

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5 _____

дисциплина: *Операционные системы*

Студент: Ниemek Яи Жак

Группа: НММБд-04-24

МОСКВА

2025__ г.

Цель работы

Выполнение лабораторной работы

Менеджер паролей pass

Установка

- Fedora
 - `pass`
 - `dnf install pass pass-otp`
 - `gopass`
 - `dnf install gopass`

Настройка

1. Ключи GPG
 - Просмотр списка ключей:
 - `gpg --list-secret-keys`
 - Если ключа нет, нужно создать новый:
 - `gpg --full-generate-key`
2. Инициализация хранилища
 - Инициализируем хранилище:
 - `pass init <gpg-id or email>`
3. 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`
 - Для синхронизации выполняется следующая команда:
 - `pass git pull`
 - `pass git push`
8. 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`
 - Проверить статус синхронизации можно командой
 - `pass git status`

Настройка интерфейса с браузером

- Для взаимодействия с браузером используется интерфейс *native messaging*.
- Поэтому кроме плагина к браузеру устанавливается программа, обеспечивающая интерфейс *native messaging*.

1. Плагин [browserpass](#)

- Репозиторий: <https://github.com/browserpass/browserpass-extension>
- Плагин для броузера
 - 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/naepdomgkenhinolocfifgehiddafch>.
- Интерфейс для взаимодействия с браузером (native messaging)
 - Репозиторий: <https://github.com/browserpass/browserpass-native>
 - Gentoo:
 - `emerge www-plugins/browserpass`
 - Fedora
 - `dnf copr enable maximbaz/browserpass`
 - `dnf install browserpass`

Сохранение пароля

1. 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`

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`
- 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`

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)"`

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`

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`

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 изменённым 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:
- `chezmoi init --apply https://github.com/<username>/dotfiles.git`
- Через ssh:
- `chezmoi init --apply git@github.com:<username>/dotfiles.git`

Ежедневные операции с `chezmoi`

1. 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`.
2. 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`
3. 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.

```

jacques@vbox:~$ sudo dnf install pass pass-otp gopass -y
[sudo] Mot de passe de jacques :
Dernière vérification de l'expiration des métadonnées effectuée il y a 0:40:08 le jeu. 28 août 2025 05:56:18.
Dépendances résolues.
=====
Paquet                Architecture  Version           Dépôt             Taille
=====
Installation:
gopass                x86_64        1.15.2-2.fc38     fedora             7.1 M
pass                  noarch        1.7.4-8.fc39      fedora             59 k
pass-otp              noarch        1.2.0-12.fc39     fedora             28 k
Installation des dépendances:
fish                  x86_64        3.7.0-1.fc39      updates            3.4 M
liboath               x86_64        2.6.12-1.fc39     updates            49 k
oathtool              x86_64        2.6.12-1.fc39     updates            45 k
qrencode              x86_64        4.1.1-5.fc39      fedora             25 k
Installation des dépendances faibles:
wl-clipboard          x86_64        2.2.1-1.fc39      updates            58 k

Résumé de la transaction
=====
Installer 8 Paquets

