Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина: Операционные среды и системное программирование

К защите допустить:
И.О. Заведующего кафедрой
информатики
_____С. И. Сиротко

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему

«МОНИТОР ОКОН»: СПИСОК ОКОН И УПРАВЛЕНИЕ ИМИ (В Т.Ч. С ФИЛЬТРАЦИЕЙ/ОТБОРОМ)

БГУИР КП 1-40 04 01 011 ПЗ

Студент Е. А. Киселёва

Руководитель Н. Ю. Гриценко

Минск 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В современном мире информационных технологий пользователи сталкиваются с растущей потребностью в эффективном управлении рабочим пространством на своих компьютерных устройствах. Один из ключевых аспектов этого управления связан с организацией и мониторингом оконных приложений. Монитор окон представляет собой программный инструмент, позволяющий пользователям управлять списком открытых окон и производить различные операции с ними, такие как фильтрация и отбор.

Можно выделить следующие причины актуальности курсовой работы:

1 Рост использования компьютерных устройств: с каждым годом количество компьютеров и мобильных устройств, используемых пользователями, продолжает расти. Следовательно, эффективное управление рабочим пространством на этих устройствах становится все более важным.

2 Увеличение числа оконных приложений: с развитием информационных технологий и программного обеспечения растет и количество установленных и одновременно открытых оконных приложений. Удобное управление ими становится необходимостью для повышения производительности и комфорта работы.

3 Необходимость в эффективной организации рабочего пространства: организация рабочего пространства является важным аспектом повседневной работы с компьютером. Монитор окон предоставляет пользователю инструменты для удобного распределения и переключения между приложениями, что способствует повышению эффективности труда.

4 Требования к персонализации и гибкости: современные пользователи ожидают от программного обеспечения возможности персонализации и гибкой настройки под свои потребности. Монитор окон, способный предоставить различные опции фильтрации и отбора оконных приложений, отвечает этим требованиям.

5 Развитие технологий пользовательского интерфейса: с развитием технологий интерфейсов пользователя становятся доступными новые методы взаимодействия с программным обеспечением. Исследование монитора окон в контексте современных тенденций в дизайне интерфейсов позволяет определить оптимальные подходы к его разработке и улучшению.

Цель данной курсовой работы состоит в изучении принципов работы монитора окон, его основных функций, анализе методов фильтрации и отбора оконных приложений, а также реализации подобного приложения. Помимо этого, будут рассмотрены современные подходы к организации

пользовательского интерфейса монитора окон с учетом требований к удобству использования и эффективности.

Для достижения цели курсовой работы, которая заключается в изучении принципов работы монитора окон, его основных функций, а также в реализации монитора окон с учетом современных требований, были поставлены следующие задачи:

1 Изучение существующих подходов к организации монитора окон: анализ существующих программных решений и исследование их основных принципов работы, функциональности и интерфейсов.

2 Анализ требований к монитору окон: определение основных потребностей пользователей в управлении рабочим пространством, выявление требований к функциональности и удобству использования монитора окон.

3 Исследование методов фильтрации и отбора оконных приложений: оценка различных подходов к фильтрации и отбору оконных приложений, а также их применимости в контексте разработки монитора окон. В ходе исследования будут проанализированы основные методы фильтрации и отбора оконных приложений, такие как поиск по названию, типу, активности и другим характеристикам.

4 Проектирование пользовательского интерфейса монитора окон: разработка концепции интерфейса пользователя с учетом современных требований к дизайну интерфейсов и оптимальной организации рабочего пространства.

5 Реализация программного прототипа монитора окон: создание программного прототипа монитора окон на основе выбранной концепции и проектирования пользовательского интерфейса с использованием современных технологий разработки.

6 Тестирование и анализ: проведение тестирования разработанного прототипа монитора окон с целью выявления ошибок, оценки его производительности и удобства использования. Анализ результатов тестирования и выявление возможных улучшений.

В итоге, данная курсовая работа нацелена на всестороннее изучение и разработку монитора окон с учетом современных требований к управлению рабочим пространством на компьютере. Посредством анализа существующих подходов, методов фильтрации и отбора оконных приложений, а также проектирования и реализации программного прототипа, предполагается создание эффективного инструмента, способного оптимизировать рабочий процесс пользователя и повысить его производительность.

1 МОНИТОР ОКОН В WINDOWS

В Windows монитор окон, или «Window Monitor», обычно относится к программному инструменту, который следит за активностью окон на компьютере. Он может быть полезен для различных целей, таких как отслеживание активности приложений, управление окнами, для автоматизации определенных действий. [1]

Функции, которые могут предоставлять мониторы окон:

1 Отслеживание активности приложений: монитор окон позволяет видеть, какие приложения открыты и активны в данный момент. Этот функционал полезен для отслеживания использования ресурсов или управления процессами.

2 Управление окнами: некоторые мониторы окон могут предоставлять возможность управления окнами, например, изменение их размеров, перемещение или закрытие.

3 Определение активности пользователя: мониторы окон могут отслеживать, какие окна активны или находятся в фокусе в данный момент, что может быть полезно для анализа активности пользователя или для автоматизации определенных задач, например, запуска определенной программы при открытии определенного окна.

4 Мониторинг использования ресурсов: монитор окон также может предоставлять информацию о том, какие приложения используют больше ресурсов, а именно процессорного времени или памяти, что может быть полезным для оптимизации производительности.

5 Отслеживание изменений состояния окон: некоторые мониторы окон могут отслеживать изменения в состоянии окон, а именно открытие окон, закрытие или изменение заголовка. Этот функционал может быть полезен для автоматического выполнения определенных действий на основе отслеженных изменений.

Мониторы окон предоставляют графический интерфейс пользователя для визуализации и управления окнами. Они могут быть как частью операционной системы Windows, так и сторонними приложениями, доступными для скачивания и установки. Если говорить о встроенных в Windows мониторах окон, то можно выделить два подобных по своей сути инструмента: переключатель между запущенными приложениями и инструмент, отображающий все рабочие столы и приложения, запущенные на них.

1.1 Переключение между запущенными приложениями

Переключение между открытыми приложениями с помощью комбинации клавиш Alt + Tab является одной из самых удобных и быстрых функций в операционной системе Windows. Этот метод позволяет пользователям мгновенно перемещаться между различными приложениями без необходимости использования мыши или поиска нужного окна на панели задач.

Когда пользователь нажимает клавишу Alt и затем клавишу Tab, Windows отображает список всех запущенных приложений в виде миниатюрных изображений, представляющих окна каждого приложения. Этот список появляется на переднем плане и позволяет пользователям быстро просматривать доступные приложения.

