МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ОТЧЕТ

по лабораторной работе

по дисциплине «Системное программирование в Linux»

на тему «Разработка системного инструмента для аудита системы»

Студент гр. 22.Б15-пу _____ Осовский Н.С

Преподаватель _____ Киямов Ж.У.

Санкт-Петербург
2023 г.

Оглавление

1. Цель работы

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

2. Задача

Изучить работу ptrace и написать программу отслеживающую работу процесса по заданному PID.

3. Теоретическая часть

"Аудит системы" - это процесс регистрации и анализа событий, происходящих в компьютерной системе, с целью обеспечения безопасности, контроля и соответствия нормативным требованиям. Аудит системы включает в себя мониторинг действий пользователей, обновлений конфигурации системы, запуска и завершения процессов, изменений файлов и других значимых операций. Результаты аудита системы могут использоваться для выявления потенциальных угроз безопасности, а также для улучшения производительности и управления ресурсами системы.

Библиотека "ptrace" - это механизм в ядре Linux, который предоставляет возможность отслеживать и изменять выполнение процессов в системе. Она позволяет отладчику или другой программе контролировать выполнение другого процесса, регистрировать системные вызовы, изменять состояние процесса и многое другое. Библиотека "ptrace" часто используется в различных инструментах для отладки, профилирования и мониторинга системы, а также для реализации механизмов безопасности, таких как аудит системы.

4. Описание программы

1. Проверка количества переданных аргументов командной строки. Если количество аргументов не равно 2, выводится сообщение об ошибке "write pid" и происходит завершение программы с возвратом значения 1.

2. Вызов функции logEvent("start ptrace", pid), который фиксирует начало отслеживания процесса с указанным идентификатором.

3. Вызов функции ptrace(PTRACE_ATTACH, pid, nullptr, nullptr) для присоединения к указанному процессу. Если операция присоединения не удалась, выводится сообщение об ошибке "Failed to attach to the process" и происходит завершение программы с возвратом значения 1.

4. Ожидание завершения указанного процесса с помощью функции waitpid(pid, &status, 0), где pid - идентификатор процесса, status - указатель на

целочисленную переменную, в которой будет сохранен статус завершения процесса.

5. Получение значений регистров процесса с помощью функции ptrace(PTRACE_GETREGS, pid, nullptr, &regs), где regs - структура user_regs_struct, содержащая значения регистров процессора. Полученные значения используются для определения системного вызова, который выполняется в процессе.

6. Запись события, соответствующего системному вызову, в журнал событий с использованием функции logEvent(get_syscall_name(regs.orig_rax), pid), где get_syscall_name() пытается определить имя системного вызова на основе его номера.

7. Отслеживание следующего системного вызова в процессе с помощью функции ptrace(PTRACE_SYSCALL, pid, nullptr, nullptr). Если операция отслеживания следующего системного вызова не удалась, выводится сообщение об ошибке "Failed to trace next system call".

8. Повторение шагов 6-8 до тех пор, пока не произойдет завершение отслеживаемого процесса.

9. Отсоединение от отслеживаемого процесса с использованием функции ptrace(PTRACE_DETACH, pid, nullptr, nullptr). В случае провала операции отсоединения, выводится сообщение об ошибке "Failed to detach from the process" и происходит завершение программы с возвратом значения

4.1. Описание функций

В программе используются 5 функций

Таблица 5.1. Описание функций

Имя функции	Описание функции
logEvent	Описание: Функция используется для сохранения событий в журнале событий. - Аргументы: - event - строка, описывающая событие, которое необходимо зарегистрировать в журнале. - pid - целочисленное значение, представляющее идентификатор процесса, с которым связано событие. - Действия: Функция открывает файл "eventlog.txt" в режиме добавления данных, получает текущее время, форматирует его и записывает вместе с идентификатором процесса и описанием события в указанный журнал. После этого файл закрывается.

