Отчёта по лабораторной работе

Лабораторная работа №5

Диана Садова Алексеевна

Содержание

# 1 Цель работы

Освоить работу с файловой системой и системой поролей

# 2 Выполнение лабораторной работы

## 2.1 Менеджер паролей pass

### 2.1.1 Установка

Fedora  
  
 pass(рис. [-@fig:001]).

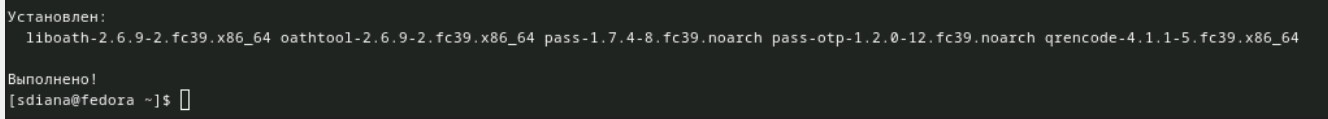


Рис. 1: Установка нового пакета

gopass(рис. [-@fig:002]).

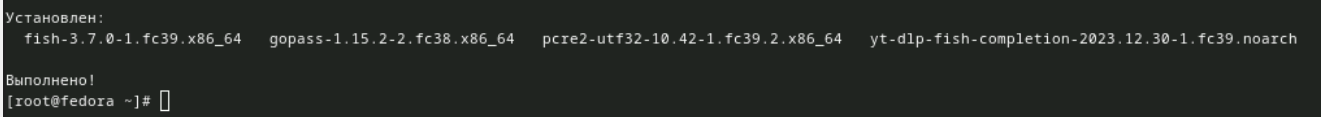


Рис. 2: Установка нового пакета

### 2.1.2 Настройка

1)Ключи GPG  
  
 Просмотр списка ключей:(рис. [-@fig:003]).

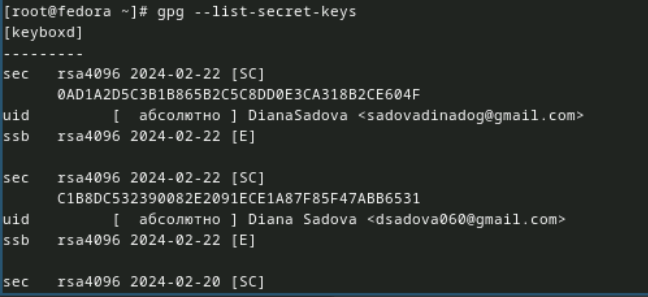


Рис. 3: Просматриваем список ключей

Если ключа нет, нужно создать новый:(рис. [-@fig:004]).

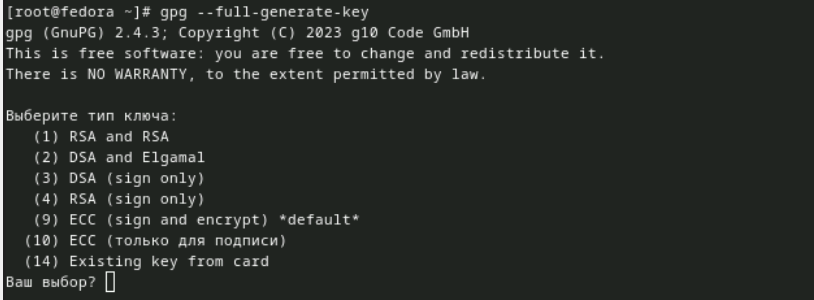


Рис. 4: Создаем новый ключ

Это был не обязательный шаг, но мы его сделали

2)Инициализация хранилища  
  
 Инициализируем хранилище:(рис. [-@fig:005]).

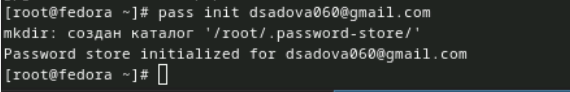


Рис. 5: Инициализация хранилища

3)Синхронизация с git  
  
 Создадим структуру git:(рис. [-@fig:006]).

