

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Политехнический институт
Факультет приборостроения,
информационных технологий и
электроники

Кафедра
«Информационная безопасность
систем и технологий»

Специальность 10.05.02 –

Специализация –

Информационная безопасность
телекоммуникационных систем
Разработка защищенных
телекоммуникационных систем

ДИПЛОМНЫЙ ПРОЕКТ

на тему

ПРОГРАММНОЕ СРЕДСТВО ШИФРОВАНИЯ ПОЛЬЗОВАТЕЛЬСКИХ
ДАННЫХ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМА ПО ГОСТ Р 34.12-2015.

Студент _____ Лопухин Илья Викторович

Руководитель _____ Калмык С.Ф.

Нормоконтролёр _____ Иванов А.П.

Рецензент Начальник УПСЗИ ООО НТП «Криптософт» _____ Вартанов А.А.

Работа допущена к защите (протокол заседания кафедры от _____ № _____)

Заведующий кафедрой _____ С.Л. Зефилов

Работа защищена с отметкой _____ (протокол заседания ГЭК
от _____ № _____)

Секретарь ГЭК _____ А.П. Иванов

Пенза 2018

Оглавление

Введение.....	4
1 Основные понятия разработки драйвера.....	6
1.1 Общие сведения о драйверах. Основные общие концепции драйверов в операционных системах.....	6
1.2 Классификация драйверов.....	7
1.3 Программное обеспечение, необходимое для разработки драйверов....	8
2 Архитектура Windows и архитектура WDM.....	10
2.1 Фундаментальные задачи Windows. Архитектура Windows.....	10
2.2 Архитектура WDM	14
3 Особенности функционирования драйвера-минифилтра.....	17
4 Разработка драйвера-минифилтра.....	21
5 Разработка утилиты «Мастер управления ключами».....	36
6 Разработка модели угроз для программного средства.....	41
7 Тестирование разработанного программного средства.....	77
8 Вредные психофизиологические факторы, влияющие на органы зрения специалиста по защите информации. Методы и средства физической культуры, снижающие их воздействие.....	83
Заключение.....	87
Список используемых источников.....	88
Приложение А Исходный код драйвера-минифилтра.....	89
Приложение Б Исходный код утилиты «Мастер управления ключами».....	128
Приложение В Модель угроз для разработанного программного средства	139
Приложение Г Акт о внедрении результатов ВКР.....	150
Приложение Д Протокол проверки в системе «Антиплагиат».....	152

Обозначения и сокращения

DDK – device driver kit;

IRP – input/output request packet;

IT – information technology;

KMD – kernel-mode driver;

VDD – virtual device driver;

WDM – windows driver model;

ВКР – выпускная квалификационная работа;

ВМ – виртуальная машина;

ОС – операционная система;

ПК – персональный компьютер;

ПО – программное обеспечение;

ТЗ – техническое задание.

Введение

В век развития информационных технологий вопросы о защите информации занимают ключевые позиции. Существует большое количество способов и методов защиты информации, например шифрование. Для шифрования могут быть использованы специальные программы.

Возможность шифрования пользовательских данных важна для современных пользователей. Даже если пользователь придумает стойкий пароль для того, чтобы защитить свой пользовательский аккаунт, безопасность данных будет под угрозой, когда кто-либо решит украсть ваше устройство. У большинства ПК локальный диск легко вытащить и поставить в другую систему, также можно загрузить ПК с внешним диском и скопировать всю информацию с диска. Даже если вы удалили информацию с локального диска полностью, программа для восстановления сможет считать удаленные файлы.

Шифрование пользовательских данных делает всё вышеперечисленное сложно выполнимым, либо вовсе невозможным. Помимо этого, шифрование защищает от кражи данных с помощью вирусных приложений и троянов. Даже если злоумышленник похитит какую-либо информацию, он не сможет ее прочитать. И если даже пользователь сотрёт свою информацию с диска, оставшаяся информация, которая может считаться с помощью программы восстановления данных, по-прежнему будет зашифрована.

Актуальность темы также подчёркивается использованием нового стандарта блочного шифрования ГОСТ Р 34.12-2015 «Кузнечик», принятого в июне 2015 года. Новый стандарт шифрования имеет ряд преимуществ по сравнению со стандартом блочного шифрования ГОСТ 28147-89, который теперь называется «Магма».

При выполнении ВКР учитывалось, что при шифровании пользовательских данных важна скорость шифрования и удобство использования программного средства шифрования для пользователя. Использование нового стандарта ГОСТ Р 34.12-2015 позволит шифровать пользовательские данные быстрее, чем при использовании ГОСТ 28147-89, так как в новом стандарте

увеличена длина блока до 128 бит и сокращено число циклов до 10, что позволяет шифровать данные за меньшее количество итераций.

Удобство использования разрабатываемого программного средства шифрования будет достигаться тем, что в состав программного средства будет входить фильтр-драйвер, который будет перехватывать все запросы к файловой системе, автоматически шифровать файлы, которые пользователь будет переносить в специальный каталог, а доступ к файлам будет получать через утилиту «Мастер управления ключами».

1 Основные понятия разработки драйвера

1.1 Общие сведения о драйверах. Основные общие концепции драйверов в операционных системах

Драйвер – это часть кода ОС (программа, предназначенная не для пользователя, а для ОС), которая отвечает за взаимодействие с аппаратной частью. Под термином «аппаратная часть» можно подразумевать как реальные физические устройства, так и логические, и виртуальные [1]. Например, существует много различных принтеров нескольких разных производителей. Для того чтобы использовать их для печати, необходимо знать, какие именно команды выбранному принтеру нужно передавать. У каждого производителя эти команды могут очень сильно отличаться один от другого. Необходимо как-то научить операционную систему работать с любыми типами принтеров. Для этого используются драйверы.

С того момента, как драйвер появился и до сегодняшнего дня, он постоянно эволюционировал, и этот процесс продолжается до сих пор.

Как отдельный и достаточно самостоятельный модуль драйвер развивался не сразу, да и в данный момент некоторые драйверы почти неотделимы от ОС. В большом числе случаев это приводит к обязательной переустановке системы (операционной системы Windows) или к переборке ядра ОС (в системах UNIX).

Список главных общих концепций драйверов в ОС Windows и системах UNIX выглядит следующим образом:

- метод работы с драйверами как с файлами, т.е. функции, которые используются при взаимодействии с определенными файлами, фактически одинаковы при взаимодействии с драйверами (лексически): open, close, read и другие;
- драйвер, как легкозаменяемая часть операционной системы;
- наличие режима ядра ОС.

1.2 Классификация драйверов

Классификация драйверов для ОС Windows:

- драйверы режима пользователя (user-mode drivers):
 - драйверы для принтеров (printer drivers);
 - драйверы для виртуальных устройств (virtual device drivers, VDD)
- необходимо использовать для поддержания программ MS-DOS;
- драйверы режима ядра ОС (kernel-mode drivers):
 - драйверы для видеоадаптеров (video drivers) - необходимы для реализации графических операций;
 - драйверы файловой системы (file system drivers) - необходимы для осуществления ввода/вывода на локальные и сетевые диски;
 - драйверы потоков устройств (streaming drivers) - необходимы для осуществления ввода/вывода потоков звука и видео;
 - унаследованные драйверы (legacy drivers) - были написаны для старых версий ОС Windows;
 - WDM драйверы (windows driver model) - необходимы для поддержания технологии Plug and Play и управления электропитанием.

Существуют одноуровневые и многоуровневые драйверы. Если драйвер многоуровневый, то обработка запросов на ввод и вывод делятся между драйверами, каждый из которых должен выполнять свою часть работы. Между этих драйверов существует возможность установить любое количество фильтр-драйверов (filter-drivers). Во время обработки запроса данные поступают от вышестоящих драйверов к нижестоящим, а во время возврата – наоборот от нижестоящих драйверов к вышестоящим. Одноуровневый драйвер (monolithic) является противоположным к многоуровневому.

Для Plug and Play технологии можно выделить три типа драйверов:

- фильтр-драйверы;
- шинные драйверы;
- функциональные драйверы.

Системное программное обеспечение, поддерживающее технологию Plug and Play предоставляет в распоряжение следующие возможности:

- распределение и перераспределение ресурсов между запрашивающими их устройствами;
- автоматическая идентификация подключенных к ОС устройств;
- загрузка нужных драйверов;
- предоставление драйверу интерфейса для взаимодействий с Plug and Play технологией;
- реализация механизмов, которые позволят драйверу и приложению получать информацию обо всех изменениях в наборе устройств, которые подключены к системе устройств, и позволят совершать необходимые действия;

1.3 Программное обеспечение, необходимое для разработки драйверов

Существует множество утилит, которые нужны при разработке драйвера. Главным средством для разработки является Microsoft Windows DDK — это пакет для разработки драйверов, который включает компилятор, редактор связей, заголовочные файлы, множество библиотек, большой набор примеров, многие из которых являются драйверами, которые на самом деле работают в ОС, и документацию. В состав входит также отладчик WinDbg, с помощью которого можно проводить отладку драйвера на двухкомпьютерной конфигурации и кроме того отладчик позволяет смотреть файлы образа памяти, которые получены при сбоях ОС (crash dump file).

В DDK используется язык программирования C. Этот язык программирования допускает вставки на ассемблере, который раньше был единственным и основным языком для программирования драйверов.

В бесплатном пакете DDK отсутствует интегрированная среда разработки, поэтому для редактирования исходного кода есть необходимость использовать Visual Studio. С помощью данной среды разработки процесс выявления синтаксических ошибок существенно облегчается, а это является неотъемлемым достоинством интегрированных сред для программирования. Компилятор и

линковщик среды разработки Visual Studio производят бинарный код, который является вполне работоспособным при указании необходимых опций компиляции, но эталоном следует считать файл с бинарным кодом, который получается при компиляции кода с использованием утилиты Build, которая имеется в составе пакета DDK. Встроенный отладчик Visual Studio и прилагаемая документация для разработки драйвера не являются нужными, так как не предназначены для работы с ПО режима ядра.

Существуют другие пакеты для разработки драйверов: WinDriver и NuMega Driver Studio, но у них есть некоторые отличия базиса функций Microsoft и много других небольших неудобств.

Также необходимо использовать сторонние утилиты, такие как: редактор реестра, средство для просмотра каталогов имен объектов, программы для просмотра и сохранений отладочных сообщений.

Для написания драйверов под 64-битные системы очень полезными будут являться такие программы, как:

- анализаторы PREfast, созданный компанией Microsoft. Это анализатор кода, обнаруживающий некоторые ошибки, которые встречаются и в драйверах, и в обычных программах, написанных на C и C++;
- статический анализатор кода для общего назначения Gimpel Software PC-lint;
- статический анализатор кода Viva64, предназначенный для поиска ошибок, которые проявляются при переносе кода с 32-битных на 64-битные системы в программах, написанных на C и C++. [1]

2 Архитектура Windows и архитектура WDM

2.1 Фундаментальные задачи Windows. Архитектура Windows

Ниже перечислены фундаментальные задачи Windows:

- совместимость. ОС должна поддерживать большое множество аппаратного и программного обеспечения;
- переносимость. ОС должна функционировать на большом количестве аппаратных платформ;
- расширяемость. Так как требования рынков увеличиваются, ОС должна уметь без труда увеличивать набор своих возможностей и перечень аппаратуры, которую она поддерживает, при небольшом вмешательстве в код операционной системы;
- надежность и устойчивость. ОС должна быть стойкой к преднамеренному или неумышленному ошибочному использованию компонентов. Приложения не должны иметь возможностей, которые приведут систему к фатальным сбоям;
- производительность. ОС должна обеспечивать высокую производительность на всех аппаратных платформах, которые поддерживаются.

Провозглашенные цели не всегда достигаются, и поэтому компромиссы общих и текущих значимых задач могут оказаться неизбежными. Операционная система Windows также подвержена этим компромиссам, также как и все остальные ОС.

Уровни аппаратных привилегий в ОС Windows.

Для того, чтобы обеспечить устойчивость в работе операционной системы, разработчики ОС Windows выбрали для ядра так называемую «архитектуру клиент-сервер». Пользовательское приложение в данном случае является клиентом служб ОС.

Пользовательские приложения функционируют в специальном режиме, который называется «user mode» — пользовательский режим. В пользовательском режиме код приложения ограничивается выполнением

безопасных инструкций. Например, через реализацию маппинга виртуальной памяти, который представляет собой страничное представление памяти, код пользовательского приложения лишается возможности получения доступа к памяти, которая предоставляется другим приложениям (кроме случаев обоюдного согласия, которое реализуется специальными для этого случая методами). Также команды аппаратного ввода/вывода не смогут быть выполнены с помощью кода приложения пользовательского режима. Большое количество команд центрального процессора, которые называются привилегированными, запрещены в Windows для выполнения кодом приложения пользовательского режима, например такие команды процессора как IN, OUT. Если приложению пользовательского режима требуется выполнить какую-либо из привилегированных команд, приложение должно сделать запрос на соответствующую службу ОС.

Код самой ОС выполняется в «kernel mode» — режиме ядра. Код данного режима может выполнять любые процессорные инструкции, в том числе инструкции ввода/вывода. Память, которая принадлежит любому приложению, доступна коду режима ядра.

Процессоры реализуют несколько форм привилегированного. Код режима ядра запускается в привилегированном контексте, а пользовательский код выполняется уже в непривилегированной среде. Разные процессоры, равно как и платформы, реализуемые на их основе, по-разному реализуют привилегированные режимы, например, для того, чтобы обеспечить переносимость разработки ОС вводят особые элементы программирования, которые дают возможность разграничить режим ядра и пользовательский режим. Код ОС использует их для переключения привилегированного и непривилегированного контекста, поэтому при перенесении ОС необходимо переписывать под специфику конкретно новой аппаратной платформы только код этих дополнительных элементов. На платформе «Intel» режим пользователя реализован из набора инструкций Ring 3 (третье кольцо), а режим уровня ядра реализован с использованием Ring 0 (нулевое кольцо).

Ring 3 – это кольцо, имеющее наименьшее количество привилегий, в котором есть много ограничений по работе с устройствами, памятью и так далее. В третьем кольце приложения не способны читать с адресного пространства других приложений, если они не имеют на это особого разрешения. Также приложения на третьем кольце не могут выполнять особые привилегированные команды процессора и напрямую обращаться к оборудованию.

Режим ядра (или защищенный режим) – это главный режим работы процессоров. Основные механизмы, которые регулируются режимом ядра:

- механизм переключения задач (такие задачи хранят в себе состояние всех регистров процессора, которые с ней связаны. Хранится это состояние на сегментах);
- механизм защиты памяти и ввода/вывода, который состоит из 4 уровней;
- особая организация адресного пространства.

При такой организации адресного пространства используется два способа для ее преобразования: сегментация и разбивка на страницы. Сегмент – это блок общего пространства памяти. В 32-битной адресации максимальный размер сегмента равен 4 Гб, а максимально возможное количество сегментов 8192. Сегментация может обеспечить защиту данных. Страничный способ организации позволяет использовать большее количество памяти.

Драйверы для уровня ядра работают в привилегированном контексте. Поэтому если драйвер плохо написан, он может стать вредоносным для ОС. Разработчик должен очень внимательно разрабатывать такой код, чтобы не сломать операционную систему. Фирма Microsoft решает проблему надежности драйверов, поставляемых в составе Windows, посредством механизма подписания и тестирования драйверов.

Переносимость.

Для решения задачи переносимости конструкторы Windows используют многослойную архитектуру.

Существует слой аппаратных абстракций, который изолирует платформенные и процессорные особенности от кода ОС. Microsoft предлагает

пользоваться слоем аппаратных абстракций разработчику драйверного кода. Возможно написать код драйвера так, что для переноса на другую платформу требуется только его перекомпилировать. В этом случае драйвер должен обращаться к использованию средств слоя аппаратных абстракций для взаимодействия с аппаратной поддержкой шин и аппаратными регистрами. В некоторых случаях разработчик может использовать код, который предоставляется диспетчером ввода/вывода, и используется для совместной работы с используемыми аппаратными ресурсами.

