Отчёта по лабораторной работе № 5

Операционные системы

Сахно Алёна Юрьевна

Содержание

Список иллюстраций

Список таблиц

# 1 Цель работы

Изучиь основные свойства структура базы паролей и уметь работать с pass. # Задание

1. Научиться работать с Pass # Теоретическое введение

Структура базы паролей

* Структура базы может быть произвольной, если Вы собираетесь использовать её напрямую, без промежуточного программного обеспечения. Тогда семантику структуры базы данных Вы держите в своей голове.
* Если же необходимо использовать дополнительное программное обеспечение, необходимо семантику заложить в структуру базы паролей.

Семантическая структура базы паролей

* Рассмотрим пользователя user в домене example.com, порт 22.
* Отсутствие имени пользователя или порта в имени файла означает, что любое имя пользователя и порт будут совпадать:
* example.com.pgp
* Соответствующее имя пользователя может быть именем файла внутри каталога, имя которого совпадает с хостом. Это полезно, если в базе есть пароли для нескольких пользователей на одном хосте:
* example.com/user.pgp
* Имя пользователя также может быть записано в виде префикса, отделенного от хоста знаком @:
* user@example.com.pgp
* Соответствующий порт может быть указан после хоста, отделённый двоеточием (:):
* example.com:22.pgp example.com:22/user.pgp user@example.com:22.pgp
* Эти все записи могут быть расположены в произвольных каталогах, задающих Вашу собственную иерархию.

Способы создания файла шаблона

* При первом добавлении файла передайте аргумент –template:
* chezmoi add –template ~/.zshrc
* Если файл уже контролируется chezmoi, но не является шаблоном, можно сделать его шаблоном:
* chezmoi chattr +template ~/.zshrc
* Можно создать шаблон вручную в исходном каталоге, присвоив ему расширение .tmpl:
* chezmoi cd $EDITOR dot\_zshrc.tmpl
* Шаблоны в каталоге .chezmoitemplates должны создаваться вручную:
* chezmoi cd mkdir -p .chezmoitemplates cd .chezmoitemplates $EDITOR mytemplate

Редактирование файла шаблона

* Используйте chezmoi edit:
* chezmoi edit ~/.zshrc
* Чтобы сделанные вами изменения сразу же применялись после выхода из редактора, используйте опцию –apply:
* chezmoi edit –apply ~/.zshrc

# 2 Выполнение лабораторной работы

Менеджер паролей pass

Установка

Fedora  
  
 pass; dnf install gopass

(рис. 1).

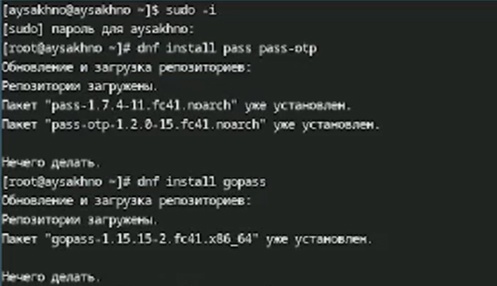


Рис. 1: Установка

Настройка

Ключи GPG  
  
 Просмотр списка ключей:  
  
 gpg --list-secret-keys  
  
 Если ключа нет, нужно создать новый:  
  
 gpg --full-generate-key

(рис. 2).

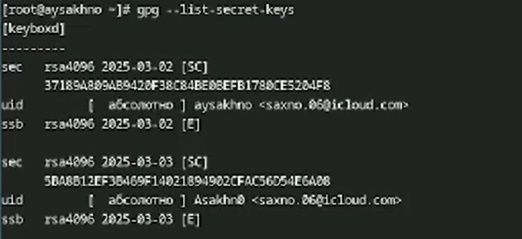


Рис. 2: Настройка

Инициализация хранилища

Инициализируем хранилище:  
  
pass init <gpg-id or email>

(рис. 3).

Рис. 3: Инциаоизация хранилища

Рис. 3: Инциаоизация хранилища

Синхронизация с git

* Создадим структуру git:
* pass git init
* Также можно задать адрес репозитория на хостинге (репозиторий необходимо предварительно создать):
* pass git remote add origin git@github.com:<git\_username>/<git\_repo>.git

(рис. 4).

