**СОДЕРЖАНИЕ**

[Введение 5](#_Toc194926764)

[1 Обзор литературы 6](#_Toc194926765)

[1.1 Утилита обнаружения и тестирования функций USB-устройства как проект 6](#_Toc194926766)

[1.2 Анализ существующих аналогов 6](#_Toc194926767)

[1.3 Постановка задачи 7](#_Toc194926768)

[2 Системное проектирование 7](#_Toc194926769)

[Требования 9](#_Toc194926770)

Введение

В современном мире USB-устройства играют ключевую роль в работе компьютеров и других электронных систем, обеспечивая подключение периферийных устройств, передачу данных и взаимодействие с различными аппаратными компонентами. Однако процесс обнаружения и диагностики USB-устройств часто требует специализированных инструментов, особенно при работе на низком уровне, где необходимо прямое взаимодействие с системными API и драйверами.

Целью данного курсового проекта является разработка утилиты на языке C++ для обнаружения и тестирования USB-устройств в операционной системе Linux. Программа должна предоставлять пользователю детальную информацию о подключённых устройствах, включая идентификаторы производителя, модель, скорость передачи данных и другие параметры, а также выполнять базовые тесты на работоспособность, такие как проверка скорости чтения, записи и целостности передаваемой информации.

Актуальность проекта заключается в необходимости создания удобного и эффективного инструмента для диагностики USB-устройств, который может быть полезен как для системных администраторов, так и для обычных пользователей. В процессе разработки будут использованы низкоуровневые библиотеки, такие как libusb, а также системные API операционных систем для обеспечения максимальной точности и контроля над процессами взаимодействия с устройствами.

Основные функции утилиты включают:

— Перечисление всех подключённых USB-устройств с выводом технических характеристик;

— Тестирование скорости передачи данных между устройством и компьютером;

— Проверку целостности данных при чтении и записи;

— Сохранение результатов тестирования в лог-файл для дальнейшего анализа.

Проект имеет практическую ценность, так как подобные инструменты востребованы при диагностике неисправностей, а также при оценке производительности USB-устройств. В перспективе программа может быть расширена за счёт добавления функций автоматического мониторинга и интеграции с другими системными утилитами.

1 Обзор литературы

1.1 Утилита обнаружения и тестирования функций USB-устройства как проект

Разработка утилиты для обнаружения и тестирования USB-устройств представляет собой важный практический проект в области системного программирования. Такой инструмент позволяет не только получать базовую информацию о подключённых устройствах, но и проводить комплексную диагностику их работоспособности. В процессе создания утилиты особое внимание уделяется взаимодействию с низкоуровневыми API операционных систем и библиотеками, такими как libusb, что обеспечивает прямой доступ к функционалу USB-контроллеров и подключённых устройств.

Актуальность проекта обусловлена необходимостью удобного и универсального инструмента для диагностики USB-устройств, который мог бы использоваться как в повседневной работе, так и при решении специализированных задач. Создаваемая утилита на языке C++ сможет не только отображать технические характеристики устройств, но и тестировать их производительность, что делает её полезной для системных администраторов, разработчиков и обычных пользователей. Реализация подобного проекта позволяет глубже понять принципы работы USB-стека в современных операционных системах и получить ценный опыт в области системного программирования.

1.2 Анализ существующих аналогов

Прежде чем приступить к разработке собственного решения, необходимо изучить существующие аналогичные утилиты для работы с USB-устройствами. Анализ функциональных возможностей, преимуществ и недостатков подобных инструментов позволит определить оптимальный подход к реализации проекта, а также выделить уникальные особенности разрабатываемой программы.

1.2.1 USBView

USBView – стандартная утилита, входящая в состав Windows SDK, которая предоставляет детальную информацию о подключённых USB-устройствах и их характеристиках. Программа отображает иерархию USB-портов, идентификаторы устройств (Vendor ID, Product ID), а также поддерживаемые скорости передачи данных. Однако USBView имеет ограниченный функционал в части тестирования производительности и не предоставляет инструментов для проверки стабильности работы устройств.

1.2.2 Isusb

Утилита lsusb, доступная в Linux, позволяет выводить список подключённых USB-устройств с их основными параметрами. Она поддерживает фильтрацию по классам устройств и вывод дополнительных дескрипторов. Несмотря на свою простоту и удобство использования в терминале, lsusb не предоставляет возможностей для тестирования скорости передачи данных или диагностики ошибок, что ограничивает её применение в задачах глубокого анализа.

1.2.3 USBDeview

USBDeview — программа от NirSoft, которая сочетает в себе функции мониторинга и управления USB-устройствами. Она позволяет не только просматривать подключённые устройства, но и отключать их, сохранять историю подключений, а также экспортировать данные в различные форматы. Однако, как и предыдущие аналоги, USBDeview не включает инструментов для тестирования производительности и проверки целостности данных, что делает её менее пригодной для задач комплексной диагностики.

1.3 Постановка задачи

Разработка утилиты для обнаружения и тестирования USB-устройств на платформе Linux ставит ряд технических задач, требующих решения средствами языка C++ и низкоуровневых Linux-интерфейсов. Основная цель проекта — создать консольное приложение, способное не только идентифицировать подключенные USB-устройства через системные механизмы ядра Linux, но и проводить комплексную проверку их функциональности и производительности.