longtostr	- Описание: Функция преобразует переданное целочисленное значение типа unsigned long long в строку. - Аргументы: - `value` - целочисленное значение типа unsigned long long, которое необходимо преобразовать в строку. - Действия: Функция использует стандартную функцию `snprintf` для преобразования целочисленного значения в строку, которая затем возвращается из функции.
getsyscallname	Описание: Функция возвращает имя системного вызова на основе его номера. - Аргументы: - `rax` - целочисленное значение типа unsigned long long, представляющее номер системного вызова. - Действия: Функция проверяет переданное значение `rax`. Если оно находится в диапазоне от 0 до 332 включительно, оно интерпретируется как индекс в массиве `table`, представляющем соответствие между номером системного вызова и его именем. Если переданное значение `rax` находится вне этого диапазона, оно преобразуется в строку с помощью функции `longtostr` и возвращается.
main	- Описание: Это основная функция программы. - Аргументы: - `argc` - целочисленное значение, представляющее количество аргументов командной строки. - `argv` - массив строк, представляющих аргументы командной строки. - Действия: Функция начинается с проверки количества аргументов командной строки. Если количество аргументов не равно 2, выводится сообщение об ошибке и программа завершается с возвратом значения 1. Далее идет преобразование второго аргумента командной строки (идентификатор процесса) из строки в целое число. Происходит запись события "start ptrace" в журнале событий. Затем программа присоединяется к указанному процессу с помощью функции ptrace(PTRACEATTACH, pid, nullptr, nullptr). После чего она выполняет отслеживание системных вызовов в процессе, регистрируя их в

	журнале. По завершении отслеживания процесса он отсоединяется соответствующим образом.
ptrace	Функция ptrace используется для контроля и отслеживания процессов. Она позволяет контролировать выполнение процессов, читать и изменять их память и регистры, а также отслеживать системные вызовы и сигналы, связанные с процессом. В данной программе она используется для присоединения к процессу, получения/установки значений регистров процесса и отслеживания системных вызовов.

5. Контрольный пример

Пример логов при чтении команды ping google.com

Sat Dec 23 13:50:36 2023 : 415013 : newfstatat

Sat Dec 23 13:50:36 2023 : 415013 : newfstatat

Sat Dec 23 13:50:36 2023 : 415013 : openat

Sat Dec 23 13:50:36 2023 : 415013 : openat

Sat Dec 23 13:50:36 2023 : 415013 : newfstatat

Sat Dec 23 13:50:36 2023 : 415013 : newfstatat

Sat Dec 23 13:50:36 2023 : 415013 : lseek

Sat Dec 23 13:50:36 2023 : 415013 : lseek

Sat Dec 23 13:50:36 2023 : 415013 : read

Sat Dec 23 13:50:36 2023 : 415013 : read

Sat Dec 23 13:50:36 2023 : 415013 : read

Sat Dec 23 13:50:36 2023 : 415013 : read

Sat Dec 23 13:50:36 2023 : 415013 : close

Sat Dec 23 13:50:36 2023 : 415013 : close

Sat Dec 23 13:50:36 2023 : 415013 : socket

Sat Dec 23 13:50:36 2023 : 415013 : socket

Sat Dec 23 13:50:36 2023 : 415013 : setsockopt

Sat Dec 23 13:50:36 2023 : 415013 : setsockopt

Sat Dec 23 13:50:36 2023 : 415013 : connect

Sat Dec 23 13:50:36 2023 : 415013 : connect

6. Рекомендации пользователя

Для отслеживания узнайте что такое PID процесса в Linux.

7. Рекомендации программиста

Для запуска программы необходим C++ 20. Запускать только на Linux.

Библиотеки: string, fstream, iostream, ctime, vector, iomanip, sys/ptrace, sys/types, sys/wait, unistd, sys/user

8. Вывод

В данной лабораторной работе была разработана программа на языке C++, использующая библиотеку ptrace для мониторинга системных вызовов указанного процесса. Программа отслеживает события, связанные с системными вызовами, выполняемыми указанным процессом, и регистрирует их в журнале событий.

Программа выполняет отслеживание процесса, регистрируя вызовы, события и их временные метки в файле event_log.txt. Она представляет собой мощный инструмент для анализа работы программ и отладки системного взаимодействия.

Таким образом, выполнение данной лабораторной работы позволило освоить работу с библиотекой ptrace и применить ее для отслеживания системных вызовов процесса, что является важным аспектом в разработке инструментов системного мониторинга и отладки программного обеспечения.

Приложение

https://github.com/naelxd/linuxspbu

Дополнительные материалы

https://man7.org/linux/man-pages/man2/ptrace.2.html