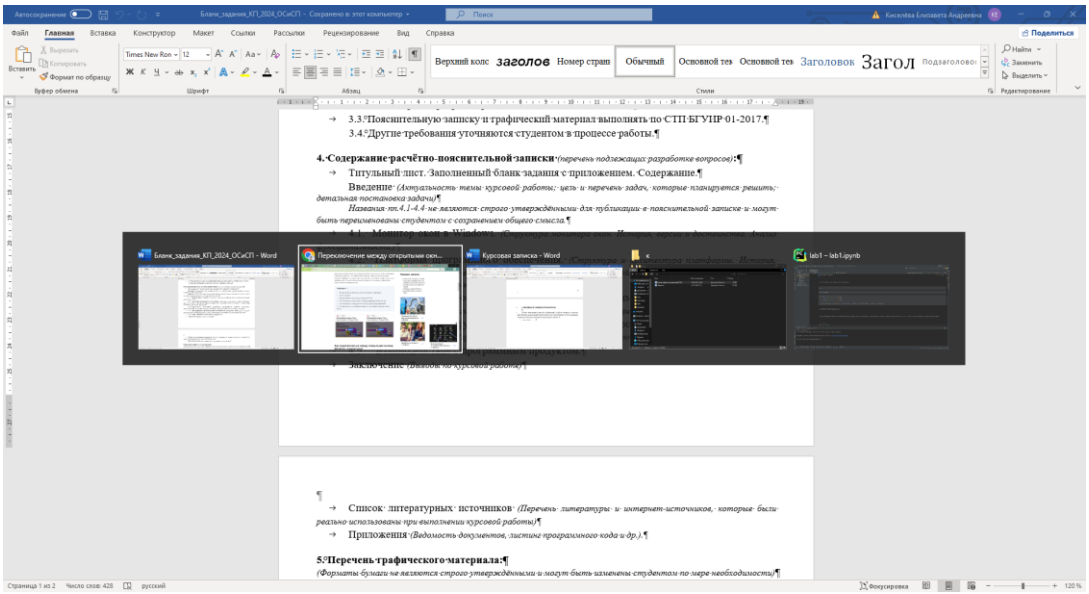

Рисунок 1.1 – Переключение между открытыми приложениями

Одна из главных преимуществ этого метода заключается в его простоте и эффективности. Пользователи могут быстро переключаться между приложениями, просто нажимая клавишу Tab несколько раз, чтобы выделить нужное приложение, а затем отпуская клавиши Alt и Tab для активации выбранного окна.

Эта функция особенно полезна, когда у пользователя открыто много приложений или когда требуется переключение между несколькими задачами. Она также помогает сэкономить время, улучшая общую производительность работы за счет исключения необходимости постоянного перемещения курсора мыши и щелчков по окнам.

Кроме того, в некоторых версиях Windows функция Alt + Tab дополнена предварительным просмотром окон, который позволяет пользователю более точно определить, к какому приложению он переключается, что дополнительно повышает удобство и навигационную эффективность этого метода.

Таким образом, комбинация клавиш Alt + Tab в операционной системе Windows представляет собой удобный инструмент для быстрого переключения между окнами. Этот метод эффективно управляет ресурсами и обеспечивает моментальный доступ к различным задачам пользователя без заметных перерывов в выполнении процесса.

Использование комбинации клавиш Alt + Tab позволяет оперативно переключаться между контекстами исполнения, представленными открытыми приложениями, без необходимости сохранения и восстановления состояния приложений. Это уменьшает временные затраты на переключение контекста и снижает вероятность допущения ошибок в работе пользователя.

Благодаря интуитивной натуре использования и высокой скорости реакции, комбинация клавиш Alt + Tab является ключевым моментом в пользовательском опыте операционной системы Windows, демонстрирует важность эффективного управления многозадачностью для повышения производительности и удовлетворенности пользователей. [2]

1.2 История, версии и достоинства переключения между открытыми приложениями

Функционал, который открывается при нажатии комбинации клавиш Alt + Tab в операционной системе Windows, известен как «переключение приложений» или «переключение окон». Этот инструмент был введен еще в ранних версиях Windows и продолжает эволюционировать в современных версиях.

История переключения окон в Windows:

1 Переключение окон было впервые введено в Windows 1.0 в 1985 году, и с тех пор стало одной из основных функций операционной системы. Изначально оно позволяло пользователям переключаться между запущенными приложениями, отображая миниатюрные изображения окон их интерфейсов.

2 С течением времени функционал был усовершенствован. В Windows 95 появилась возможность переключения между окнами приложений с помощью комбинации клавиш Alt + Tab, что значительно улучшило пользовательский опыт.

3 В последующих версиях Windows этот функционал был доработан и оптимизирован, чтобы обеспечить более быстрое и эффективное переключение между приложениями, в том числе добавлена поддержка переключения между окнами одного и того же приложения.

Версии переключения окон в Windows:

1 Windows XP: в Windows XP был значительно улучшен интерфейс переключения приложений. Была добавлена поддержка переключения между окнами одного приложения с помощью комбинации клавиш Ctrl + Tab. Это облегчило работу с множеством открытых окон в одном приложении, например, в браузере.

2 Windows Vista/7: в этих версиях Windows были внесены дополнительные улучшения в интерфейс переключения приложений, включая более плавную анимацию и возможность предварительного просмотра окон приложений, а также поддержка Aero Flip.

3 Windows 8/10: в Windows 8 и Windows 10 были добавлены новые функции, такие как поддержка множества виртуальных рабочих столов (Desktops), а также возможность группировки окон в приложении на одной панели в переключателе приложений.

4 Windows 11: в Windows 11 были внесены некоторые изменения в дизайн и интерфейс переключения приложений, чтобы сделать его более современным и удобным для использования на современных устройствах с сенсорными экранами.

Основные достоинства современного функционала переключения приложений в Windows:

1 Простота использования: комбинация клавиш Alt + Tab обеспечивает быстрый доступ к переключателю приложений, что делает работу с несколькими приложениями более удобной и эффективной.

2 Многооконная поддержка: позволяет легко переключаться между различными открытыми окнами и приложениями, что особенно полезно при работе с многозадачными задачами.

3 Быстродействие: функционал переключения приложений обычно хорошо оптимизирован, что обеспечивает плавное и быстрое переключение между окнами даже при большом количестве открытых приложений.

4 Улучшенный мультитаскинг: позволяет пользователям эффективно управлять своими задачами и оперативно переключаться между ними без необходимости использования мыши.

В целом, функционал переключения приложений в Windows является важной частью пользовательского опыта, которая обеспечивает удобство и эффективность работы с множеством приложений и окон. [3]

1.3 Отображение всех рабочих столов и приложений на них

Комбинация клавиш Win + Tab в Windows открывает функцию под названием «Task View» или «Просмотр задач». Это удобный инструмент для управления открытыми приложениями и рабочими столами, обеспечивающий более эффективную навигацию и организацию рабочего пространства.

При нажатии Win + Tab экран переключается на Task View, который отображает миниатюры всех открытых приложений и рабочих столов. Каждая миниатюра представляет собой окно открытого приложения или рабочего стола, что позволяет легко и быстро переключаться между ними.