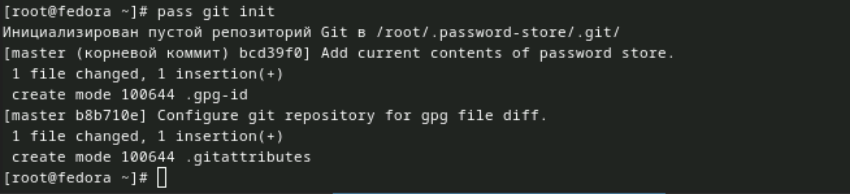


Рис. 6: Создали структуру в git

### 2.1.3 Настройка интерфейса с броузером

Для взаимодействия с броузером используется интерфейс native messaging.  
Поэтому кроме плагина к броузеру устанавливается программа, обеспечивающая интерфейс native messaging.  
  
1)Плагин browserpass  
 Репозиторий: https://github.com/browserpass/browserpass-extension  
 Плагин для брoузера  
 Плагин для Firefox: https://addons.mozilla.org/en-US/firefox/addon/browserpass-ce/.  
 Плагин для Chrome/Chromium: https://chrome.google.com/webstore/detail/browserpass-ce/naepdomgkenhinolocfifgehidddafch.  
  
 Интерфейс для взаимодействия с броузером (native messaging)  
 Репозиторий: https://github.com/browserpass/browserpass-native  
  
 Gentoo:  
  
 emerge www-plugins/browserpass  
  
 Fedora (рис. [-@fig:007]),(рис. [-@fig:008]).

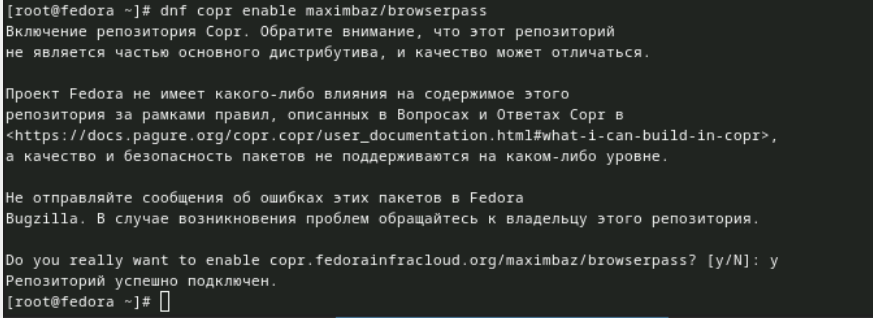


Рис. 7: Добавление плагинов

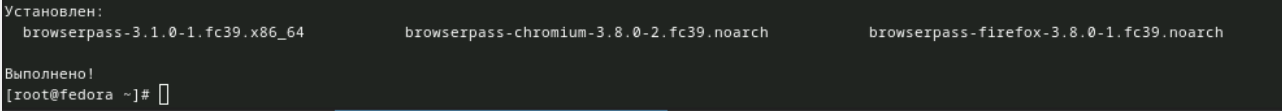


Рис. 8: Добавление плагинов

### 2.1.4 Сохранение пароля

1)Добавить новый пароль  
  
 Выполните:(рис. [-@fig:009]).

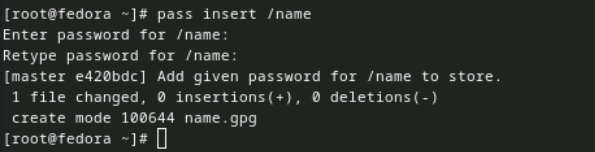


Рис. 9: Добавили новый пароль

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:(рис. [-@fig:010]).

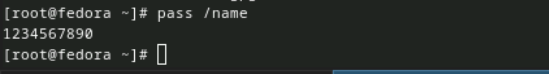


Рис. 10: Наш пароль на данное имя

Замените существующий пароль:(рис. [-@fig:011]).