Расширяемость.

Одной из функций ядра является планирование активности программных потоков. Для каждого из потоков необходимо сохранить уникальный потоковый контекст, чтобы сохранить независимость от других потоков. Потоковый контекст включает в себя состояния регистров процессора, сохраненного идентификатора потока (Thread ID), распределения памяти, связанной с потоком (Thread Local Storage), значения приоритета, и другой информации, которая относится к данному потоку.

Планировщик потоков определяет, какой поток должен выполняться в текущий момент времени. В среде, которая имеет единственный процессор, в каждый момент времени процессор получает в распоряжение только один поток. В многопроцессорной конфигурации потоки имеют возможность выполняться одновременно на разных процессорах, реализуя параллельность выполнения кода. Планировщик чаще всего выделяет потоку процессор на определенный временной интервал, который называется потоковым временным квантом. В основном предоставление процессора происходит на основе величины приоритета потока.

Так как главной задачей ядра является управление потоками, то работа с управлением памятью, действиями по вводу/выводу, вопросы доступа возлагаются на другие компоненты ОС. Эти компоненты называются исполнительными компонентами. Такой компонент представляет из себя модульное программное обеспечение.

Производительность.

Так как многие участники сложных и объемных программных проектов хорошо знают о такой особенности многослойного ПО как его низкая производительность, особое внимание разработчики Windows уделили ускорению межслойного взаимодействия.

Во-первых, все слои, выполняются в аппаратном режиме (в режиме ядра). Поэтому межслойные вызовы не могут использовать ничего тяжелее, чем инструкция процессора CALL. Предоставляемые же слоем аппаратных абстракций средства обычно представляют собой макроопределения, которые являются inline-включаемым кодом.

Во-вторых, разработчики постарались над тем, чтобы параллельно заработало максимально большое число программных потоков. Вспомогательные процедуры блокируются редко или находятся в режиме ожидания, что существенно уменьшает время простоя процессора.

Вопросы повышения производительности касаются в большой степени и разработчиков драйверов. Когда системный поток или пользовательское приложение посылает запрос к устройству, которое обслуживается драйвером, очень важно, чтобы код драйвера не совершал блокирование выполнения запроса. Поэтому если запрос не может быть обработан в данный момент (например, если устройство медленно работает или занято), запрос должны поставить в очередь для дальнейшей обработки запроса. Диспетчер ввода/вывода предоставляет все необходимые для этого процедуры. [2]

Архитектура Windows изображена на рисунке 1.

2.2 Архитектура WDM

WDM (windows driver model) – это драйверная модель от компании Microsoft для ОС Windows, которая пришла на смену предыдущей среде написания драйверов для ОС Windows – VDD (Virtual Device Driver).



Рисунок 1 – Архитектура Windows

WDM в настоящий момент времени – одна из самых важных концепций в написании драйверов.

Главными особенностями WDM являются:

- поддержка для управления питанием;
- совместимость на уровне бинарных кодов между драйверами для Windows 98 и Windows NT;
- поддержка для Plug and Play;
- поддержка для «продвинутого» шинного управления.

Жизненный цикл среднестатистического WDM драйвера следующий:

- а) обнаружение устройства драйвером шины;
- б) Plug and Play Manager определяет, где находится ключ устройства в ветке Enum реестра. Данный ключ содержит ссылку на другой ключ реестра, который определяет функциональный драйвер. PnP-менеджер динамически загружает функциональный драйвер;
- в) PnP-менеджер вызывает метод драйвера AddDevice для того, чтобы создать объект драйвера. Если драйвер соответствует более, чем одному устройству, PnP-менеджер вызовет AddDevice для каждого из них. С этого момента драйвер взаимодействует с окружением с использованием IRP (I/O Request Packet) – пакетов;

г) PnP-менеджер выделяет необходимые ресурсы ввода/вывода (номера портов, запросы на прерывание и т.д.) и посылает запрос для инициализации устройства;

д) если данное устройство может быть удалено из системы без выключения компьютера, то PnP-менеджер посылает драйверу специальный IRP-пакет, после чего объект устройства уничтожается;

е) когда все устройства удалены, менеджер ввода/вывода (I/O Manager) вызывает функцию Driver_unload, удаляющую образ драйвера из памяти. [3]

3 Изучение особенностей функционирования драйвера-минифilterа

Фильтрация осуществляется с помощью так называемого Filter Manager, который включен в операционную систему Windows. Filter Manager активируется только при загрузке мини-фильтров. Filter Manager подключается к стеку файловой системы напрямую.

Мини-фильтры регистрируются на обработку данных по операциям ввода-вывода, используя функционал Filter Manager, таким образом получая косвенный доступ к файловой системе. После процесса регистрации и запуска мини-фильтр получает данные о операциях ввода-вывода, которые указывались при конфигурировании. Также при необходимости мини-фильтр может внести изменения в данные о операциях ввода-вывода, таким образом влияя на работу файловой системы.[4]

Так же, как и унаследованные фильтр-драйверы драйверы-минифilterы подключаются в определенном порядке. Данный порядок подключения определяется уникальным идентификатором, называемым высотой. Каждый экземпляр драйвера-минифilterа подключается на определенном значении высоты. Значение высоты мини-фильтра определяет последовательность, в которой Filter Manager вызывает драйвер-минифilter для обработки запросов по операциям ввода-вывода. Значение высоты выделяется и управляется операционной системой.

На рисунке 2 изображена схема функционирования Filter Manager.

Для каждой операции ввода-вывода, выбранной для фильтрации, драйвер-минифilter может регистрировать процедуру, которая будет вызываться перед операцией ввода-вывода и (или) процедуру которая будет вызываться после операции ввода-вывода. При обработке операции ввода-вывода Filter Manager вызывает соответствующую процедуру для каждого драйвера мини-фильтра, зарегистрированного для определенной операции ввода-вывода. Когда эта процедура выполняется, менеджер фильтров вызывает соответствующую процедуру для следующего драйвера мини-фильтра, зарегистрированного для данной операции ввода-вывода.

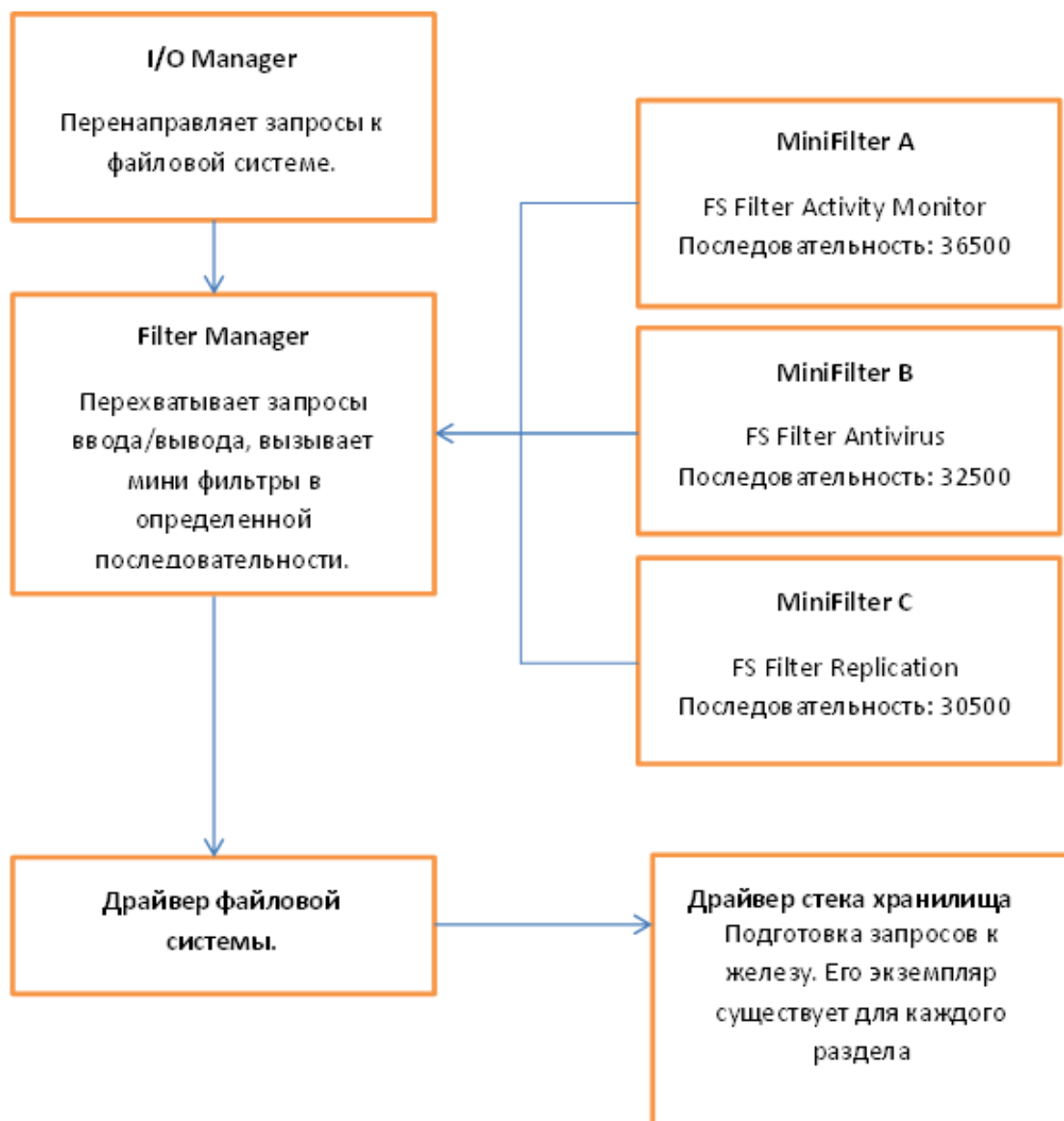


Рисунок 2 – Схема функционирования Filter Manager.

Например, при условии, что все три драйвера мини-фильтра, приведенные на рисунке 2 зарегистрированы для одной и той же операции ввода-вывода, Filter Manager будет вызывать свои процедуры перед операцией ввода-вывода в порядке высоты от наивысшего до самого низкого (А, В, С), а затем пропускает запрос ввода-вывода к следующему ниже драйверу для дальнейшей обработки. Когда Filter Manager получает запрос на завершение операции ввода-вывода, он вызывает каждую послеоперационную процедуру мини-фильтра, но

уже в обратном порядке, от самого низкого до самого высокого (C, B, A).

Для взаимодействия драйверов-минифильтров с унаследованными фильтр-драйверами Filter Manager может присоединять объекты фильтрующего устройства к стеку ввода-вывода файловой системы. Каждый из фильтрующих устройств менеджера фильтра называется фреймом. С точки зрения унаследованного фильтр-драйвера каждый фрейм менеджера фильтров является еще одним фильтр-драйвером.

Каждый фрейм менеджера фильтра представляет собой диапазон высот. Менеджер фильтров может настроить существующий фрейм или создать новый фрейм, чтобы драйверы-минифильтры могли подключаться в нужном месте и в нужном порядке.

Менеджер фильтров не может подключить драйвер-минифильтр между двумя прикрепленными унаследованными фильтрами файловой системы, если между ними нет фрейма менеджера фильтров. Если драйвер-минифильтр предназначен для соединения с унаследованным фильтр-драйвером файловой системы, он может быть прикреплен под ним, в зависимости от наличия второго прикрепленного фильтр-драйвера. Драйвер-минифильтр, предназначенный для прикрепления к фильтр-драйверу, может быть прикреплен над этим фильтром файловой системы.

Всегда следует проверять совместимость унаследованных фильтр-драйверов с драйверами-минифильтрами или в другом случае заменять унаследованные фильтр-драйверы файловой системы на минифильтры.

На рисунке 3 в упрощенном виде показана схема функционирования Filter Manager при взаимодействии фильтр-драйвера с драйверами-минифильтрами посредством двух фреймов менеджера фильтров. [5]

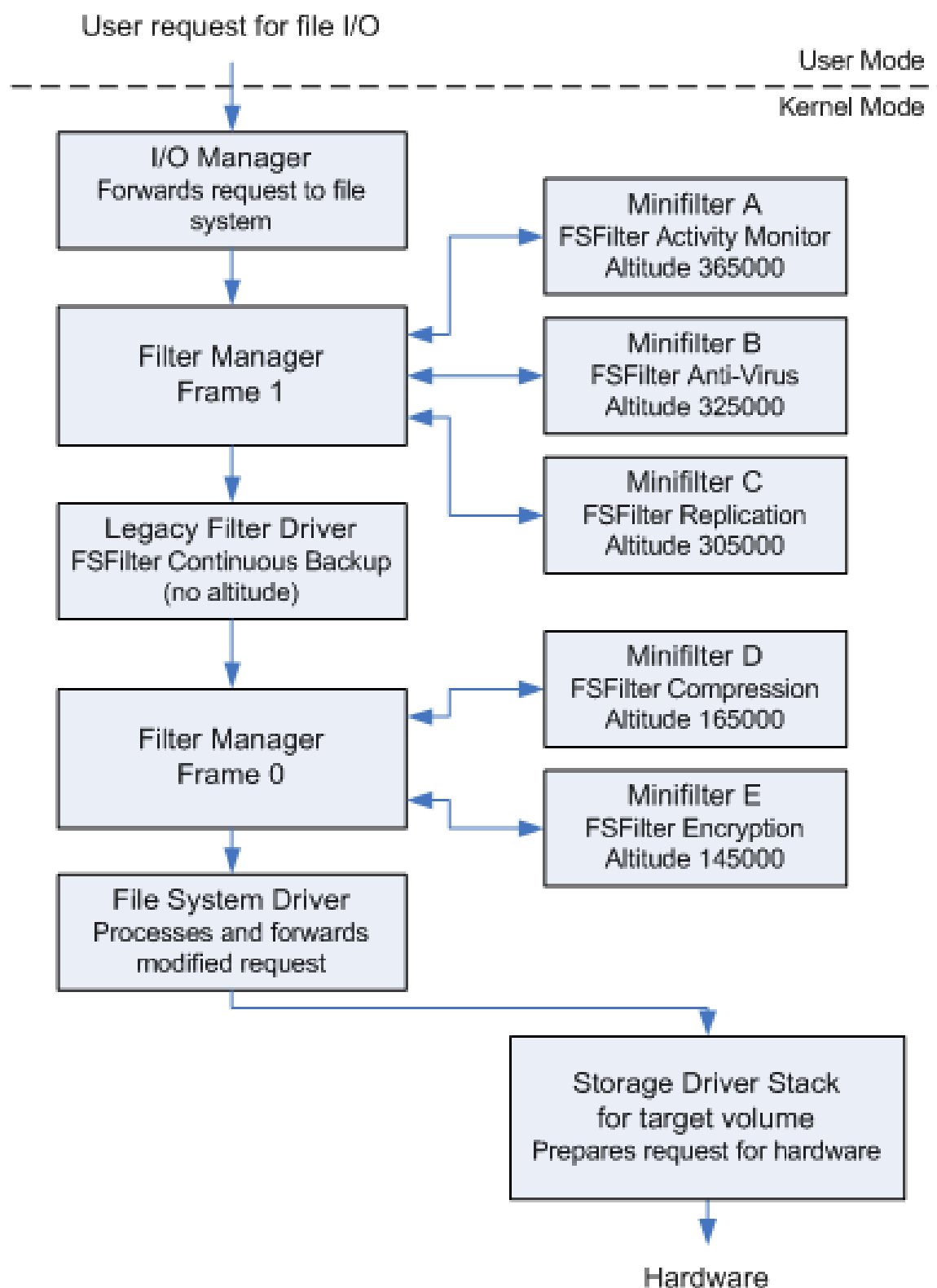


Рисунок 3 - Схема функционирования Filter Manager при взаимодействии фильтр-драйвера с драйверами-минифilterами

4 Разработка драйвера-минифильтра

Для разработки драйвера был использован комплект разработки драйверов Windows Driver Kit для Windows 10 и интегрированная среда разработки программного обеспечения Microsoft Visual Studio 2015.

