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**

[1 Постановка задачи 3](#_Toc157722973)

[2 Краткие теоретические сведения 4](#_Toc157722974)

[3 Результаты выполнения лабораторной работы 10](#_Toc157722975)

[Выводы 15](#_Toc157722976)

[Список использованных источников 13](#_Toc157722977)

[Приложение А (обязательное) Листинг исходного кода 14](#_Toc157722978)

# **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.

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.

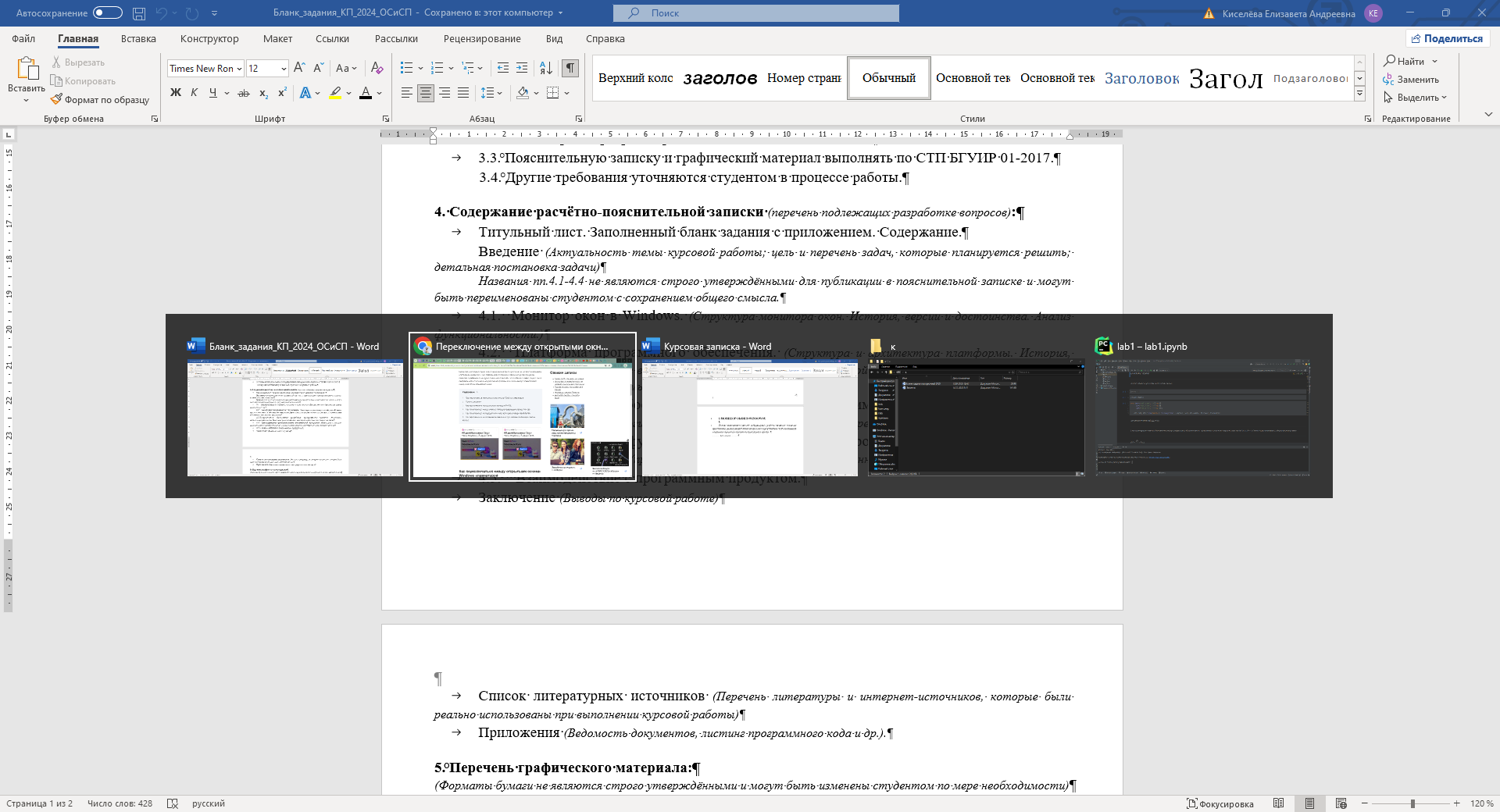


Рисунок 1.1 – Переключение между открытыми приложениями

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.