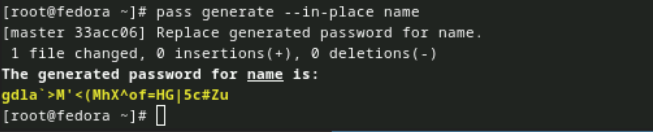


Рис. 11: Создание нового пароля

## 2.2 Управление файлами конфигурации

## 2.3 Дополнительное программное обеспечение

Установите дополнительное программное обеспечение:(рис. [-@fig:012]),(рис. [-@fig:013]),(рис. [-@fig:014]),(рис. [-@fig:015]),(рис. [-@fig:016]),(рис. [-@fig:017]),(рис. [-@fig:018]),(рис. [-@fig:019]),(рис. [-@fig:020]),(рис. [-@fig:021]),(рис. [-@fig:022]),(рис. [-@fig:023]).

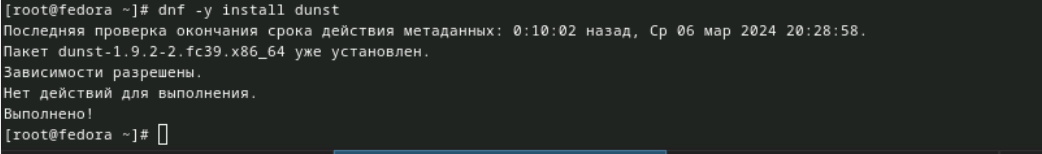


Рис. 12: Установили дополнительное ПО

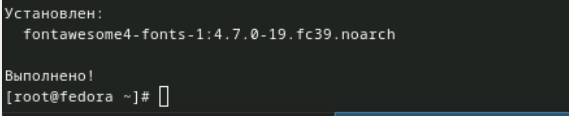


Рис. 13: Установили дополнительное ПО

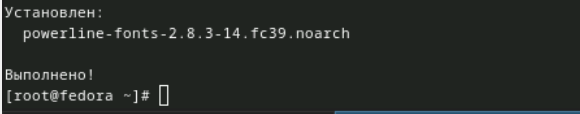


Рис. 14: Установили дополнительное ПО

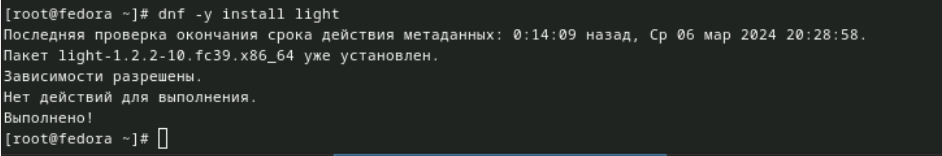


Рис. 15: Установили дополнительное ПО