Рис. 4: Инциаоизация хранилища

Рис. 4: Инциаоизация хранилища

Для синхронизации выполняется следующая команда:

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:

cd ~/.password-store/  
 git add .  
 git commit -am 'edit manually'  
 git push

(рис. 5).

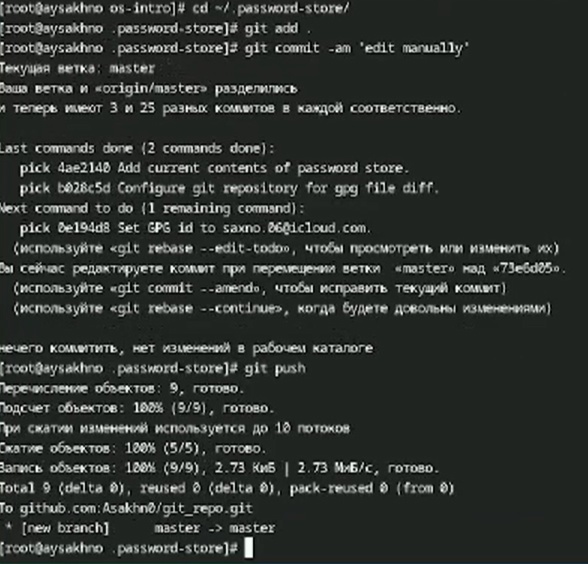


Рис. 5: Прямые изменения

* Проверить статус синхронизации модно командой
* pass git status

Настройка интерфейса с броузером

* Для взаимодействия с броузером используется интерфейс native messaging.
* Поэтому кроме плагина к броузеру устанавливается программа, обеспечивающая интерфейс native messaging.
* Плагин browserpass Fedora
* dnf copr enable maximbaz/browserpass  
   dnf install browserpass

(рис. 6).

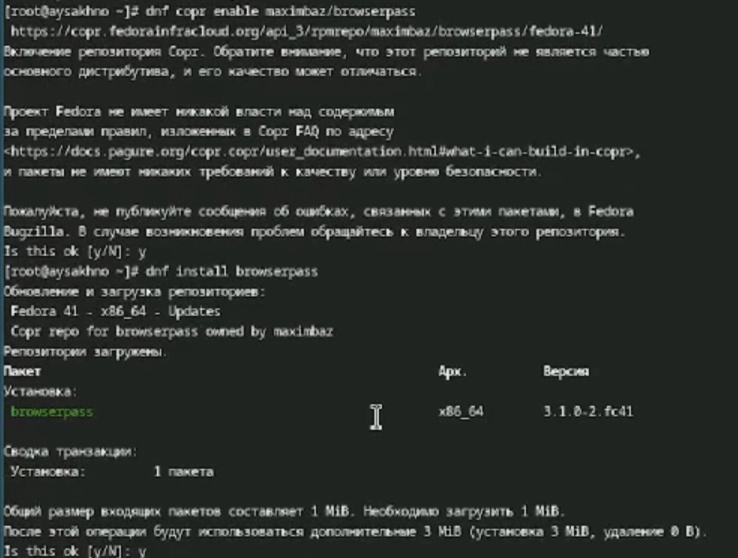


Рис. 6: Настройка интерфейса с броузером

Сохранение пароля

* Добавить новый пароль
* Выполните:  
    
   pass insert [OPTIONAL DIR]/[FILENAME]  
    
   OPTIONAL DIR: необязательное имя каталога, определяющее файловую структуру для вашего хранилища паролей;  
   FILENAME: имя файла, который будет использоваться для хранения пароля.  
    
   Отобразите пароль для указанного имени файла:  
    
   pass [OPTIONAL DIR]/[FILENAME]  
    
   Замените существующий пароль:  
    
   pass generate --in-place FILENAME

(рис. 7).

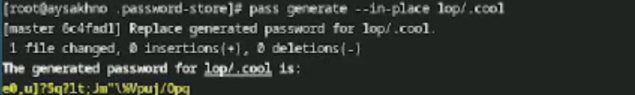


Рис. 7: Сохранение пароля

Управление файлами конфигурации

Дополнительное программное обеспечение

Установите дополнительное программное обеспечение:  
  
sudo dnf -y install \  
 dunst \  
 fontawesome-fonts \  
 powerline-fonts \  
 light \  
 fuzzel \  
 swaylock \  
 kitty \  
 waybar swaybg \  
 wl-clipboard \  
 mpv \  
 grim \  
 slurp

(рис. 8).