Так как даже небольшая ошибка в коде драйвера может привести к нестабильности операционной системы и «синему экрану смерти», для запуска разработанного драйвера была использована виртуальная машина, созданная с помощью приложения VMware Workstation, с установленной на ней операционной системой Windows 10.

Сначала была объявлена глобальная переменная DriverObject, которая понадобится позднее. Точка входа необходима в любом драйвере. Ниже приведен код входной точки драйвера.

```
////////////////////////////////////
// Global data

DRIVER_OBJECT    g_fs_filter_driver_object=NULL;

////////////////////////////////////
// driver_entry - Точка входа драйвера

NTSTATUS driver_entry(
    __inout DRIVER_OBJECT  driver_object,
    __in    UNICODE_STRING registry_path
)
{
    STATUS Status_=STATUS_SUCCESS;
    ULONG    i      =0;

    //ASSERT(FALSE);
    //
```

```

// Сохраняем наш объект драйвера.
//

g_fs_filterDriverObject=driver_object;
...
}

```

Следующим шагом было заполнение таблицы отправки IRP-запросов указателями на функции для обработчиков IRP-запросов. В драйвере-филт্রে объявлен общий IRP-обработчик, который отправляет запрос следующему драйверу. Также был написан обработчик для IRP-запроса IRP_MJ_CREATE. Данный запрос посылается при каждом создании файла или каталога, а также при открытии файлов. Обработчик этого запроса будет извлекать имена открывающихся файлов. Реализация обработчиков IRP-запросов описана ниже.

```

////////////////////////////////////
// driver_entry - Точка входа драйвера

NTSTATUS driver_entry(
    __inout DRIVER_OBJECT driver_object,
    __in    UNICODE_STRING registry_path
)
{
    ...

    for (i=0; i <= IRP_MJ_MAXIMUM_FUNCTION; i++)
    {
        driver_object                                     ->
MajorFunction[i]=fs_filter_dispatch_pass_through;
    }

    driver_object                                     ->

```

```
MajorFunction[IRP_MJ_CREATE]=fs_filter_dispatch_create;
    ...
}
```

Драйвер фильтра файловой системы должен иметь таблицу быстрого ввода-вывода. Если не настроить таблицу быстрого ввода-вывода, это приведет к сбою системы. Быстрый ввод-вывод – это альтернативный способ инициировать операции ввода-вывода. Данный способ быстрее, чем IRP-запросы. Операции быстрого ввода-вывода всегда синхронны. Если обработчик быстрого ввода-вывода возвращает FALSE, то быстрый ввод-вывод невозможен, и будет создан IRP-запрос.

```
////////////////////////////////////
// Global data

FAST_IO_DISPATCH g_fast_io_dispatch =
{
    sizeof(FAST_IO_DISPATCH),
    fs_filter_fast_io_check_if_possible,
    ...
};
////////////////////////////////////
// driver_entry - Точка входа драйвера

NTSTATUS driver_entry(
    __inout DRIVER_OBJECT driver_object,
    __in    UNICODE_STRING registry_path
)
{
    ...
    //
    // Set fast-io dispatch table.
```

```

//

driver_object                                ->
fast_io_dispatch=&g_fast_io_dispatch;

...

}

```

Также должно отслеживаться, активирована ли файловая система, чтобы выполнить прикреплeние/отключение мини-филътра файловой системы. Регистрация уведомления об изменениях файловой системы показано ниже.

```

////////////////////////////////////
// driver_entry - Точка входа драйвера

STATUS driver_entry(
    __inout DRIVER_OBJECT driver_object,
    __in    UNICODE_STRING registry_path
)
{
    ...

status_ = io_register_fs_registration_change(driver_object,
    fs_filter_notification_callback);
    if (!SUCCESS(status_))
    {
        return status_;
    }
    ...
}

```

Установка разгрузки драйверов.

Далее для полной инициализации драйвера была установлена процедура выгрузки драйвера из системы. Установка процедуры выгрузки драйверов даёт возможность загружать и выгружать драйвер мини-фильтр несколько раз без перезагрузки операционной системы.

```
////////////////////////////////////  
// driver_entry - Точка входа драйвера  
  
NTSTATUS driver_entry(  
    __inout DRIVER_OBJECT driver_object,  
    __in    UNICODE_STRING registry_path  
)  
{  
    ...  
    driver_object->driver_unload=fs_filter_unload;  
  
    return STATUS_SUCCESS;  
}
```

Процедура выгрузки драйверов отвечает за очистку памяти и освобождение ресурсов. Далее была отменена регистрация изменений файловой системы.

```
////////////////////////////////////  
// Unload routine  
  
VOID fs_filter_unload(  
    __in DRIVER_OBJECT driver_object  
)  
{  
    ...  
}
```

```
io_unregister_fs_registration_change(driver_object,
fs_filter_notification_callback);
```

```
...
```

```
}
```

По завершению работы мини-фильтр драйвера каждое созданное нами устройство должно быть отсоединено и удалено.

```
////////////////////////////////////
// Unload routine
```

```
VOID fs_filter_unload(
    __in DRIVER_OBJECT driver_object
)
{
    ...

    for (;;)
    {
        io_enumerate_device_objectList(
            driver_object,
            dev_list,
            sizeof(dev_list),
            &num_devices);

        if (0 == num_devices)
        {
            break;
        }

        num_devices=min(num_devices,
```

```

RTL_NUMBER_OF(dev_list));

        for (i=0; i < num_devices; ++i)
        {

fs_filter_detach_from_device(dev_list[i]);
            ob_dereference_object(dev_list[i]);
        }

        ke_delay_execution_thread(kernel_mode, FALSE,
&interval);
    }
}

```

Обработчик IRP-запросов, описанный ниже осуществляет передачу запросов следующему драйверу-фильтру. Для этого используется объект драйвера-фильтра, который будет подключаться следующим.

```

////////////////////////////////////
// PassThrough IRP Handler

STATUS fs_filter_dispatch_pass_through(
    __in DEVICE_OBJECT device_object,
    __in IRP          irp
)
{
    FSFILTER_DEVICE_EXTENSION p_dev_ext=
        (FSFILTER_DEVICE_EXTENSION)device_object    ->
device_extension;

    io_skip_current_irp_stack_location(irp);
    return          io_call_driver(p_dev_ext        ->

```

```
attached_to_device_object, irp);
}
```

Следующий обработчик IRP-запросов вызывается при каждой операции с файлом. Обработчик извлекает имя и путь к файлу FILE_OBJECT. Затем проверяется выставлен ли флаг, отвечающий за разрешение доступа к файлу в данном каталоге. Если доступ разрешен, то вызывается описанный выше обработчик, который осуществляет передачу IRP-запроса следующему фильтру-драйверу.

```

////////////////////////////////////
// IRP_MJ_CREATE IRP Handler

STATUS fs_filter_dispatch_create(
    __in DEVICE_OBJECT device_object,
    __in IRP             irp
)
{
    FILE_OBJECT
p_file_object=io_get_current_irp_stack_location(irp)-
>file_object;

    STRING file_name=p_file_object->FileName;
    If (flag_access == true)
        return
fs_filter_dispatch_pass_through(device_object, irp);
    else if (file_name.InCatalog == false)
fs_filter_dispatch_pass_through(device_object, irp);
    else return error_access_denied;
}

```

Чтобы проверить верность таблицы быстрого ввода-вывода для нашего фильтр-драйвера, был использован вспомогательный макрос, код которого

представлен ниже.

```
#define
VALID_FAST_IO_DISPATCH_HANDLER(_fast_io_dispatch_ptr,
_field_name) \
    (((_fast_io_dispatch_ptr) != NULL) && \
    (((_fast_io_dispatch_ptr)                                ->
size_of_fast_io_dispatch) >= \
    (FIELD_OFFSET(FAST_IO_DISPATCH, _field_name) +
sizeof(void *))) && \
    ((_fast_io_dispatch_ptr)->_field_name != NULL))
```

Код обработки запросов быстрого ввода-вывода более объёмный по сравнению с кодом обработки IRP-запросов, поскольку каждая функция быстрого ввода-вывода имеет свой собственный набор параметров. Типичная функция быстрого ввода-вывода показана ниже.

```
BOOL fs_filter_fast_io_query_basic_info(
    __in FILE_OBJECT      file_object,
    __in BOOL              wait,
    __out FILE_BASIC_INFORMATION buffer,
    __out IO_STATUS_BLOCK io_status,
    __in DEVICE_OBJECT     device_object
)
{
    DEVICE_OBJECT      next_device_object=
        ((FSFILTER_DEVICE_EXTENSION)device_object    ->
device_extension) -> attached_to_device_object;
    FAST_IO_DISPATCH   fast_io_dispatch
=next_device_object -> driver_object -> fast_io_dispatch;

    if
    (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
```

```

fast_io_query_basic_info))
    {
        return (fast_io_dispatch ->
fast_io_query_basic_info)(
        file_object,
        wait,
        buffer,
        io_status,
        next_device_object);
    }

    return FALSE;
}

```

Быстрое отсоединение устройства ввода-вывода

Далее был написан обработчик запроса быстрого ввода-вывода на отсоединение устройства. Это специальный запрос быстрого ввода-вывода, который нужно обработать, но затем уже не вызываем следующий драйвер. Следует отсоединить наше фильтрующее устройство от стека устройств файловой системы и удалить наше устройство. Это было сделано с помощью следующего кода.

```

VOID fs_filter_fast_io_detach_device(
    __in DEVICE_OBJECT    source_device,
    __in DEVICE_OBJECT    target_device
)
{
    io_detach_device(target_device);
    io_delete_device(source_device);
}

```

Когда файловая система активируется, происходит присоединение драйвера-фильтра к управляющему устройству, но перед этим проверяется не

подключен ли уже данный драйвер. При деактивации файловой системы проверяется стек устройства управления файловой системой, находится наше устройство и оно отключается. Отсоединение от устройств тома файловой системы выполняется функцией `fs_filter_fast_io_detach_device`, которая была описана выше.

```
////////////////////////////////////
VOID fs_filter_notification_callback(
    __in DEVICE_OBJECT device_object,
    __in BOOL          fs_active
)
{

    if (fs_active)
    {

fs_filter_attach_to_file_system_device(device_object);
    }
    else
    {

fs_filter_detach_from_file_system_device(device_object);
    }
}
```

В файле `AttachDetach.c` содержатся вспомогательные процедуры для подключения драйвера-фильтра, отключения драйвера-фильтра и проверки того, прикреплен ли разработанный драйвер-фильтр.

Для выполнения подключения драйвера-фильтра, был создан новый объект устройства (с помощью вызова метода `io_create_device`). Затем вызывается метод `io_attach_device_to_device_stack_safe`, который содержит цикл с задержкой в случае отказа. Запрос на подключение может завершиться неудачно в том случае,

если объект устройства не завершил инициализацию. Когда установка завершена, сохраняется объект, подключенный к расширенному устройству и обнуляем флаг DO_DEVICE_INITIALIZING.

Далее был написан метод отключения драйвера-фильтра. Сначала был получен подключенный объект устройства из расширенного устройства, а затем были вызваны методы `io_detach_device` и `io_delete_device`, которые, соответственно, отключают и удаляют фильтр-драйвер из файловой системы.

```
void fs_filter_detach_from_device(
    __in DEVICE_OBJECT device_object
)
{
    FSFILTER_DEVICE_EXTENSION p_dev_ext=
        (FSFILTER_DEVICE_EXTENSION)device_object ->
device_extension;

    io_detach_device(p_dev_ext ->
attached_to_device_object);
    io_delete_device(device_object);
}
```

Далее был написан метод, проверяющий подключен ли наш фильтр-драйвер. Чтобы проверить это, следует проверить наш объект на совпадение с каждым подключенным устройством в стеке устройств, используя методы `io_get_attached_device_reference` и `io_get_lower_device_object`, и если объект будет найден метод возвращает значение TRUE. Проверить устройства на совпадение можно сравнивая объект драйвера устройства с нашим объектом-драйвером (`g_fs_filter_driver_object`).

```
////////////////////////////////////
```

```
BOOL fs_filter_is_my_device_object(
    __in DEVICE_OBJECT device_object
```

```

    )
    {
        return    device_object    ->    driver_object    ==
g_fs_filter_driver_object;
    }

```

Для получения данных с третьего кольца выполняется подключение к виртуальному устройству. Так как ключ хранится в зашифрованном виде, следует хэшировать его с помощью полученного пароля пользователя, для получения ключа шифрования. После получения ключа производится дешифрование файлов специального каталога и устанавливается новое значение флага flag_access.

```

    int i;
    RtlInitUnicodeString(&devName,
L"\\Device\\devExampleDriver"); //имя устр
    RtlInitUnicodeString(&symLinkName,
L"\\DosDevices\\slExampleDriver"); //симв. ссылка

    DriverObject->Driver_unload=OnUnload;
    //открытие устройства - OnCreate;
        DriverObject->MajorFunction[IRP_MJ_CREATE]=
OnCreate;
    //закрытие устройства - OnClose;
    DriverObject->MajorFunction[IRP_MJ_CLOSE]=OnClose;
    //запись данных в устройство - OnReadWrite;
        DriverObject-
>MajorFunction[IRP_MJ_READ]=OnReadWrite;
    //чтение ответа устройства - OnReadWrite;
        DriverObject-
>MajorFunction[IRP_MJ_WRITE]=OnReadWrite;
    //управление вводом/выводом - OnDeviceControl.

```

```

DriverObject->MajorFunction[IRP_MJ_DEVICE_CONTROL]=OnDeviceControl;

        status=IoCreateDevice(DriverObject,
sizeof(DEVICE_EXTENSION), &devName, FILE_DEVICE_UNKNOWN,
0, FALSE, &pdo);
        if(!SUCCESS(status))//Создание устройства
        {
            DbgPrint("IoCreateDevice error!");
            return status;
        }
        dx=(DEVICE_EXTENSION)pdo->DeviceExtension;
        dx->pdo=pdo;
        dx->ustrSymLinkName=symLinkName;//назначение
СИМВОЛЬНОГО ИМЕНИ
        status=IoCreateSymbolicLink(&symLinkName,
&devName);
        UNICODE_STRING buffer=dx->buffer;
        k=k.gambling(buffer);
        Decryption(k);
        flag_access=true;
        k=k.gambling(buffer);
        if(!SUCCESS(status))
        {
            IoDeleteDevice(pdo);
            return status;
        }

```

Чтобы произвести сборку драйвера нужно создать еще два файла: sources.txt и makefile.txt. Эти текстовые файлы используются встроенной утилитой для сборки драйвера. Файл sources.txt содержит параметры проекта и

имена исходных файлов.

Содержимое файла sources.txt представлено ниже:

```
TARGETNAME=FsFilter
```

```
TARGETPATH=obj
```

```
DRIVERTYPE=FS
```

```
TARGETTYPE=DRIVER
```

```
SOURCES      =
```

```
    Main.c
```

```
    FastIo.c
```

```
    AttachDetach.c
```

```
    IrpDispatch.c
```

```
    Notification.c
```

Содержимое файла makefile.txt является стандартным:

```
!include $(NTMAKEENV) \makefile.def
```

Полный исходный код фильтр-драйвера представлен в Приложении А.

5 Разработка утилиты «Мастер управления ключами»

Для разработки утилиты была использована среда разработки Microsoft Visual Studio 2017. Был создан проект для создания оконного приложения Windows на языке C#.