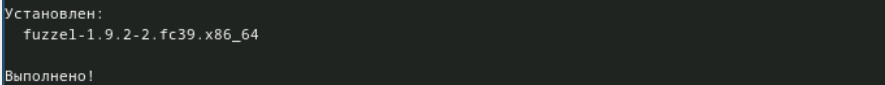


Рис. 16: Установили дополнительное ПО

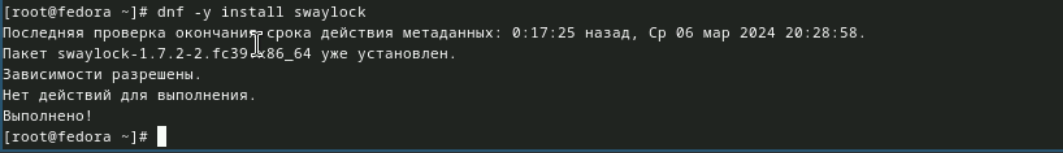


Рис. 17: Установили дополнительное ПО

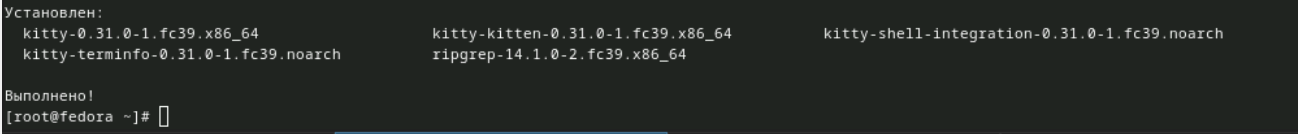


Рис. 18: Установили дополнительное ПО

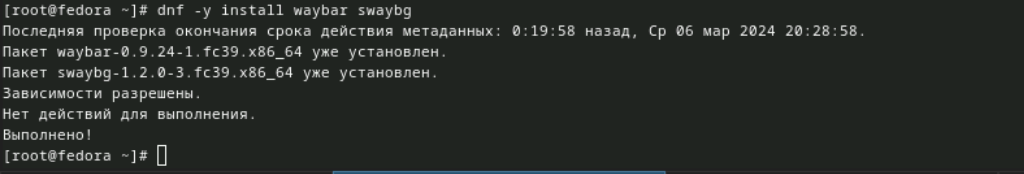


Рис. 19: Установили дополнительное ПО

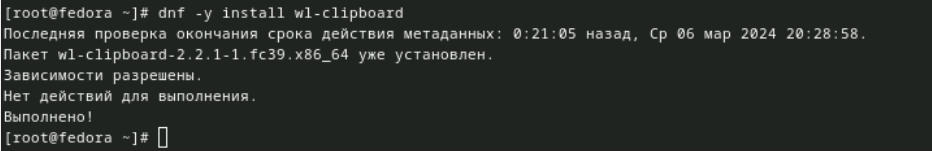


Рис. 20: Установили дополнительное ПО

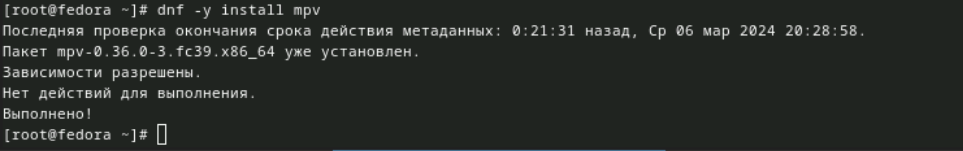


Рис. 21: Установили дополнительное ПО

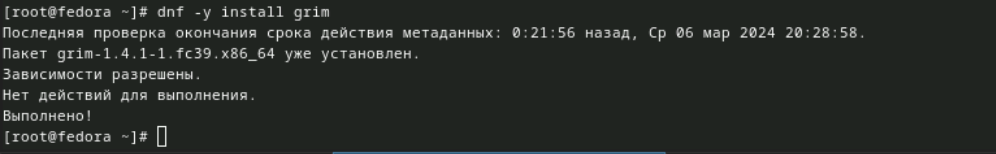


Рис. 22: Установили дополнительное ПО

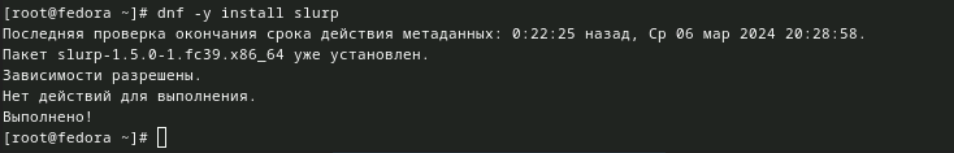


Рис. 23: Установили дополнительное ПО

Установите шрифты:(рис. [-@fig:024]),(рис. [-@fig:025]),(рис. [-@fig:026]).