**1.2** **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.

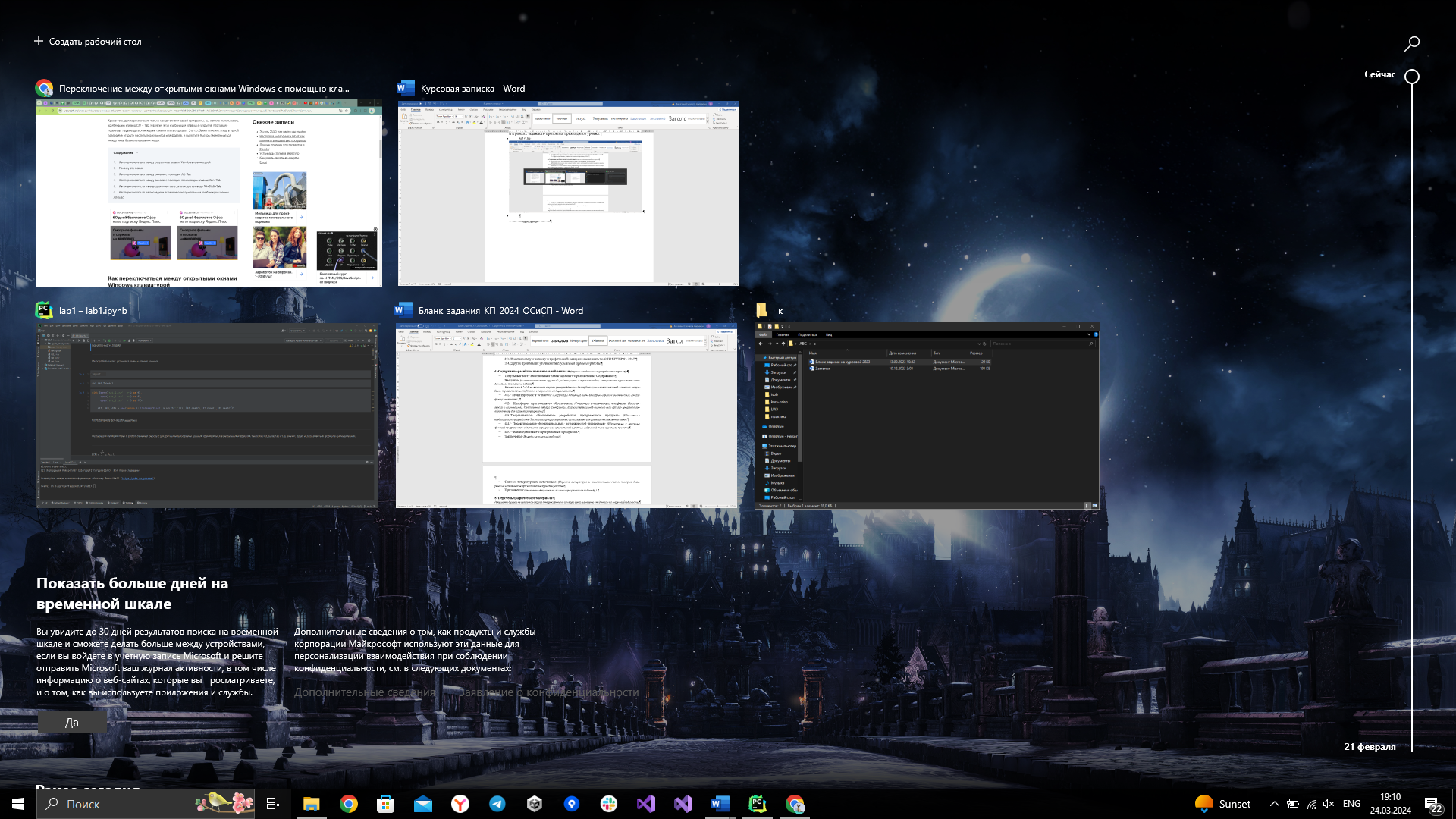


Рисунок 1.2 – Просмотр задач

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.

