Отчет по лабораторной работе №5

Шубина София Антоновна

Содержание

# 1 Цель работы

Научиться работать с менеджером паролей.

# 2 Выполнение лабораторной работы

[1] Менеджер паролей pass Установка Fedora pass dnf install pass pass-otp gopass dnf install gopass

(рис. **¿fig:001?**,**¿fig:002?**).

Установка Установка

Настройка Ключи GPG Просмотр списка ключей: gpg –list-secret-keys (рис. 1).

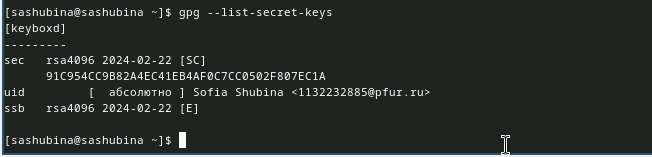


Рис. 1: Просмотр списка ключей

Если ключа нет, нужно создать новый: gpg –full-generate-key Инициализация хранилища Инициализируем хранилище: pass init (рис. 2).

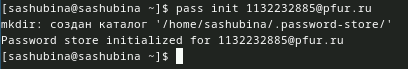


Рис. 2: Инициализация хранилища

Синхронизация с git Создадим структуру git: pass git init (рис. 3).

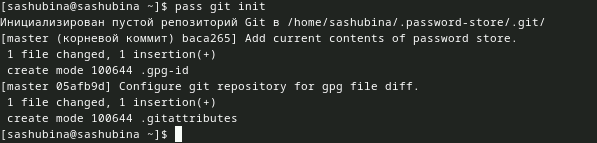


Рис. 3: Создем структуру git

Также можно задать адрес репозитория на хостинге (репозиторий необходимо предварительно создать): pass git remote add origin git@github.com:/.git (рис. 4, 5).

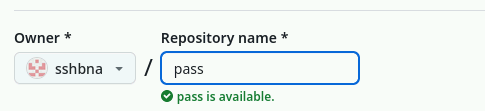


Рис. 4: Создаем репозиторий

Задаем адрес репозитория на хостинге

Рис. 5: Задаем адрес репозитория на хостинге

Для синхронизации выполняется следующая команда: pass git pull pass git push (рис. 6).

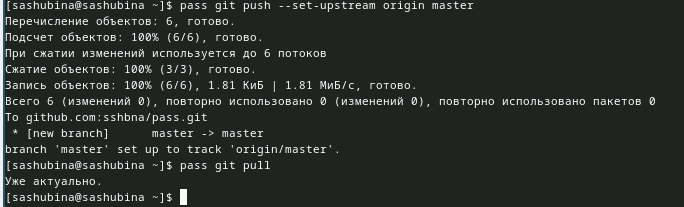


Рис. 6: Синхронизация

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. 7).

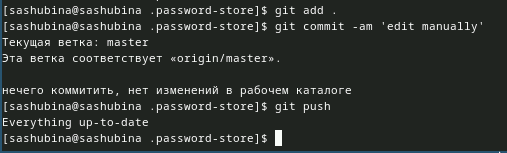


Рис. 7: Выкладываем изменения

Проверить статус синхронизации модно командой pass git status (рис. 8).