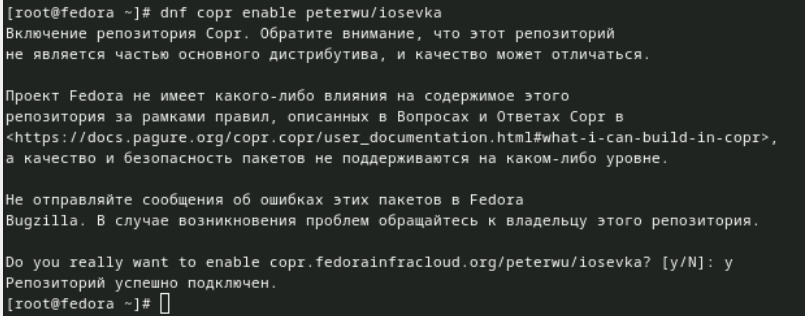


Рис. 24: Установка шривта



Рис. 25: Установка шривта

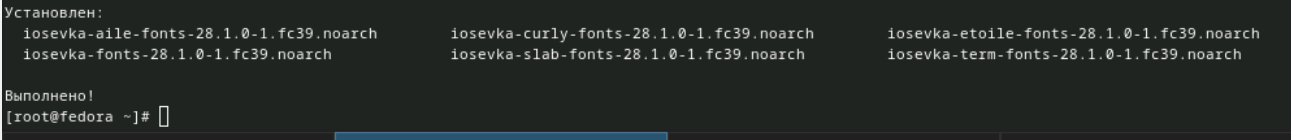


Рис. 26: Установка шривта

### 2.3.1 Установка

Установка бинарного файла. Скрипт определяет архитектуру процессора и операционную систему и скачивает необходимый файл:  
  
 с помощью wget:(рис. [-@fig:027]).

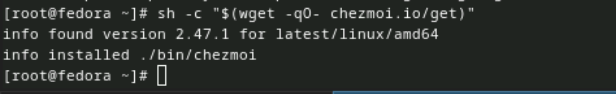


Рис. 27: Установим chezmoi

### 2.3.2 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. [-@fig:028]).

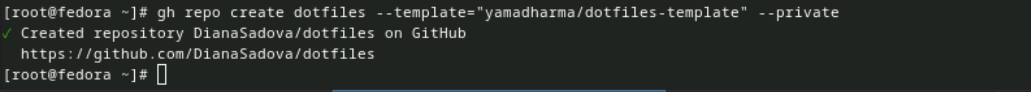


Рис. 28: Создали новый репридиторий

### 2.3.3 Подключение репозитория к своей системе

Инициализируйте chezmoi с вашим репозиторием dotfiles:(рис. [-@fig:029]),(рис. [-@fig:030]).

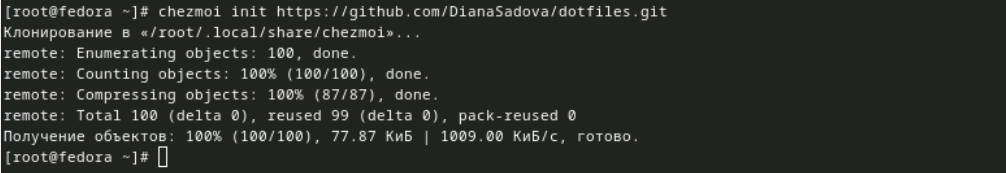


Рис. 29: Инициализируем chezmoi к репридиторию

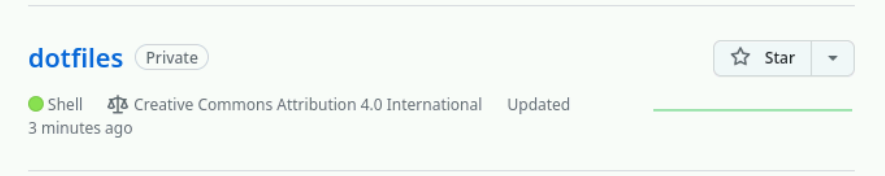


Рис. 30: Новый репридиторий

Проверьте, какие изменения внесёт chezmoi в домашний каталог, запустив:(рис. [-@fig:031]).

