**.МИНОБРНАУКИ РОССИИ**

**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 «Разработка системного инструмента для аудита системы»**

**Студент гр. 23Б15-пу**

**Кубякин Н.А.**

**Преподаватель**

**Киямов Ж. У.**

**Санкт-Петербург**

**2024 г.**

***Оглавление***

[1. Цель работы 3](#_Toc178797423)

[2. Описание задачи (формализация задачи) 3](#_Toc178797424)

[3. Теоретическая часть 4](#_Toc178797431)

[4. Основные шаги программы 4](#_Toc178797431)

[5. Описание программы 4](#_Toc178797442)

[6. Рекомендации пользователя 4](#_Toc178797443)

[7. Рекомендации программиста 5](#_Toc178797444)

[8. Исходный код программы 5](#_Toc178797445)

[9. Контрольный пример 6](#_Toc178797447)

[10. Вывод 8](#_Toc178797447)

[11. Источники 8](#_Toc178797448)

# Цель работы

Разработать системный инструмент, который будет использоваться для аудита и мониторинга системы Linux.

# Описание задачи (формализация задачи)

* Создать программу, которая будет регистрировать события в системе.
* Программа должна сохранять информацию о событиях в журнале событий.
* Реализовать возможность фильтрации и поиска событий в журнале по различным критериям, таким как пользователь, тип события, время и другие.,

# Теоретическая часть

# Системные вызовы в Linux — это механизм взаимодействия пользовательских программ с ядром операционной системы. Когда программе требуется системный ресурс, она отправляет запрос на этот ресурс ядру через системный вызов.

# Модуль psutil - это кроссплатформенная библиотека для получения информации о запущенных процессах и использовании системы (ЦП, память, диски, сеть, датчики) в Python. Этот модуль полезен для мониторинга системы, профилирования, ограничения ресурсов процессов и управления запущенными процессами. Он реализует множество функций, предлагаемых инструментами командной строки UNIX.

# Основные шаги программы

1. Настраивается логирование в файл с указанием формата записи.
2. Перебираются все процессы системы и логируются в нужном формате в указанный файл с частотой в 20 секунд
3. Список процессов может быть отфильтрован по имени пользователя, идентификатору процесса или дате

# Описание программы

Описание основных компонентов программы представлено в таблице 4.

*Таблица 4. Описание функций*

|  |  |  |
| --- | --- | --- |
| Имя | Параметры | Описание |
| escape\_string\_for\_json | s: Переменная, содержащая строку лога | Вспомогательная функция для переносов строк в json |
| format\_and\_log\_process | process\_info: Переменная, содержащая строку с информацией о процессе | Функция для форматирования строк в json и их сохранения |
| monitor\_processes | - | Функция для мониторинга системы |

# Рекомендации пользователя

# **1.** sudo python3 system\_monitor.py

**2.** Логи сохранятся в отдельный файл system\_audit.log, а так же выведутся в терминал

3. sudo python3 system\_log\_search.py

* + - * 1. 4. Откроется окно для поиска и фильтрации лога по имени, идентификатору и/или дате

5. Для завершения закройте окно программы и терминала.

# Рекомендации программиста

# Для запуска программы необходима 64-битная операционная система Linux. Для работы с кодом необходима среда разработки, совместимая с python 3.1 и библиотеки psutil[1], tkinter[2], logging[3], json[4]

# Минимальное необходимое место на диске: 1 МБ. Минимальное необходимое количество оперативной памяти: 25 МБ.

# Исходный код программы

# Исходный код программы и необходимые текстовые файлы доступны по ссылке: <https://github.com/NikiTaku1/spbu_linux/tree/main/lab3>

# Контрольный пример

В данном разделе представлены контрольные примеры для каждой из задач, демонстрирующие способность выполнять комплексную обработку данных.