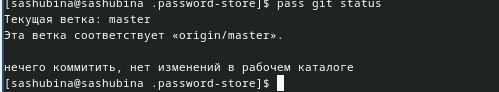
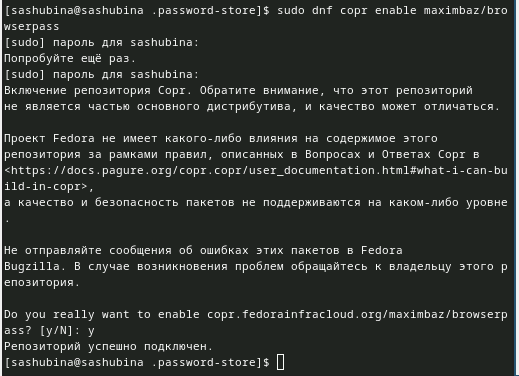
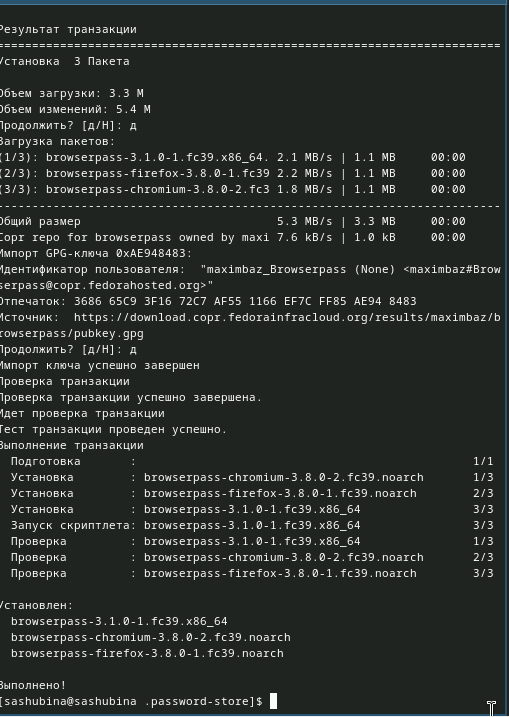


Рис. 8: Проверка статуса синхронизации

Настройка интерфейса с броузером Для взаимодействия с броузером используется интерфейс native messaging. Поэтому кроме плагина к броузеру устанавливается программа, обеспечивающая интерфейс native messaging. Fedora dnf copr enable maximbaz/browserpass dnf install browserpass (рис. **¿fig:011?**, **¿fig:012?**).

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. 9).

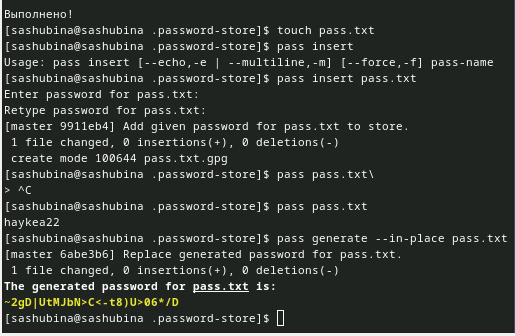


Рис. 9: Манипуляции с паролем

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. 10).

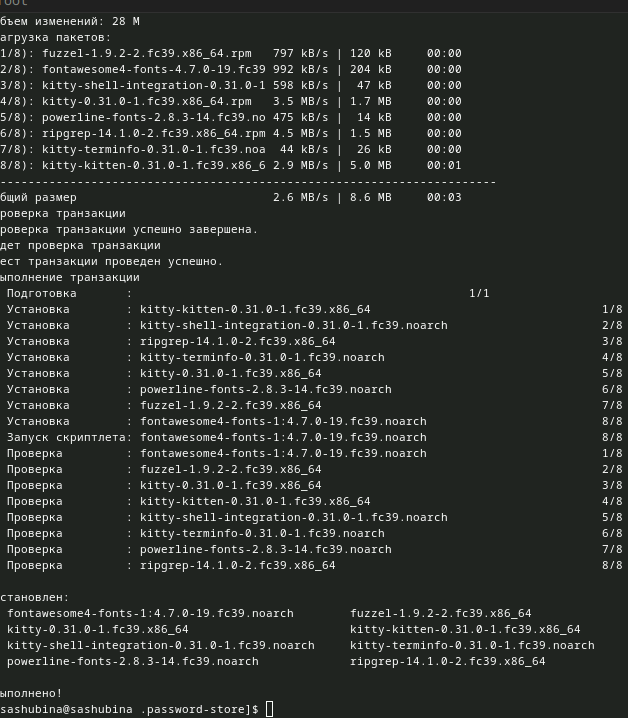


Рис. 10: Установка доп. программного обеспечения

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. 11).