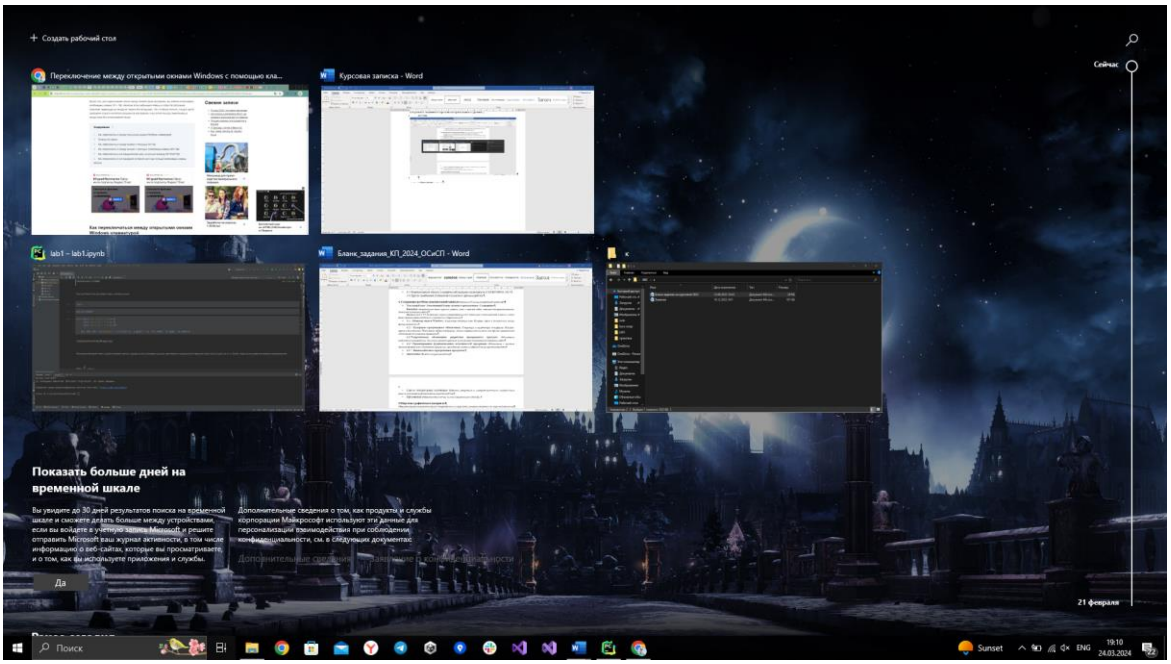

Рисунок 1.2 – Просмотр задач

В Task View можно использовать клавиши стрелок вверх, вниз, влево и вправо, чтобы выбрать нужное приложение или рабочий стол. Также для переключения между приложениями можно использовать мышь. Такие способы навигации позволяют быстро переходить между различными задачами без необходимости искать нужное окно среди множества открытых.

Кроме того, Task View также облегчает организацию рабочего пространства, позволяя создавать новые рабочие столы и переключаться между несколькими рабочими столами. Это особенно полезно, когда имеется большое количество открытых приложений или когда есть необходимость в разделении задач на разные рабочие области для повышения производительности.

После завершения работы с Task View, можно закрыть его, щелкнув в любом месте экрана за пределами области Task View или нажав клавишу Esc на клавиатуре. Это вернет систему к обычному режиму работы с открытыми приложениями и рабочим столом.

Функция Task View, открываемая с помощью комбинации клавиш Win + Tab, представляет собой удобный инструмент для управления множеством открытых приложений и рабочих столов в операционной системе Windows. Она способствует повышению эффективности переключения между приложениями для пользователей, обеспечивая быструю навигацию и организацию рабочего пространства. Этот инструмент помогает минимизировать временные затраты на переключение между задачами, что в свою очередь может улучшить общую производительность и скорость работы пользователя. [4]

1.4 История, версии и достоинства Task View

Функционал, который открывается при нажатии комбинации клавиш Win + Tab в операционной системе Windows, известен как «Task View». Этот инструмент позволяет пользователям просматривать и переключаться между виртуальными рабочими столами, а также просматривать историю запущенных приложений.

История Task View в Windows не длинна, так как просмотр задач в таком формате является достаточно новой функцией в Windows. Task View был впервые представлен в Windows 10 в 2015 году как часть новых функций для улучшения многозадачности и организации рабочего пространства. Этот инструмент был разработан для того, чтобы облегчить управление множеством открытых приложений и повысить производительность пользователей.

Версии функционала Task View в Windows:

1 Windows 10: в Windows 10 Task View представлял собой способ просмотра всех открытых приложений и окон на текущем рабочем столе, а также возможность создания и переключения между виртуальными рабочими столами. Это позволяло пользователям организовывать свою рабочую область более эффективно и улучшало многозадачность.

2 Windows 10 Creators Update: в этом обновлении Windows 10 были добавлены новые функции Task View, такие как «Timeline» (Хронология), которая позволяет пользователям просматривать историю своих действий на компьютере за определенный период времени. История включает в себя открытые приложения, файлы и веб-сайты, которые пользователь посещал.

3 Windows 10 Fall Creators Update: в этом обновлении были внесены улучшения в интерфейс Task View, включая новый дизайн и более интуитивно понятные элементы управления. Были также добавлены новые возможности, такие как возможность перемещать окна между виртуальными рабочими столами с помощью мыши.

4 Windows 11: в Windows 11 Task View был доработан и интегрирован в новый дизайн операционной системы. В этой версии были внесены изменения в интерфейс и добавлены новые функции, такие как возможность группировки открытых приложений на виртуальных рабочих столах и улучшенная интеграция с функцией Snap Layouts для упорядочивания окон.

Достоинства функционала Task View в Windows:

1 Удобство многозадачности: позволяет легко управлять множеством открытых приложений и окон путем организации их на разных виртуальных рабочих столах.

2 Повышенная производительность: упрощает навигацию между приложениями и рабочими столами, что помогает пользователям быстрее находить нужную информацию и выполнять задачи.

3 Организация рабочего пространства: позволяет пользователям создавать разные рабочие среды для разных задач или проектов, что повышает организованность и эффективность работы.

4 История действий: функция Timeline (Хронология) в Task View позволяет пользователям легко возвращаться к предыдущим действиям и открытым приложениям за определенный период времени, что может быть полезно для восстановления работы или поиска потерянной информации.

В целом, Task View представляет собой полезный инструмент для управления многозадачностью и организации рабочего пространства в операционной системе Windows, который продолжает развиваться и улучшаться с каждым обновлением. [5]

2 ПЛАТФОРМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1 Структура и архитектура платформы

2.1.1 Выбранная платформа

В рамках выполнения курсовой работы была принята решающая роль в выборе определенных компонентов платформы для разработки программного продукта. Операционная система *Windows* 10 была выбрана в качестве основной, так как она обеспечивает широкий спектр совместимости с различными программными и аппаратными решениями. Эта операционная система также предоставляет удобную интеграцию с инструментарием разработки и обеспечивает стабильную среду для проведения экспериментов и тестирования.

Для написания кода и выполнения проекта была выбрана интегрированная среда разработки *Visual Studio*. *Visual Studio* обеспечивает удобный интерфейс, возможности отладки, а также поддержку различных языков программирования. Эта среда разработки обладает широким функционалом и является стандартом в индустрии для разработки программного обеспечения под *Windows*.

Язык программирования C++ выбран для создания монитора окон из-за его высокой производительности, возможности прямого доступа к системным ресурсам через библиотеку WinAPI, а также из-за широких возможностей стандартной библиотеки для работы с различными аспектами приложений.

Таким образом, выбранная платформа, объединяющая *Windows* 10, *Visual Studio* и язык программирования *C++*, обеспечивает оптимальные условия для эффективной разработки и тестирования монитора окон, а именно списка окон с возможностью управления ими.

2.1.2 Операционная система

Операционная система – это программное обеспечение, управляющее аппаратными и программными ресурсами компьютера. Она предоставляет интерфейс для взаимодействия пользователя с компьютером, обеспечивает выполнение прикладных программ и эффективное управление аппаратурой, такой как процессор, память, устройства ввода-вывода и хранения данных. Операционная система выполняет ряд основных функций, таких как загрузка системы, управление процессами, обеспечение безопасности данных, файловая система и другие.