# **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 разработочной платформы. Операционная система *Windows* 10 была выбрана в качестве основной, обеспечивая широкий спектр совместимости с различными программными и аппаратными решениями. Это также предоставляет удобную интеграцию с инструментарием разработки и обеспечивает стабильную среду для проведения экспериментов и тестирования.

Интегрированная среда разработки *Visual* *Studio* была предпочтительным выбором для написания кода и выполнения проекта. *Visual* *Studio* обеспечивает удобный интерфейс, богатые возможности отладки, а также поддержку различных языков программирования. Это средство разработки обладает широким функционалом и является стандартом в индустрии для разработки программного обеспечения под *Windows*.

Язык программирования *C/C*++ был выбран в связи с его эффективностью, близким уровнем доступа к аппаратным ресурсам и поддержкой параллельных вычислений. *C/C*++ предоставляет высокую производительность и контроль над памятью, что особенно важно при реализации распределенных вычислений на нескольких процессорах (*CPU*, *GPU*).

Таким образом, выбранная платформа, объединяющая *Windows* 10, *Visual Studio* и язык программирования *C/C*++, обеспечивает оптимальные условия для эффективной разработки, тестирования и анализа распределенных вычислений в рамках поставленных задач курсовой работы.

**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.

С точки зрения программиста операционная система – это программное окружение, которое предоставляет набор абстракций и сервисов для управления аппаратными ресурсами компьютера. Программист взаимодействует с операционной системой для выполнения своих программных задач. [10]

**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. [11]

**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. [12]

**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 ПК, 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 ПК, 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 надежным выбором для разработчиков, удовлетворяя современным требованиям и обеспечивая эффективность в процессе работы. [13]

**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 его универсальным инструментом для разработчиков.

3 Отладка. Мощные инструменты отладки, включая 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, что позволяет разработчикам 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 программистов. Основные преимущества выбранной интегрированной среды разработки включают многозадачность, языковую поддержку, мощные средства отладки, графические средства, тестирование, совместную разработку, расширяемость, и регулярные обновления, делая ее универсальным и современным инструментом для разработчиков. [14]

**2.2.3** Язык программирования *C/C*++

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

История языков программирования *C/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 выбранных языков 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 языков программирования *C/C*++:

1 Эффективность. *C*/*C*++ известны своей высокой производительностью и эффективностью выполнения.

2 Близость к аппаратуре. Эти языки предоставляют низкоуровневый доступ к памяти и аппаратным ресурсам, что важно для системного программирования.

3 Обширные библиотеки. Оба языка обладают обширными библиотеками, предоставляя разработчикам множество функций и инструментов.

4 Переносимость. Код на *C*/*C*++ легко переносится между различными платформами.

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

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

7 Сообщество разработчиков. Оба языка имеют огромное сообщество разработчиков, что обеспечивает поддержку и ресурсы для программистов.

Использование *C*/*C*++ остается актуальным в различных областях программирования, обеспечивает мощные инструменты для разработки высокопроизводительных приложений. [15]

**2.3 Обоснование выбора платформы**

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

Язык программирования *C/C*++ выбран для обеспечения высокой производительности и низкоуровневого управления ресурсами, что является важным аспектом для эффективных распределенных вычислений. Его близость к аппаратному обеспечению позволяет оптимизированно использовать вычислительные ресурсы.

Интегрированная среда разработки *Visual Studio* предоставляет полный инструментарий для написания, отладки и тестирования кода, обеспечивая эффективную разработку сложных программных проектов. Многозадачность и средства совместной разработки делают ее удобным выбором для командной работы.

