**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики-процессов управления**

**Программа бакалавриата**

**“Большие данные и распределенная цифровая платформа”**

**ОТЧЕТ**

**по лабораторной работе №1**

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

**на тему «****Создание демона резервного копирования»**

**Студент гр. 23Б16-пу**

**Аникин Р. С.**

**Преподаватель**

**Киямов Ж. У.**

**Санкт-Петербург**

**2024 г.**

**Оглавление**

[**Цель работы** 2](#_Toc185238724)

[**Описание задачи (формализация задачи)** 2](#_Toc185238725)

[**Теоретическая часть** 2](#_Toc185238726)

[**Описание программы** 3](#_Toc185238727)

[**Рекомендации пользователю** 4](#_Toc185238728)

[**Рекомендации программисту** 4](#_Toc185238729)

[**Исходный код программы** 4](#_Toc185238730)

[**Контрольный пример** 4](#_Toc185238731)

[**Вывод** 6](#_Toc185238732)

[**Источники** 6](#_Toc185238733)

# **Цель работы**

Цель лабораторной работы – разработка демона для регулярного резервного копирования данных с одного каталога в другой с учётом конфигурационных настроек и минимизацией системной нагрузки.

# **Описание задачи (формализация задачи)**

Необходимо разработать демон для автоматического резервного копирования данных с использованием языка программирования Python. Задача включает:

1. Чтение параметров из конфигурационного файла.

2. Регулярное создание резервных копий с добавлением временной метки.

3. Журналирование операций в системный лог.

4. Запуск демона при старте системы.

5. Возможность управления и мониторинга демона через командную строку.

6. Обеспечение безопасности данных и конфигурационных файлов.

# **Теоретическая часть**

Для реализации задачи используется Python 3.12 с библиотеками:

- os – для работы с файловой системой.

- shutil – для копирования каталогов.

- schedule – для планирования задач.

- daemon – для работы с системными процессами.

- logging – для ведения системного журнала.

- atexit – для регистрации действий при завершении работы демона.

**Основные шаги программы**

1) Загрузка конфигурации из файла `config.json`.

2) Планирование задач резервного копирования с использованием библиотеки `schedule`.

3) Регулярное выполнение задач в фоновом режиме.

4) Журналирование успешных операций и ошибок.

5) Управление остановкой и перезапуском демона.

# **Описание программы**

Программная реализация написана на языке Python 3.12.6 с использованием библиотек os, shutil, schedule, daemon, logging, atexit. Программа организована в виде одного файла с шестью подпрограммами:

Таблица 1. Описание программы

|  |  |
| --- | --- |
| load\_config() | Подгрузка настроек из файла конфигурации |
| backup\_files() | Создание резервной копии |
| run\_daemon() | Запуск планировщика в фоновом режиме |
| main() | Основная функция для запуска демона |

Конфигурационный файл `config.json` включает параметры:

- Исходный каталог.

- Каталог для резервных копий.

- Частота выполнения резервного копирования.

Пример конфигурационного файла:

json

{

"source\_directory": "/path/to/source",

"backup\_directory": "/path/to/backup",

"backup\_interval": 3600

}

## **Рекомендации пользователю**

1. Убедитесь, что директории для исходных данных и резервных копий существуют.

2. Проверьте права доступа к конфигурационному файлу.

3. Для управления демоном используйте команды:

- `start`: запуск демона.

- `stop`: остановка демона.

- `restart`: перезапуск

## **Рекомендации программисту**

* Поддерживайте актуальность используемых библиотек (Random, Time, CSV) и версии Python (минимальная версия - 3.12.6) для сохранения актуальности и работоспособности кода на современных системах.
* Если пользователь попросит вас добавить его данные в программу-генератор, организуйте эти данные в соответствии с примерами в соответствующих папках в папке input-data. Все три файла назовите одинаково. Сообщите пользователю название файлов, чтобы он мог ими воспользоваться.

# **Исходный код программы**

[**---**](https://github.com/RomanMuteki/simple_dataset_generator)

# **Контрольный пример**

Проверка статуса демона (systemctl status backup\_daemon)