Структурные компоненты операционной системы:

1 Ядро – это «сердце» операционной системы. Именно оно контролирует запуск всех программ и компьютерных компонентов.

2 Командный модуль – обеспечивает выполнение команд, поступающих от пользователя.

3 Набор драйверов – система специальных программ для корректной работы операционной системы.

4 Сервисные утилиты – дополнительные программы для выполнения различных задач.

5 Оболочка – графический интерфейс операционной системы, именно его видит пользователь компьютера.

Операционные системы предназначены не только для работы на персональных компьютерах и ноутбуках, но и для смартфонов, планшетов, смарт-часов, смарт-телевизоров, маршрутизаторов и других электронных устройств. [6]

2.1.3 Интегрированная среда разработки

Интегрированная среда разработки – это программное средство, предоставляющее программисту интегрированный набор инструментов и функций для удобного создания, редактирования, компиляции, отладки и управления программным кодом. *IDE* обычно включает в себя следующие ключевые элементы:

1 Текстовый редактор: для написания и редактирования исходного кода. Обеспечивает подсветку синтаксиса, автодополнение и другие удобства.

2 Компилятор/Интерпретатор: позволяет преобразовывать исходный код в исполняемый файл или промежуточный код.

3 Отладчик: предоставляет средства для пошагового выполнения кода, выявления ошибок и отслеживания значений переменных в процессе выполнения программы.

4 Система управления версиями: инструменты для отслеживания изменений в коде, совместной работы нескольких разработчиков и восстановления предыдущих версий кода.

5 Панели инструментов: сокращают количество действий для выполнения основных операций, таких как компиляция или запуск программы.

6 Менеджер проектов: позволяет организовать файлы и ресурсы проекта, управлять зависимостями и настройками проекта.

7 Интеграция с отладочными и профилирующими инструментами: для более глубокого анализа и оптимизации кода. [7]

2.1.4 Язык программирования

Язык программирования представляет собой формальный инструмент, используемый для написания компьютерных программ и обеспечивающий взаимодействие между человеком и компьютером. Он предоставляет программистам средства для выражения идей и логики работы программы, используя читаемый синтаксис. Эти инструкции, созданные на языке программирования, являются понятными для компьютера и предоставляют ему путь выполнения задач.

Язык программирования выполняет описательную функцию, позволяя программистам выражать свои концепции и логику работы программы так, чтобы было понятно человеку. Он также служит для написания инструкций, которые могут быть выполнены компьютером, предоставляя необходимые средства для создания программного кода.

Основной целью языка программирования является создание абстракции, позволяющей программистам работать на более высоком уровне, избегая прямого взаимодействия с аппаратными ресурсами компьютера. Это уровень абстракции делает возможным более удобное и эффективное программирование.

Кроме того, язык программирования обеспечивает переносимость программ между различными платформами и системами, предоставляя возможность запускать одинаковый код на разных устройствах. [8]

2.2 История, версии и достоинства

2.2.1 Операционная система *Windows* 10

Операционная система *Windows* 10 была представлена *Microsoft* в июле 2015 года. Это было важное обновление для предыдущих версий *Windows*, таких как *Windows* 8 и *Windows* 8.1, и представляло собой попытку объединения функциональности для персональных компьютеров, планшетов и мобильных устройств в одну универсальную платформу.

Windows 10 изначально выпущена как одна общая версия, но *Microsoft* предоставляет регулярные обновления функций два раза в год. Эти обновления включают в себя улучшения безопасности, новые функции и улучшения производительности.

Достоинства операционной системы, выбранной в курсовом проекте:

1 Универсальность. *Windows* 10 предназначена для работы на широком спектре устройств, включая персональные компьютеры, планшеты, ноутбуки и мобильные устройства.

2 Интерфейс. Введение стартового меню возвращает привычную среду для пользователей, привыкших к более ранним версиям *Windows*.

3 Безопасность. *Windows* 10 включает улучшенные средства безопасности, такие как *Windows Defender* и система обновлений для борьбы с вирусами и вредоносным программным обеспечением.

4 Обновления. Регулярные обновления обеспечивают улучшения функциональности, новые возможности и патчи безопасности.

5 Совместимость с приложениями. *Windows* 10 обеспечивает поддержку широкого спектра приложений, включая множество классических приложений *Windows* и современных *Universal Windows Platform* (*UWP*) приложений.

6 Облако и интеграция с онлайн-сервисами. *Windows* 10 включает интеграцию с облачными службами *Microsoft*, такими как *OneDrive*, обеспечивая удобный доступ к данным из любой точки мира.

Windows 10, введенная в 2015 году, представляет собой эффективную операционную систему для курсового проекта, предоставляет универсальность в использовании на различных устройствах, обеспечивает безопасность данных и регулярные обновления для улучшения производительности. Ее совместимость с различными приложениями, включая современные и традиционные, а также интеграция с облачными сервисами, делают *Windows* 10 удовлетворяющей современным требованиям и обеспечивающей эффективность в процессе работы. [9]

2.2.2 Интегрированная среда разработки *Visual Studio*

Интегрированная среда разработки (*IDE*) *Visual Studio* была представлена компанией *Microsoft* в 1997 году и с тех пор прошла долгий путь эволюции и усовершенствования. Она стала одним из основных инструментов для разработки программного обеспечения под платформу *Windows* и другие технологии. *Visual Studio* имеет множество версий, каждая из которых вносит улучшения в среду разработки и предоставляет новые инструменты для программистов. Каждая последующая версия вносила новые функциональности, интегрировала последние технологии и улучшала процесс разработки.

Достоинства выбранной интегрированной среды разработки:

1 Многозадачность. *Visual Studio* обеспечивает разработчиков множеством инструментов и функций, улучшающих процесс создания программного обеспечения, включая редактор кода, отладчик, дизайнер форм, и многие другие.

2 Языковая поддержка. Поддержка различных языков программирования, включая *C++*, *C#*, *Visual Basic*, *F#*, *Python*, и другие, что делает ее универсальной средой разработки.

3 Отладка. Мощные инструменты отладки, в том числе возможность пошагового выполнения кода, анализа переменных и состояния приложения.

4 Графические средства разработки. Интегрированные средства для создания графических интерфейсов приложений, веб-сайтов и других проектов.

5 Тестирование. Инструменты для автоматизации тестирования, включая модульное, функциональное и нагрузочное тестирование.

6 Совместная разработка. Возможности совместной разработки, поддержка систем контроля версий и средства для работы в команде.

7 Расширяемость. Возможность установки и использования различных плагинов и расширений для адаптации среды разработки под конкретные потребности разработчика.

8 Обновления. Регулярные обновления и поддержка последних технологий, позволяющие оставаться в курсе современных требований и тенденций в программировании.

Visual Studio, представленная *Microsoft* в 1997 году, стала неотъемлемым инструментом разработки программного обеспечения под *Windows* и другие технологии. Ее эволюция привела к созданию многочисленных версий, каждая из которых предоставляет улучшения и новые инструменты для разработки программного кода. Основные преимущества выбранной интегрированной среды разработки включают многозадачность, языковую поддержку, мощные средства отладки, графические средства, тестирование, совместную разработку, расширяемость, и регулярные обновления. По вышеперечисленным причинам было принято решение использовать Visual Studio для разработки монитора окон. [10]

2.2.3 Язык программирования *C++*

C++ – это один из наиболее влиятельных и широко используемых языков программирования. Его история, версии и достоинства формировались на протяжении десятилетий.

