**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики-процессов управления**

**Программа бакалавриата**

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

**ОТЧЕТ**

**по лабораторной работе №3**

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

**на тему «****Создание службы аудита системы»**

**Студент гр. 23Б16-пу**

**Аникин Р. С.**

**Преподаватель**

**Киямов Ж. У.**

**Санкт-Петербург**

**2024 г.**

**Оглавление**

[**Цель работы** 2](#_Toc179554877)

[**Описание задачи (формализация задачи)** 2](#_Toc179554878)

[**Теоретическая часть** 3](#_Toc179554879)

[**Основные шаги программы** 4](#_Toc179554880)

[**Блок-схема программы** 5](#_Toc179554881)

[Рис. 1. Блок-схема основной программы 5](file:///A:\educ\algos\Отчёт%201.docx#_Toc179554882)

[Рис. 2. Блок-схема подпрограмм 1 6](file:///A:\educ\algos\Отчёт%201.docx#_Toc179554883)

[Рис. 3. Блок-схема подпрограмм 2 7](file:///A:\educ\algos\Отчёт%201.docx#_Toc179554884)

[Рис. 4. Блок-схема подпрограмм 3 8](file:///A:\educ\algos\Отчёт%201.docx#_Toc179554885)

[**Описание программы** 9](#_Toc179554886)

[**Рекомендации пользователю** 9](#_Toc179554887)

[**Рекомендации программисту** 10](#_Toc179554888)

[**Исходный код программы** 10](#_Toc179554889)

[**Контрольный пример** 10](#_Toc179554890)

[**Вывод** 12](#_Toc179554891)

[**Источники** 12](#_Toc179554892)

# **Цель работы**

Цель лабораторной работы – разработать инструмент для аудита и мониторинга системы Linux с возможностью регистрации событий, фильтрации, оповещения и анализа журнала событий.

# **Описание задачи (формализация задачи)**

Разработать программу, которая будет:

1. Регистрировать события: запуск и завершение процессов, изменения файлов, сетевые операции.
2. Сохранять информацию о событиях в журнале.
3. Предоставлять возможность фильтрации событий по различным критериям (пользователь, тип, время).
4. Реализовывать оповещения об избранных событиях.
5. Генерировать отчёты и статистику на основе журнала.
6. Обеспечивать безопасность данных и ограниченные привилегии доступа.

# **Теоретическая часть**

Для выполнения задачи используются:

* **Watchdog** – для мониторинга изменений файлов.
* **/proc** – для отслеживания процессов и сетевых соединений.
* **Logging** – для ведения журналов.
* **Tkinter** – для разработки интерфейса анализа логов.
* **Threading** – для многопоточного выполнения мониторинга.

**Основные шаги программы**

1. Настройка системы логирования для событий файловой системы, процессов и сетевых операций.
2. Мониторинг файловой системы с использованием FileSystemEventHandler из watchdog.
3. Отслеживание процессов через /proc и логирование их состояния.
4. Сбор сетевых соединений из /proc/net.
5. Создание интерфейса для анализа логов с фильтрацией и поиском.
6. Обеспечение безопасности работы программы через ограничения прав доступа.

## Рис. 3. Блок-схема подпрограмм 2

# **Описание программы**

Программная реализация написана на языке Python 3.12.6 с использованием библиотек os, sys, logging, threading, watchdog, time, socket, struct, getpass. Программа организована в виде одного файла с шестью подпрограммами:

Таблица 1. Описание программы auditor.py

|  |  |
| --- | --- |
| Функция | Описание |
| setup\_logger | Настройка логгера для записи сообщений в файл |
| monitor\_processes | Мониторинг процессов в системе |
| monitor\_network | Мониторинг сетевых соединений |
| hex\_to\_ip | |  | | --- | |  |  |  | | --- | | Преобразование IP-адреса из шестнадцатеричного формата в строковый вид | |
| FileAuditHandler | |  | | --- | |  |  |  | | --- | | Обработчик событий файловой системы | |
| on\_created | Обработка события создания файла или папки |
| on\_deleted | Обработка события удаления файла или папки |
| on\_moved | Обработка события перемещения или переименования файла |
| find\_original\_file | Поиск оригинального файла для временного файла |

