Отчёт по лабораторной работе №5

Операционные системы

Лисенков Е.Р.

Содержание

# Цель работы

Приобретение практических навыков взаимодействия пользователя с системой посредством командной строки.

# Задание

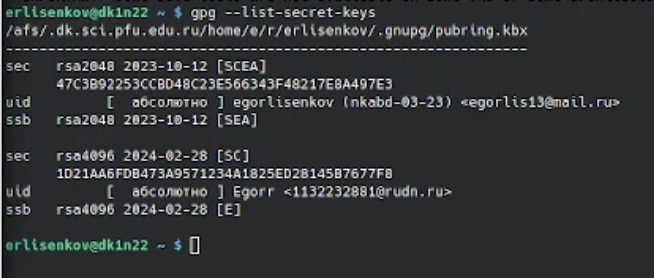
1. Установить необходимое программное обеспечение.
2. Изучить принцип работы менеджера паролей pass.
3. Установить дополнительное программное обеспечение.

# Теоретическое введение

Менеджер паролей pass создан в рамках идеалогии Unix. В нём данные хранятся в файловой системе в виде каталогов и файлов. Файлы шифруются с помощью GPG-ключа. Структура базы паролей может быть произвольной. Если необходимо использовать дополнительное программное обеспечение, семантику необходимо заложить в структуру базы паролей. На данный момент существует 2 основных реализации: pass и gopass.

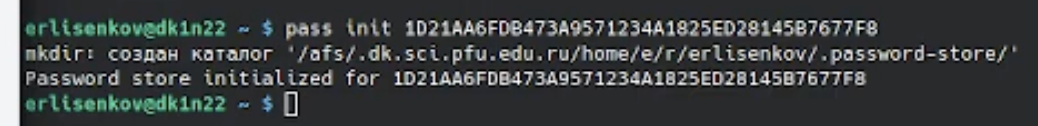
# Выполнение лабораторной работы

Выполним команду для просмотра списка ключей (рис.1).



Просмотрим список ключей

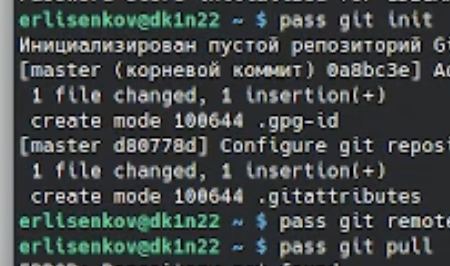
Инициализируем хранилище (рис.2).



Инициализируем хранилище

Синхронизация с git (рис.3 рис.4)

Создадим структуру git:  
  
1. pass git init  
  
Также можно задать адрес репозитория на хостинге (репозиторий необходимо предварительно создать):  
  
2. pass git remote add origin git@githubcom:<git\_username>/<git\_repo>.git  
  
Для синхронизации выполняется следующая команда:  
  
3. pass git pull  
  
4. pass git push



Синхронизация с git

Синхронизация с git

Синхронизация с git

Скачаем интерфейс для взаимодействия с броузером (native messaging) (рис.5).

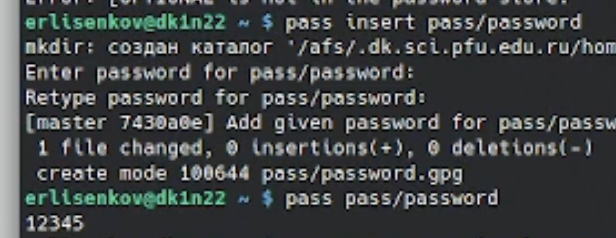
Репозиторий: https://github.com/browserpass/browserpass-native  
  
Gentoo:  
  
emerge www-plugins/browserpass

Синхронизация с git

Синхронизация с git

Сохранение пароля (рис.6 рис.8)

1.Добавим новый пароль:   
  
pass insert [OPTIONAL DIR]/[FILENAME]  
  
2.Отобразите пароль для указанного имени файла:  
  
ass [OPTIONAL DIR]/[FILENAME]  
  
3.Замените существующий пароль:  
  