История языка программирования *C++*:

1 Язык программирования *C* был создан в 1972 году в лабораториях *Bell Telephone* (*Bell Labs*) Деннисом Ритчи. Он был разработан как язык программирования для написания операционной системы *UNIX*. *C* был отмечен своей эффективностью и низкоуровневым характером, что делало его предпочтительным выбором для системного программирования.

2 В 1983 году Бьёрн Страуструп расширил язык *C*, создав *C++*. *C++* добавил объектно-ориентированные возможности, что сделало его более гибким и удобным для разработки сложных программных систем.

Версии выбранного языка программирования:

1 Язык *C* не имеет формальных версий, но стандарты были выпущены. Один из самых важных – *ANSI C* (*C*89), затем был выпущен стандарт *C*99, а затем *C*11.

2 *C++* также прошел через несколько версий. Одной из первых была *C++*98, затем *C++*03. Существенные изменения внесены в стандарт *C++*11, который внедрил множество новых функций и улучшений. Последующие стандарты включают *C++*14, *C++*17 и *C++*20.

Достоинства языка программирования *C++*:

1 Эффективность: *C++* известен своей высокой производительностью и эффективностью выполнения.

2 Близость к аппаратуре: этот язык предоставляет низкоуровневый доступ к памяти и аппаратным ресурсам, что важно для системного программирования.

3 Обширные библиотеки: язык обладает обширными библиотеками, предоставляет разработчикам множество функций и инструментов.

4 Переносимость: код на *C++* легко переносится между различными платформами.

5 Объектно-ориентированное программирование: *C++* добавил объектно-ориентированные концепции, что делает его подходящим для создания сложных и масштабируемых систем.

6 Широкое использование в индустрии: *C++* широко используются в разработке операционных систем, встроенных систем, игр, высокопроизводительных приложений и других областях.

7 Сообщество разработчиков. Язык имеет огромное сообщество разработчиков, что обеспечивает поддержку и ресурсы для программистов.

Использование *C++* является актуальным в различных областях программирования, обеспечивает мощные инструменты для разработки высокопроизводительных приложений, в том числе монитора окон. [11]

2.3 Обоснование выбора платформы

Рассмотрим обоснование выбора *C++*, *Visual Studio* и *Windows* 10, выбранных для выполнения данной курсовой работы.

Язык программирования C++ выбран для создания монитора окон из-за его высокой производительности, возможности прямого доступа к системным ресурсам через библиотеку WinAPI, а также из-за широких возможностей стандартной библиотеки для работы с различными аспектами приложений.

Среда разработки Visual Studio была выбрана из-за и широкого спектра инструментов, предоставляемых для создания приложений на языке C++ под операционную систему Windows. Среда разработки обладает мощными функциями, такими как автодополнение кода, отладка, анализ кода и поддержка интеграции с системами контроля версий. Все эти преимущества обеспечивают эффективную разработку монитора окон.

Windows 10 выбрана для разработки монитора окон в силу доступа к современным API и набору инструментов, предоставляемых операционной системой для создания приложений, работающих с окнами и процессами. Операционная система обеспечивает широкие возможности для взаимодействия с графическим пользовательским интерфейсом, включая функции для отслеживания активности окон, мониторинга процессов и управления рабочим пространством. Кроме того, высокий уровень распространения Windows 10 среди пользователей предоставляет потенциал для широкого применения созданного приложения.

Можно сказать, что *C++* обеспечивает эффективность и контроль над ресурсами, *Visual Studio* предоставляет удобную среду разработки, а *Windows* 10 обеспечивает совместимость с большинством вычислительных систем, что соответствует требованиям для разработки монитора окон, предоставляющего список окон и управление ими, в том числе с фильтрацией и отбором.

2.4 Анализ программного обеспечения для написания программы

В контексте разработки курсового проекта на тему монитора окон, представляющего собой список окон с управлением ими, выбор операционной системы, среды разработки и языка программирования играют важную роль. Рассмотрим операционную систему *Windows* 10, среду разработки *Visual Studio* и язык программирования *C++*, выделяя их характеристики и вклад в обеспечение успешной реализации проекта.

Операционная система *Windows* 10 выделяется своей высокой совместимостью с современным программным обеспечением, предоставляя широкий доступ к инновационным технологиям. Её интуитивный пользовательский интерфейс и гибкие возможности настройки облегчают рабочий процесс, а поддержка *DirectX* становится ключевым фактором для разработки графически интенсивных приложений.

Особое внимание следует уделить интеграции *Windows* 10 с средой разработки *Visual Studio*. Это обеспечивает не только плавное взаимодействие, но и оптимизацию производительности, предоставляя разработчикам удобные инструменты для создания программного обеспечения под платформу *Windows*. Тесная интеграция со средой разработки от *Microsoft* обеспечивает эффективное использование возможностей операционной системы, сокращая временные затраты на разработку и повышая качество конечного продукта.

Преимущества среды разработки *Visual Studio* проявляются в многозадачности, предоставляя разработчикам широкий инструментарий, включающий редактор кода, отладчик и дизайнер форм. Это упрощает процесс создания программного обеспечения. Поддержка различных языков программирования, включая *C++*, делает *Visual Studio* универсальным инструментом для разработчиков. Мощные инструменты отладки, такие как пошаговое выполнение кода и анализ переменных, обеспечивают эффективную отладку приложений. Интегрированные графические средства упрощают создание графических интерфейсов приложений.

Совместимость и расширяемость *Visual Studio* обеспечивают удобные возможности совместной разработки, поддержку систем контроля версий и средства для командной работы. Возможность установки различных плагинов и расширений делает среду разработки гибкой и адаптируемой к индивидуальным потребностям разработчика.

Использование языка программирования *C++* придает проекту высокую производительность, что особенно ценно для системного программирования и разработки высокопроизводительных приложений. Кроме того, прямой доступ к системным ресурсам, предоставляемый *C++*, важен для создания приложений, которым нужна высокая производительность из-за частого использования конечным пользователем.

Интеграция Windows 10 с Visual Studio, обеспечивающая оптимальную совместимость, и выбор C++ позволяет создавать высокопроизводительное программное обеспечение с прямым доступом к системным ресурсам. Совместимость, расширяемость и регулярные обновления делают этот стек технологий привлекательным выбором при работе в среде Windows.

3 ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ ПРОГРАММНОГО ПРОДУКТА

3.1 Обоснование необходимости разработки

Разработка монитора окон представляет собой важную задачу, особенно в современном мире, где многозадачность и эффективное управление рабочим пространством становятся все более важными для повышения производительности и удобства пользователей. Обоснование необходимости такого инструмента можно представить следующим образом:

1 Управление рабочим пространством: с ростом количества приложений и задач на компьютере становится сложно эффективно управлять открытыми окнами. Монитор окон позволяет быстро переключаться между приложениями, организовывать их в удобном для пользователя виде и экономить время на поиск нужного окна.

2 Многозадачность: современные пользователи часто выполняют несколько задач одновременно. Монитор окон дает возможность отслеживать все открытые приложения и быстро переключаться между ними без необходимости минимизировать или закрывать окна.