Таблица 2. Описание функций программы filter.py

|  |  |
| --- | --- |
| Функция | Описание |
| \_\_init\_\_ | Инициализирует приложение и создает интерфейс пользователя. |
| load\_file | Загружает выбранный лог-файл и отображает его содержимое. |
| display\_file\_content | Отображает содержимое текущего выбранного лог-файла в текстовой области. |
| apply\_filter | Применяет фильтры по пользователю, событию и временному интервалу к лог-файлу. |
| search | Выполняет поиск по ключевому слову в содержимом лог-файла. |
| display\_filtered\_content | Отображает отфильтрованные строки в текстовой области. |
| show\_error | Отображает сообщение об ошибке в диалоговом окне. |

## **Рекомендации пользователю**

**Запуск программы**:

* Откройте терминал или командную строку и перейдите в папку с программой.
* Запустите программу с помощью команды:



Здесь <путь\_к\_папке> — это путь к папке, за которой вы хотите вести мониторинг.

· **Использование программы**:

* После запуска программа начнет отслеживать изменения в указанной папке. Она будет логировать события, такие как создание, удаление и перемещение файлов и папок, а также запуск и завершение процессов и сетевые соединения.
* Логи будут записываться в файлы:
  + file\_system\_log.txt — для событий файловой системы.
  + process\_log.txt — для мониторинга процессов.
  + network\_log.txt — для сетевых соединений.
* Вы можете просмотреть эти логи, открыв их с помощью текстового редактора или командой cat в терминале.
* При желании можно воспользоваться фильтром запустив в терминале интерфейс при помощи команды: python3 filtering.py

## **Рекомендации программисту**

Для успешного запуска и работы программы важно проверить несколько аспектов и убедиться, что установлены необходимые библиотеки. Вот шаги, которые нужно выполнить:

### **1.** **Проверка установленной версии Python**

* Убедитесь, что у вас установлен Python 3.x. Проверьте версию Python с помощью команды:
* Если Python не установлен, загрузите и установите его с официального сайта [python.org](https://www.python.org/downloads/).

### **2.** **Установка необходимых библиотек**

* Проверьте, что у вас установлены все необходимые библиотеки. В программе используются следующие модули:
  + **os**, **sys**, **logging**, **time**, **socket**, **struct** — встроенные модули Python, устанавливать их не нужно.
  + **threading** — также встроенный модуль.
  + **watchdog** — внешний модуль для отслеживания изменений файловой системы.
* Установите **watchdog**, если он не установлен.

