**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**Факультет прикладной математики – процессов управления**

**отчет**

**по лабораторной работе**

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

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

|  |  |  |
| --- | --- | --- |
| Студент гр. 22.Б15 |  | Агишев А.Б. |
| Преподаватель |  | Киямов Ж.У. |

**Санкт-Петербург**

**2023 г.**

**Содержание**

[1. Цель работы 2](#_Toc154235591)

[2. Задача 2](#_Toc154235592)

[3. Теоретическая часть 2](#_Toc154235593)

[4. Алгоритм метода 3](#_Toc154235594)

[5. Описание программы 3](#_Toc154235595)

[6. Рекомендации пользователю 3](#_Toc154235596)

[7. Рекомендации программисту 3](#_Toc154235597)

[8. Контрольный пример 4](#_Toc154235598)

[9. Заключение 4](#_Toc154235599)

[10. Литература 4](#_Toc154235600)

# **Цель работы**

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

# **Задача**

1. Определить список системных вызовов.
2. Написать программу для аудита и мониторинга.
3. Проверить работоспособность программы

# **Теоретическая часть**

Аудит и мониторинг системы Linux представляют собой процессы наблюдения и анализа активности в операционной системе Linux с целью обеспечения безопасности, целостности данных и эффективности работы системы.

*Аудит*:

Определение событий: Наблюдение за различными событиями в системе, такими как системные вызовы, файловые операции, аутентификация и другие действия пользователей.

Запись событий: Фиксирование информации о событиях в аудиторских журналах для последующего анализа.

Анализ событий: Анализ журналов аудита для выявления потенциальных угроз безопасности и отслеживания активности пользователей.

*Мониторинг*:

Слежение за ресурсами: Отслеживание использования системных ресурсов, таких как ЦП, память, сеть и диски, для обеспечения эффективности и предотвращения проблем.

Анализ производительности: Оценка производительности системы, выявление узких мест и оптимизация работы приложений.

Выявление нештатных ситуаций: Мониторинг для обнаружения отклонений от нормального функционирования, таких как сбои в работе приложений или аномальное использование ресурсов.

# **Алгоритм метода**

После запуска программы алгоритм будет мониторить новые и истекшие процессы в /proc/[pid] и выводить их в консоль и в журнал.

# **Описание программы**

Описание функций программы *«main.cpp»* представлено в таблице 5.1.

*Таблица 5.1. Описание функций программы «main.cpp»*

|  |  |  |
| --- | --- | --- |
| Входные данные | Имя | Описание |
| Старый и новый процесс | getNewProcesses | Мониторинг новых процессов |
| Старый и новый процесс | getExpiredProcesses | Мониторинг истекших процессов |
| Процесс, флаг | makeMessage | Создание записи |
| PID, токен | getProcessStatusValue | Создание движений для фигур |
| PID | getUptime | Получение времени работы процесса |
| - | getpids | Управление получением информации |
| Номер сигнала | signalHandler | Обработчик сигналов завершения работы |
| - | openLog | Создание журнала |
| - | main | Мониторинг, вызов других функций |

# **Рекомендации пользователю**

Скомпилируйте программу, запустите ее, отслеживаете процессы.

# **Рекомендации программисту**

Для запуска программы необходим C++ 20.

# **Контрольный пример**



*Рисунок 8.1 Пример работы программы.*

# **Заключение**

В результате выполнения работы был разработан системный инструмент для аудита и мониторинга системы Linux. Этот инструмент включает в себя определение списка системных вызовов, написание программы для аудита и мониторинга, а также проверку её работоспособности.

# **Литература**

1. <https://github.com/finstape/system-programming-in-linux>