3 Оптимизация рабочего процесса: правильное управление окнами способствует оптимизации рабочего процесса. Пользователь может организовать окна по своему усмотрению, что повышает комфорт работы и позволяет концентрироваться на существенных задачах.

4 Повышение производительности: эффективное использование монитора окон сокращает время, затрачиваемое на поиск и переключение между приложениями. Это способствует повышению производительности работы и снижению утомляемости пользователя.

5 Улучшение пользовательского опыта: использование удобного и интуитивно понятного монитора окон улучшает пользовательский опыт. Пользователи получают доступ к инструментам, которые делают работу с окнами более эффективной и удобной.

6 Адаптация к различным сценариям использования: монитор окон может быть настроен под различные сценарии использования, такие как работа с несколькими мониторами, группировка окон по проектам или задачам, использование горячих клавиш для быстрого доступа к функциям и т. д.

Таким образом, разработка монитора окон оправдана из-за необходимости улучшения управления рабочим пространством, повышения производительности и комфорта работы пользователей, а также создания инструмента, который адаптирован под современные требования и стандарты использования компьютерных систем. [12]

3.2 Технологии программирования, используемые для решения поставленных задач

Для разработки монитора окон в ходе курсового проекта были применены несколько ключевых технологий программирования. Основными инструментами, которые были использованы, являются Visual Studio, язык программирования C++ и библиотека WinAPI. Visual Studio предоставляет всю необходимую среду для разработки и отладки приложений, а C++ является мощным языком программирования, позволяющим эффективно управлять памятью и ресурсами системы. Библиотека WinAPI, в свою очередь, предоставляет широкие возможности для работы с операционной системой Windows, включая создание и управление окнами, обработку сообщений и многое другое.

Одним из первых шагов в разработке монитора окон было изучение документации WinAPI и освоение основных функций для работы с окнами. Эти функции включают в себя создание окна, установку его параметров (например, размера, положения, стиля), обработку сообщений, поступающих от операционной системы, и так далее. Visual Studio предоставляет удобный интерфейс для написания кода на C++ и отладки приложения, что значительно облегчает процесс разработки.

Для создания пользовательского интерфейса монитора окон были использованы элементы управления, предоставляемые WinAPI, такие как кнопки, меню, полосы прокрутки и др. Это позволяет пользователю удобно взаимодействовать с монитором окон, управляя параметрами отображения окон и процессами.

Также важным аспектом является сообщений от операционной системы. Это включает в себя отслеживание событий, происходящих с окнами (например, открытие, закрытие, изменение размера) и реагирование на них соответствующим образом. Эффективная обработка сообщений позволяет создать монитор окон, который отзывается на действия пользователя быстро и точно.

Также потребовалось использование многопоточности для обеспечения производительности и отзывчивости монитора окон. Например, один поток может отвечать за обновление интерфейса, а другой - за обработку сообщений и взаимодействие с WinAPI. Это позволит распределить нагрузку и оптимизировать работу приложения.

3.2.1 Язык программирования C++

Языки программирования C++ выбран в качестве основного языка разработки. Он предоставляет ряд преимуществ, которые делают его привлекательным выбором для разработки данного приложения.

Прежде всего, C++ обладает высокой производительностью и эффективностью, что особенно важно для монитора окон, который должен обеспечивать быстрый отклик на действия пользователя и эффективно управлять множеством окон и процессов. Благодаря низкоуровневым возможностям языка, можно оптимизировать производительность своего приложения и достичь высокой отзывчивости.

Кроме того, C++ предоставляет прямой доступ к системным ресурсам компьютера через библиотеку WinAPI. Прямой доступ позволяет создать монитор окон, который эффективно управляет процессами и окнами на уровне операционной системы. Разработчик имеет полный контроль над созданием, управлением и обработкой окон, что делает приложение более мощным и гибким.

Благодаря широким возможностям стандартной библиотеки C++, можно легко реализовать различные функции монитора окон, такие как управление размером и положением окон, обработка событий и сообщений от операционной системы, а также взаимодействие с другими приложениями.

Портативность C++ также является важным фактором при разработке монитора окон. При правильном написании кода приложение на C++ может быть легко перенесено на различные платформы, что дает возможность создавать кроссплатформенные решения или легко адаптировать разработанное приложение под различные операционные системы.

Таким образом, язык программирования C++ предоставляет мощные инструменты для создания эффективного и гибкого монитора окон. Благодаря высокой производительности, доступу к системным ресурсам и широким возможностям стандартной библиотеки, C++ является хорошим выбором для написания монитора окон. [13]

3.2.2 Основные этапы развития C++

Первые версии языка программирования C++ (тогда он назывался "Си с классами") были разработаны в начале 80-х годов Бьярном Страуструпом, сотрудником знаменитой AT&T Bell Labs, где ранее были разработаны такие шедевры программирования, как операционная система UNIX и язык программирования C.

По признанию самого автора языка, C++ никогда не разрабатывался на бумаге. Проектирование, реализация и документирование новых возможностей происходили фактически одновременно. Единственной целью разработки было создание языка, на котором было бы удобно программировать автору и его друзьям. Если вспомнить историю создания языка Си, то прослеживаются явные аналогии.

За основу был взят популярный в среде профессиональных разработчиков язык программирования C. Первыми средствами, которыми был расширен C, стали средства поддержки абстракций данных и объектно-ориентированного программирования.

Как это принято в AT&T, описание нового языка не было опубликовано сразу. Первыми его пользователями стали сами сотрудники Bell Labs. В 1993 впервые была реализована коммерческий транслятор, и сам язык был назван "C++", что можно (имея в виду операцию инкрементирования языка C) трактовать как увеличенный или расширенный язык C.

Первым транслятором языка был препроцессор cfront, транслирующий программу на C++ в эквивалентную программу на C. И только в конце 80-х годов были реализованы прямые трансляторы, не использующие C в качестве промежуточного языка. Пионером среди таких трансляторов стал GNU CC.

Если не считать документацию к транслятору cfront, первой книгой с описанием языка стала "The C++ Programming Language" (Addison-Wesley, 1985), переведенная на русский язык и изданная в 1991 году (Страуструп Б. Язык программирования C++. М.: Радио и Связь, 1991).

С этого момента началось его бурное распространение и создание многочисленных реализаций.

Модель реализации ООП была частично позаимствована из языка программирования Simula67 и ориентировалась в основном на возможность эффективной реализации на вычислительных машинах со стандартной архитектурой. Некоторые возможности языка Simula были отклонены, так как, по мнению автора C++, подталкивали разработчика к плохому стилю программирования. Так, в первых версиях C++ полностью отсутствовала возможность динамической идентификации типа объекта. Основные

концепции поддержки ООП в C++ были изложены Страуструпом в статье "What is Object Oriented Programming".

С 1985 года в язык были введены новые возможности: множественное и виртуальное наследование, шаблоны функций и классов, обработка исключительных ситуаций. Кардинально изменена семантика совместного использования оператора new, изменен синтаксис для вложенных классов.

С момента опубликования и до настоящего момента язык постоянно усовершенствовался и расширялся. Важным этапом в его развитии стала публикация в 1990 году подробного и достаточно строгого описания языка. Сокращенно эту книгу часто называют ARM. Фактически одновременно с этим началась стандартизация языка.