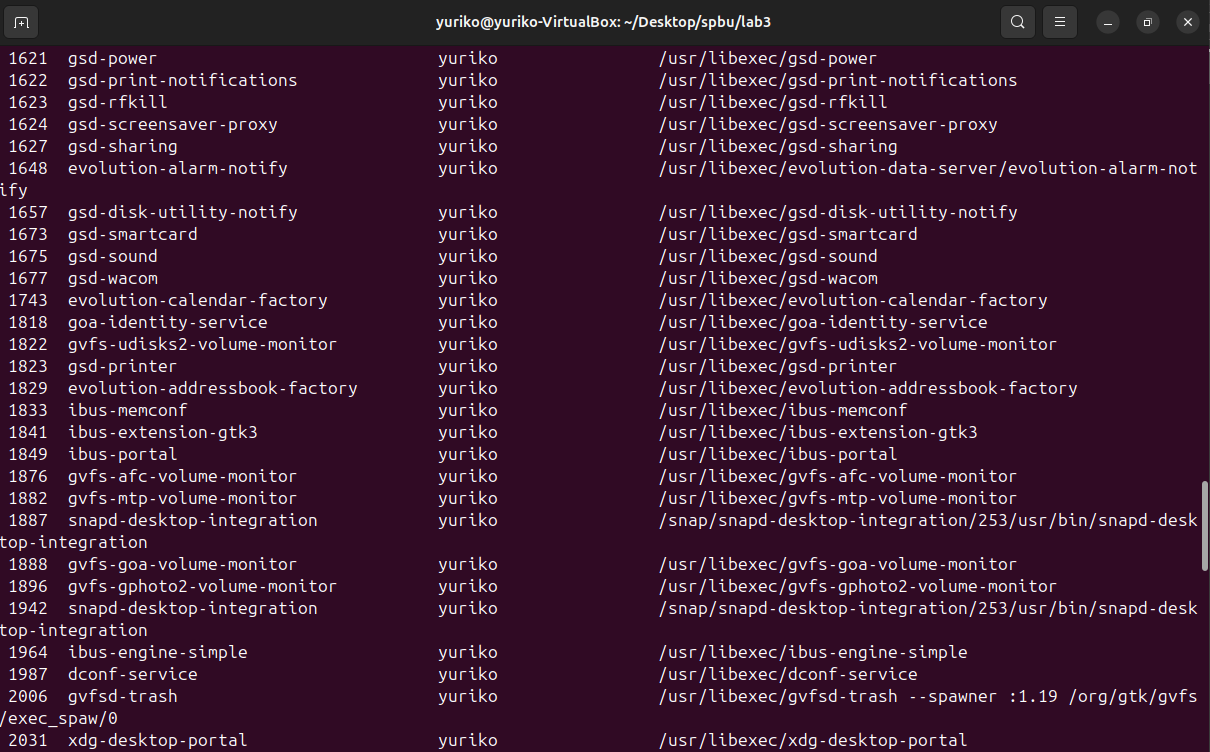


Рис. 9.1 Вывод в терминал.

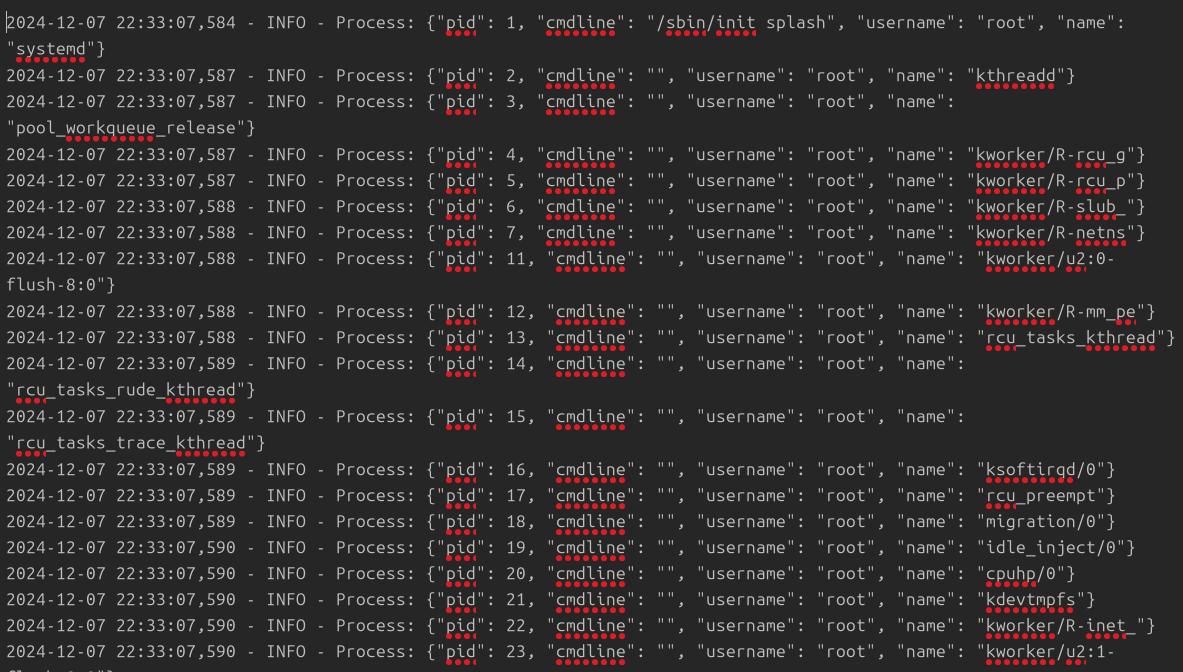


Рис. 9.2 Логи.

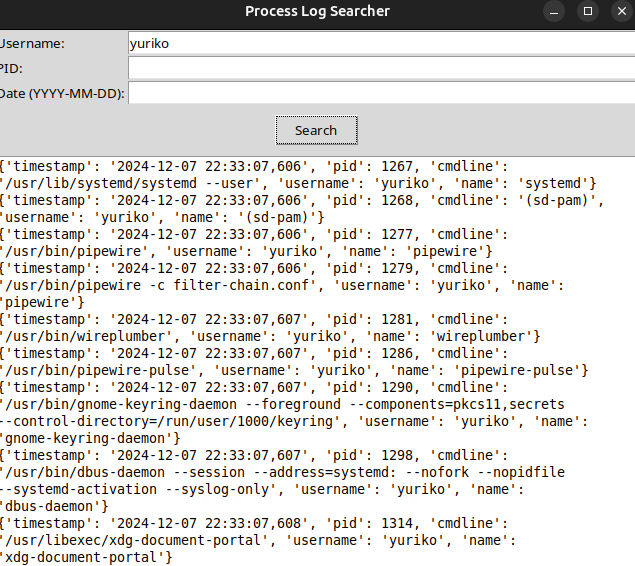


Рис. 9.3 Работа фильтра по имени.

# Вывод

# В ходе разработки программы был реализован функционал для аудита системы Linux с помощью библиотеки psutil. Основной функционал программы заключается в логировании системных процессов и поиска и фильтрации по списку с логами.

# Источники

# psutil — psutil documentation // Documentation URL: <https://psutil.readthedocs.io/en/latest/index.html> (дата обращения: 07.12.2024).

# tkinter — tkinter documentation // Documentation URL: <https://docs.python.org/3/library/tkinter.html> (дата обращения: 07.12.2024).

# logging — logging documentation // Documentation URL: <https://docs.python.org/3/library/logging.html> (дата обращения: 07.12.2024).

# json — json documentation // Documentation URL: <https://docs.python.org/3/library/json.html> (дата обращения: 07.12.2024).