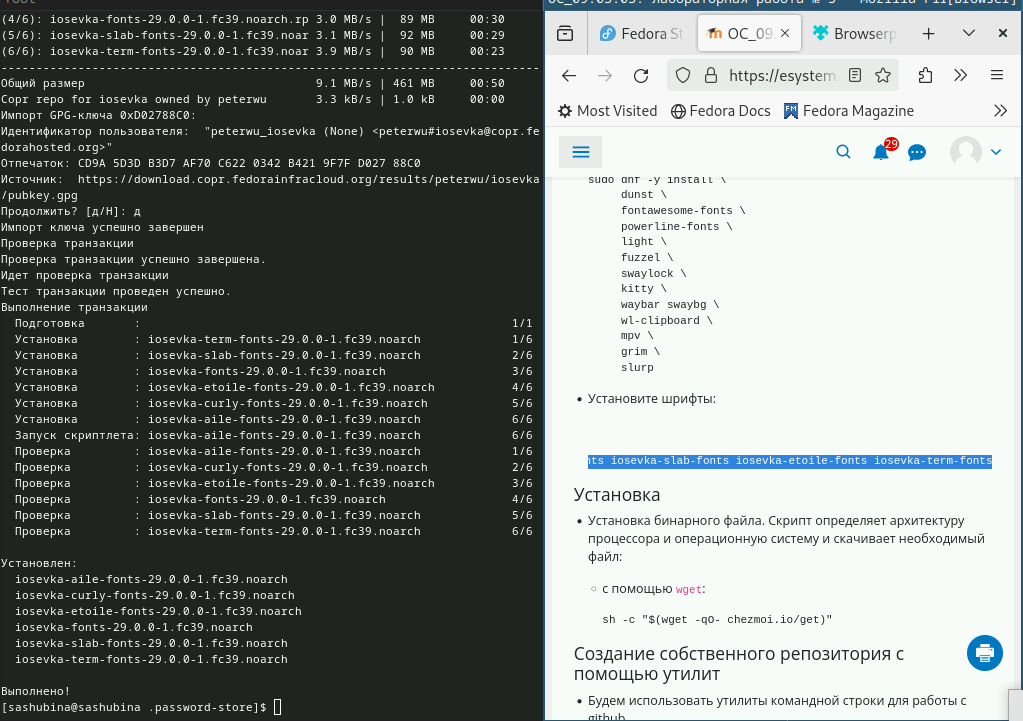


Рис. 11: Установка шрифтов

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. 12).

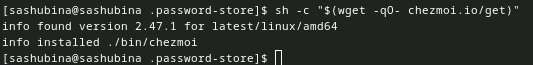


Рис. 12: Установка бинарного файла

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. 13).

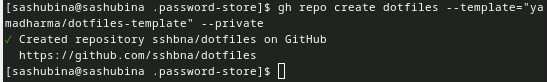


Рис. 13: Создание репозитория с помощью утилит

Подключение репозитория к своей системе Инициализируйте chezmoi с вашим репозиторием dotfiles: chezmoi init git@github.com:/dotfiles.git (рис. 14).

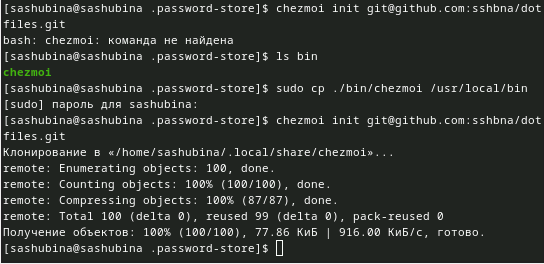


Рис. 14: Инициализация

Проверьте, какие изменения внесёт chezmoi в домашний каталог, запустив: chezmoi diff Если вас устраивают изменения, внесённые chezmoi, запустите: chezmoi apply -v (рис. 15).