```

```
jacques@vbox:~127$ sh -c "$(curl -fsLS get.chezmoi.io)"
info found chezmoi version 2.65.0 for latest/linux/amd64
info found glibc version 2.38
info installed bin/chezmoi
jacques@vbox:~$ chezmoi --version
chezmoi version v2.65.0, commit 39e5ad1b8bb7c35a8a23aa1bdda5343224e77329, built
at 2025-08-26T13:34:51Z, built by goreleaser
jacques@vbox:~$
```

```
jacques@vbox:~$ gpg --list-secret-keys
[Keyboxd]
-----
sec   rsa4096 2025-08-27 [SC]
      69467C81DE786D0E7CE4449AED92BC4B4568035B
uid   [  ultime ] jacques (gpg key) <nyemeckyaijacques@gmail.com>
ssb   rsa4096 2025-08-27 [E]

jacques@vbox:~$ gpg --list-secret-keys --keyid-format=long
[keyboxd]
-----
sec   rsa4096/ED92BC4B4568035B 2025-08-27 [SC]
      69467C81DE786D0E7CE4449AED92BC4B4568035B
uid   [  ultime ] jacques (gpg key) <nyemeckyaijacques@gmail.com>
ssb   rsa4096/6F1786E2C88CD5FD 2025-08-27 [E]
```

```
jacques@vbox:~1$ chezmoi --version
bash: chezmoi: commande inconnue...
jacques@vbox:~127$ sh -c "$(curl -fsLS get.chezmoi.io)"
info found chezmoi version 2.65.0 for latest/linux/amd64
info found glibc version 2.38
info installed bin/chezmoi
jacques@vbox:~$ chezmoi --version
chezmoi version v2.65.0, commit 39e5ad1b8bb7c35a8a23aa1bdda5343224e77329, built at 2025-08-2
6T13:34:51Z, built by goreleaser
jacques@vbox:~$ chezmoi init
jacques@vbox:~$ chezmoi add ~/.bashrc
jacques@vbox:~$ chezmoi apply
jacques@vbox:~$ chezmoi init --apply https://github.com/Jacques-003/dotfiles.git
jacques@vbox:~$ chezmoi update
Pas d'information de suivi distant pour la branche actuelle.
Veuillez spécifier sur quelle branche vous souhaitez rebaser.
Référez-vous à git-pull(1) pour de plus amples détails.

git pull <distant> <branche>

Si vous souhaitez indiquer l'information de suivi distant pour cette branche, vous pouvez le
faire avec :

git branch --set-upstream-to=origin/<branche> master

chezmoi: git: exit status 1
jacques@vbox:~1$ chezmoi diff
jacques@vbox:~$
```

```

jacques@vbox:~$ pass init "nyemeckyaijacques@gmail.com"
mkdir: création du répertoire '/home/jacques/.password-store/'
Password store initialized for nyemeckyaijacques@gmail.com
jacques@vbox:~$ pass git init
Dépôt Git vide initialisé dans /home/jacques/.password-store/.git/
erreur : gpg n'a pas pu signer les données :
gpg: « Jacques <nyemeckyaijacques@gmail.com> » a été ignorée : Pas de clef secrète
[GNUPG:] INV_SGNR 9 Jacques <nyemeckyaijacques@gmail.com>
[GNUPG:] FAILURE sign 17
gpg: signing failed: Pas de clef secrète

fatal : échec de l'écriture de l'objet commit
erreur : gpg n'a pas pu signer les données :
gpg: « Jacques <nyemeckyaijacques@gmail.com> » a été ignorée : Pas de clef secrète
[GNUPG:] INV_SGNR 9 Jacques <nyemeckyaijacques@gmail.com>
[GNUPG:] FAILURE sign 17
gpg: signing failed: Pas de clef secrète

fatal : échec de l'écriture de l'objet commit
jacques@vbox:~$

```

```

jacques@vbox:~$ pass git remote add origin git@github.com:Jacques-003/repo.git
jacques@vbox:~$ pass git pull
ssh: connect to host github.com port 22: Connection refused
fatal : Impossible de lire le dépôt distant.

Veuillez vérifier que vous avez les droits d'accès
et que le dépôt existe.
jacques@vbox:~$ pass git push
fatal : La branche courante master n'a pas de branche amont.
Pour pousser la branche courante et définir la distante comme amont, utilisez

    git push --set-upstream origin master

Pour que cela soit fait automatiquement pour les branches sans
suivi distant, voir "push.autoSetupRemote" dans 'git help config'.

jacques@vbox:~$ pass github.com

```

```

jacques@vbox:~$ pass github.com
Error: github.com is not in the password store.
jacques@vbox:~$ gh repo create dotfiles --template="yamadharma/dotfiles-template" --private
✓ Created repository Jacques-003/dotfiles on GitHub
https://github.com/Jacques-003/dotfiles
jacques@vbox:~$ chezmoi init git@github.com:Jacques-003/dotfiles.git
jacques@vbox:~$ chezmoi diff
jacques@vbox:~$ chezmoi apply -v
jacques@vbox:~$ chezmoi update

```