В соответствии с ТЗ был разработан интерфейс приложения, обеспечивающий возможность выбора пользователем каталога в файловой системе. Окно для выбора каталога должно появляться при нажатии на кнопку «Выбрать папку». Для выбора каталога используется класс FolderBrowserDialog, а именно его метод ShowDialog() и поле SelectedPath. Код метода-обработчика нажатия на кнопку «Выбрать папку» представлен ниже.

```
private void button1_C(object sender_, EventArgs e_)
{
    if(folderBrowserDialog1.ShowDialog()==DialogResult.OK
    {
        lab1.Location=new Point(10,70);
        lab1.Text=folderBrowserDialog1.SelectedPath;
    }
}
```

Также в соответствии с ТЗ в программу добавлена возможность ввода пароля пользователем. Для этого на главном окне был сгенерирован TextBox для ввода пароля. Для визуального скрытия пароля во время ввода используется поле PasswordChar класса TextBox, которое позволяет задавать символ для маскирования вводимых символов пароля. Код генерации TextBox представлен ниже.

```
//
// textB1
//
this->textB1.Font=new System.Drawing.Font("Microsoft
Sans Serif", 8.25F,
```

```

System.Drawing.FontStyle.Regulars,
System.Drawing.GraphicsUnit.Points, ((byte) (205)));
this->textB1.Location=new System.Drawing.Point(100,
159);
this->textB1.Name="textB1";
this->textB1.PasswordChar='☹';
this->textB1.Size=new System.Drawing.Size(230, 20);
this->textB1.TabIndex=2;

```

Еще одной важной функцией утилиты будет являться передача данных драйверу-минифilterу. Для передачи данных на нулевое кольцо создаётся виртуальное устройство и символьная ссылка. Передача данных производится по нажатию кнопки «Получить доступ». Код метода-обработчика и вспомогательных методов представлен ниже.

```

private void button2_Click(object sender, EventArgs e)
{
    string data=textB1.Text;

    SafeFileHandle _hdev>CreateFileR(data);
    if (_hdev.IsClosed | _hdev.IsInvalid)
    {
        MessageBox.Show("Ошибка открытия
виртуального устройства");
        return;
    }

    bool result=false;
    int bytesReturned;
    int sessionId=0;

```

```

        result=DeviceIoControl(_hdev, 0, IntPtr.Zero,
0, (IntPtr)sessionId, Marshal.SizeOf(sessionId), out
bytesReturned, IntPtr.Zero);

```

```

        if (result == false)
        {
            int
error_code=Marshal.GetLastWin32Error();
            MessageBox.Show("Result: " + result +
"error code: " + error_code);
        }
    }

```

```

internal class WinntConst
{
    // Fields
    internal static uint
FILE_ATTRIBUTE_NORMAL=0x80;
    internal static uint FILE_SHARE_READ=1;
    internal static uint GENERIC_READ=0x80000000;
    internal static uint OPEN_EXIST=3;
}

```

```

[DllImport ("kernel32.dll" CharSet=CharSet.Auto;
SetLastError=true)]
private static IntPtr CreatingFile(string
lFileName, uint dDesiredAccess, uint dShareMode, IntPtr

```

```

lSecurityAttributes, uint dCreationDisposition, uint
dFlagsAndAttributes, IntPtr TemplateFilePtr);

    public static SafeFileHandle CreateFileR(string
device)
    {
        string str=device.EndsWith(@"\") ?
device.Substring(0, device.Length - 1) : device;
        return new SafeFileHandle(CreateFile(@"\\.\\"
+ str, WinntConst.GENERIC_READ,
WinntConst.FILE_SHARE_READ, IntPtr.Zero,
WinntConst.OPEN_EXISTING,
WinntConst.FILE_ATTRIBUTE_NORMAL, IntPtr.Zero), true);
    }

    [return: MarshalAs(UnmanagedType.Bool)]
    [DllImport ("kernel32.dll"
Charset=Charset.Auto; SetLastError=true)]
    private static bool DeviceIoControl([In]
SafeFileHandle hDevice,
        [In] int dwIoControlCode, [In] IntPtr
lpInBuffer,
        [In] int nInBufferSize, [Out] IntPtr
lpOutBuffer,
        [In] int nOutBufferSize, out int
lpBytesReturned,
        [In] IntPtr lpOverlapped);

```

```

[DllImport ("kernel32.dll",
EntryPoint="DeviceIoControl" CharSet=CharSet.Auto;
SetLastError=true)]
private static bool DeviceIoControlAlt(
    Microsoft.Win32.SafeHandles.SafeFileHandle
hDevice,
    IOControlCode IoControlCode,
    [MarshalAs (UnmanagedType.AsAny) ] [In] object
InBuffer,
    uint nInBufferSize,
    [MarshalAs (UnmanagedType.AsAny) ] [Out] object
OutBuffer,
    uint nOutBufferSize,
    ref uint pBytesReturned,
    [In] ref System.Threading.NativeOverlapped
Overlapped
    );

```

Полный исходный код утилиты «Мастер управления ключами» представлен в Приложении Б.

6 Разработка модели угроз для программного средства

При разработке модели угроз учитывалось, что программное средство написано для информационной системы, являющейся автономным автоматизированным рабочим местом.

Также учитывались следующие исходные данные, относящиеся к информационной системе:

- система имеет прикладные программы, повышающие защищенность системы;
- система не взаимодействует с внешними системами;
- система подключена к сети;
- система не подключена к сетям общего пользования;
- компоненты системы размещены в пределах одной контролируемой зоны;
- многопользовательский режим обработки информации в системе;
- права доступа к системе разграничиваются;
- функции по управлению информационной системой не разделены;
- информационная система не сегментирована.

Затем были выявлены источники угроз и для каждого из источников определен потенциал нарушителя и его мотивация. Источники угроз, потенциал нарушителя и мотивация нарушителя представлены в таблице 1.

Таблица 1 – Источники угроз

Источники угроз	Потенциал нарушителя	Мотивация
Внешние субъекты (физические лица)	Низкий	Идеологические или политические мотивы, причинение имущественного ущерба путем мошенничества или иным преступным путем,

Продолжение таблицы 1

		любопытство или желание самореализации (подтверждение статуса), выявление уязвимостей с целью их дальнейшей продажи и получения финансовой выгоды.
Административный, обслуживающий персонал, охрана	Низкий	Причинение имущественного ущерба путем обмана или злоупотребления доверием, непреднамеренные, неосторожные или неквалифицированные действия.
Пользователи информационной системы	Низкий	Причинение имущественного ущерба путем мошенничества или иным преступным путем, любопытство или желание самореализации (подтверждения статуса), месть за ранее совершённые действия, непреднамеренные, неосторожные или неквалифицированные действия.

Затем была проведена оценка уровня ущерба. Был оценён уровень ущерба от возможного нарушения конфиденциальности, целостности и доступности информации. Результаты приведены в таблице 2.

Таблица 2 – Оценка уровня ущерба

Уровень ущерба от возможного нарушения конфиденциальности информации	Низкий
Уровень ущерба от возможного нарушения целостности информации	Низкий
Уровень ущерба от возможного нарушения доступности информации	Низкий

Были рассмотрены все угрозы из банка данных угроз безопасности информации ФСТЭК России. Для каждой из угроз было принято решение о том, следует ли её рассматривать для данной информационной системы. В таблице 3 приведены все угрозы банка данных угроз безопасности информации ФСТЭК России, а также приводится обоснование для каждой угрозы, которую решено не рассматривать.

Таблица 3 – Угрозы безопасности информации

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Актуальность (УБИ _j ^A)	Обоснование
1	Угроза автоматического распространения вредоносного кода в грид-системе	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Ресурсные центры грид-системы	не рассматривается	Данная технология не применяется в ИС
2	Угроза агрегирования данных, передаваемых в грид-системе	Внешний нарушитель со средним потенциалом	Сетевой трафик	не рассматривается	Данная технология не применяется в ИС

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
3	Угроза анализа криптографических алгоритмов и их реализации	Внешний нарушитель со средним потенциалом	Метаданные, системное программное обеспечение	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
4	Угроза аппаратного сброса пароля BIOS	Внутренний нарушитель с низким потенциалом	Микропрограммное и аппаратное обеспечение BIOS/UEFI	актуальная	
5	Угроза внедрения вредоносного кода в BIOS	Внутренний нарушитель с высоким потенциалом	Микропрограммное и аппаратное обеспечение BIOS/UEFI	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
6	Угроза внедрения кода или данных	Внешний нарушитель с низким потенциалом	Системное программное обеспечение, прикладное программное обеспечение	актуальная	
7	Угроза воздействия на программы с высокими привилегиями	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Информационная система, виртуальная машина, сетевое программное обеспечение, сетевой трафик	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
8	Угроза восстановления аутентификационной информации	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, Микропрограммное обеспечение, учётные данные пользователя	актуальная	

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ ₁ ^A)	Обосно- вание
9	Угроза восстановления предыдущей уязвимой версии BIOS	Внутренний нарушитель с низким потенциалом	Микропрограммное обеспечение BIOS/UEFI	акту- альная	
10	Угроза выхода процесса за пределы виртуальной машины	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Информацион- ная система, сетевой узел, носитель информации, объекты файловой системы, учётные данные пользователя, образ виртуальной машины	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
11	Угроза деавторизации санкционированного клиента беспроводной сети	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Сетевой узел	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
12	Угроза деструктивного изменения конфигурации/среды окружения программ	Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение, Микропрограммное обеспечение, метаданные, объекты файловой системы, реестр	акту- альная	
13	Угроза деструктивного использования декларированного функционала BIOS	Внутренний нарушитель с низким потенциалом	Микропрограммное обеспечение BIOS/UEFI	акту- альная	

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
14	Угроза длительного удержания вычислительных ресурсов пользователями	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Информационная система, сетевой узел, носитель информации, системное программное обеспечение, сетевое программное обеспечение, сетевой трафик	акту- альная	
15	Угроза доступа к защищаемым файлам с использованием обходного пути	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Объекты файловой системы	акту- альная	
16	Угроза доступа к локальным файлам сервера при помощи URL	Внешний нарушитель со средним потенциалом	Сетевое программное обеспечение	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
17	Угроза доступа/перехвата/изменения HTTP cookies	Внешний нарушитель с низким потенциалом	Прикладное программное обеспечение, сетевое программное обеспечение	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
18	Угроза загрузки нештатной операционной системы	Внутренний нарушитель с низким потенциалом	Микропрог- раммное обеспечение BIOS/UEFI	акту- альная	
19	Угроза заражения DNS-кеша	Внешний нарушитель с низким потенциалом	Сетевой узел, сетевое программное обеспечение, сетевой трафик	акту- альная	
20	Угроза злоупотребления возможностями, предоставленными потребителям облачных услуг	Внутренний нарушитель с низким потенциалом	Облачная система, виртуальная машина	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
21	Угроза злоупотребления доверием потребителей облачных услуг	Внешний нарушитель с низким потенциалом	Облачная система	не рассматривается	Данная технология не применяется в ИС
22	Угроза избыточного выделения оперативной памяти	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Аппаратное обеспечение, системное программное обеспечение, сетевое программное обеспечение	акту- альная	
23	Угроза изменения компонентов системы	Внутренний нарушитель с низким потенциалом	Информацион- ная система, сервер, рабочая станция, виртуальная машина, системное программное обеспечение, прикладное программное обеспечение, аппаратное обеспечение	акту- альная	
24	Угроза изменения режимов работы аппаратных элементов компьютера	Внутренний нарушитель с высоким потенциалом	Микропрог- раммное и аппаратное обеспечение BIOS/UEFI	не рассматривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы
25	Угроза изменения системных и глобальных переменных	Внутренний нарушитель со средним потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение	не рассматривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
26	Угроза искажения XML-схемы	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Сетевой узел, сетевое программное обеспечение, сетевой трафик	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
27	Угроза искажения вводимой и выводимой на периферийные устройства информации	Внешний нарушитель с высоким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение, аппаратное обеспечение	акту- альная	
28	Угроза использования альтернативных путей доступа к ресурсам	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Сетевой узел, объекты файловой системы, прикладное программное обеспечение, системное программное обеспечение	акту- альная	
29	Угроза использования вычислительных ресурсов суперкомпьютера «паразитными» процессами	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Вычислительны е узлы суперкомпьютер а	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
30	Угроза использования информации идентификации/аутентификации, заданной по умолчанию	Внешний нарушитель со средним потенциалом, Внутренний нарушитель с низким потенциалом	Средства защиты информации, системное программное обеспечение, сетевое программное обеспечение, Микропрог- раммное обеспечение.	акту- альная	

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^A)	Обосно- вание
31	Угроза использования механизмов авторизации для повышения привилегий	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение	акту- альная	
32	Угроза использования поддельных цифровых подписей BIOS	Внешний нарушитель со средним потенциалом	Микропрог- раммное и аппаратное обеспечение BIOS/UEFI	не рассмат- ривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы
33	Угроза использования слабостей кодирования входных данных	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение, Микропрог- раммное обеспечение, реестр	не рассмат- ривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы
34	Угроза использования слабостей протоколов сетевого/локальног о обмена данными	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, сетевое программное обеспечение, сетевой трафик	акту- альная	
35	Угроза использования слабых криптографических алгоритмов BIOS	Внешний нарушитель с высоким потенциалом	Микропрог- раммное обеспечение BIOS/UEFI	не рассмат- ривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
36	Угроза исследования механизмов работы программы	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение, Микропрог- раммное обеспечение	не рассмат- ривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы
37	Угроза исследования приложения через отчёты об ошибках	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение, Микропрог- раммное обеспечение	не рассмат- ривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы
38	Угроза истощения вычислительных ресурсов хранилища больших данных	Внутренний нарушитель с низким потенциалом	Информационна я система	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
39	Угроза истощения запаса ключей, необходимых для обновления BIOS	Внешний нарушитель со средним потенциалом	Микропрог- раммное обеспечение BIOS/UEFI	не рассмат- ривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы
40	Угроза конфликта юрисдикций различных стран	Внешний нарушитель с низким потенциалом	Облачная система	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
41	Угроза межсайтового скриптинга	Внешний нарушитель с низким потенциалом	Сетевой узел, сетевое программное обеспечение	не рассматривается	Данная технология не применяется в ИС
42	Угроза межсайтовой подделки запроса	Внешний нарушитель со средним потенциалом	Сетевой узел, сетевое программное обеспечение	не рассматривается	Данная технология не применяется в ИС
43	Угроза нарушения доступности облачного сервера	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Облачная система, облачный сервер	не рассматривается	Данная технология не применяется в ИС
44	Угроза нарушения изоляции пользовательских данных внутри виртуальной машины	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Виртуальная машина, гипервизор	не рассматривается	Данная технология не применяется в ИС
45	Угроза нарушения изоляции среды исполнения BIOS	Внутренний нарушитель с низким потенциалом	Микропрограммное и аппаратное обеспечение BIOS/UEFI	акту- альная	
46	Угроза нарушения процедуры аутентификации субъектов виртуального информационного взаимодействия	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Сетевой узел, сетевое программное обеспечение, метаданные, учётные данные пользователя	не рассматривается	Данная технология не применяется в ИС
47	Угроза нарушения работоспособности грид-системы при нетипичной сетевой нагрузке	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Грид-система, сетевой трафик	не рассматривается	Данная технология не применяется в ИС

