Шаблон отчёта по лабораторной работе

Простейший вариант

Хань Цзянтао

Содержание

# 1 Цель работы

Знаком Менеджер паролей pass

# 2 Задание

1. Менеджер паролей pass
2. Управление файлами конфигурации
3. Общая информация
4. Выполнение лабораторной работы

# 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 @:

Соответствующий порт может быть указан после хоста, отделённый двоеточием (:):

Эти все записи могут быть расположены в произвольных каталогах, задающих Вашу собственную иерархию.

# 4 Выполнение лабораторной работы

### 4.0.1 **4.1Перенаправление ввода-вывода**

В системе по умолчанию открыто три специальных потока: – stdin — стандартный поток ввода (по умолчанию: клавиатура), файловый дескриптор 0; – stdout — стандартный поток вывода (по умолчанию: консоль), файловый дескриптор 1; – stderr — стандартный поток вывод сообщений об ошибках (по умолчанию: консоль), файловый дескриптор 2. Большинство используемых в консоли команд и программ записывают результаты своей работы в стандартный поток вывода stdout. Например, команда ls выводит в стан- дартный поток вывода (консоль) список файлов в текущей директории. Потоки вывода и ввода можно перенаправлять на другие файлы или устройства. Проще всего это делается с помощью символов >, >>, <, <<. Рассмотрим пример.

ls -lR > dir-tree.list  
1>filename  
1>>filename  
2>filename  
2>>filename  
&>filename

|  |
| --- |
| 01 |

01

### 4.0.2 4.2Конвейер

* Конвейер (pipe) служит для объединения простых команд или утилит в цепочки, в ко- торых результат работы предыдущей команды передаётся последующей. Синтаксис следующий: команда 1 | команда 2 Конвейеры можно группировать в цепочки и выводить с помощью перенаправления в файл, например: ls -la |sort > sortilg\_list 10 ### 4.3Поиск файла

​Команда find используется для поиска и отображения на экран имён файлов, соответ- ствующих заданной строке символов. Формат команды:

find путь [-опции]

12 Путь определяет каталог, начиная с которого по всем подкаталогам будет вестись поиск. Примеры: 1. Вывести на экран имена файлов из вашего домашнего каталога и его подкаталогов, начинающихся на f:

find ~ -name "f\*" -print

13 Здесь ~ — обозначение вашего домашнего каталога, -name — после этой опции указы- вается имя файла, который нужно найти, “f\*” — строка символов, определяющая имя файла, -print — опция, задающая вывод результатов поиска на экран. 2. Вывести на экран имена файлов в каталоге /etc, начинающихся с символа p:

find /etc -name "p\*" -print

lab4 3. Найти в Вашем домашнем каталоге файлы, имена которых заканчиваются символом ~ и удалить их:

find ~ -name "\*~" -exec rm "{}" \;

lab5 Здесь опция -exec rm “{}” ; задаёт применение команды rm ко всем файлам, име- на которых соответствуют указанной после опции -name строке символов. Для просмотра опций команды find воспользуйтесь командой man.

### 4.0.3 4.4Фильтрация текста

​ Найти в текстовом файле указанную строку символов позволяет команда grep. Формат команды:

grep строка имя\_файла

роме того, команда grep способна обрабатывать стандартный вывод других команд (любой текст). Для этого следует использовать конвейер, связав вывод команды с вводом grep. Примеры: 1. Показать строки во всех файлах в вашем домашнем каталоге с именами, начинающи- мися на f, в которых есть слово begin:

|  |
| --- |
| 14 |

14

grep begin f\*

lab6

1. Найти в текущем каталоге все файлы, содержащих в имени «лаб»:

ls -l | grep лаб

### 4.0.4 4.5Фильтрация текста

Команда df показывает размер каждого смонтированного раздела диска. Формат команды:

df [-опции] [файловая\_система]

Пример:

df -vi

lab7 Команда du показывает число килобайт, используемое каждым файлом или каталогом. Формат команды:

du [-опции] [имя\_файла...]

Пример.

du -a ~/

На afs можно посмотреть использованное пространство командой

fs quota

### 4.0.5 4.6Управление задачами

Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Для этого следует в конце имени команды указать знак амперсанда &. Например:

gedit &

Будет запущен текстовой редактор gedit в фоновом режиме. Консоль при этом не будет заблокирована. Запущенные фоном программы называются задачами (jobs). Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент задач. Для завершения задачи необходимо выполнить команду

kill %номер задачи

### 4.0.6 4.7Управление процессами

Любой команде, выполняемой в системе, присваивается идентификатор процесса (process ID). Получить информацию о процессе и управлять им, пользуясь идентифи- катором процесса, можно из любого окна командного интерпретатора. ### 4.8Получение информации о процессах Команда ps используется для получения информации о процессах. Формат команды:

ps [-опции]

Для получения информации о процессах, управляемых вами и запущенных (работаю- щих или остановленных) на вашем терминале, используйте опцию aux. Пример:

ps aux

Для запуска команды в фоновом режиме необходимо в конце командной строки ука- зать знак & (амперсанд). Пример работы, требующей много машинного времени для выполнения, и которую целесообразно запустить в фоновом режиме:

find /var/log -name "\*.log" -print > l.log &

# 5 Выводы

# Список литературы

::: {#refs}

1. GDB: The GNU Project Debugger. — URL: https://www.gnu.org/software/gdb/.
2. GNU Bash Manual. — 2016. — URL: https://www.gnu.org/software/bash/manual/.
3. Midnight Commander Development Center. — 2021. — URL: https://midnight

commander. org/.

1. NASM Assembly Language Tutorials. — 2021. — URL: https://asmtutor.com/.
2. Newham C. Learning the bash Shell: Unix Shell Programming. — O’Reilly

Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL:

http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658.

1. Robbins A. Bash Pocket Reference. — O’Reilly Media, 2016. — 156 с. — ISBN

978-1491941591.

1. The NASM documentation. — 2021. — URL: https://www.nasm.us/docs.php.
2. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN
3. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
4. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. :

Солон-Пресс, 2017.

1. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
2. Расширенный ассемблер: NASM.— 2021.—URL: https://www.opennet.ru/docs/RUS/nasm/.
3. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е

изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.

1. Столяров А. Программирование на языке ассемблера NASM для ОС Unix.— 2-

е изд.—М. : МАКС Пресс, 2011.—URL: http://www.stolyarov.info/books/asm\_unix.:::