**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 |  | Добренкова Л.С. |
| Преподаватель |  | Киямов Ж.У. |

**Санкт-Петербург**

**2023 г.**

Содержание

[1. Цель работы 2](#__RefHeading___Toc836_2594690726)

[2. Задача 2](#__RefHeading___Toc838_2594690726)

[3. Теоретическая часть 2](#__RefHeading___Toc840_2594690726)

[4. Описание программы 3](#__RefHeading___Toc842_2594690726)

[5. Рекомендации пользователю 4](#__RefHeading___Toc844_2594690726)

[6. Рекомендации программисту 4](#__RefHeading___Toc846_2594690726)

[7. Контрольный пример 4](#__RefHeading___Toc848_2594690726)

[8. Заключение 5](#__RefHeading___Toc850_2594690726)

[9. Литература 5](#__RefHeading___Toc852_2594690726)

[https://habr.com/ru/articles/111266/ 5](#__RefHeading___Toc794_1655828954)

[https://ru.wikipedia.org/wiki/Сигнал\_(Unix) 5](#__RefHeading___Toc796_1655828954)

# **Цель работы**

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

# **Задача**

#### **Проектирование:**

* + Разработка алгоритма мониторинга системных вызовов в процессах.
  + Параметризированный запуск.

#### **Работа с ptrace:**

* + Изучение основ работы с ptrace.
  + Интеграция ptrace в проект.
  + Реализация механизма обработки ошибок.

#### **Логгирование и фильтрация:**

* + Выбор способа логгирования.
  + Изучение способов фильтрации.
  + Реализация graceful shutdown с обработкой сигналов.

# **Теоретическая часть**

#### **Ptrace PTRACE\_ATTACH:**

Позволяет подключиться к процессу по его pid.

#### ***Ptrace PTRACE\_SETOPTIONS:***

Использование системного вызова ptrace для установки опций отслеживания для указанного процесса с идентификатором pid.

#### **waitpid:**

Ожидание процесса с идентификатором pid.

#### **Ptrace PTRACE\_GETREGS:**

Системный вызов, который используется для получения значений регистров процессора дочернего процесса.

#### Ptrace PTRACE\_SYSCALL:

Используется для продолжения выполнения процесса, остановленного в результате системного вызова.

#### Ptrace PTRACE\_DETACH:

Отключается от процесса.

# **Описание программы**

Описание основных компонентов программы *«main.cpp»* представлено в таблице 4.1.

*Таблица 4.1. Описание переменных программы «main.cpp»*

|  |  |  |
| --- | --- | --- |
| Имя | Параметры | Описание |
| onSigterm | int signum – номер сигнала | Обработчик сигнала SIGINT (Ctrl+C). Детачит процесс, используя ptrace, и выводит сообщение о детаче. |
| initPidArg | int argc, char\*\* argv – аргументы main | Инициализация PID из аргументов командной строки. |
| getFormatTimestamp | - | Генерирует отформатированную временную метку в формате "%d.%m.%YT%H:%M:%S". |
| getPrefix | int pid - PID процесса. | Генерирует префикс для логгирования, включая PID и временную метку. |
| logSystemCall | unsigned long long system\_code - Код системного вызова.  int pid - PID процесса. | Логгирует информацию о системном вызове. |
| logCommand | const char\* line - Строка с информацией о команде.  int pid - PID процесса. | Логгирует сообщения мониторинга. |
| ptraceInit | int pid - PID процесса. | Инициализация отслеживания процесса с использованием ptrace. |

# **Рекомендации пользователю**

https://github.com/v131v/linux\_labs\_3\_sem/blob/main/lab3/README.md

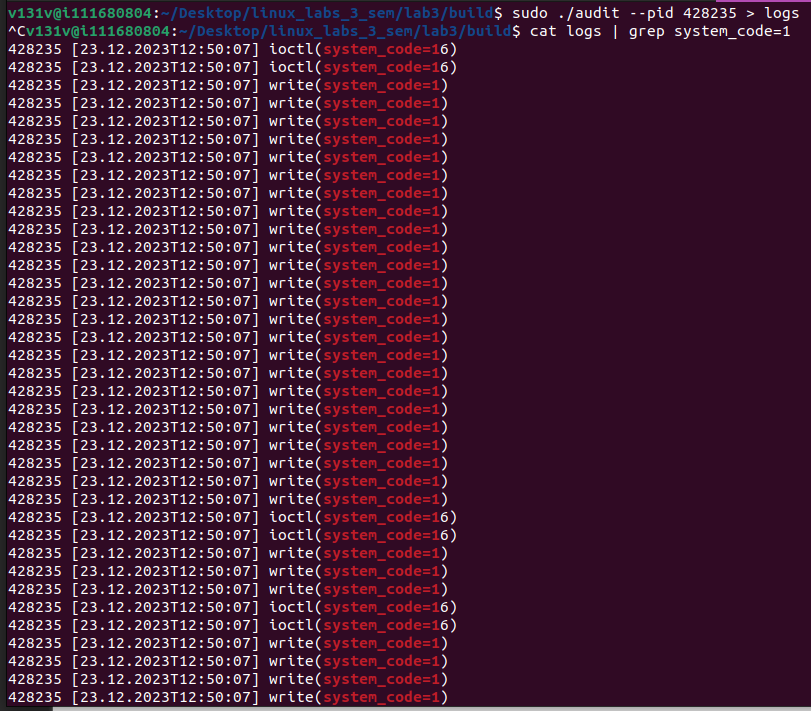
# **Рекомендации программисту**

* Компилятор c++, поддерживающий 11 стандарт
* ОС linux
* Могут потребоваться права sudo

# **Контрольный пример**

В данном разделе представлены контрольный пример, демонстрирующий способность выполнять работу.

*Рисунок 7.1. «audit»*

**

# **Заключение**

В ходе выполнения лабораторной работы был реализован механизм отслеживания системных вызовов процесса на уровне системного вызова с использованием библиотеки ptrace. Программа успешно прикрепляется к целевому процессу, следит за его системными вызовами и логгирует информацию о них. Использование сигнала SIGINT (Ctrl+C) обеспечивает корректное завершение программы и детач процесса.

Код доступен по ссылке:

[https://github.com/v131v/linux\_labs\_3\_sem/tree/main/lab](https://github.com/v131v/linux_labs_3_sem/tree/main/lab2)3

# **9. Литература**

* https://habr.com/ru/articles/111266/
* [https://ru.wikipedia.org/wiki/Сигнал\_(Unix)](https://github.com/jtilly/inih/blob/master/INIReader.h)