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
48	Угроза нарушения технологии обработки информации путём несанкционированного внесения изменений в образы виртуальных машин	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Образ виртуальной машины, сетевой узел, сетевое программное обеспечение, виртуальная машина	не рассматривается	Данная технология не применяется в ИС
49	Угроза нарушения целостности данных кеша	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Сетевое программное обеспечение	акту- альная	
50	Угроза неверного определения формата входных данных, поступающих в хранилище больших данных	Внутренний нарушитель с низким потенциалом	Хранилище больших данных, метаданные	не рассматривается	Данная технология не применяется в ИС
51	Угроза невозможности восстановления сессии работы на ПЭВМ при выводе из промежуточных состояний питания	Внутренний нарушитель с низким потенциалом	Рабочая станция, носитель информации, системное программное обеспечение, метаданные, объекты файловой системы, реестр	акту- альная	
52	Угроза невозможности миграции образов виртуальных машин из-за несовместимости аппаратного и программного обеспечения	Внешний нарушитель с низким потенциалом	Облачная инфраструктура, виртуальная машина, аппаратное обеспечение, системное программное обеспечение	не рассматривается	Данная технология не применяется в ИС
53	Угроза невозможности управления правами пользователей BIOS	Внутренний нарушитель с низким потенциалом	Микропрограммное обеспечение BIOS/UEFI	акту- альная	

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
54	Угроза недобросовестного исполнения обязательств поставщиками облачных услуг	Внешний нарушитель с низким потенциалом	Информационная система, сервер, носитель информации, метаданные, объекты файловой системы	не рассматривается	Данная технология не применяется в ИС
55	Угроза незащищённого администрирования облачных услуг	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Облачная система, рабочая станция, сетевое программное обеспечение	не рассматривается	Данная технология не применяется в ИС
56	Угроза некачественного переноса инфраструктуры в облако	Внешний нарушитель с низким потенциалом	Информационная система, иммигрированная в облако, облачная система	не рассматривается	Данная технология не применяется в ИС
57	Угроза неконтролируемого копирования данных внутри хранилища больших данных	Внутренний нарушитель с низким потенциалом	Хранилище больших данных, метаданные, защищаемые данные	не рассматривается	Данная технология не применяется в ИС
58	Угроза неконтролируемого роста числа виртуальных машин	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Облачная система, консоль управления облачной инфраструктурой, облачная инфраструктура	не рассматривается	Данная технология не применяется в ИС
59	Угроза неконтролируемого роста числа зарезервированных вычислительных ресурсов	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Информационная система, сервер	не рассматривается	Данная технология не применяется в ИС
60	Угроза неконтролируемого уничтожения информации хранилищем больших данных	Внутренний нарушитель с низким потенциалом	Хранилище больших данных, метаданные, защищаемые данные	не рассматривается	Данная технология не применяется в ИС

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
61	Угроза некорректного задания структуры данных транзакции	Внутренний нарушитель со средним потенциалом	Сетевой трафик, база данных, сетевое программное обеспечение	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
62	Угроза некорректного использования прозрачного прокси-сервера за счёт плагинов браузера	Внешний нарушитель с низким потенциалом	Сетевое программное обеспечение	не рассматривается	Данная технология не применяется в ИС
63	Угроза некорректного использования функционала программного обеспечения	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение, Микропрограммное обеспечение, аппаратное обеспечение	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
64	Угроза некорректной реализации политики лицензирования в облаке	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение	не рассматривается	Данная технология не применяется в ИС
65	Угроза неопределённости в распределении ответственности между ролями в облаке	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение	не рассматривается	Данная технология не применяется в ИС

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^A)	Обосно- вание
66	Угроза неопределённости ответственности за обеспечение безопасности облака	Внешний нарушитель с низким потенциалом	Облачная система	не рассматривается	Данная технология не применяется в ИС
67	Угроза неправомерного ознакомления с защищаемой информацией	Внутренний нарушитель с низким потенциалом	Аппаратное обеспечение, носители информации, объекты файловой системы	акту- альная	
68	Угроза неправомерного/некорректного использования интерфейса взаимодействия с приложением	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение, Микропрограммное обеспечение, реестр	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
69	Угроза неправомерных действий в каналах связи	Внешний нарушитель с низким потенциалом	Сетевой трафик	акту- альная	
70	Угроза непрерывной модернизации облачной инфраструктуры	Внутренний нарушитель со средним потенциалом	Облачная инфраструктура	не рассматривается	Данная технология не применяется в ИС
71	Угроза несанкционированного восстановления удалённой защищаемой информации	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Машинный носитель информации	акту- альная	
72	Угроза несанкционированного выключения или обхода механизма защиты от записи в BIOS	Внутренний нарушитель с низким потенциалом	Микропрограммное и аппаратное обеспечение BIOS/UEFI	акту- альная	

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
73	Угроза несанкционированного доступа к активному и (или) пассивному виртуальному и (или) физическому сетевому оборудованию из физической и (или) виртуальной сети	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Сетевое оборудование, Микропрограммное обеспечение, сетевое программное обеспечение, виртуальные устройства	не рассматривается	Данная технология не применяется в ИС
74	Угроза несанкционированного доступа к аутентификационной информации	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, объекты файловой системы, учётные данные пользователя, реестр, машинные носители информации	акту- альная	
75	Угроза несанкционированного доступа к виртуальным каналам передачи	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Сетевое программное обеспечение, сетевой трафик, виртуальные устройства	не рассматривается	Данная технология не применяется в ИС
76	Угроза несанкционированного доступа к гипервизору из виртуальной машины и (или) физической сети	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Гипервизор	не рассматривается	Данная технология не применяется в ИС
77	Угроза несанкционированного доступа к данным за пределами зарезервированного адресного пространства	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Сервер, рабочая станция, виртуальная машина, гипервизор, машинный носитель информации, метаданные	не рассматривается	Данная технология не применяется в ИС

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^A)	Обосно- вание
78	Угроза несанкционированного доступа к защищаемым виртуальным машинам из виртуальной и (или) физической сети	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Виртуальная машина	не рассматривается	Данная технология не применяется в ИС
79	Угроза несанкционированного доступа к защищаемым виртуальным машинам со стороны других виртуальных машин	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Виртуальная машина	не рассматривается	Данная технология не применяется в ИС
80	Угроза несанкционированного доступа к защищаемым виртуальным устройствам из виртуальной и (или) физической сети	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Виртуальные устройства хранения, обработки и передачи данных	не рассматривается	Данная технология не применяется в ИС
81	Угроза несанкционированного доступа к локальному компьютеру через клиента грид-системы	Внешний нарушитель со средним потенциалом	Узлы грид-системы	не рассматривается	Данная технология не применяется в ИС
82	Угроза несанкционированного доступа к сегментам вычислительного поля	Внутренний нарушитель со средним потенциалом	Вычислительный узел суперкомпьютера	не рассматривается	Данная технология не применяется в ИС
83	Угроза несанкционированного доступа к системе по беспроводным каналам	Внешний нарушитель с низким потенциалом	Сетевой узел, учётные данные пользователя, сетевой трафик, аппаратное обеспечение	не рассматривается	Данная технология не применяется в ИС
84	Угроза несанкционированного доступа к системе хранения данных из виртуальной сети	Внешний нарушитель с низким потенциалом, Внутренний нарушитель	Виртуальные устройства хранения данных, виртуальные диски	не рассматривается	Данная технология не применяется в ИС

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ ₁ ^A)	Обосно- вание
85	Угроза несанкционированного доступа к хранимой в виртуальном пространстве защищаемой информации	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Носитель информации, объекты файловой системы	не рассматривается	Данная технология не применяется в ИС
86	Угроза несанкционированного изменения аутентификационной информации	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, объекты файловой системы, учётные данные пользователя, реестр	акту- альная	
87	Угроза несанкционированного использования привилегированных функций BIOS	Внешний нарушитель с высоким потенциалом, Внутренний нарушитель с низким потенциалом	Аппаратное обеспечение, Микропрограммное обеспечение BIOS/UEFI	акту- альная	
88	Угроза несанкционированного копирования защищаемой информации	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Объекты файловой системы, машинный носитель информации	акту- альная	
89	Угроза несанкционированного редактирования реестра	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, использующее реестр, реестр	акту- альная	
90	Угроза несанкционированного создания учётной записи пользователя	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение	акту- альная	

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
91	Угроза несанкционированного удаления защищаемой информации	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Метаданные, объекты файловой системы, реестр	акту- альная	
92	Угроза несанкционированного удалённого внеполосного доступа к аппаратным средствам	Внешний нарушитель с высоким потенциалом	Информацион- ная система, аппаратное обеспечение	не рассмат- ривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы
93	Угроза несанкционированного управления буфером	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение	акту- альная	
94	Угроза несанкционированного управления синхронизацией и состоянием	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение, Микропрог- раммное обеспечение	не рассмат- ривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы
95	Угроза несанкционированного управления указателями	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение	не рассмат- ривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
96	Угроза несогласованности политик безопасности элементов облачной инфраструктуры	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, облачная система	не рассматривается	Данная технология не применяется в ИС
97	Угроза несогласованности правил доступа к большим данным	Внутренний нарушитель с низким потенциалом	Хранилище больших данных	не рассматривается	Данная технология не применяется в ИС
98	Угроза обнаружения открытых портов и идентификации привязанных к нему сетевых служб	Внешний нарушитель с низким потенциалом	Сетевой узел, сетевое программное обеспечение, сетевой трафик	акту- альная	
99	Угроза обнаружения хостов	Внешний нарушитель с низким потенциалом	Сетевой узел, сетевое программное обеспечение, сетевой трафик	акту- альная	
100	Угроза обхода некорректно настроенных механизмов аутентификации	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, сетевое программное обеспечение	акту- альная	
101	Угроза общедоступности облачной инфраструктуры	Внешний нарушитель со средним потенциалом	Объекты файловой системы, аппаратное обеспечение, облачный сервер	не рассматривается	Данная технология не применяется в ИС
102	Угроза опосредованного управления группой программ через совместно используемые данные	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^A)	Обосно- вание
103	Угроза определения типов объектов защиты	Внешний нарушитель с низким потенциалом	Сетевой узел, сетевое программное обеспечение, сетевой трафик	акту- альная	
104	Угроза определения топологии вычислительной сети	Внешний нарушитель с низким потенциалом	Сетевой узел, сетевое программное обеспечение, сетевой трафик	акту- альная	
105	Угроза отказа в загрузке входных данных неизвестного формата хранилищем больших данных	Внутренний нарушитель с низким потенциалом	Хранилище больших данных, метаданные	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
106	Угроза отказа в обслуживании системой хранения данных суперкомпьютера	Внутренний нарушитель с низким потенциалом	Система хранения данных суперкомпьютера	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
107	Угроза отключения контрольных датчиков	Внешний нарушитель с высоким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
108	Угроза ошибки обновления гипервизора	Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, гипервизор	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
109	Угроза перебора всех настроек и параметров приложения	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение, Микропрог- раммное обеспечение, реестр	не рассмат- ривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^A)	Обосно- вание
110	Угроза перегрузки грид-системы вычислительными заданиями	Внутренний нарушитель с низким потенциалом	Ресурсные центры грид- системы	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
111	Угроза передачи данных по скрытым каналам	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Сетевой узел, сетевое программное обеспечение, сетевой трафик	не рассмат- ривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы
112	Угроза передачи запрещённых команд на оборудование с числовым программным управлением	Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, прикладное программное обеспечение	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
113	Угроза перезагрузки аппаратных и программно- аппаратных средств вычислительной техники	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, аппаратное обеспечение	акту- альная	
114	Угроза переполнения целочисленных переменных	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение	не рассмат- ривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы
115	Угроза перехвата вводимой и выводимой на периферийные устройства информации	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, прикладное программное обеспечение, аппаратное обеспечение	акту- альная	

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
116	Угроза перехвата данных, передаваемых по вычислительной сети	Внешний нарушитель с низким потенциалом	Сетевой узел, сетевой трафик	акту- альная	
117	Угроза перехвата привилегированного потока	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
118	Угроза перехвата привилегированного процесса	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
119	Угроза перехвата управления гипервизором	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Системное программное обеспечение, гипервизор, консоль управления гипервизором	не рассматривается	Данная технология не применяется в ИС
120	Угроза перехвата управления средой виртуализации	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Информационная система, системное программное обеспечение	не рассматривается	Данная технология не применяется в ИС
121	Угроза повреждения системного реестра	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Объекты файловой системы, реестр	акту- альная	

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
122	Угроза повышения привилегий	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Системное программное обеспечение, сетевое программное обеспечение, информационная система	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
123	Угроза подбора пароля BIOS	Внутренний нарушитель с низким потенциалом	Микропрограммное обеспечение BIOS/UEFI	актуальная	
124	Угроза подделки записей журнала регистрации событий	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение	актуальная	
125	Угроза подключения к беспроводной сети в обход процедуры аутентификации	Внешний нарушитель с низким потенциалом	Сетевой узел, сетевое программное обеспечение	не рассматривается	Данная технология не применяется в ИС
126	Угроза подмены беспроводного клиента или точки доступа	Внешний нарушитель с низким потенциалом	Сетевой узел, сетевое программное обеспечение, аппаратное обеспечение, точка беспроводного доступа	не рассматривается	Данная технология не применяется в ИС
127	Угроза подмены действия пользователя путём обмана	Внешний нарушитель со средним потенциалом	Прикладное программное обеспечение, сетевое программное обеспечение	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
128	Угроза подмены доверенного пользователя	Внешний нарушитель с низким потенциалом	Сетевой узел, сетевое ПО	актуальная	

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^A)	Обосно- вание
129	Угроза подмены резервной копии программного обеспечения BIOS	Внутренний нарушитель с низким потенциалом	Микропро- граммное и аппаратное обеспечение BIOS/UEFI	акту- альная	
130	Угроза подмены содержимого сетевых ресурсов	Внешний нарушитель с низким потенциалом	Прикладное программное обеспечение, сетевое программное обеспечение, сетевой трафик	акту- альная	
131	Угроза подмены субъекта сетевого доступа	Внешний нарушитель со средним потенциалом	Прикладное программное обеспечение, сетевое программное обеспечение, сетевой трафик	не рассмат- ривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы
132	Угроза получения предварительной информации об объекте защиты	Внешний нарушитель со средним потенциалом	Сетевой узел, сетевое программное обеспечение, сетевой трафик, прикладное программное обеспечение	не рассмат- ривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы
133	Угроза получения сведений о владельце беспроводного устройства	Внешний нарушитель с низким потенциалом	Сетевой узел, метаданные	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
134	Угроза потери доверия к поставщику облачных услуг	Внутренний нарушитель со средним потенциалом	Объекты файловой системы, Информацион- ная система, иммигрированны я в облако	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
135	Угроза потери и утечки данных, обрабатываемых в облаке	Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, метаданные, объекты файловой системы	не рассматривается	Данная технология не применяется в ИС
136	Угроза потери информации вследствие несогласованности работы узлов хранилища больших данных	Внутренний нарушитель с низким потенциалом	Информационная система, узлы хранилища больших данных	не рассматривается	Данная технология не применяется в ИС
137	Угроза потери управления облачными ресурсами	Внешний нарушитель с высоким потенциалом	Сетевой трафик, объекты файловой системы	не рассматривается	Данная технология не применяется в ИС
138	Угроза потери управления собственной инфраструктурой при переносе её в облако	Внутренний нарушитель со средним потенциалом	Информационная система, иммигрированная в облако, системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение	не рассматривается	Данная технология не применяется в ИС
139	Угроза преодоления физической защиты	Внешний нарушитель со средним потенциалом	Сервер, рабочая станция, носитель информации, аппаратное обеспечение	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
140	Угроза приведения системы в состояние «отказ в обслуживании»	Внешний нарушитель с низким потенциалом, Внутренний нарушитель	Информационная система, сетевой узел, системное программное обеспечение	актуальная	