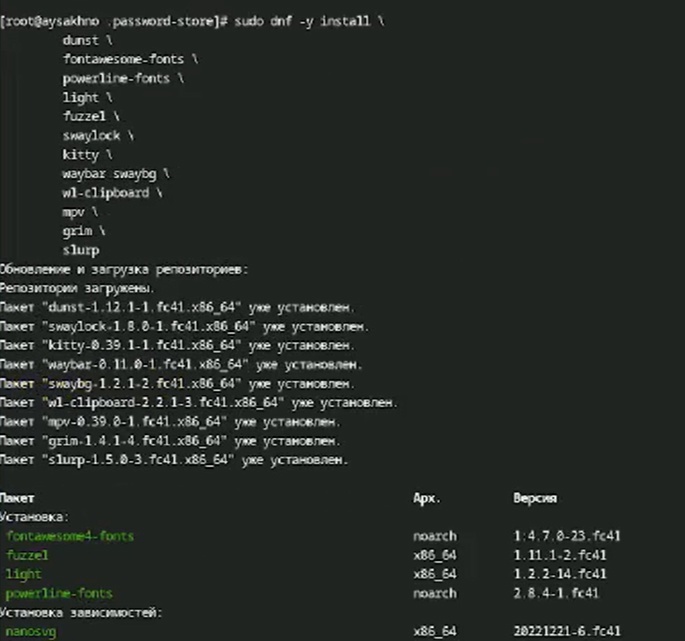


Рис. 8: Управление файлами конфигурации

Установите шрифты:

sudo dnf copr enable peterwu/iosevka  
sudo dnf search iosevka  
sudo dnf install iosevka-fonts iosevka-aile-fonts iosevka-curly-fonts iosevka-slab-fonts iosevka-etoile-fonts iosevka-term-fonts

(рис. 9).

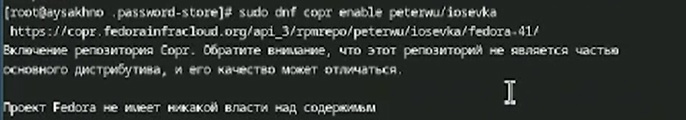


Рис. 9: Установите шифр

Установка

Установка бинарного файла. Скрипт определяет архитектуру процессора и операционную систему и скачивает необходимый файл:  
  
 с помощью wget:  
  
 sh -c "$(wget -qO- chezmoi.io/get)"

(рис. 10).

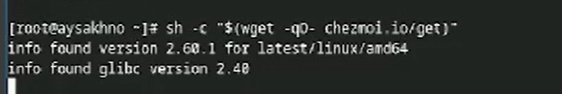


Рис. 10: Установка

Создание собственного репозитория с помощью утилит

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

(рис. 5).

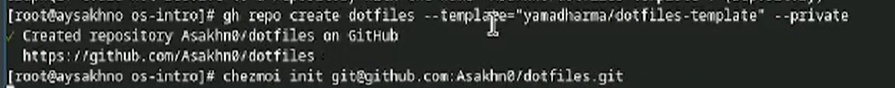


Рис. 11: Создание собственного репозитория с помощью утилит

Подключение репозитория к своей системе

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

(рис. 12).

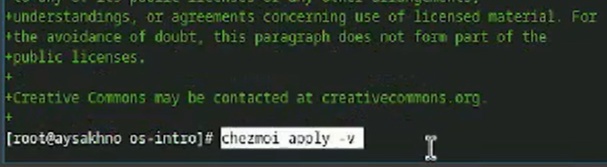


Рис. 12: Подключение репозитория к своей системе

Использование chezmoi на нескольких машинах

На второй машине инициализируйте chezmoi с вашим репозиторием dotfiles:  
  
chezmoi init https://github.com/<username>/dotfiles.git  
  
Или через ssh:  
  
chezmoi init git@github.com:<username>/dotfiles.git  
  
Проверьте, какие изменения внесёт chezmoi в домашний каталог, запустив:  
  
chezmoi diff  
  
Если вас устраивают изменения, внесённые chezmoi, запустите:  
  
chezmoi apply -v  
  
Если вас не устраивают изменения в файле, отредактируйте его с помощью:  
  
chezmoi edit file\_name  
  
Также можно вызвать инструмент слияния, чтобы объединить изменения между текущим содержимым файла, файлом в вашей рабочей копии и измененным содержимым файла:  
  
chezmoi merge file\_name  
  
При существующем каталоге chezmoi можно получить и применить последние изменения из вашего репозитория:  
  
chezmoi update -v

(рис. **¿fig:0013?**).