# **Исходный код программы**

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

# **Контрольный пример**

Переход в директорию программы и запуск

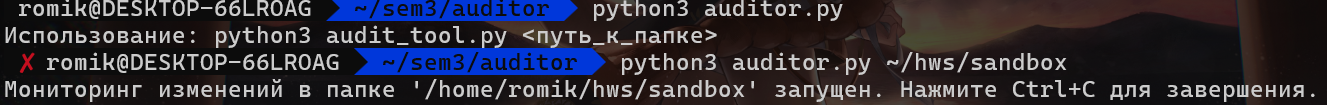


Рис. 1. Запуск программы

Производим изменения в отслеживаемой папке



Рис. 2. Изменение файлов в отслеживаемой директории

Запускаем filter.py для просмотра логов

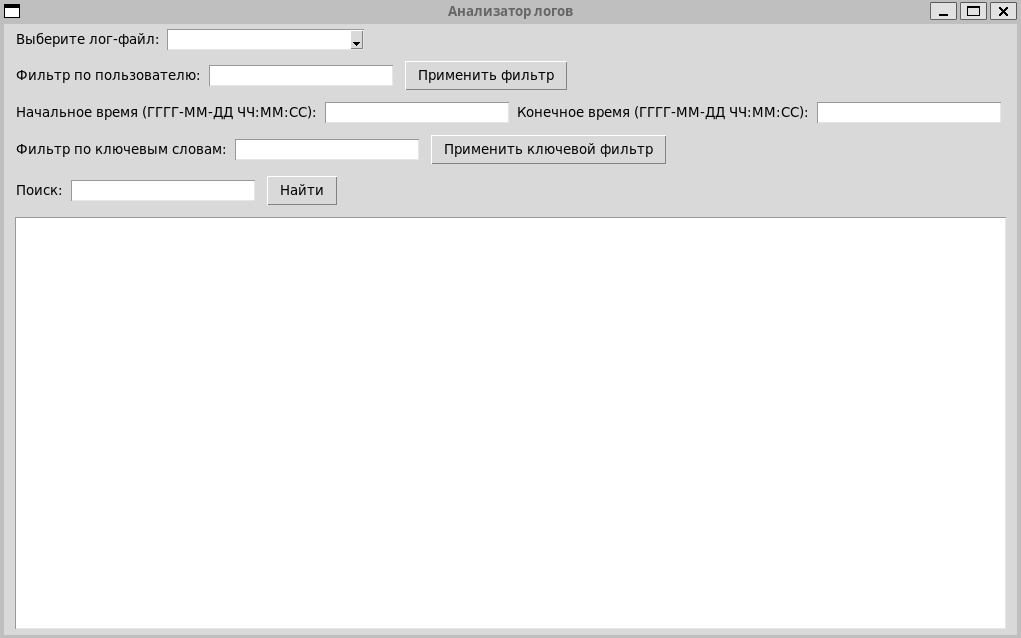


Рис. 3. Запущенный filter.py

Выбираем файл логов, и он открывается. Можно применить фильтры

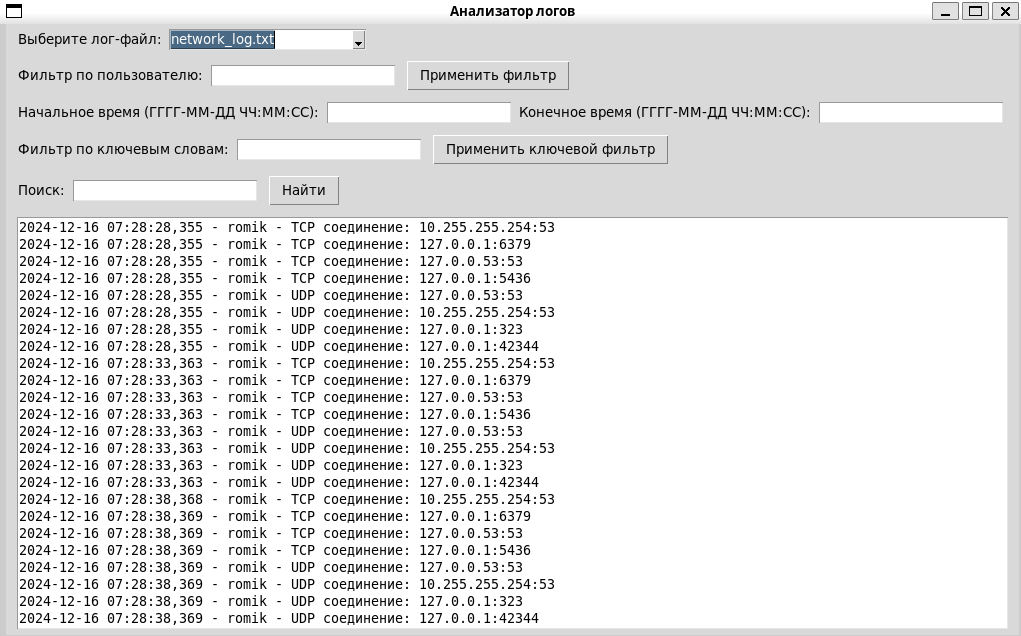


Рис. 4. Отображены логи

Проверяем содержимое некоторых папок с бэкапами

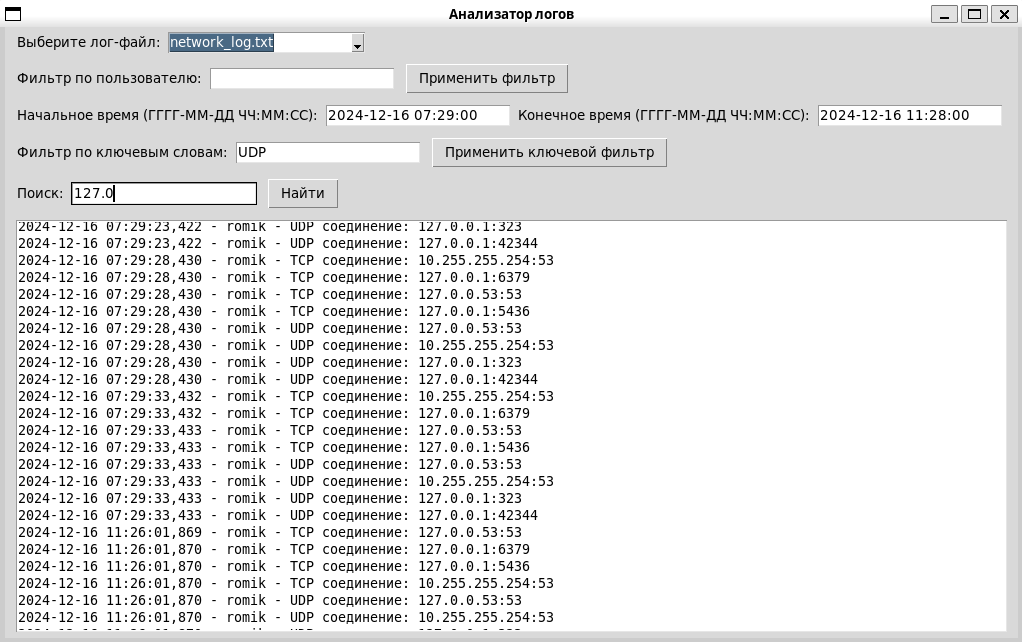


Рис. 4. Применены фильтры

Проверяем почту на факт наличия уведомлений об удалении файла

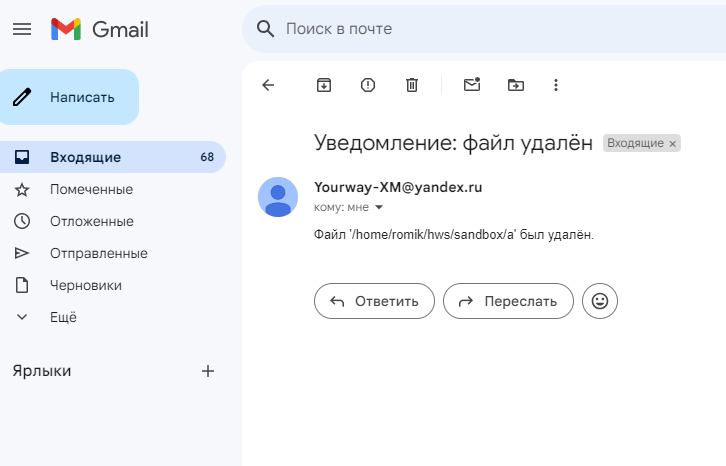


Рис.5 пример уведомления

По завершении исполнения программа автоматически формирует отчёт с историей изменений:

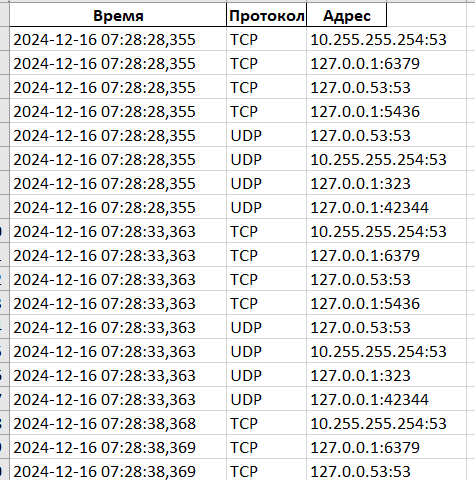


Рис.6. Пример отчёта

# **Вывод**

Разработанная программа является полезным инструментом для мониторинга изменений в файловой системе, процессов и сетевых соединений на **Linux**. Она позволяет в реальном времени отслеживать создание, удаление и перемещение файлов, а также запуск и завершение процессов, что может быть полезно для мониторинга активности на сервере или в рабочей среде.

Программа эффективно использует многозадачность, что позволяет одновременно мониторить процессы и сетевые операции, минимизируя при этом нагрузку на основной процесс. Все события логируются в файлы, что позволяет сохранять информацию для дальнейшего анализа.

# **Источники**

1) **watchdog** — Библиотека для мониторинга изменений в файловой системе.  
[*https://python-watchdog.readthedocs.io/en/stable/*](https://python-watchdog.readthedocs.io/en/stable/)  
Дата обращения: 15.12.2024

2) **logging** — Модуль для ведения журналов в Python.  
[*https://docs.python.org/3/library/logging.html*](https://docs.python.org/3/library/logging.html)  
Дата обращения: 15.12.2024

3) **socket** — Библиотека для работы с сетевыми соединениями в Python.  
[*https://docs.python.org/3/library/socket.html*](https://docs.python.org/3/library/socket.html)Дата обращения: 15.12.2024

4) **struct** — Библиотека для работы с двоичными данными и упаковкой/распаковкой данных в Python.  
[*https://docs.python.org/3/library/struct.html*](https://docs.python.org/3/library/struct.html)  
Дата обращения: 15.12.2024