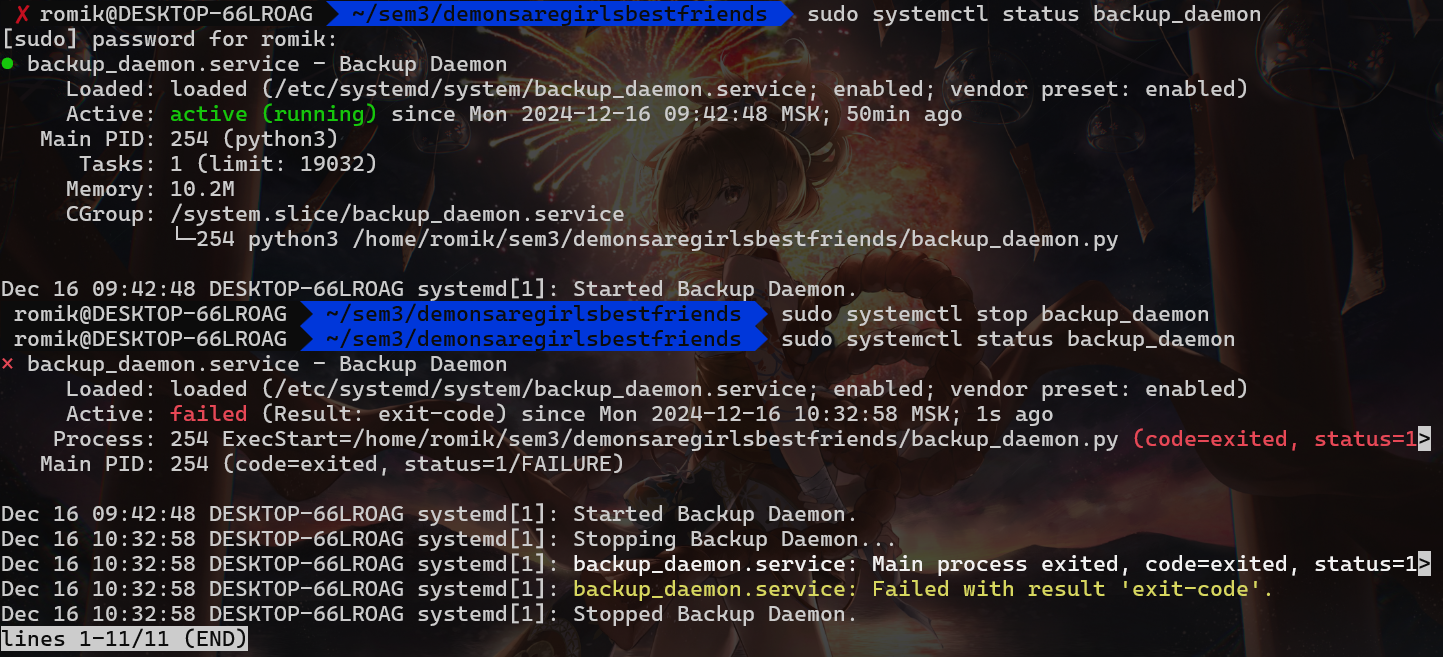


Рис. 1. Вывод консоли

Если демон не запущен, запускаем его

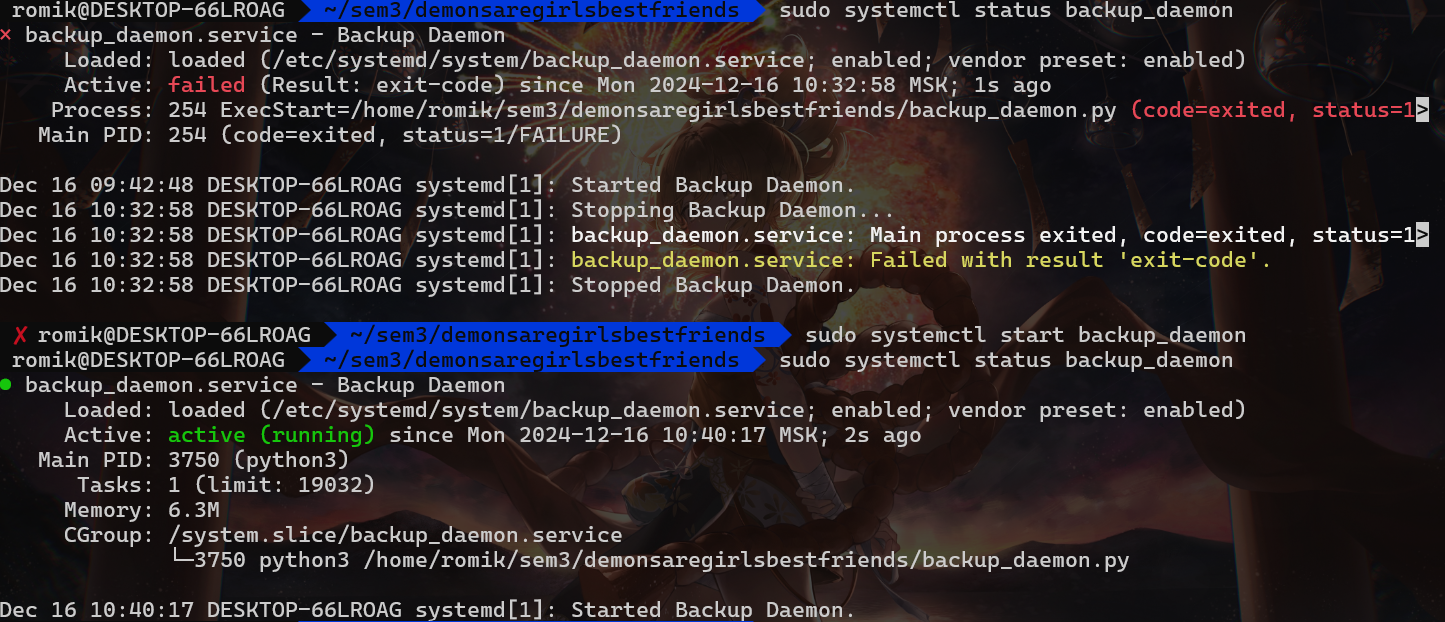


Рис. 2. Запуск демона

Проводим махинации с файлами, находящимися в папке, которая резервно копируется.



Рис. 3. Изменены файлы в резервируемой директории

Проверяем директорию, в которую происходит резервное копирование

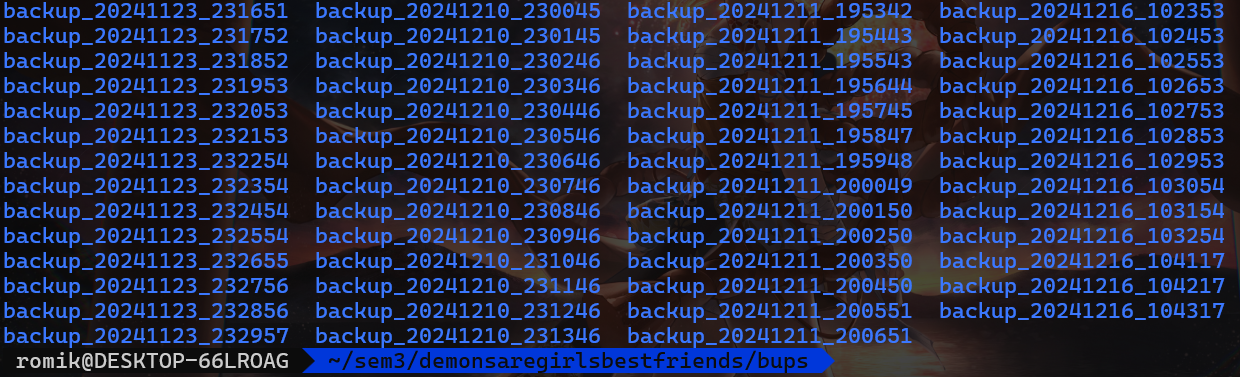


Рис. 3. Директория с бэкапами

Проверяем содержимое некоторых папок с бэкапами

