

Отчёт по лабораторной работе №5

Операционные системы

Терещенкова М.В.

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	8
4.1	Менеджер паролей pass	8
4.1.1	Установка	8
4.1.2	Настройка	9
4.1.3	Настройка формата коммитов	9
4.2	Конфигурация git-flown	10
4.3	Создание релиза	11
4.4	Работа с репозиторием	12
4.4.1	Разработка новой функциональности	12
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	Установка Pass	8
4.2	Установка gopass	8
4.3	Просмотр списка ключей	9
4.4	Инициализация хранилища	9
4.5	Инициализация package.json	9
4.6	Добавление конфигурации	10
4.7	Добавление файлов и выполнение коммита	10
4.8	Добавление файлов и выполнение коммита	11
4.9	Создание релиза 1.0.0	11
4.10	Создание релиза 1.0.0	12
4.11	Создание релиза на GitHub	12
4.12	Работа с репозиторием	13

1 Цель работы

Познакомиться с pass, gopass, native messaging, chezmoi. Научиться пользоваться этими утилитами, синхронизировать их с git.

2 Задание

- Установить дополнительное ПО - Установить и настроить pass - Настроить интерфейс с браузером - Сохранить пароль - Установить и настроить chezmoi - Настроить chezmoi на новой машине - Выполнить ежедневные операции с chezmoi

3 Теоретическое введение

Менеджер паролей pass Менеджер паролей pass — программа, сделанная в рамках идеологии Unix. Также носит название стандартного менеджера паролей для Unix (The standard Unix password manager).

Основные свойства

Данные хранятся в файловой системе в виде каталогов и файлов. Файлы шифруются с помощью GPG-ключа.

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.

4 Выполнение лабораторной работы

4.1 Менеджер паролей pass

4.1.1 Установка

Устанавливаю pass

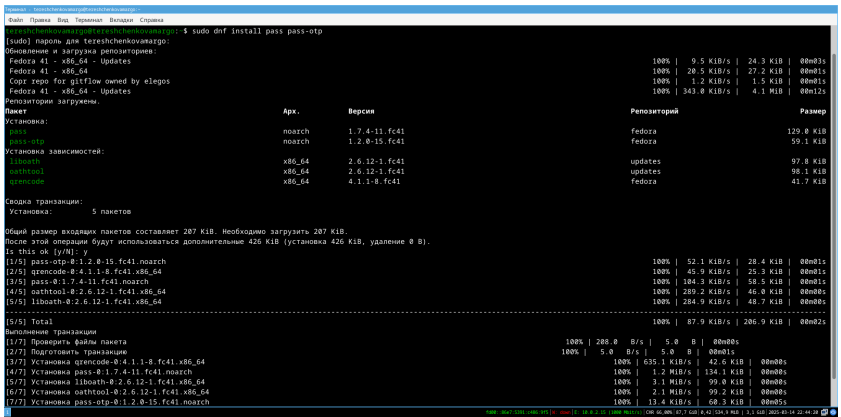


Рис. 4.1: Установка Pass

Устанавливаю gorpass

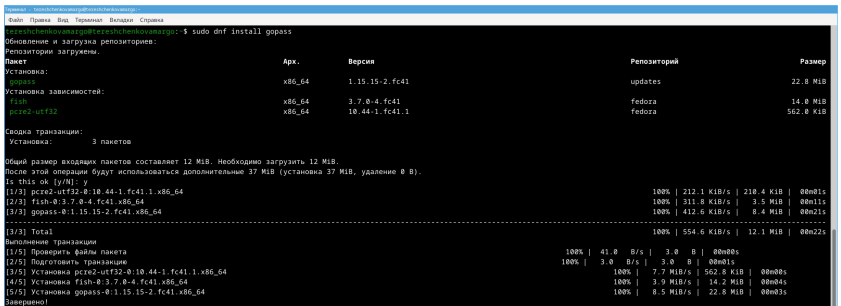


Рис. 4.2: Установка gorpass

4.1.2 Настройка

Просматриваю списки ключей gpg.

```
tereshchenkovamargo@tereshchenkovamargo:~$ gpg --list-secret-keys
[keyboxd]
-----
sec   rsa4096 2025-03-06 [SC]
      B28CAA2839BECBF804B7F389561F9B0F818847B
uid    [ абсолютно ] RitaTereshchenkova <1132246723@pfur.ru>
ssb    rsa4096 2025-03-06 [E]
```

Рис. 4.3: Просмотр списка ключей

Вижу, что ключ есть, поэтому инициализирую хранилище.

```
tereshchenkovamargo@tereshchenkovamargo:~$ pass init 1132246723@pfur.ru
mkdir: создан каталог '/home/tereshchenkovamargo/.password-store/'
Password store initialized for 1132246723@pfur.ru
tereshchenkovamargo@tereshchenkovamargo:~$ pass git init
Инициализирован пустой репозиторий Git в /home/tereshchenkovamargo/.password-store/.git/
[master (корневой коммит) ceb4db2] Add current contents of password store.
1 file changed, 1 insertion(+)
create mode 100644 .gpg-id
[master b6b2b83] Configure git repository for gpg file diff.
1 file changed, 1 insertion(+)
create mode 100644 .gitattributes
```