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
141	Угроза привязки к поставщику облачных услуг	Внутренний нарушитель с низким потенциалом	Информационная система, иммигрированная в облако, системное программное обеспечение, сетевое программное обеспечение, сетевой трафик, объекты файловой системы	не рассматривается	Данная технология не применяется в ИС
142	Угроза приостановки оказания облачных услуг вследствие технических сбоев	Внешний нарушитель с низким потенциалом	Системное программное обеспечение, аппаратное обеспечение, канал связи	не рассматривается	Данная технология не применяется в ИС
143	Угроза программного выведения из строя средств хранения, обработки и (или) ввода/вывода/передачи информации	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Носитель информации, Микропрограммное обеспечение, аппаратное обеспечение	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
144	Угроза программного сброса пароля BIOS	Внутренний нарушитель с низким потенциалом	Микропрограммное обеспечение BIOS/UEFI, системное программное обеспечение	актуальная	
145	Угроза пропуска проверки целостности программного обеспечения	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение	актуальная	

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
146	Угроза прямого обращения к памяти вычислительного поля суперкомпьютера	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Вычислительные узлы суперкомпьютера, каналы передачи данных суперкомпьютера, системное программное обеспечение	не рассматривается	Данная технология не применяется в ИС
147	Угроза распространения несанкционированного повышения прав на всю грид-систему	Внутренний нарушитель со средним потенциалом	Ресурсные центры грид-системы, узлы грид-системы, грид-система, сетевое программное обеспечение	не рассматривается	Данная технология не применяется в ИС
148	Угроза сбоя автоматического управления системой разграничения доступа хранилища больших данных	Внешний нарушитель с низким потенциалом	Информационная система, система разграничения доступа хранилища больших данных	не рассматривается	Данная технология не применяется в ИС
149	Угроза сбоя обработки специальным образом изменённых файлов	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Метаданные, объекты файловой системы, системное программное обеспечение	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
150	Угроза сбоя процесса обновления BIOS	Внутренний нарушитель со средним потенциалом	Микропрограммное и аппаратное обеспечение BIOS/UEFI, каналы связи	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
151	Угроза сканирования веб-сервисов, разработанных на основе языка описания WSDL	Внешний нарушитель с низким потенциалом	Сетевое программное обеспечение, сетевой узел	не рассматривается	Данная технология не применяется в ИС
152	Угроза удаления аутентификационной информации	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, Микропрограммное обеспечение, учётные данные пользователя	акту- альная	
153	Угроза усиления воздействия на вычислительные ресурсы пользователей при помощи сторонних серверов	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Информационная система, сетевой узел, системное программное обеспечение, сетевое программное обеспечение	акту- альная	
154	Угроза установки уязвимых версий обновления программного обеспечения BIOS	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Микропрограммное обеспечение BIOS/UEFI	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
155	Угроза утраты вычислительных ресурсов	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Информационная система, сетевой узел, носитель информации, системное программное обеспечение, сетевое программное обеспечение, сетевой трафик	акту- альная	
156	Угроза утраты носителей информации	Внутренний нарушитель с низким потенциалом	Носитель информации	акту- альная	

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ ₁ ^A)	Обосно- вание
157	Угроза физического выведения из строя средств хранения, обработки и (или) ввода/вывода/перед ачи информации	Внешний нарушитель с низким потенциалом	Сервер, рабочая станция, носитель информации, аппаратное обеспечение	акту- альная	
158	Угроза форматирования носителей информации	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Носитель информации	акту- альная	
159	Угроза «форсированного веб-браузинга»	Внешний нарушитель с низким потенциалом	Сетевой узел, сетевое программное обеспечение	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
160	Угроза хищения средств хранения, обработки и (или) ввода/вывода/перед ачи информации	Внешний нарушитель с низким потенциалом	Сервер, рабочая станция, носитель информации, аппаратное обеспечение	акту- альная	
161	Угроза чрезмерного использования вычислительных ресурсов суперкомпьютера в ходе интенсивного обмена межпроцессорными сообщениями	Внутренний нарушитель с низким потенциалом	Вычислительны е узлы суперкомпьютер а	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
162	Угроза эксплуатации цифровой подписи программного кода	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, прикладное программное обеспечение	акту- альная	

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^A)	Обосно- вание
163	Угроза перехвата исключения/сигнала из привилегированного блока функций	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Системное программное обеспечение	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
164	Угроза распространения состояния «отказ в обслуживании» в облачной инфраструктуре	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Облачная инфраструктура, созданная с использованием технологий виртуализации	не рассматривается	Данная технология не применяется в ИС
165	Угроза включения в проект не достоверно испытанных компонентов	Внутренний нарушитель со средним потенциалом	Программное обеспечение, техническое средство, Информационная система, ключевая система информационно й инфраструктуры	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
166	Угроза внедрения системной избыточности	Внутренний нарушитель со средним потенциалом	Программное обеспечение, Информационная система, ключевая система информационно й инфраструктуры	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
167	Угроза заражения компьютера при посещении неблагонадёжных сайтов	Внутренний нарушитель с низким потенциалом	Сетевой узел, сетевое программное обеспечение	не рассматривается	Угроза не актуальна, так система не подключена к Интернету

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
168	Угроза «кражи» учётной записи доступа к сетевым сервисам	Внешний нарушитель с низким потенциалом	Сетевое программное обеспечение	акту- альная	
169	Угроза наличия механизмов разработчика	Внутренний нарушитель со средним потенциалом	Программное обеспечение, техническое средство	не рассмат- ривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы
170	Угроза неправомерного шифрования информации	Внешний нарушитель с низким потенциалом	Объект файловой системы	акту- альная	
171	Угроза скрытного включения вычислительного устройства в состав бот-сети	Внешний нарушитель с низким потенциалом	Сетевой узел, сетевое программное обеспечение	не рассмат- ривается	Угроза не акту- альна, так система не подклю- чена к сети связи общего пользо- вания (Интерн ет)
172	Угроза распространения «почтовых червей»	Внешний нарушитель с низким потенциалом	Сетевое программное обеспечение	акту- альная	
173	Угроза «спама» веб-сервера	Внешний нарушитель с низким потенциалом	Сетевое программное обеспечение	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
174	Угроза «фарминга»	Внешний нарушитель с низким потенциалом	Рабочая станция, сетевое программное обеспечение, сетевой трафик	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
175	Угроза «фишинга»	Внешний нарушитель с низким потенциалом	Рабочая станция, сетевое программное обеспечение, сетевой трафик	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
176	Угроза нарушения технологического/п роизводственного процесса из-за временных задержек, вносимых средством защиты	Внешний нарушитель с низким потенциалом	Средство защиты информации	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС
177	Угроза неподтверждённого ввода данных оператором в систему, связанную с безопасностью	Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, сетевое программное обеспечение, прикладное программное обеспечение, аппаратное обеспечение	акту- альная	
178	Угроза несанкционированн ого использования системных и сетевых утилит	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение	акту- альная	
179	Угроза несанкционированн ой модификации защищаемой информации	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Объекты файловой системы	акту- альная	
180	Угроза отказа подсистемы обеспечения температурного режима	Внешний нарушитель со средним потенциалом, Внутренний нарушитель с низким потенциалом	Технические средства воздушного кондиционирова ния, включая трубопроводные системы для циркуляции охлаждённого воздуха в ЦОД	акту- альная	

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
181	Угроза перехвата одноразовых паролей в режиме реального времени	Внешний нарушитель со средним потенциалом	Сетевое программное обеспечение	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
182	Угроза физического устаревания аппаратных компонентов	Внутренний нарушитель с низким потенциалом	Аппаратное средство	акту- альная	
183	Угроза перехвата управления автоматизированной системой управления технологическими процессами	Внешний нарушитель с высоким потенциалом, Внутренний нарушитель со средним потенциалом	Программное обеспечение автоматизированной системы управления технологическими процессами	не рассматривается	Данная технология не применяется в ИС
184	Угроза агрегирования данных, обрабатываемых с помощью мобильного устройства	Внутренний нарушитель со средним потенциалом	Мобильное устройство	не рассматривается	Данная технология не применяется в ИС
185	Угроза несанкционированного изменения параметров настройки средств защиты информации	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Средство защиты информации	акту- альная	
186	Угроза внедрения вредоносного кода через рекламу, сервисы и контент	Внутренний нарушитель с низким потенциалом	Сетевое программное обеспечение	не рассматривается	Угроза не актуальна, так как система не подключена к Интернету)

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ _г ^А)	Обосно- вание
187	Угроза несанкционированного воздействия на средство защиты информации	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Средство защиты информации	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
188	Угроза подмены программного обеспечения	Внутренний нарушитель со средним потенциалом	Прикладное программное обеспечение, сетевое программное обеспечение, системное программное обеспечение	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
189	Угроза маскирования действий вредоносного кода	Внешний нарушитель со средним потенциалом	Системное программное обеспечение, сетевое программное обеспечение	не рассматривается	Потенциал нарушителя недостаточен для реализации угрозы
190	Угроза внедрения вредоносного кода за счет посещения зараженных сайтов в сети Интернет	Внешний нарушитель со средним потенциалом	Сетевое программное обеспечение	не рассматривается	Угроза не актуальна, так система не подключена к сети связи общего пользования (Интернет)

Продолжение таблицы 3

ID УБ И	Название угрозы	Источники угрозы	Объект воздействия	Акту- альность (УБИ ₁ ^A)	Обосно- вание
191	Угроза внедрения вредоносного кода в дистрибутив программного обеспечения	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Прикладное программное обеспечение, сетевое программное обеспечение, системное программное обеспечение	акту- альная	
192	Угроза использования уязвимых версий программного обеспечения	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Прикладное программное обеспечение, сетевое программное обеспечение, системное программное обеспечение	акту- альная	
193	Угроза утечки информации за счет применения вредоносным программным обеспечением алгоритмов шифрования графика	Внешний нарушитель со средним потенциалом	Информационны е ресурсы, объекты файловой системы	не рассмат- ривается	Потен- циал наруши- теля недоста- точен для реали- зации угрозы
194	Угроза несанкционированн ого использования привилегированных функций мобильного устройства	Внешний нарушитель с высоким потенциалом	Мобильное устройство	не рассмат- ривается	Данная техноло- гия не приме- няется в ИС

Модель угроз, состоящая из выбранных актуальных угроз представлена в Приложении В.

7 Тестирование разработанного программного средства

Тестирование драйверов, написанных вручную, не безопасно для операционной системы, поэтому для тестирования разработанного программного обеспечения с помощью программы VMware Workstation Pro была создана тестовая виртуальная машина.

После успешной сборки драйвера-минифilterа был получен системный файл FsFilter.sys. Этот файл был загружен на тестовую виртуальную машину.

Теперь с помощью программы Kernel-Mode Driver Manager, интерфейс которой представлен на рисунке 4, следует зарегистрировать драйвер в системе, а затем запустить его.

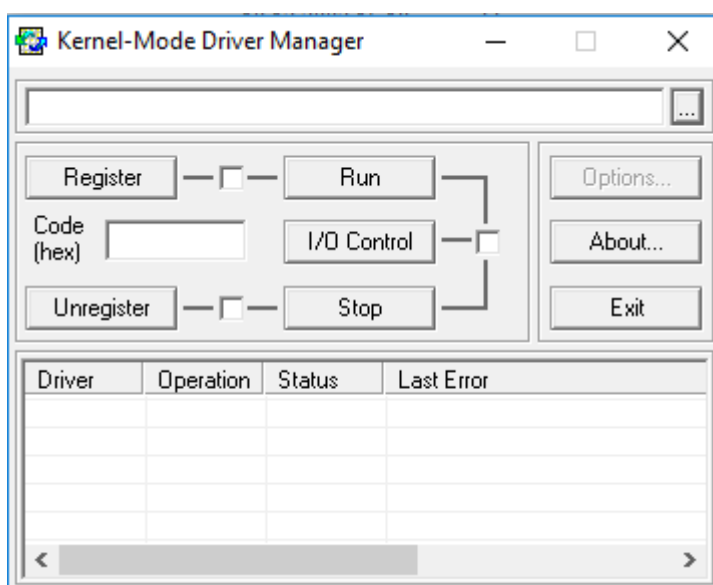


Рисунок 4 – Интерфейс программы Kernel-Mode Driver Manager

Для регистрации драйвера в системе следует выбрать системный файл FsFilter.sys и нажать на кнопку Register. В нижней части окна приложения появится имя регистрируемого драйвера, имя операции и результат операции. Если в столбце «Status» появится строка «Success», а в столбце «Last Error» строка «Операция успешно выполнена», значит регистрация драйвера была успешно произведена. Диалоговое окно выбора файла для регистрации и запуска драйвера представлено на рисунке 5. Результат успешной регистрации драйвера в системе изображен на рисунке 6.

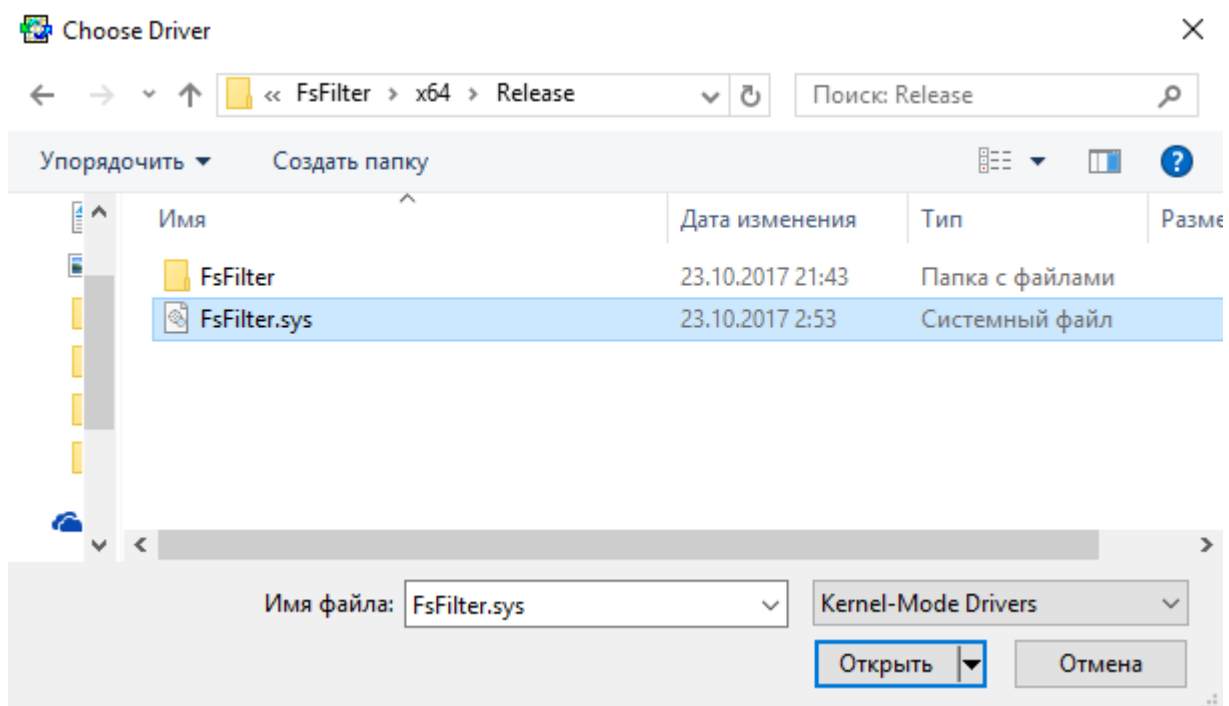


Рисунок 5 – Диалоговое окно выбора файла драйвера

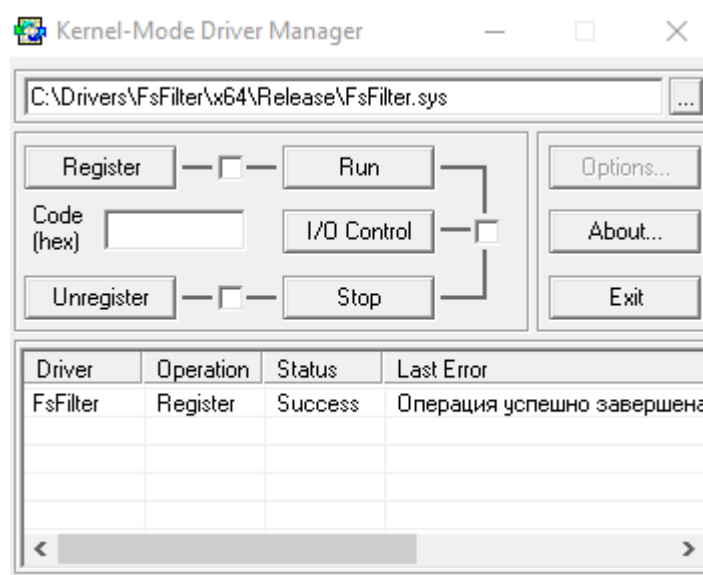


Рисунок 6 – Успешная регистрация драйвера в системе

После успешной регистрации драйвера в системе следует запустить драйвер. Для запуска драйвера нужно нажать кнопку «Run». Если после нажатия кнопки в столбце «Status» появится строка «Success», а в столбце «Last Error» строка «Операция успешно выполнена», значит запуск драйвера был успешно произведен. Результат успешного запуска драйвера изображен на рисунке 7.

