Отчёт по лабораторной работе №5

Архитектура компьютеров и операционные системы

Мутале Чали

Содержание

[1 Цель работы 1](#_Toc192548196)

[2 Задание 1](#_Toc192548197)

[3 Выполнение лабораторной работы 2](#_Toc192548198)

[3.1 Установка и Настройка Pass 2](#_Toc192548199)

[3.2 Настройка интерфейса с броузером 3](#_Toc192548200)

[3.3 Сохранение пароля 3](#_Toc192548201)

[3.4 Управление файлами конфигурации 4](#_Toc192548202)

[3.4.1 Дополнительное программное обеспечение 4](#_Toc192548203)

[3.5 Установка бинарного файла 5](#_Toc192548204)

[3.6 Создание собственного репозитория 5](#_Toc192548205)

[3.7 Подключение репозитория к своей системе 5](#_Toc192548206)

[3.8 Использование chezmoi на нескольких машинах 6](#_Toc192548207)

[3.9 Ежедневные операции c chezmoi 7](#_Toc192548208)

[4 Выводы 8](#_Toc192548209)

[Список литературы 8](#_Toc192548210)

# 1 Цель работы

Цель данной работы – настроика рабочей среды с помощью менеджера паролей pass и конфигурации chezmoi.

# 2 Задание

1. Установитв и настроить Pass
2. Настроить интерфейс с броузером
3. Управление файлами конфигурации
4. Использовать chezmoi на нескольких машинах

# 3 Выполнение лабораторной работы

## 3.1 Установка и Настройка Pass

С помощью командой dnf install через суперпользователя устанавливаю pass, pass-otp и gopass:

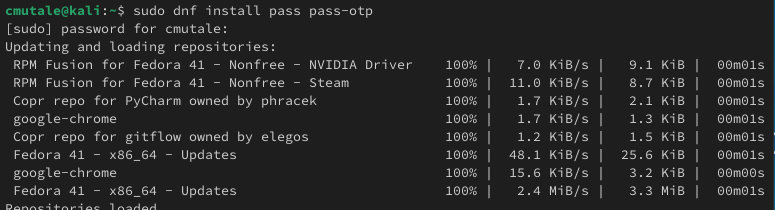


Рис. 1: Установка pass pass-otp

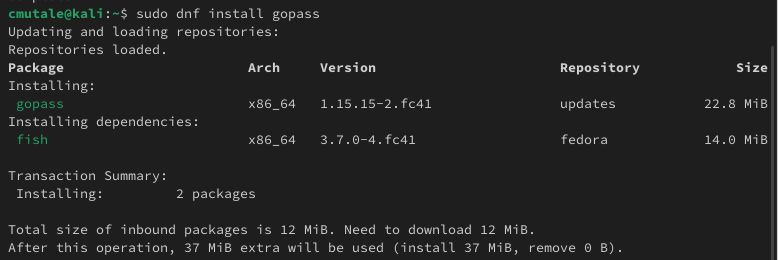


Рис. 2: Установка gopass

Создаю новый ключ GPG поскольку команда gpg –list-secret-keys ничего не выводила:



Рис. 3: Создание ключа GPG

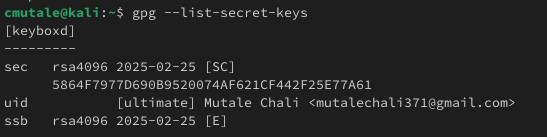


Рис. 4: Созданные ключи

С помощью pass init, инициализирую хранилище указывая свой адрес электроной почты и создаю структуру git (pass git init) для синхронирования:

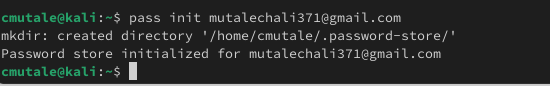


Рис. 5: Инициализирование хранилище

Перехожу в ~/.password-store для посмотра статуса синхронизации:

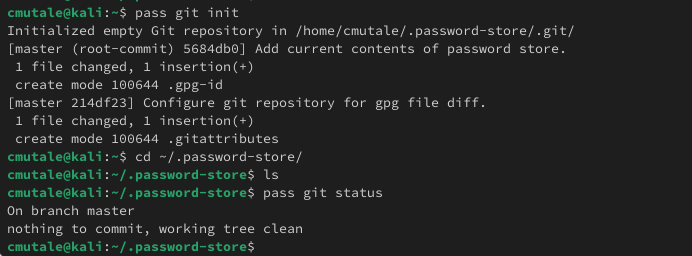


Рис. 6: посмотр статуса синхронизации

## 3.2 Настройка интерфейса с броузером

Добавляю плагин browserpass для firefox через броузера:

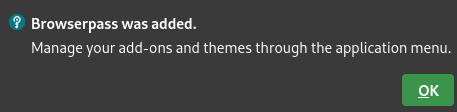


Рис. 7: Добавлен плагин

Для взаимодействия с броузером используется интерфейс native messaging. Поэтому кроме плагина к броузеру устанавливается программа, обеспечивающая интерфейс native messaging. Делаю это через терминала

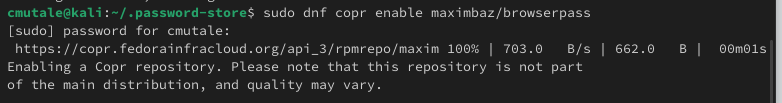


Рис. 8: Включение репозитория Copr

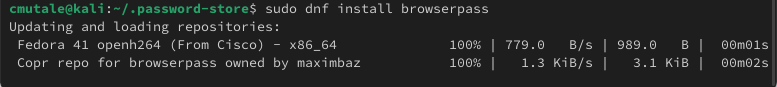


Рис. 9: Установка browserpass native

## 3.3 Сохранение пароля

Добавляю новый пароль в файле, который будет использоваться для хранения пароля. Этот файл находится в каталоге, определяющее файловую структуру для вашего хранилища паролей.

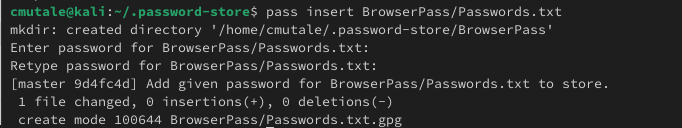


Рис. 10: Добавление пароля

При запуске pass BrowserPass/Passwords.txt, выводится запрос подтвердить созданный пароль:

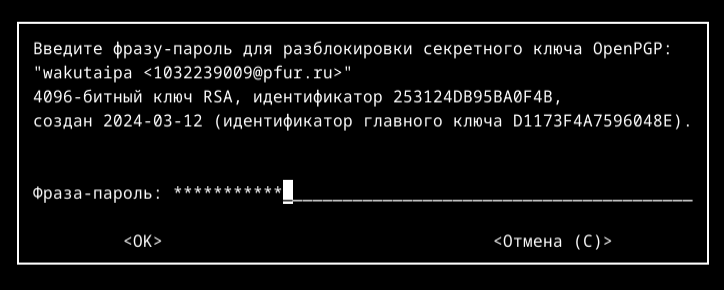


Рис. 11: Подтверждение созданного пароля

Заменяю существующий пароль с помощью pass generate –in-place Passwords.txt

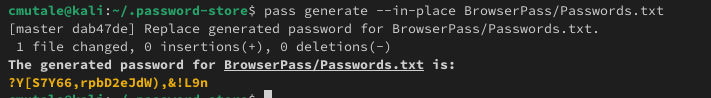


Рис. 12: Замена пароля

## 3.4 Управление файлами конфигурации

### 3.4.1 Дополнительное программное обеспечение

Устанавливаю дополнительное программное обеспечение:

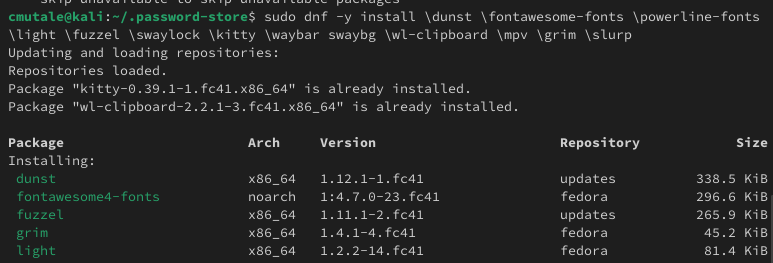


Рис. 13: Установка дополнительного программного обеспечения

Установливаю шрифты iosevka. Для этого надо включть copr peterwu/iosevka и искать iosevka:

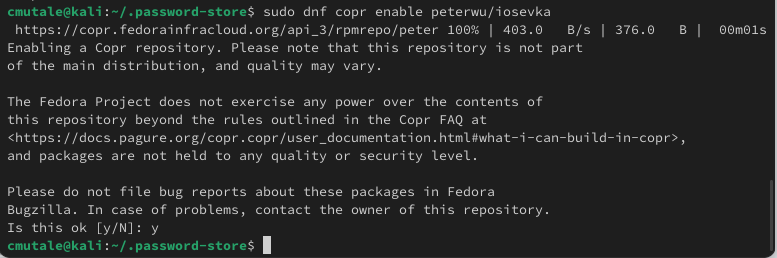


Рис. 14: Включение copr peterwu/iosevka

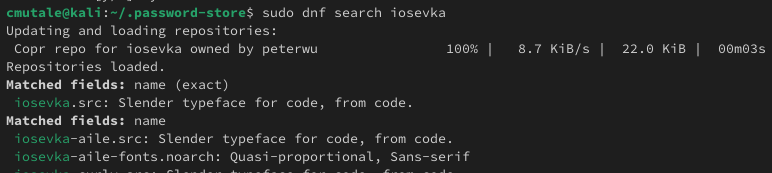


Рис. 15: Поиск iosevka



Рис. 16: Установка шрифты

## 3.5 Установка бинарного файла

Скрипт определяет архитектуру процессора и операционную систему и скачивает необходимый файл, поэтому я просто запускаю sh -c “$(wget -qO- chezmoi.io/get)”:

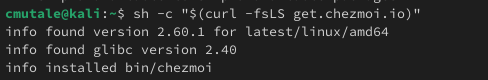


Рис. 17: Установка chezmoi

## 3.6 Создание собственного репозитория

С помощью утилита gh я создаю новый репозиторий dotfiles по шаблону yamadharma

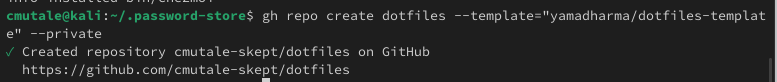


Рис. 18: Создание репозиторий

## 3.7 Подключение репозитория к своей системе

Сhezmoi является клоном репозитория dotfiles. Инициализирую его:

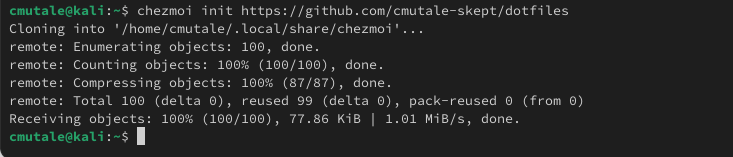


Рис. 19: Инициализирование Сhezmoi

Проверяю какие изменения внесёт chezmoi в домашний каталог, запуская chezmoi diff:

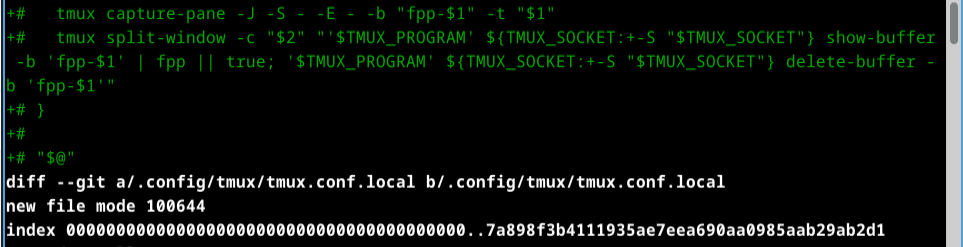


Рис. 20: Проверка изменений

Применяю изменения запускав chezmoi apply -v:



Рис. 21: Применение изменений

## 3.8 Использование chezmoi на нескольких машинах

На другой машине я устанавливаю chezmoi

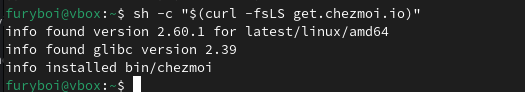


Рис. 22: Установка chezmoi на другой машине

Далее инициализирую chezmoi со своём репозиторием dotfiles:

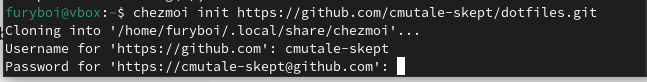


Рис. 23: Инициализирование Сhezmoi на другой машине

Проверяю какие изменения внесёт chezmoi в домашний каталог, запуская chezmoi diff:

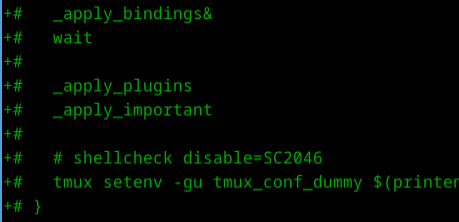


Рис. 24: Проверка изменений на второй машине

Применяю изменения запускав chezmoi apply -v:

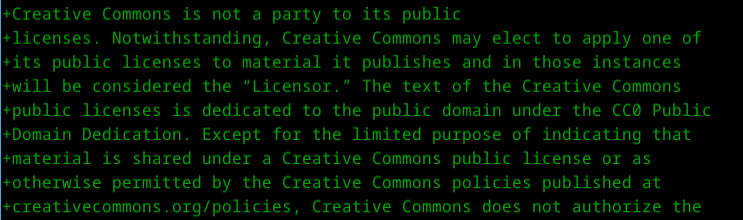


Рис. 25: Применение изменений на второй машине

Меня устраивают изменения, внесённые chezmoi, поэтомму я просто обновляю chezmoi :



Рис. 26: Обновление chezmoi

## 3.9 Ежедневные операции c chezmoi

На моей основной машине я обновляю chezmoi на всякий случай:



Рис. 27: Обновление chezmoi

Выполняю chezmoi git pull – –autostash –rebase && chezmoi diff. Это запускается git pull –autostash –rebase в исходном каталоге, а chezmoi diff затем показывает разницу между целевым состоянием, вычисленным из исходного каталога, и фактическим состоянием.

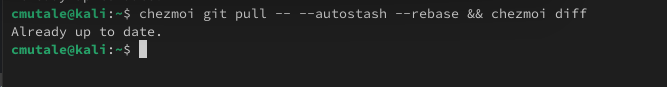


Рис. 28: Извлечение изменений

Далее применяю “изменения”:

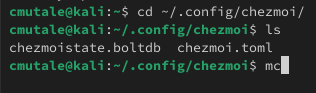


Рис. 29: Применение изменений после обновления

Когда в исходный каталог вносятся изменения, chezmoi фиксирует изменения с помощью автоматически сгенерированного сообщения фиксации и отправляет их в репозиторий. Эта функция отключена по умолчанию но у меня уже была включено. Можно это увидеть в файле конфигурации ~/.config/chezmoi/chezmoi.toml:

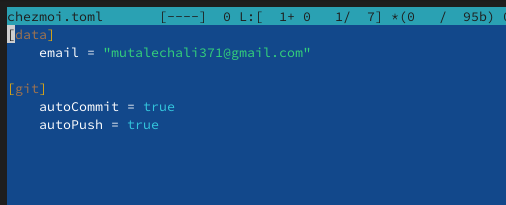


Рис. 30: Функция фиксирование изменений автоматически

# 4 Выводы

При выполнении данной работы я настроила рабочую среду с помощью менеджера паролей pass и конфигурации chezmoi.

# Список литературы

[Архитектура ЭВМ](https://esystem.rudn.ru/mod/page/view.php?id=1098796)