Рис. 4.4: Инициализация хранилища

4.1.3 Настройка формата коммитов

Был инициализирован package.json с настройками:

```
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ pnpm init
Wrote to /home/tereshchenkovamargo/git-extended/package.json

{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ nano package.json
```

Рис. 4.5: Инициализация package.json

Добавлена конфигурация для commitizen:

```
Терминал - tereshchenkovamargo@tereshchenkovamargo:~/git-extended
Файл  Правка  Вид  Терминал  Вкладки  Справка
GNU nano 8.1
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
"config": {
  "commitizen": {
    "path": "cz-conventional-changelog"
  }
}
}
```

Рис. 4.6: Добавление конфигурации

После этого были добавлены файлы и выполнен коммит через git cz:

```
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ nano package.json
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git add .
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git cz
cz-cli@4.3.1, cz-conventional-changelog@3.3.0

? Select the type of change that you're committing: feat:      A new feature
? What is the scope of this change (e.g. component or file name): (press enter to skip)
? Write a short, imperative tense description of the change (max 94 chars):
(10) added json
? Provide a longer description of the change: (press enter to skip)

? Are there any breaking changes? Yes
? A BREAKING CHANGE commit requires a body. Please enter a longer description of the commit itself:
vjhifkhhk
? Describe the breaking changes:
bn
? Does this change affect any open issues? No
[master 984d6f0] feat: added json
1 file changed, 14 insertions(+)
create mode 100644 package.json
```

Рис. 4.7: Добавление файлов и выполнение коммита

4.2 Конфигурация git-flown

Выполняю инициализацию. Устанавливаю префикс для версий v. Проверяю, что активная ветка — develop и загружаю весь репозиторий.

```

tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git flow init

Which branch should be used for bringing forth production releases?
- master
Branch name for production releases: [master] v
Fatal: Local branch 'v' does not exist.
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git branch
* master
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git push all
fatal: 'all' does not appear to be a git repository
fatal: Не удалось прочитать из внешнего репозитория.

Удостоверьтесь, что у вас есть необходимые права доступа
и репозиторий существует.
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git push --all
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.20 КиБ | 1.20 МБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/ritondriy/git-extended.git
abd02a8..984d6f0 master -> master

```

Рис. 4.8: Добавление файлов и выполнение коммита

4.3 Создание релиза

Создаю релиз 1.0.0;

```

tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git flow release start 1.0.0
Переключились на новую ветку «release/1.0.0»

Summary of actions:
- A new branch 'release/1.0.0' was created, based on 'develop'
- You are now on branch 'release/1.0.0'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish '1.0.0'

```

Рис. 4.9: Создание релиза 1.0.0

```

tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git add CHANGELOG.md
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git commit -am 'chore(site): add changelog'
[release/1.0.0 567815d] chore(site): add changelog
1 file changed, 14 insertions(+)
create mode 100644 CHANGELOG.md
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git flow release finish 1.0.0
Переключились на ветку «master»
Эта ветка соответствует «origin/master».
Merge made by the 'ort' strategy.
CHANGELOG.md | 14 ++++++
1 file changed, 14 insertions(+)
create mode 100644 CHANGELOG.md
Уже на «master»
Ваша ветка опережает «origin/master» на 2 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
fatal: нет описания метки?
Fatal: Tagging failed. Please run finish again to retry.
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git push --all
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (4/4), 2.00 Киб | 1.00 Миб/с, готово.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/ritondriy/git-extended.git
984d6f0..7bd3e8e master -> master
* [new branch] release/1.0.0 -> release/1.0.0
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git push --tags
Everything up-to-date

```

Рис. 4.10: Создание релиза 1.0.0

Создаю релиз на GitHub с помощью команды: **gh release create v1.0.0 -F CHANGELOG.md**

```

tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/ritondriy/git-extended/releases/tag/v1.0.0
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$

```

Рис. 4.11: Создание релиза на GitHub

4.4 Работа с репозиторием

4.4.1 Разработка новой функциональности

Создана ветка новой функциональности с помощью команды: **git flow feature start feature_branch**. После внесения изменений выполнено объединение с develop **git flow feature finish feature_branch**. Создаю новый релиз 1.2.3: **git flow release start 1.2.3**. Обновляю package.json, затем выполняю генерацию журнала изменений и коммит Завершаю релиз и отправляю на GitHub.

```

tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git flow release start 1.2.3
Fatal: There is an existing release branch '1.0.0'. Finish that one first.
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ nano package.json
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ standard-changelog
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git add CHANGELOG.md
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git commit -am 'chore(site): update changelog'
[develop bf0beb8] chore(site): update changelog
2 files changed, 16 insertions(+), 1 deletion(-)
create mode 100644 CHANGELOG.md
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git flow release finish 1.2.3
Fatal: Branch 'release/1.2.3' does not exist and is required.
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (4/4), 1.20 КиБ | 205.00 КиБ/с, готово.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ritondriy/git-extended.git
 984d6f0..bf0beb8 develop -> develop
tereshchenkovamargo@tereshchenkovamargo:~/git-extended$ git push --tags
Everything up-to-date

```

Рис. 4.12: Работа с репозиторием

5 Выводы

Познакомилась с pass, gopass, native messaging, chezmoi. Научилась пользоваться этими утилитами, синхронизировать их с git.

Список литературы