Инициатором стандартизации выступил не автор языка. Более того, Страуструп всегда довольно прохладно относился к попытке его полной стандартизации и выступал за реализации, в которых базовые возможности языка расширялись бы средствами и библиотеками, характерными только для данной реализации. [14]

3.2.3 История стандартизации C++

Объединенный ANSI-ISO комитет начал функционировать в конце 1989 года. Целью его работы является создание единого стандарта для языка Си++ и его библиотечных средств. За основу проекта стандарта было взято описание языка, данное в ARM.

В работе объединенного комитета, функционирующего и по сей день, значительное место занимает изучение (с последующим принятием или отказом от) возможных изменений текста проекта стандарта, а также уточнение различных правил языка. Работа по его стандартизации завершилась в 1989 году, и стандартизованный вариант сейчас известен под именем ANSI C.

Работа по стандартизации C++ осложнялась тем, что язык долгое время был открыт для расширений. Сами формулировки правил ARM были недостаточно строги и часто требовали уточнения. C++ стал довольно громоздким.

Изначально планировалось, что окончательная редакция проекта стандарта будет опубликована в 1994 году. Эти сроки были безнадежно провалены. Можно сказать, что последние 3 года процесс стандартизации постоянно находится в состоянии «2 года до завершения».

В ранних версиях проекта стандарта не было раздела, описывающего стандартные библиотеки. Не было описаний библиотеки и в ARM. В то же

время реализация библиотеки потокового ввода/вывода, предложенная Andrew Koenig, была повторена в нескольких реализациях и стала фактическим стандартом. В 1993-1994 годах в проекте стандарта было введено около семи новых разделов для описания библиотеки.

Принципиально важным событием в истории развития стандарта стандартной библиотеки стало включение библиотеки STL (Standard Template Library) разработанной сотрудником Hewllet-Packard Александром Степановым. В своей статье об истории STL он упоминает, что изначально стремился использовать в C++ только возможности шаблонов, аналогичные generic пакетам и процедурам языка Ada, но после обсуждений со Страуструпом существующих возможностей C++ изменил свое мнение. Комитет по стандартизации пошел им навстречу, вплоть до того, что в семантику шаблонов были внесены изменения. Таким образом получилось, что не библиотека написана для языка, а сам язык претерпел изменения под влиянием библиотеки, причем разработанной не автором языка.

В итоге официальная стандартизация языка началась в 1998 году, когда был опубликован стандарт языка ISO/IEC 14882:1998 (известный как C++98), разработанный комитетом по стандартизации C++ (ISO/IEC JTC1/SC22/WG21 working group). Стандарт C++ не описывал способов именования объектов, некоторых деталей обработки исключений и других возможностей, связанных с деталями реализации, что делает несовместимым объектный код, созданный различными компиляторами.

В 2003 году опубликован стандарт C++ ISO/IEC 14882:2003, где были исправлены выявленные ошибки и недочёты предыдущей версии стандарта. В 2005 году опубликован отчёт Library Technical Report (TR1). Не являясь официально частью стандарта, отчёт описывает расширения стандартной библиотеки, которые, по мнению авторов, должны были быть включены в следующую версию стандарта.

С 2009 года велась работа по обновлению предыдущего стандарта. Предварительная версия называлась C++09, в следующем году её переименовали в C++0x. Стандарт был опубликован в 2011 году под названием C++11. В него включены дополнения в ядре языка и расширение стандартной библиотеки, в том числе большая часть TR1.

Следующая версия стандарта, C++14, вышла в августе 2014 года. Она содержит в основном уточнения и исправления ошибок предыдущей версии.

Стандарт C++17, опубликованный в декабре 2017 года, включил в стандартную библиотеку параллельные версии стандартных алгоритмов и удалил некоторые устаревшие и крайне редко используемые элементы.

Последняя стабильная на текущий момент действующая версия стандарта – C++20. Помимо прочего, он содержит поддержку модулей. [15]

3.2.2 Библиотека WinAPI

Библиотека Win32API (Windows API) играет ключевую роль в разработке монитора окон на платформе Windows. Далее будут рассмотрены основные причины, по которыми обусловлен выбор WinAPI для реализации этого приложения.

Прежде всего, WinAPI обеспечивает непосредственный доступ к функциям операционной системы Windows. Такой доступ позволяет эффективно взаимодействовать с окнами, процессами и другими системными ресурсами. Благодаря этому монитор окон на основе WinAPI может обеспечить высокую производительность и полный контроль над процессами в системе.

Кроме того, WinAPI предоставляет широкий набор функций и классов для работы с окнами. Есть возможность создавать, управлять и обрабатывать окна, устанавливать их параметры (размер, положение, стиль) и реагировать на события, происходящие с окнами. Эти функции и классы позволяют создавать гибкий и функциональный пользовательский интерфейс монитора окон.

WinAPI также предоставляет средства для обработки сообщений от операционной системы. Обработка сообщений от операционной системы включает в себя получение и обработку события открытия и закрытия окон, перемещение и изменение размера окон и другие подобные сообщения. Обработка сообщений от операционной системы позволяет монитору окон реагировать на действия пользователя и изменения в системе в реальном времени.

WinAPI поддерживает многопоточность, а значит позволяет создавать многопоточные приложения, также полезно для монитора окон. Например, один поток может быть выделен для обновления интерфейса, а другой – для обработки сообщений и взаимодействия с WinAPI. Такой подход позволяет оптимизировать производительность и отзывчивость приложения.

В целом библиотека WinAPI предоставляет все необходимые средства для создания мощного и эффективного монитора окон на платформе Windows. Благодаря своей функциональности, гибкости и возможностям, WinAPI является хорошим выбором для разработки монитора окон. [16]

3.2.3 Многозадачность и многопоточность

Многозадачность (multitasking) – свойство операционной системы или среды выполнения обеспечивать возможность параллельной (или псевдопараллельной) обработки нескольких задач.

Многопоточность (multithreading) – свойство платформы (например, операционной системы, виртуальной машины и т. д.) или приложения, состоящее в том, что процесс, порождённый в операционной системе, может состоять из нескольких потоков, выполняющихся «параллельно», то есть без предписанного порядка во времени. При выполнении некоторых задач такое разделение может достичь более эффективного использования ресурсов вычислительной машины.

Многозадачность и многопоточность играют ключевую роль в разработке монитора окон, обеспечивая его отзывчивость, производительность и эффективное управление ресурсами компьютера.

Во-первых, многозадачность позволяет одновременно выполнять несколько задач на компьютере. Для монитора окон это означает возможность следить за несколькими процессами и окнами одновременно. Благодаря многозадачности, пользователь может взаимодействовать с различными приложениями и окнами, не переключаясь между ними.

Во-вторых, многозадачность важна для обработки параллельных событий и задач в мониторе окон. Например, приложение может одновременно отслеживать активность нескольких окон и обрабатывать события, происходящие в них, что обеспечивает плавное и непрерывное функционирование монитора окон.

Многопоточность в свою очередь позволяет разделить задачи на отдельные потоки выполнения, работающие параллельно. Для монитора окон это означает, что различные аспекты приложения могут работать независимо друг от друга, улучшая отзывчивость и производительность приложения.