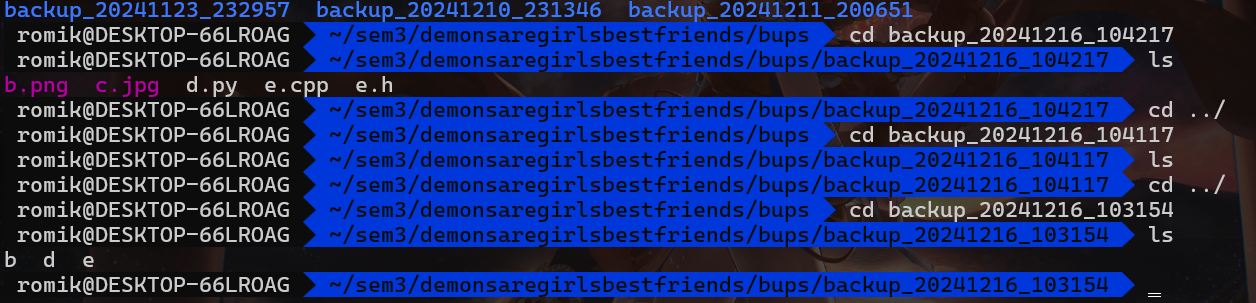


Рис. 4. Содержимое некоторых бэкапов

# **Вывод**

В ходе лабораторной работы был разработан демон для регулярного резервного копирования данных. Реализован функционал чтения конфигурации, автоматического выполнения задач, журналирования и управления процессом. Программа протестирована на различных сценариях, включая обработку ошибок и мониторинг.

# **Источники**

1. Daemons documentation // daemons URL: https://daemons.readthedocs.io/en/latest/ (дата обращения: 10.10.2024).