Очет по лабораторной работе № 5

Операционные системы

Абронина Алиса Кирилловна

Содержание

Список иллюстраций

Список таблиц

# 1 Цель работы

Познакомиться с pass, gopass, native messaging, chezmoi. Научиться пользоваться этими утилитами, синхронизировать их с гит.

# 2 Задание

1. Установка pass и gopass   
2. Настройка интерфейса с броузером.   
3. Добавление нового пароля  
4. Дополнительное ПО  
5. Создание собственного репозитория с помощью утилит  
6. Подключение репозитория к своей системе  
7. Ежедневные операции с chezmoi

# 3 Выполнение лабораторной работы

## 3.1 Установка pass и gopass

Установка pass и gopass (рис. 1) (рис. 2).

![Рис. 1: Установка pas](data:application/octet-stream;base64,)

Рис. 1: Установка pas

![Рис. 2: Установка gopass](data:application/octet-stream;base64,)

Рис. 2: Установка gopass

Настройка ключа GPG. Просмотр списка ключей и инициализация хранилища ( рис. 3).

![Рис. 3: list secret and pass init](data:application/octet-stream;base64,)

Рис. 3: list secret and pass init

Синхронизация с git. Создадим структуру git с помощью pass git init ( рис. 4)

![Рис. 4: pass git pull, pass git push](data:application/octet-stream;base64,)

Рис. 4: pass git pull, pass git push

Прямые изменения. Следует заметить, что отслеживаются только изменения, сделанные через сам gopass (или pass). Если изменения сделаны непосредственно на файловой системе, необходимо вручную закоммитить и выложить изменения( рис. 5) ( рис. 6)

![Рис. 5: git add? git commit](data:application/octet-stream;base64,)

Рис. 5: git add? git commit

![Рис. 6: git push](data:application/octet-stream;base64,)

Рис. 6: git push

## 3.2 Настройка интерфейса с броузером.

Для взаимодействия с броузером используем интерфейс native messaging. Поэтому кроме плагина к броузеру устанавливаем программа, обеспечивающая интерфейс native messaging. ( рис. 7)

![Рис. 7: browserpass](data:application/octet-stream;base64,)

Рис. 7: browserpass

## 3.3 Добавление нового пароля

Добавляем новый пароль( рис. 8).

![Рис. 8: pass insert [OPTIONAL DIR]/[FILENAME]](data:application/octet-stream;base64,)

Рис. 8: pass insert [OPTIONAL DIR]/[FILENAME]

Отображаем пароль для указанного имени файла ( рис. 9).

![Рис. 9: pass [OPTIONAL DIR]/[FILENAME]](data:application/octet-stream;base64,)

Рис. 9: pass [OPTIONAL DIR]/[FILENAME]

## 3.4 Дополнительное ПО

Установливаем дополнительное программное обеспечение( рис. 10).

![Рис. 10: sudo dnf -y install  ](data:application/octet-stream;base64,)

Рис. 10: sudo dnf -y install

Устанавливаем шрифты ( рис. 11) ( рис. 12).

![Рис. 11: ч1](data:application/octet-stream;base64,)

Рис. 11: ч1

![Рис. 12: ч2](data:application/octet-stream;base64,)

Рис. 12: ч2

Устанавливаем бинарный файл. Скрипт определяет архитектуру процессора и операционную систему и скачивает необходимый файл( рис. 13).

![Рис. 13: sh -c](data:application/octet-stream;base64,)

Рис. 13: sh -c

## 3.5 Создание собственного репозитория с помощью утилит

Будем использовать утилиты командной строки для работы с github. Создадим свой репозиторий для конфигурационных файлов на основе шаблона( рис. 14).

![Рис. 14: gh repo create dotfiles –template=“yamadharma/dotfiles-template” –private](data:application/octet-stream;base64,)

Рис. 14: gh repo create dotfiles –template=“yamadharma/dotfiles-template” –private

## 3.6 Подключение репозитория к своей системе

Инициализируем chezmoi с вашим репозиторием dotfiles( рис. 15).

![Рис. 15: chezmoi init](data:application/octet-stream;base64,)

Рис. 15: chezmoi init

Проверяем, какие изменения внесёт chezmoi в домашний каталог. Если устраивают изменения, внесённые chezmoi( рис. 16) ( рис. 17).

![Рис. 16: chezmoi diff](data:application/octet-stream;base64,)

Рис. 16: chezmoi diff

![Рис. 17: chezmoi apply -v](data:application/octet-stream;base64,)

Рис. 17: chezmoi apply -v

## 3.7 Ежедневные операции c chezmoi

Извлекаем последние изменения из репозитория и применяем их. Извлеките последние изменения из своего репозитория и посмотрите, что изменится, фактически не применяя изменения. Если вы довольны изменениями, вы можете применить их( рис. 18).

![Рис. 18: chezmoi update, git pull, apply](data:application/octet-stream;base64,)

Рис. 18: chezmoi update, git pull, apply

Можно автоматически фиксировать и отправлять изменения в исходный каталог в репозиторий. Эта функция отключена по умолчанию.

Чтобы включить её, добавьте в файл конфигурации ~/.config/chezmoi/chezmoi.toml следующее( рис. 19).

![Рис. 19: изменение файла](data:application/octet-stream;base64,)

Рис. 19: изменение файла

# 4 Выводы

Мы познакомились с pass, gopass, native messaging, chezmoi. Научились пользоваться этими утилитами, синхронизировали их с гит.