Например, один поток может быть выделен для обновления графического интерфейса монитора окон, позволяя пользователю мгновенно видеть изменения в окнах и процессах. В это же время другие потоки могут обрабатывать сообщения от операционной системы, мониторить активность окон и выполнять другие задачи.

Использование многозадачности и многопоточности позволяет создать монитор окон, который работает быстро, эффективно управляет ресурсами компьютера и обеспечивает плавное и отзывчивое пользовательское взаимодействие. Эти концепции важны для создания современных

приложений, способных эффективно работать в многозадачной среде операционной системы. [17]

3.2.4 Инструменты разработки и отладка среды разработки

При реализации монитора окон использовались инструменты разработки и отладки в среде Visual Studio. Именно они обеспечивали эффективную разработку, отладку и сопровождение приложения.

Visual Studio предоставляет удобную и мощную среду для написания кода на языке программирования C++, который используется для создания монитора окон в контексте данной курсовой работы. Интегрированная система IntelliSense предоставляет автодополнение кода, подсказки и предупреждения об ошибках, что значительно ускоряет процесс написания кода и уменьшает количество ошибок.

Также Visual Studio обладает инструментами для отладки приложений. В частности, встроенный отладчик позволяет отслеживать выполнение программы, устанавливать точки останова, просматривать значения переменных и анализировать стек вызовов, что позволяет эффективно выявлять и исправлять ошибки в коде.

Visual Studio поддерживает интеграцию с системой контроля версий, что позволяет управлять версиями исходного кода и возвращаться к предыдущим версиям кода, если в текущей произошел непоправимый сбой. Это важно для работы над любым приложением, в том числе и над монитором окон, ведь помогает не терять весь прогресс при неожиданных сбоях системы.

Инструменты анализа кода в Visual Studio помогают выявлять потенциальные проблемы и улучшать качество кода монитора окон. Встроенные инструменты статического анализа и проверки стиля кода помогают выявлять ошибки и соответствовать стандартам кодирования, что способствует созданию более надежного и поддерживаемого приложения.

Таким образом, выбор *Visual Studio* и связанных с ним инструментов обеспечивает полноценную интегрированную среду для разработки, отладки и оптимизации монитора окон, предоставляющего список окон и управление ими, в том числе с фильтрацией и отбором. [18]

4 ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ ПРОГРАММЫ

Целью выполнения данной лабораторной работы является создание приложения, реализующего атаки на протокол при установке TCP-соединения и в рамках заданного протокола прикладного уровня.

5 ВЗАИМОДЕЙСТВИЕ С ПРОГРАММНЫМ ПРОДУКТОМ

Целью выполнения данной лабораторной работы является создание приложения, реализующего атаки на протокол при установке TCP-соединения и в рамках заданного протокола прикладного уровня.

ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы было создано приложение, реализующее атаки на протокол при установке TCP-соединения и в рамках заданного протокола прикладного уровня.

СПИСОК ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ

[1] Быстрое переключение между окнами в Windows [Электронный ресурс]. – Режим доступа: https://uchet-jkh.ru/i/kak-pereklyucatsya-mezdu-otkrytymi-oknami-windows-s-pomoshhyu-klaviatury/#:~:text=Win– Дата доступа: 23.02.2024.

[2] Переключатель окон [Электронный ресурс]. – Режим доступа: https://www.deskmodder.de/wiki/index.php?title=Alt%2BTab_Alte_Ansicht_aktivieren_Windows_10 – Дата доступа: 23.02.2024.

[3] История переключателя окон [Электронный ресурс]. – Режим доступа: https://neolurk.org/Alt-Tab. – Дата доступа: 24.02.2024.

[4] Task View в Windows [Электронный ресурс]. – Режим доступа: https://www.cu.edu/blog/tech-tips/using-task-view-windows. – Дата доступа: 24.02.2024.

[5] Описания и нововведения Windows 10 [Электронный ресурс]. – Режим доступа: https://www.softmagazin.ru/blog/vstrechayte_windows_10_opisanie_i_novovvedeniya/. – Дата доступа: 24.02.2024.

[6] Что такое операционная система [Электронный ресурс]. – Режим доступа: https://help.reg.ru/support/servery-vps/oblachnyye-servery/ustanovka-programmnogo-obespecheniya/chto-takoye-operatsionnaya-sistema/ – Дата доступа: 03.03.2024

[7] Интегрированная среда разработки [Электронный ресурс]. – Режим доступа: https://blog.skillfactory.ru/glossary/ide// – Дата доступа 05.03.2023

[8] Язык программирования [Электронный ресурс]. – Режим доступа: https://blog.skillfactory.ru/glossary/yazyk-programmirovaniya/ – Дата доступа 05.03.2023

[9] Microsoft Windows 10 – полный обзор [Электронный ресурс]. – Режим доступа: https://xn--80aa0aebnilejl.xn--p1ai/%D0%9F%D0%BE%D0%BB%D0%B5%D0%B7%D0%BD%D0%B0%D1%8F_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F/Microsoft_Windows_10/ – Дата доступа 06.03.2023

[10] Microsoft Learn [Электронный ресурс]. – Режим доступа: https://learn.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2022 – Дата доступа 09.03.2023

[11] Web proger C/C++ [Электронный ресурс]. – Режим доступа: http://web.spt42.ru/index.php/chto-takoe-c-plus-plus – Дата доступа 09.03.2023

[12] Представление задач в Windows [Электронный ресурс]. – Режим доступа: https://blogs.windows.com/russia/2017/02/16/sovety-po-windows-10-task-view-snap/ – Дата доступа 11.03.2023

[13] Преимущества C++ [Электронный ресурс]. – Режим доступа: https://blog.skillfactory.ru/cplus-komu-i-dlya-chego-nuzhen/ – Дата доступа 12.03.2023

[14] Этапы развития C++ [Электронный ресурс]. – Режим доступа: http://citforum.ru/programming/prg96/76.shtml – Дата доступа 13.03.2023

[15] Стандарты языка C++ [Электронный ресурс]. – Режим доступа: https://ejudge.179.ru/tasks/cpp/total/251.html – Дата доступа 14.03.2023

[16] Win32 в C++ [Электронный ресурс]. – Режим доступа: https://learn.microsoft.com/ru-ru/windows/win32/learnwin32/learn-to-program-for-windows – Дата доступа 14.03.2023

[17] Многозадачность и многопоточность [Электронный ресурс]. – Режим доступа: https://habr.com/ru/companies/otus/articles/549814/ – Дата доступа 15.03.2023

[18] Отладка в Visual Studio [Электронный ресурс]. – Режим доступа: https://learn.microsoft.com/ru-ru/visualstudio/get-started/visual-basic/tutorial-debugger?view=vs-2022 – Дата доступа 15.03.2023

ПРИЛОЖЕНИЕ А
(обязательное)
Листинг исходного кода

Листинг 1 – Программный код

```
import time


def print_package(package):
    try:
        time.sleep(1.5)
    except KeyboardInterrupt:
        raise BaseException("Interrupted by user")
```

ПРИЛОЖЕНИЕ Б
(обязательное)
Функциональная схема алгоритма, реализующего программное средство

ПРИЛОЖЕНИЕ В
(обязательное)
Блок схема алгоритма, реализующего программное средство

ПРИЛОЖЕНИЕ Г
(обязательное)
Графический интерфейс пользователя

ПРИЛОЖЕНИЕ Д
(обязательное)
Ведомость документов