Операционная система *Windows* 10 была выбрана из-за широкой поддержки пользователей и совместимости с разнообразными вычислительными системами. Драйвера и библиотеки обеспечивают необходимый доступ к ресурсам, что важно при работе с различными процессорами, включая *CPU* и *GPU*.

Можно сказать, что комбинация *C/C*++ обеспечивает эффективность и контроль над ресурсами, *Visual Studio* предоставляет удобную среду разработки, а *Windows* 10 обеспечивает совместимость с большинством вычислительных систем. Также этот выбор соответствует требованиям задачи по распределенным вычислениям на различных процессорах (*CPU*, *GPU*).

**2.3 Анализ программного обеспечения для написания программы**

В контексте разработки курсового проекта, фокус которого направлен на распределенные вычисления, выбор операционной системы, среды разработки и языка программирования играют важную роль. Рассмотрим операционную систему *Windows* 10, среду разработки *Visual Studio* и языки программирования *C/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* и *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 языков программирования *C/C++* придает проекту высокую производительность, что особенно ценно для системного программирования и разработки высокопроизводительных приложений. Кроме того, прямой доступ к системным ресурсам, предоставляемый *C/C*++, важен для создания приложений, требующих близкого взаимодействия с аппаратным обеспечением.

Использование операционной системы Windows 10, Visual Studio и языка программирования C/C++ дает возможность создания производительных и эффективных приложений. Интеграция Windows 10 с Visual Studio обеспечивает оптимальную совместимость и производительность, а выбор C/C++ позволяет разработчикам 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**

В ходе выполнения данной лабораторной работы было создано приложение, реализующее атаки на протокол при установке TCP-соединения и в рамках заданного протокола прикладного уровня. В ходе выполнения программа может производить четыре вида атак, а именно RST Flood, TCP Reset, IP-spoofing и TCP/IP hijacking.

В ходе RST Flood атаки сервер-жертва на высокой скорости получает поддельные RST-пакеты, не имеющие отношения к любой из сессий в базе данных сервера. Сервер-жертва вынужден выделять значительное количество системных ресурсов для сопоставления входящих пакетов с текущими соединениями, что приводит к потере производительности сервера и к его частичной недоступности. Пример RST Flood атаки представлен на рисунке 3.1.

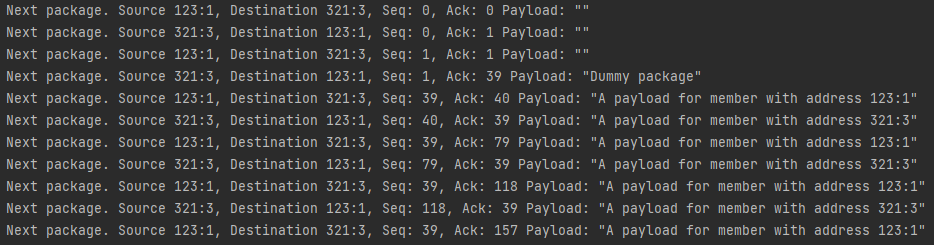


Рисунок 3.1 – Пример RST Flood атаки

При атаке TCP Reset нападающий разрывает соединение между двумя жертвами, отправляя одной или обеим фальшивые сообщения, приказывающие им немедленно прервать соединение. Пример TCP Reset атаки представлен на рисунке 3.2.

# **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] TCP Reset [Электронный ресурс]. – Режим доступа: https://habr.com/ru/articles/499902/. – Дата доступа: 23.02.2024.

[3] TCP hijacking [Электронный ресурс]. – Режим доступа: https://encyclopedia.kaspersky.ru/glossary/tcp-ip-hijacking/. – Дата доступа: 14.03.2024.

[4] TCP Reset [Электронный ресурс]. – Режим доступа: https://habr.com/ru/articles/499902/. – Дата доступа: 14.03.2024.

[10] 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

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

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

[13] 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

[14] 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

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

## **ПРИЛОЖЕНИЕ А**

## **(обязательное)**

## **Листинг исходного кода**

Листинг 1 – Программный код main.py

import time

def print\_package(package):

try:

time.sleep(1.5)

except KeyboardInterrupt:

raise BaseException("Interrupted by user")