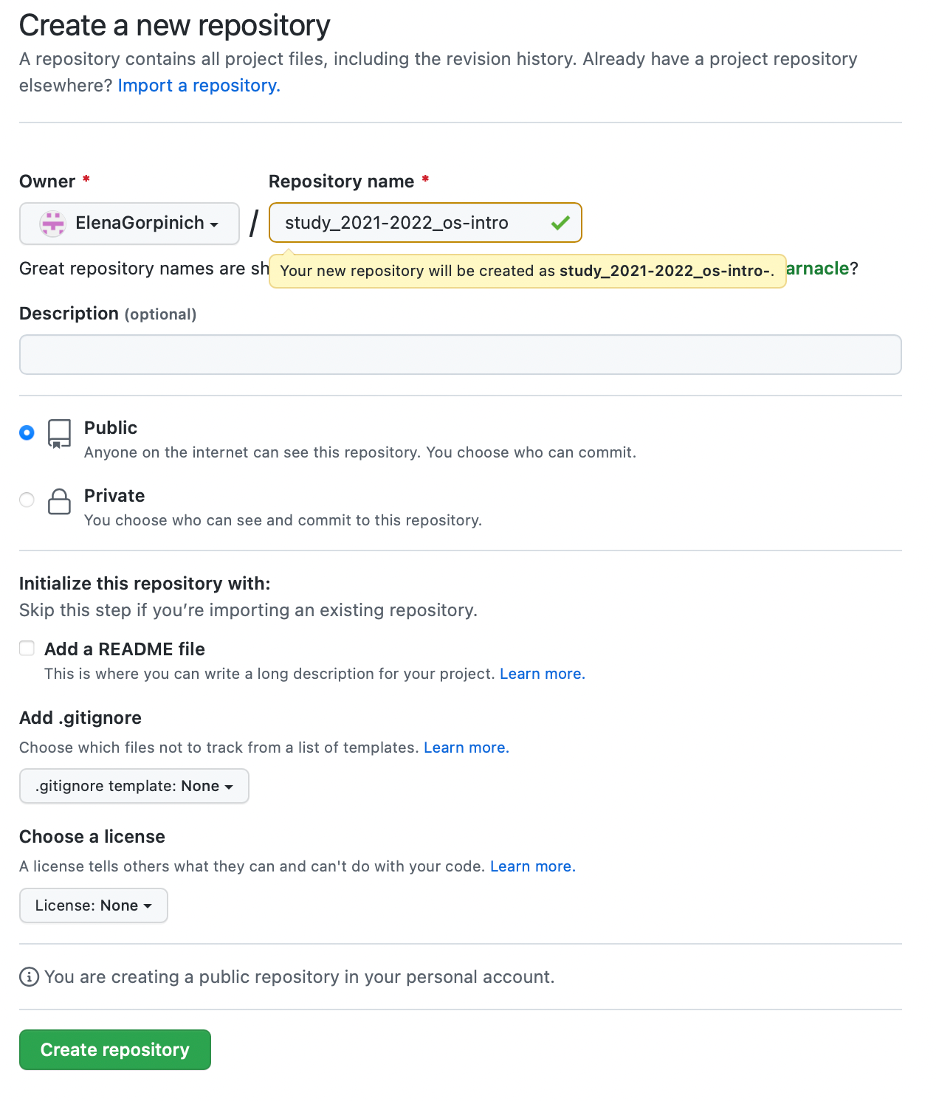
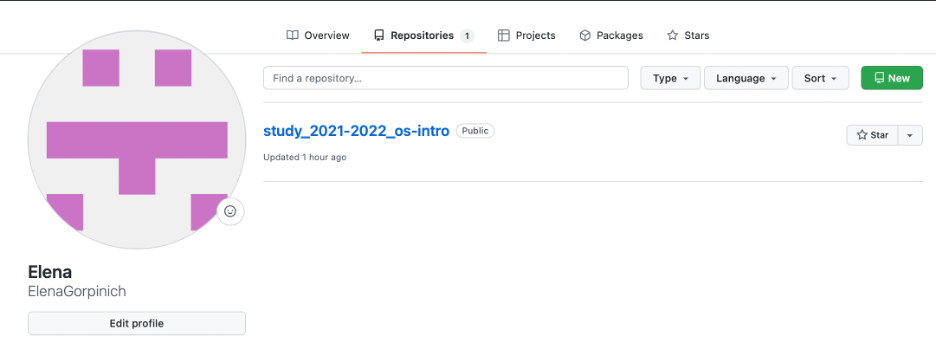
git rm Команда git rm используется в Git для удаления файлов из индекса и рабочей копии.

git mv Команда git mv — это всего лишь удобный способ переместить файл, а затем выполнить git add для нового файла и git rm для старого.

git clean Команда git clean используется для удаления мусора из рабочего каталога.

# Выполнение лабораторной работы

## Настройка GitHub

1. Изначально создаём учетную запись и заполняем основные данные на официальном сайте. https://github.com.(рис.[1]) 
2. Затем синхронизируем учетную запись с компьютером с помощью терминала. (рис.[2]) 2
3. Начинаем базовую настройку git: настроим utf-8 в выводе сообщений git, верификацию и подписание коммитов git, зададим имя начальной̆ ветки (будем называть её master), параметр autocrlf, параметр safecrlf (рис.[3]) 
4. Создадим ключ ssh по алгоритму rsa с ключевым размером 4096 бит(рис.[4]) 
5. Создадим ключ pgp. Генерируем ключ и из предложенных опций выбираем опции описанные в условии лабораторной работы. (рис.[5]) 
6. Добавим PGP ключ в GitHub. Для этого выводим список ключей и скопируем отпечаток приватного ключа. Скопируем наш сгенерированный PGP ключ в буфер обмена. С помощью настроек GitHub (https://github.com/settings/keys), добавим наш скопированный ключ, вставив его в поле ввода, которое появилось после нажатия кнопки New GPG key (рис.[7-9])   
7. Настроим автоматических подписей коммитов git (рис.[10]) 
8. Создадим новый репозиторий курса на основе шаблона и далее настроим каталог курса, удалив ненужные файлы, создав необходимые каталоги и отправив файлы на сервер (рис.[11-12])   # Вывод

В данной лабораторной работе я научилась работать с Github (создавать и привязывать учетную запись к компьютеру). Разобрала основные команды git и рассмотрела, как их применять их при работе с Github. Изучила идеологию и научилась применять средства контроля версий

# Список литературы