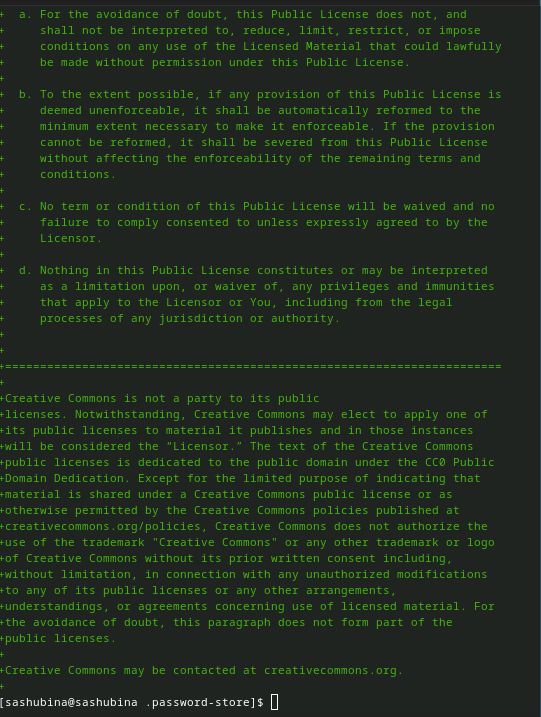


Рис. 15: Проверка и одобрение изменений

Использование chezmoi на нескольких машинах На второй машине инициализируйте chezmoi с вашим репозиторием dotfiles: chezmoi init https://github.com//dotfiles.git Или через ssh: chezmoi init git@github.com:/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 Настройка новой машины с помощью одной команды Можно установить свои dotfiles на новый компьютер с помощью одной команды: сhezmoi init –apply https://github.com//dotfiles.git (рис. 16).

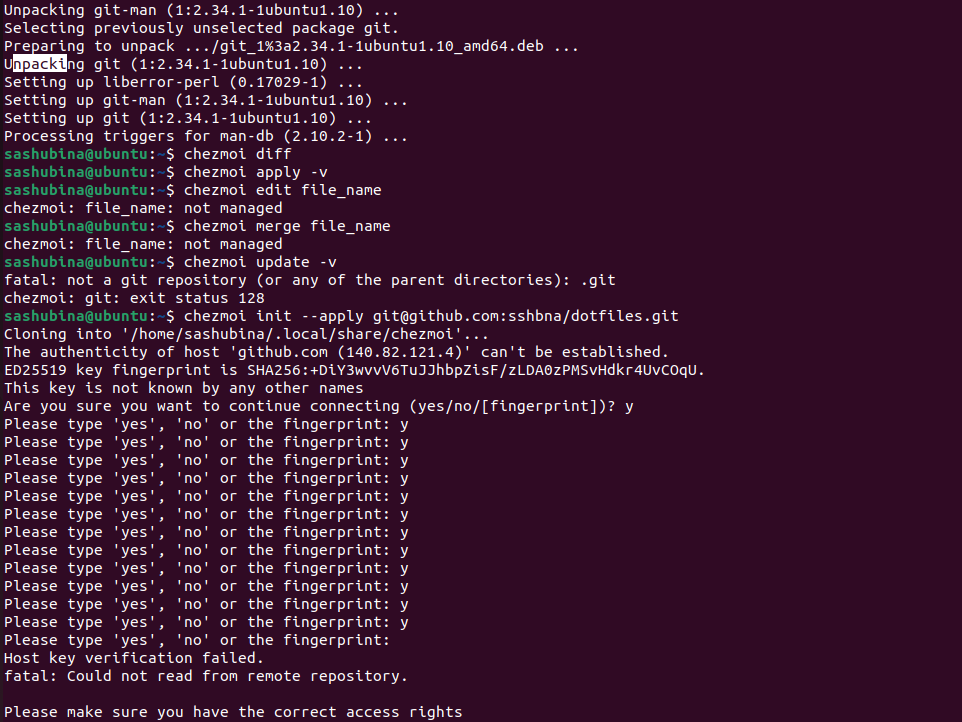


Рис. 16: Использованиие chemoi на второй виртуальной машине

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. (рис. 17).