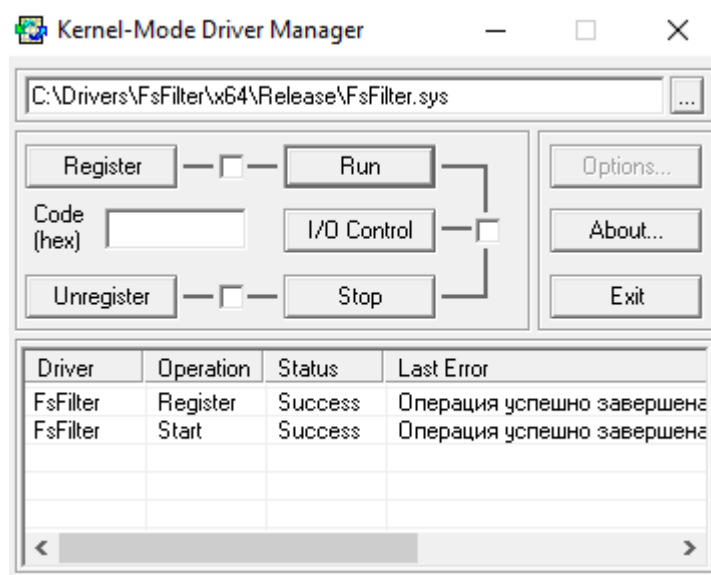


Рисунок 7 – Результат успешного запуска драйвера

Затем для проверки того, что драйвер запущен и работает в системе была использована программа DeviceTree. На рисунке 8 можно увидеть, что зарегистрированный и запущенный драйвер FsFilter появился в списке.

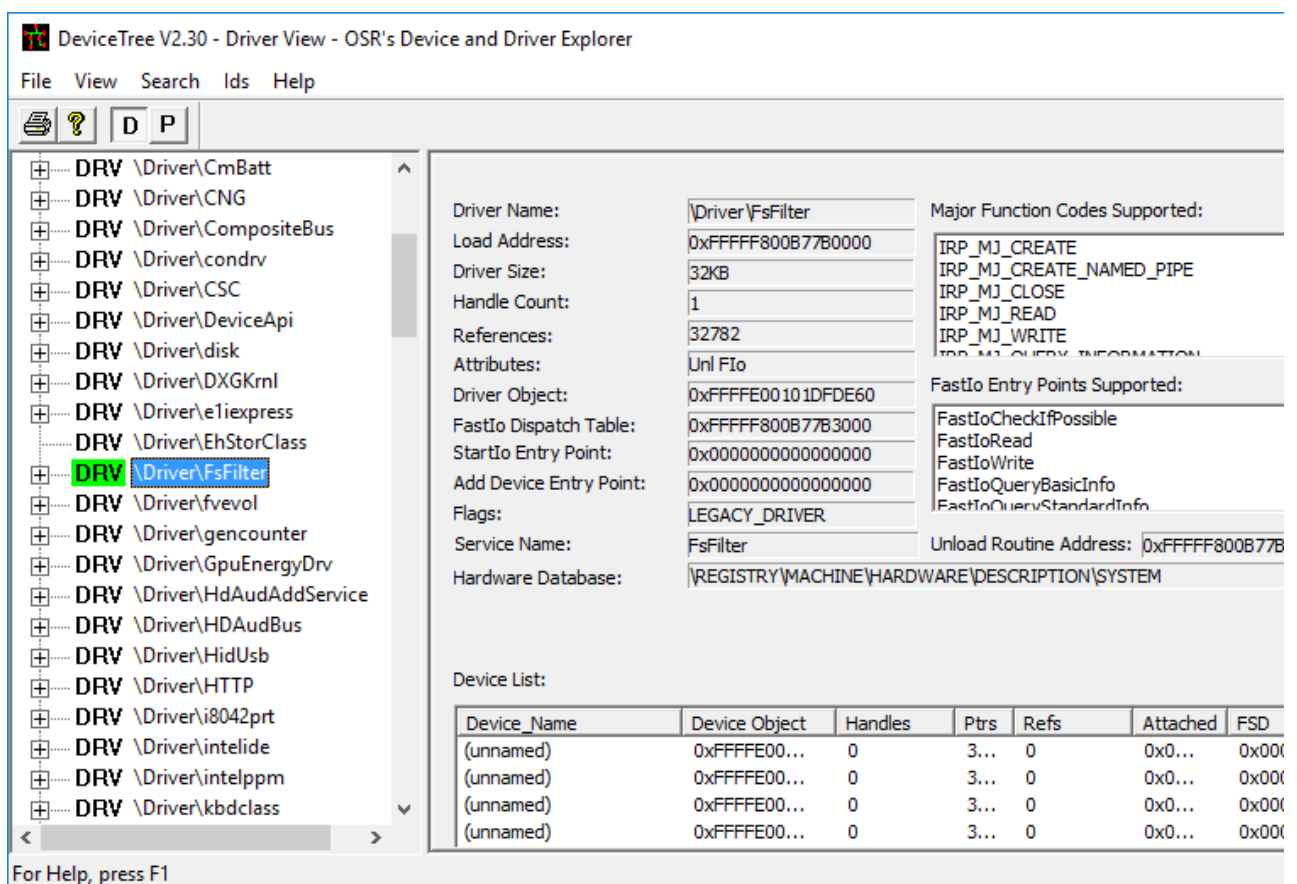


Рисунок 8 – Список запущенных драйверов в системе

Затем была проверена работа драйвера. Была проведена попытка доступа к файлу, находящемуся в защищенном каталоге. Результат попытки открытия файла, находящегося в защищенном каталоге изображен на рисунке 9.

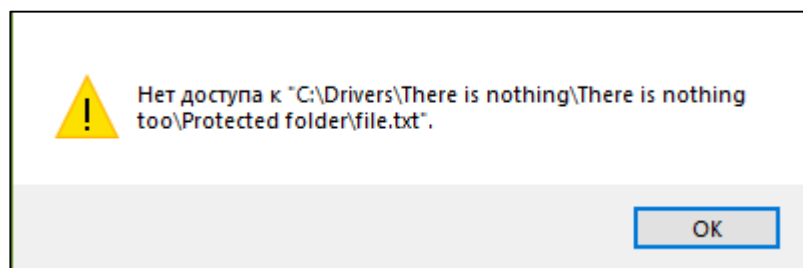


Рисунок 9 – Попытка открытия файла, находящегося в защищенном каталоге

Для получения доступа к файлам в папке следует воспользоваться разработанной утилитой «Мастер управления ключами», интерфейс которой представлен на рисунке 10.

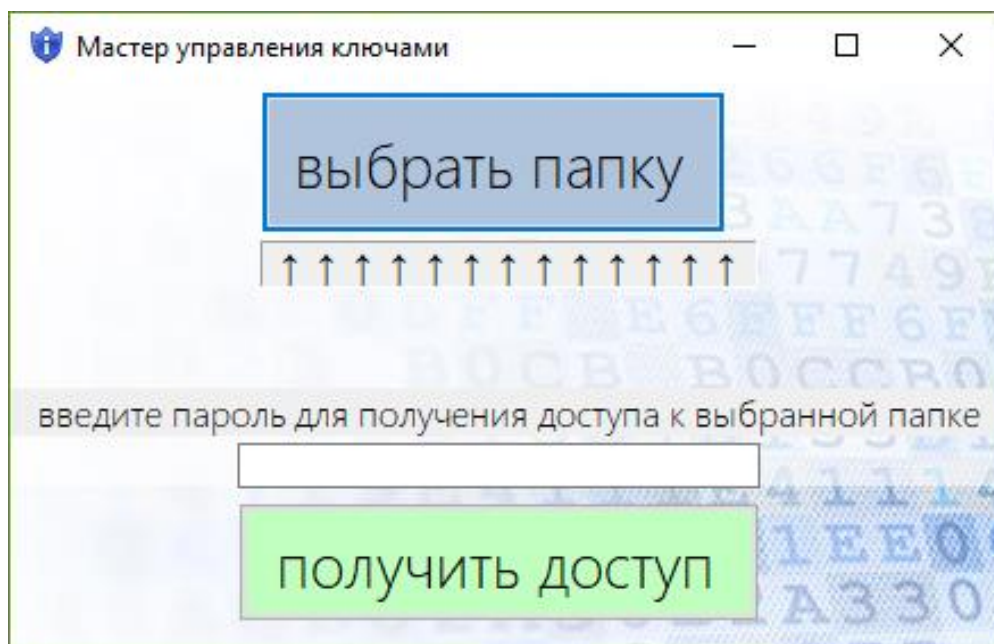


Рисунок 10 – Интерфейс утилиты «Мастер управления ключами»

После нажатия кнопки «выбрать папку» открывается стандартное диалоговое окно выбора каталога, представленное на рисунке 11. В данном окне следует выбрать нужный каталог и нажать кнопку «Ок», после чего на главном окне утилиты появится выбранная папка и путь к ней. Результат выбора каталога представлен на рисунке 12.

Далее пользователь должен ввести пароль для получения доступа к файлам выбранной папки. При вводе пароля вводимые символы будут изменяться на

маскирующие символы, как показано на рисунке 13. Это необходимо для того, чтобы избежать во время ввода пароля его просмотр другими лицами и таким образом обеспечить конфиденциальность.

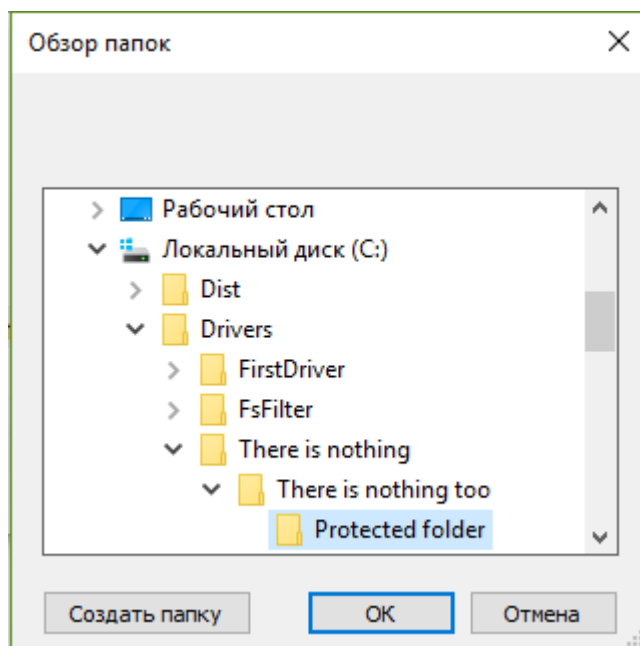


Рисунок 11 – Диалоговое окно утилиты для выбора каталога

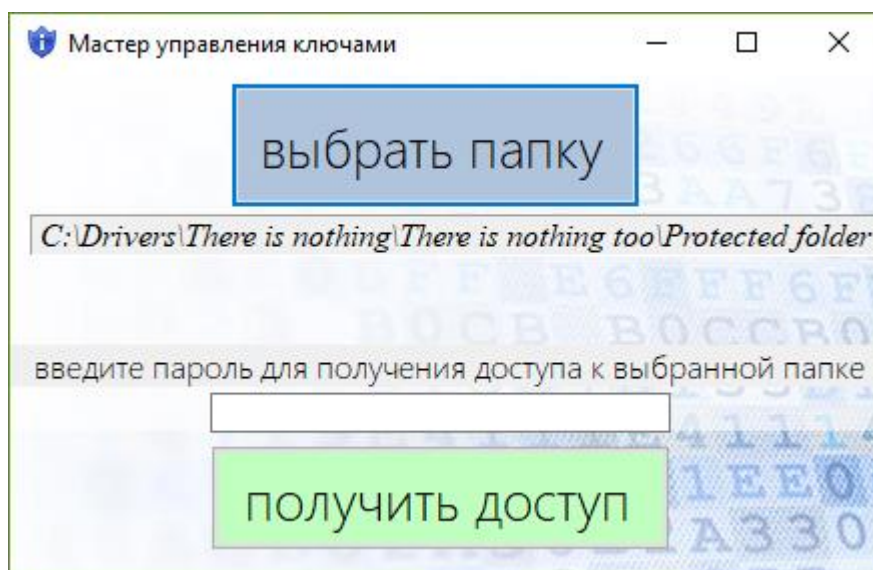


Рисунок 12 – Результат выбора каталога

После ввода пароля следует нажать кнопку «получить доступ». Если пароль и папка, к которой следует получить доступ, введены корректно, то данные с паролем передадутся драйверу на нулевое кольцо, а на главном окне утилиты выбранная папка и путь к ней подсвечиваются зелёным цветом, как показано на рисунке 14.

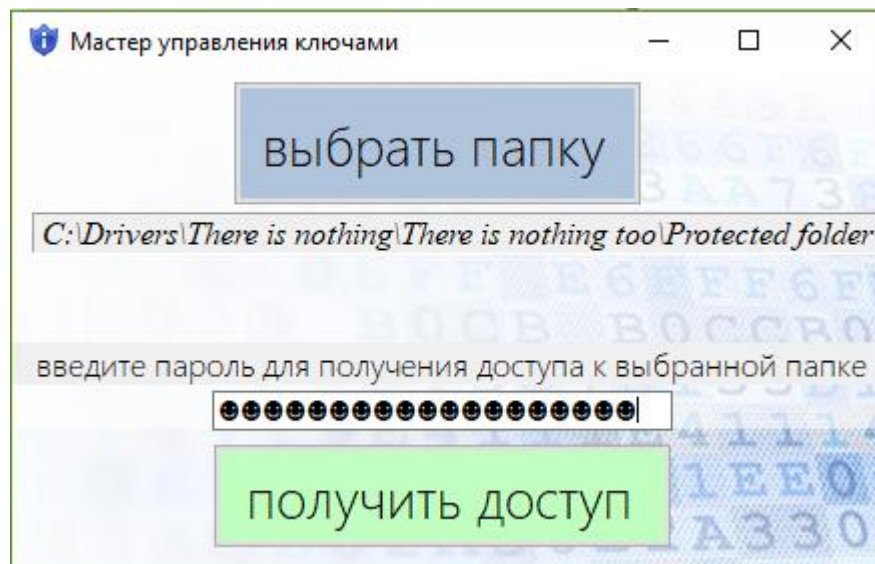


Рисунок 13 – Маскирование символов при вводе пароля

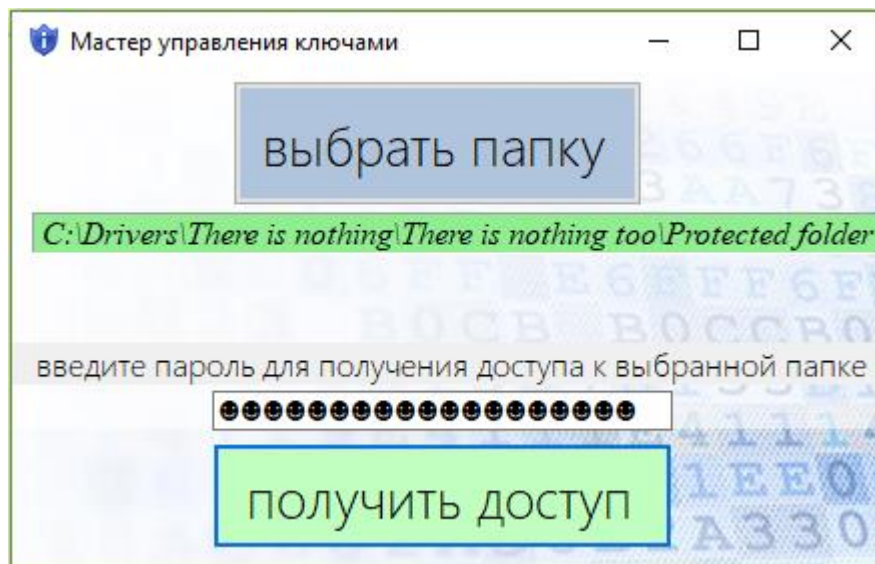


Рисунок 14 – Результат корректного выбора папки и ввода пароля

После выполнения вышеуказанных действий была выполнена ещё одна попытка открытия файла из защищенного каталога. Файл был успешно открыт, как показано на рисунке 15.

