**Санкт-Петербургский государственный УНИВЕРСИТЕТ**

**Факультет прикладной математики и процессов управления**

**отчет**

**по лабораторной работе №2**

**по дисциплине «системное программирование в Linux»**

**на тему «Демон резервного копирования»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 22Б16 |  | Шувалов Ф.В. |
| Преподаватель |  | Киямов Ж.У. |

**Санкт-Петербург**

**2023 г.**

**Оглавление**

[1. Цель 4](#_Toc152349454)

[2. Задача 4](#_Toc152349455)

[3. Теория 4](#_Toc152349456)

[4. Решение 4](#_Toc152349457)

[5. Рекомендации программиста 5](#_Toc152349459)

[6. Рекомендации пользователю 5](#_Toc152349460)

[7. Пример 5](#_Toc152349461)

[8. Вывод 6](#_Toc152349467)

[9. Список литературы 7](#_Toc152349468)

# Цель

Создать демона (системный процесс, который работает в фоновом режиме) для автоматического регулярного резервного копирования данных с одного каталога в другой.

# Задачи

Задача включает в себя следующие шаги:

1. **Настройка конфигурации:** Разработайте конфигурационный файл, в котором можно указать исходный каталог, каталог для резервных копий, частоту резервного копирования и другие параметры.
2. **Создание демона**: Напишите код демона на языке C++, который будет выполнять следующие задачи:

* Считывать конфигурацию из конфигурационного файла.
* В фоновом режиме ожидать наступления времени для следующей резервной копии.
* Создавать резервные копии файлов из исходного каталога в каталог для резервных копий с добавлением временной метки.
* Журналировать выполнение операций в системный журнал.

1. **Интеграция с системой:** Внедрите демона в систему, чтобы он запускался при старте операционной системы и регулярно выполнял резервное копирование согласно настройкам.
2. **Управление и мониторинг:** Создайте простой механизм для управления демоном, например, возможность остановки, перезапуска или изменения настроек через командную строку. Также предусмотрите механизм мониторинга, чтобы следить за статусом демона и успешностью резервного копирования.
3. **Документация:** Создайте документацию для пользователя, в которой описаны настройки, команды управления и процесс резервного копирования.
4. **Тестирование и оптимизация:** Протестируйте демона на различных сценариях и оптимизируйте его для эффективной работы и минимизации нагрузки на систему.
5. **Безопасность:** Обеспечьте безопасность демона, ограничив доступ к конфигурационным файлам и резервным копиям.

# Теория

Демон — это фоновый процесс в операционной системе, который выполняется независимо от активности пользователя и обычно предназначен для выполнения служебных задач. Он отличается от обычных пользовательских программ тем, что работает в фоновом режиме, не взаимодействует напрямую с пользователем и часто запускается при старте системы.

Концепция Backup Daemon (демона для резервного копирования) представляет собой программный процесс, который работает в фоновом режиме и отвечает за автоматическое регулярное создание резервных копий данных. Демоны резервного копирования часто используются для обеспечения безопасности данных, сохранения целостности информации и возможности восстановления в случае сбоев или потери данных.

# Решение

*Таблица 4.2 Таблица Функций*

|  |  |
| --- | --- |
| Имя | Описание |
| main | Основная функция, запускается при вызове программы из консоли |
| Signal\_handler | Функция для непосредственного сканирования порта |
| Scan\_host | Функция для непосредственного сканирования IP |
| Setup\_ui | Функция для настройки интерфейса |
| Scan\_ip | Функция для считывания введенной информации и сканирования IP |
| Scan\_ports | Функция для считывания введенной информации и для сканирования портов |
| Save\_results | Функция, сохраняет результаты |
| Display\_results | Функция, которая выводит результаты на экран |

# Рекомендации программисту

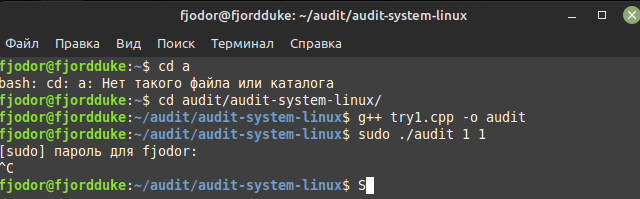
Для запуска программы необходима 64-битная операционная система Linux и python версии не ниже 3.10. Для корректной работы программы требуются библиотеки, указанные в README.md файла github .

# Рекомендации пользователю

Выполните действия, указанные на github (<https://github.com/zer0rbt/Linux/course2/scanner>). Для запуска Вам нужно иметь права superuser’a - иначе не гарантируется корректная работа программы. При наличии ошибок проверьте консоль. При невозможности устранения ошибок самостоятельно рекомендуется переустановить программу и написать разработчику.

# Контрольный Пример

Запустив программу, пользователь увидит окно. В нем можно ввести IP адрес и диапазон портов для сканирования. Для начала сканирования и сохранения результатов следует нажать на соответствующие кнопки.



*Рисунок 7.1 Основное окно программы*

# Вывод

Разработанный порт-сканнер представляет собой эффективный инструмент для администраторов сети и специалистов по безопасности для обнаружения потенциальных уязвимостей в сетевой инфраструктуре. Программа обладает простым, но мощным интерфейсом, что делает ее доступной широкому кругу пользователей.

# Список литературы

<https://github.com/zer0rbt/Linux/course2/scanner>