Ключевыми задачами являются: реализация парсинга данных из подсистемы udev для получения полной информации об устройствах; разработка модуля тестирования с использованием библиотеки libusb для проверки скорости передачи данных и целостности соединения; создание системы логирования результатов в машиночитаемом формате. Особое внимание уделяется оптимизации работы с raw-устройствами через интерфейсы HID и массовой памяти, а также обработке возможных ошибок ввода-вывода.

Проект разрабатывается исключительно для Linux с использованием специфичных для этой ОС технологий: udev для мониторинга устройств и др. Результатом станет специализированный инструмент, превосходящий по функциональности стандартные утилиты в части тестирования производительности USB-устройств под Linux.

2 Системное проектирование

Системное проектирование консольной утилиты для обнаружения и тестирования USB-устройств направлено на создание эффективного инструмента для диагностики USB-оборудования в Linux-системах. Основной акцент сделан на реализации функций обнаружения устройств, тестирования их работоспособности и логгировании. Ниже представлено описание основных модулей программы.

**2.1 Интерфейс командной строки (CLI)**

Модуль интерфейса командной строки отвечает за взаимодействие с пользователем. Он принимает команды и параметры запуска, анализирует их и передает соответствующим модулям для выполнения. Также выводит результаты тестирования в удобочитаемом формате.

**2.2 Модуль обнаружения устройств**

Данный модуль отвечает за сканирование и идентификацию подключенных USB-устройств. Он использует системные интерфейсы Linux (udev, sysfs) для получения полной информации об устройствах, включая идентификаторы производителя, класс устройства, скорость подключения и другие характеристики.

**2.3 Модуль тестирования**

Основной функциональный модуль, который выполняет:

— Тестирование скорости передачи данных;

— Проверку целостности соединения;

Для работы использует низкоуровневый доступ через библиотеку libusb.

**2.4 Модуль логгирования**

Реализует систему журналирования с единственным лог-файлом. Основные функции:

— Запись событий в лог-файл, представленный на фото;

— Отслеживание времени событий и их категоризация;

— Форматированный вывод информации для удобства анализа.

**2.5 Модуль конфигурации**

Обеспечивает работу с настройками программы, включая:

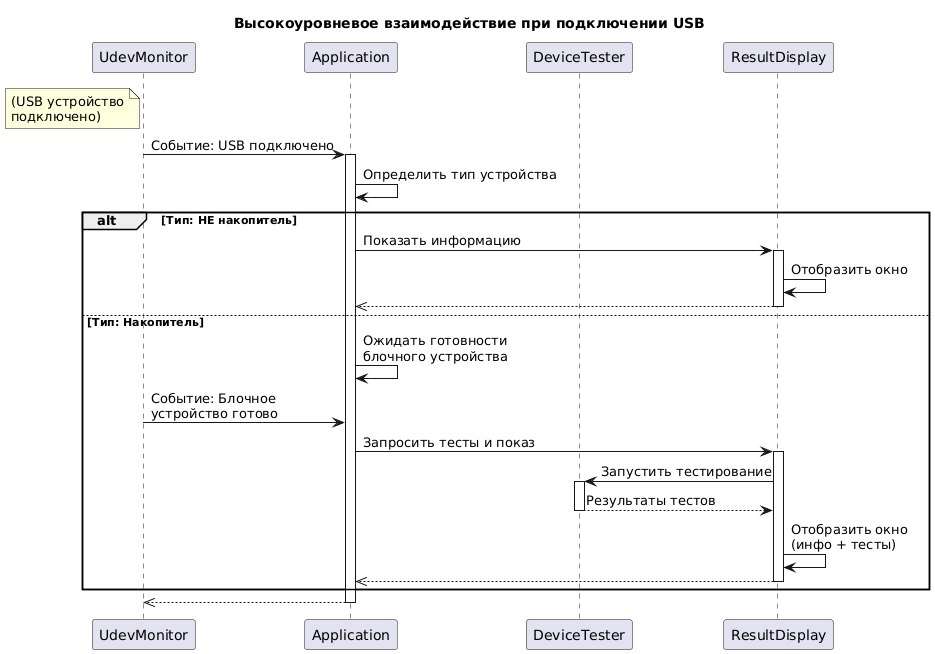
— Парсинг аргументов командной строки;

— Управление параметрами тестирования;

— Валидацию введенных параметров.

В заключение, представленная архитектура обеспечивает создание надежного инструмента для диагностики USB-устройств в Linux. Каждый модуль выполняет строго определенные функции и взаимодействует с другими компонентами системы через четко определенные интерфейсы.

Структурная схема проекта представлена в приложении А.



Требования

A black text on a white background

AI-generated content may be incorrect.

По поводу КП. 2-ой процентовки. Думаю можно провести 7 апреля, когда будет общее занятие. После общего занятия тогда можно провести процентовку.

Надо сделать около 30% КП. Если брать за основу содержание ПЗ, которое на картинке, то я буду засчитывать за 30% процентов: введение, обзор литературы, системное проектирование (структурную схему к нему). Системное проектирование это описание ваших основных модулей (блоков), которые отображены на структурной схеме.

Плюс показать код программы, на каком этапе, хотя бы придумать какие сущности у вас будут (классы), описание базовых методов.