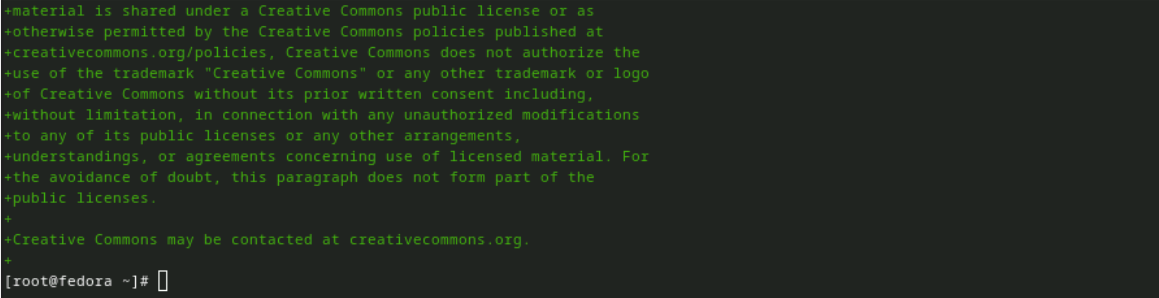


Рис. 31: Проверяем измениения в каталог

Если вас устраивают изменения, внесённые chezmoi, запустите:(рис. [-@fig:032]).



Рис. 32: Соглашаемся с изменениями

### 2.3.4 Использование chezmoi на нескольких машинах

На второй машине инициализируйте chezmoi с вашим репозиторием dotfiles:(рис. [-@fig:033]).

Инициализируем chezmoi на второй машине

Рис. 33: Инициализируем chezmoi на второй машине

Проверьте, какие изменения внесёт chezmoi в домашний каталог, запустив:(рис. [-@fig:034]).

Проверяем измениения в каталог

Рис. 34: Проверяем измениения в каталог

Если вас устраивают изменения, внесённые chezmoi, запустите:(рис. [-@fig:035]).

Соглашаемся с изменениями

Рис. 35: Соглашаемся с изменениями

При существующем каталоге chezmoi можно получить и применить последние изменения из вашего репозитория:(рис. [-@fig:036]).

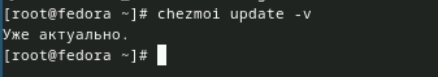


Рис. 36: Обновляем репридиторий

### 2.3.5 Настройка новой машины с помощью одной команды

Можно установить свои dotfiles на новый компьютер с помощью одной команды:(рис. [-@fig:037]).

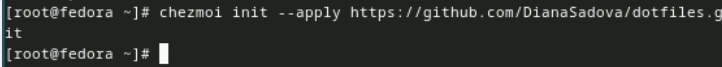


Рис. 37: Устанавливаем dotfiles одной командой

### 2.3.6 Ежедневные операции c chezmoi

Извлеките последние изменения из репозитория и примените их  
  
 Можно извлечь изменения из репозитория и применить их одной командой:(рис. [-@fig:038]).

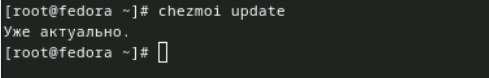


Рис. 38: Извликаем изменения и применяеи их

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:(рис. [-@fig:039]).

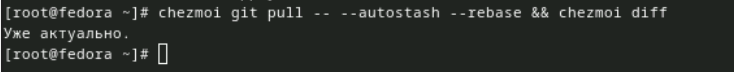


Рис. 39: Извликаем изменения, но не применяеи их

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:(рис. [-@fig:040]).

Приминяем изминения

Рис. 40: Приминяем изминения

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:(рис. [-@fig:041]).

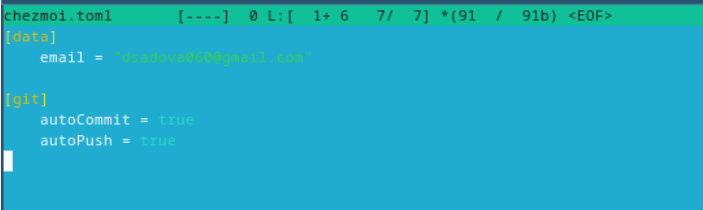


Рис. 41: Меняем код для автоматической фиксации изменений

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

Получили новый опыт работы с репридиториями github, файловой системой и системой поролей

# Список литературы