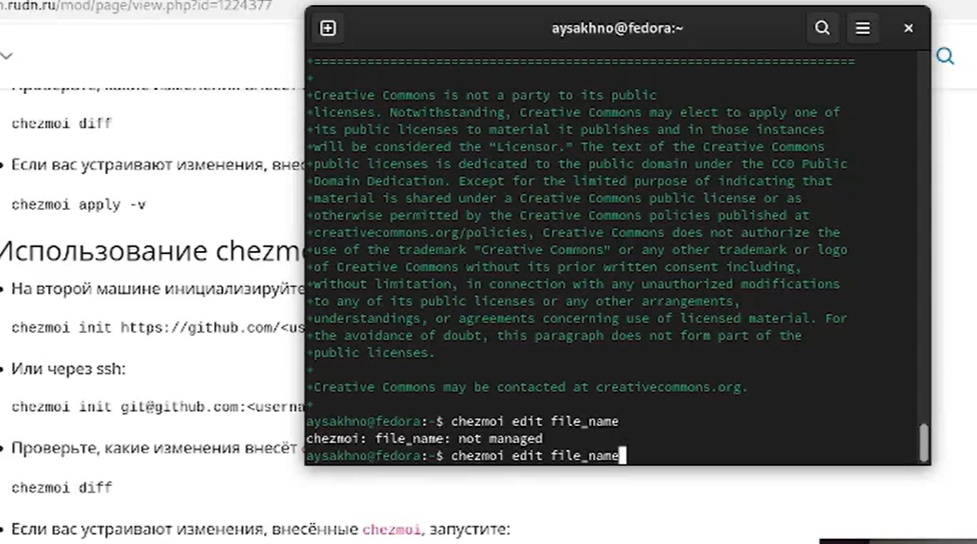


Рис. 13: На второй машине

Настройка новой машины с помощью одной команды

Можно установить свои dotfiles на новый компьютер с помощью одной команды:  
  
chezmoi init --apply https://github.com/<username>/dotfiles.git  
  
Через ssh:  
  
chezmoi init --apply git@github.com:<username>/dotfiles.git

Ежедневные операции c chezmoi

Извлеките 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.

* Извлеките последние изменения из своего репозитория и посмотрите, что изменится, фактически не применяя изменения
* Выполните:  
    
   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.

* Если вы довольны изменениями, вы можете применить их:
* chezmoi apply

(рис. 14).

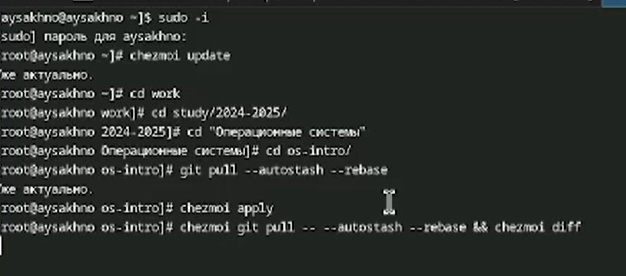


Рис. 14: Ежедневные операциир

Автоматически фиксируйте и отправляйте изменения в репозиторий 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:
* [git]  
   autoCommit = true  
   autoPush = true
* Всякий раз, когда в исходный каталог вносятся изменения, chezmoi фиксирует изменения с помощью автоматически сгенерированного сообщения фиксации и отправляет их в ваш репозиторий.
* Будьте осторожны при использовании autoPush. Если ваш репозиторий dotfiles является общедоступным, и вы случайно добавили секрет в виде обычного текста, этот секрет будет отправлен в ваш общедоступный репозиторий.

(рис. 15).



Рис. 15: Изменение в репозиторий

# 3 Выводы

Я изучила основные свойства структура базы паролей и сумела по- работать с pass.

# Список литературы

https://esystem.rudn.ru/mod/page/view.php?id=1224377