pass generate --in-place FILENAME

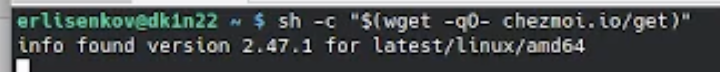


Сохранение пароля

Сохранение пароля

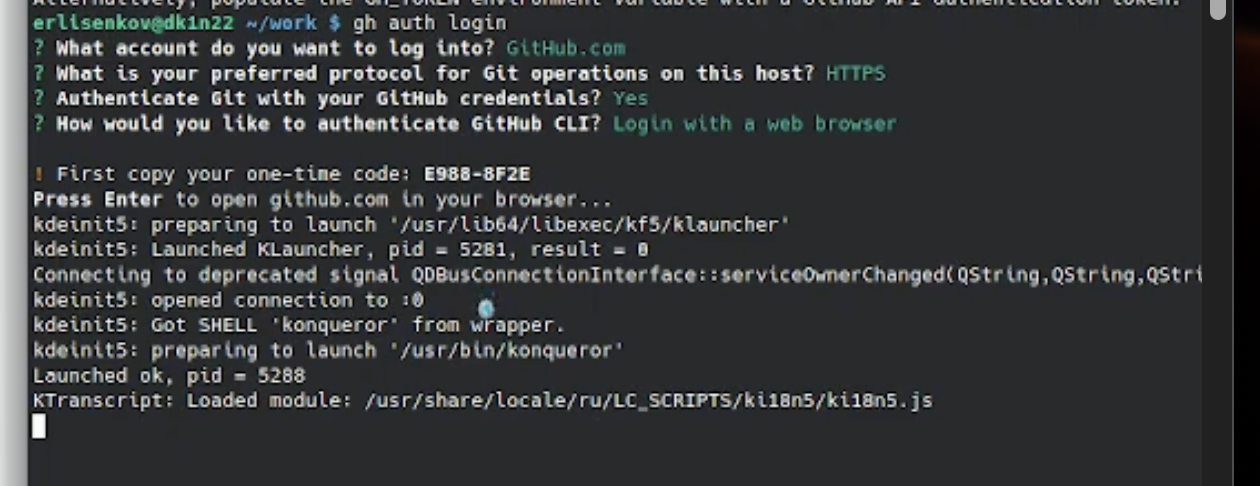
Сохранение пароля

Установим бинарный файл. Скрипт определяет архитектуру процессора и операционную систему и скачивает необходимый файл (рис.8).



Установим бинарный файл

Установим соединение с git (рис.9)



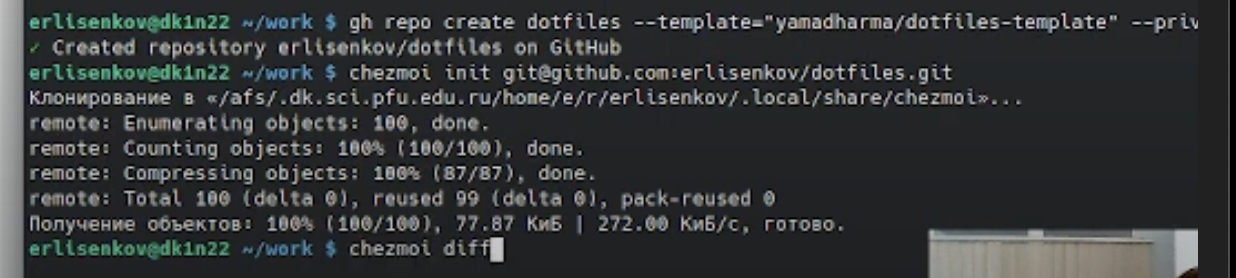
Установим соединение с git

Создам собственный репозиторий с помощью утилит

Буду использовать утилиты командной строки для работы с github.  
  
Создам свой репозиторий для конфигурационных файлов на основе шаблона:  
  
gh repo create dotfiles --template="yamadharma/dotfiles-template" --private

Подключу репозитория к своей системе

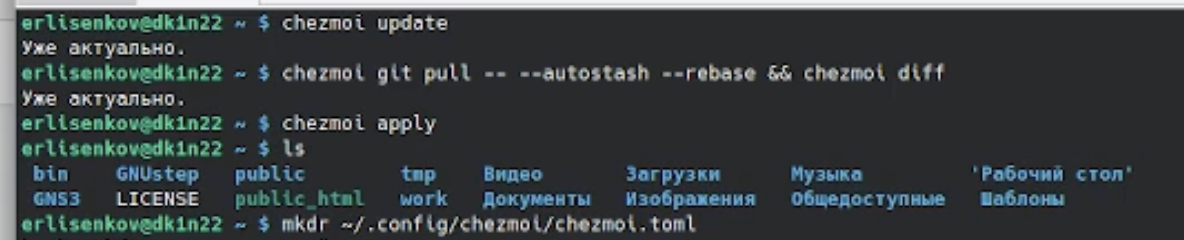
Инициализирую chezmoi с вашим репозиторием dotfiles:  
  
chezmoi init git@github.com:<username>/dotfiles.git  
  
Проверю, какие изменения внесёт chezmoi в домашний каталог, запустив:  
  
chezmoi diff  
  
Если устраит изменения, внесённые chezmoi, запустите:  
  
chezmoi apply -v



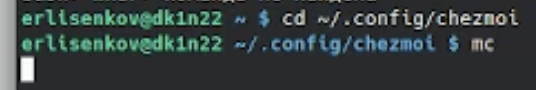
Создам собственный репозиторий

Ежедневные операции c chezmoi

Извлеку последние изменения из репозитория и примените их  
  
 Можно извлечь изменения из репозитория и применить их одной командой:  
  
 chezmoi update  
  
 Это запускается git pull --autostash --rebase в вашем исходном каталоге, а затем chezmoi apply.  
  
Извлеку последние изменения из своего репозитория и посмотрите, что изменится, фактически не применяя изменения  
  
 Выполню:  
  
 chezmoi git pull -- --autostash --rebase && chezmoi diff  
  
 Это запускается git pull --autostash --rebase в вашем исходном каталоге, а chezmoi diff затем показывает разницу между целевым состоянием, вычисленным из вашего исходного каталога, и фактическим состоянием.  
  
 применяем их:  
  
 chezmoi apply



Ежедневные операции c chezmoi



Ежедневные операции c chezmoi

# Выводы

В ходе лабораторной работы я освоила навыки использования менеджера паролей pass.