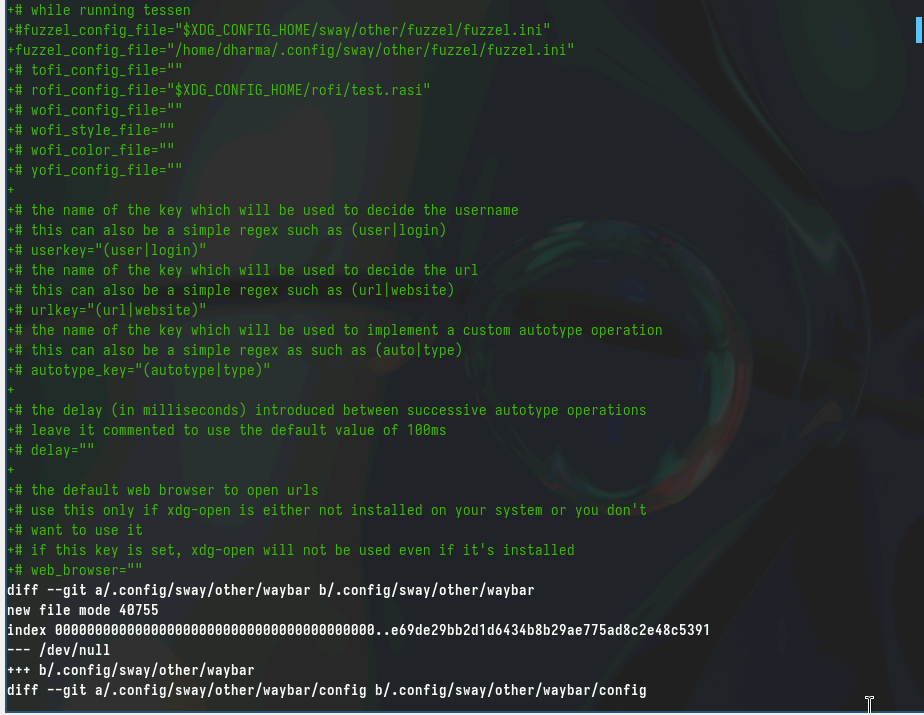


Рис. 17: Извлечение последних изменений

Если вы довольны изменениями, вы можете применить их: chezmoi 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: [git] autoCommit = true autoPush = true (рис. 18).

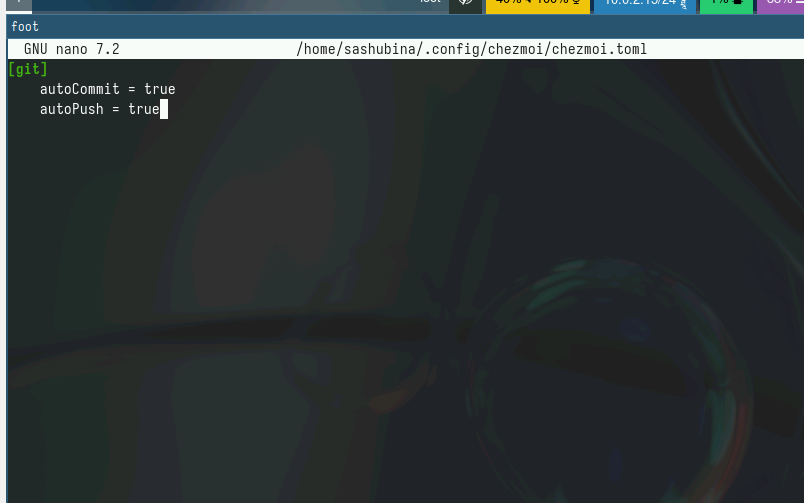


Рис. 18: Листинг

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.

# 3 Выводы

Я научилась работать с менеджером паролей

# Список литературы

1. [Learning the bash Shell: Unix Shell Programming](https://esystem.rudn.ru/user/policy.php).