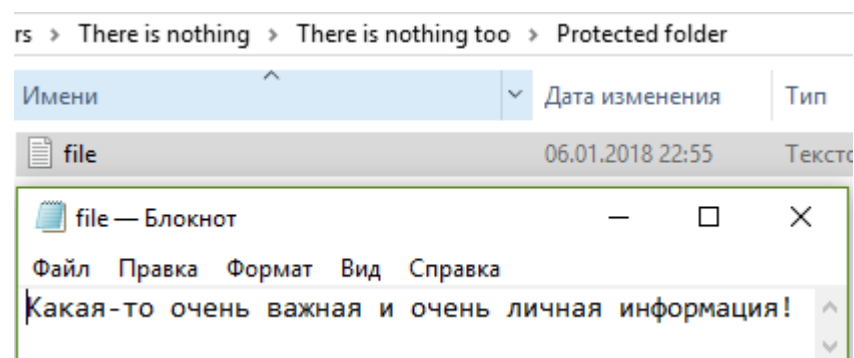


Рисунок 15 – Успешное открытие файла из защищенного каталога

8 Вредные психофизиологические факторы, влияющие на органы зрения специалиста по защите информации. Методы и средства физической культуры, снижающие их воздействие

Главным фактором, влияющим на производительность труда специалистов по защите информации, которые работают с компьютером, являются безопасные и комфортные условия для труда.

Условия для труда специалиста по защите информации определяются:

- организацией рабочего места;
- производственной средой;
- информационным взаимодействием человека и компьютера. [7]

При выполнении работ на ПЭВМ согласно ГОСТ 12.0.003-74 «ССБТ. Опасные и вредные производственные факторы. Классификация» [8] могут существовать следующие факторы:

- высокая температура поверхностей компьютера;
- высокая или низкая температура воздуха рабочей зоны;
- выделение в воздух химических веществ;
- высокая или низкая влажность воздуха;
- высокий или низкий уровень отрицательных и положительных аэроионов;
- высокое значение напряжения в электрической цепи, короткие замыкания;
- высокий уровень статического электричества;
- высокий уровень электромагнитных излучений;
- высокая напряженность электрического поля;
- недостаток или отсутствие естественного света;
- недостаточно яркая искусственная освещенность рабочей зоны;
- высокая яркость света;
- высокая контрастность;
- прямая и отраженная блёскость;

- напряжение зрительных мышц;
- монотонность процесса труда;
- нервно-эмоциональные перегрузки.

Работа на персональном компьютере сопровождается значительным и постоянным органов зрения. Важной особенностью является немного другой принцип чтения информации, который отличается от принципа при обычном чтении. При обычном чтении текст на бумаге, расположенный на горизонтальном столе, читается работником с наклоненной головой при наличии светового потока, падающего на текст. При работе за компьютером специалист по защите информации считывает текст, почти не наклоняя голову, текст появляется по иную сторону экрана, поэтому пользователь не читает отраженный текст, а смотрит именно на источник света. Это вынуждает глаза работать в несвойственном ему режиме длительное время.

Проблемы с органами зрения резко возрастают при работе более четырёх часов в день. Напряжение специалиста по защите информации при работе за компьютером возникает вследствие нехватки времени, большого объема информации. Продолжительная работа за ПК может привести к нервно-эмоциональному перенапряжению, ухудшению состояния, нарушению сна, снижению работоспособности и концентрации внимания, головной боли, депрессии.

Помимо этого, при высоких нервно-психических нагрузках в сочетании с другими факторами происходит потеря из организма витаминов и минеральных веществ. Поэтому при работе на ПК рекомендуется такая мера безопасности, как защита организма с помощью витаминно-минеральных комплексов.

Многие специалисты по защите информации, которые проводят более 8 часов за монитором, жалуются на проблемы со зрением. Им необходимы перерывы для снятия напряжения, иначе возможно развитие серьезных заболеваний. Для этого существуют разнообразные методы и средства физической культуры, которые будут снижать действие вредных факторов, влияющих на органы зрения, например специальная гимнастика для глаз. Такая

гимнастика поможет расслабить глаза и сохранить зрение.

Упражнения для глаз необходимо выполнять, когда Вы находитесь в расслабленном состоянии. Если после долгой работы появились боли в спине или шее, следует сначала выполнить гимнастику для этих зон. Для того чтобы позволить глазам хорошо отдохнуть, нужно пересест с рабочего места, следует снять очки или контактные линзы, если они имеются. Во время выполнения упражнений не следует напрягать взгляд. Нужно смотреть вдаль или просто закрыть глаза. Помимо этого следует глубоко дышать, так как при работе глазные мышцы поглощают кислород.

Чтобы уменьшить вредное влияние компьютера на глаза и сохранить хорошее зрение, необходимо придерживаться некоторых правил, которые заключаются в следующем:

- расстояние от пользователя до монитора должно составлять минимум 60 сантиметров. Если информацию на экране при этом будет видно плохо, следует поменять размер шрифта;
- монитор нужно располагать прямо перед собой. Он не должен стоять сбоку, иначе для просмотра информации в мониторе придётся постоянно поворачивать голову. Это может привести к напряжению и усталости глаз;
- необходимо правильно установить монитор. Он должен быть установлен так, чтобы угол падения взгляда на него был равен около 15 градусов. То есть верхняя часть экрана должна быть немного ниже уровня глаз;
- необходимо протирать монитор и не трогать его руками. Для этого можно воспользоваться сухими салфетками;
- следует обратить внимание на освещение в рабочей зоне. Оно не должно быть очень ярким или тусклым. Не следует работать за компьютером в темноте;
- следует установить небольшой уровень яркости монитора. Не следует увеличивать контрастность изображения;
- при работе за компьютером необходимо чаще моргать. Таким образом можно обеспечить глазам постоянное увлажнение и предотвратить высыхание роговицы;

– взрослый человек может непрерывно проводить за ПК не более 2 часов.
При этом в день специалисту по защите информации рекомендуется работать за компьютером не более 6 часов.

Заключение

В результате выполнения ВКР было разработано программное средство шифрования пользовательских данных с использованием алгоритма по ГОСТ Р 34.12-2015.

В процессе выполнения дипломного проектирования были изучены основные понятия разработки драйвера, архитектура Windows, архитектура Windows Driver Model и особенности функционирования фильтр-драйвера. Был разработан фильтр-драйвер, осуществляющий перехват запросов к файловой системе на запись и чтение, шифрование данных с использованием алгоритма по ГОСТ Р 34.12-2015. Также была разработана утилита «Мастер управления ключами», которая представляет из себя оконное приложение и дает возможность пользователю получить доступ к защищенной папке. После разработки программного средства, была составлена модель угроз. На последнем этапе разработанное программное средство было протестировано.

Таким образом, все пункты технического задания на выпускную квалификационную работу выполнены полностью.

Результаты выпускной квалификационной работы были внедрены и использованы в работе отдела №2 управления программных средств защиты информации ООО НТП «Криптософт». Акт о внедрении результатов выпускной квалификационной работы приведен в приложении Г.

Список используемых источников

- 1 Комиссарова В. Программирование драйверов для Windows – СПб.: БХВ-Петербург, 2007. – 256 с.
- 2 Архитектура операционных систем семейства Windows NT [Электронный ресурс]. Режим доступа: https://drp.su/ru/driver_dev/04_01-5.htm, свободный. – Загл. с экрана.
- 3 Драйверы. Архитектура WDM [Электронный ресурс]. Режим доступа: <https://sites.google.com/site/vladislavsajdasev/drajvery>, свободный. – Загл. с экрана.
- 4 Разработка MiniFilter драйвера. [Электронный ресурс] — <https://habrahabr.ru/post/176739>, свободный. – Загл. с экрана.
- 5 Filter Manager Concepts. [Электронный ресурс] — <https://docs.microsoft.com/ru-ru/windows-hardware/drivers/ifs/filter-manager-concepts>, свободный. – Загл. с экрана.
- 6 ГОСТ Р 34.12-2015. Информационная технология. Криптографическая защита информации. Блочные шифры [Электронный ресурс] — https://www.tc26.ru/standard/gost/GOST_R_3412-2015.pdf, свободный. – Загл. с экрана.
- 7 Упражнения для глаз при работе за компьютером [Электронный ресурс] — <https://www.horosheezrenie.ru/uprazhneniya-dlya-glaz-za-kompyuterom>, свободный. – Загл. с экрана.
- 8 ГОСТ 12.0.003-2015. Система стандартов по безопасности труда. Опасные и вредные производственные факторы. Классификация [Электронный ресурс] . — М.: Стандартинформ, 2016. - Режим доступа: <http://www.internet-law.ru/gosts/gost/62075/>. — Загл. с экрана.

Приложение А

(обязательное)

Исходный код драйвера-минифильтра

Main.c

```
#include "FsFilter.h"
```

```
////////////////////////////////////
```

```
VOID Fs_filter_unload(  
    __in DRIVER_OBJECT DriverObject  
);
```

```
////////////////////////////////////
```

```
DRIVER_OBJECT    g_fsFilterDriverObject=NULL;
```

```
FAST_IO_DISPATCH g_fast_io_dispatch =  
{  
    sizeof(FAST_IO_DISPATCH),  
    Fs_filter_fast_io_check_if_possible,  
    FsFilterFastIoRead,  
    FsFilterFastIoWrite,  
    FsFilterFastIoQueryBasicInfo,  
    FsFilterFastIoQueryStandardInfo,  
    FsFilterFastIoLock,  
    FsFilterFastIoUnlockSingle,  
    FsFilterFastIoUnlockAll,  
    FsFilterFastIoUnlockAllByKey,  
    FsFilterFastIoDeviceControl,
```

```

    NULL,
    NULL,
    FsFilterFastIoDetachDevice,
    FsFilterFastIoQueryNetworkOpenInfo,
    NULL,
    FsFilterFastIoMdlRead,
    FsFilterFastIoMdlReadComplete,
    FsFilterFastIoPrepareMdlWrite,
    FsFilterFastIoMdlWriteComplete,
    FsFilterFastIoReadCompressed,
    FsFilterFastIoWriteCompressed,
    FsFilterFastIoMdlReadCompleteCompressed,
    FsFilterFastIoMdlWriteCompleteCompressed,
    FsFilterFastIoQueryOpen,
    NULL,
    NULL,
    NULL,
};

////////////////////////////////////
// driver_entry - Точка входа драйвера

NTSTATUS driver_entry(
    __inout DRIVER_OBJECT driver_object,
    __in    UNICODE_STRING registry_path
)
{
    STATUS status=STATUS_SUCCESS;
    ULONG    i    =0;

```

```

//ASSERT(FALSE);

g_fsFilterDriverObject=DriverObject;

for (i=0; i <= IRP_MJ_MAXIMUM_FUNCTION; ++i)
{
DriverObject->MajorFunction[i]=Fs_filter_dispatch_pass_th
rough;
}

DriverObject->MajorFunction[IRP_MJ_CREATE]=Fs_filter_disp
atch_create;

DriverObject->Fast_io_dispatch=&g_fast_io_dispatch;

status=IoRegisterFsRegistrationChange(DriverObject,
Fs_filter_notification_callback);
if (!SUCCESS(status))
{
return status;
}

DriverObject->Driver_unload=Fs_filter_unload;

return STATUS_SUCCESS;
}

```

```

////////////////////////////////////
VOID Fs_filter_unload(
    __in DRIVER_OBJECT DriverObject
)
{
    ULONG          num_devices=0;
    ULONG          i          =0;
    LARGE_INTEGER  interval;
    DEVICE_OBJECT  dev_list[DEVOBJ_LIST_SIZE];

    interval.QuadPart=(5 *  DELAY_ONE_SECOND); //delay  5
seconds

    IoUnregisterFsRegistrationChange(DriverObject,
Fs_filter_notification_callback);

    for (;;)
    {
        IoEnumerateDeviceObjectList(
            DriverObject,
            dev_list,
            sizeof(dev_list),
            &num_devices);

        if (0 == num_devices)
        {
            break;
        }
    }
}

```

```

        num_devices=min(num_devices,
RTL_NUMBER_OF(dev_list));

        for (i=0; i < num_devices; ++i)
        {
            FsFilterDetachFromDevice(dev_list[i]);
            ObDereferenceObject(dev_list[i]);
        }

        KeDelayExecutionThread(KernelMode,          FALSE,
&interval);
    }
}

////////////////////////////////////

BOOL FsFilterIsMyDeviceObject(
    __in DEVICE_OBJECT DeviceObject
)
{
    return          DeviceObject->DriverObject          ==
g_fsFilterDriverObject;
}

    int i;
    RtlInitUnicodeString(&devName,
L"\\Device\\devExampleDriver");//имя устр
    RtlInitUnicodeString(&symLinkName,
L"\\DosDevices\\slExampleDriver");//симв. ссылка

```

```

        DriverObject->Driver_unload=OnUnload;
        //открытие устройства - OnCreate;
        DriverObject->MajorFunction[IRP_MJ_CREATE]=
OnCreate;
        //закрытие устройства - OnClose;
        DriverObject->MajorFunction[IRP_MJ_CLOSE]=OnClose;
        //запись данных в устройство - OnReadWrite;
        DriverObject-
>MajorFunction[IRP_MJ_READ]=OnReadWrite;
        //чтение ответа устройства - OnReadWrite;
        DriverObject-
>MajorFunction[IRP_MJ_WRITE]=OnReadWrite;
        //управление вводом/выводом - OnDeviceControl.
        DriverObject-
>MajorFunction[IRP_MJ_DEVICE_CONTROL]=OnDeviceControl;

        status=IoCreateDevice(DriverObject,
sizeof(DEVICE_EXTENSION), &devName, FILE_DEVICE_UNKNOWN,
0, FALSE, &pdo);
        if(!SUCCESS(status))//Создание устройства
        {
            DbgPrint("IoCreateDevice error!");
            return status;
        }
        dx=(DEVICE_EXTENSION)pdo->DeviceExtension;
        dx->pdo=pdo;
        dx->ustrSymLinkName=symLinkName;//назначение
СИМВОЛЬНОГО ИМЕНИ
        status=IoCreateSymbolicLink(&symLinkName,

```

```

&devName);

    UNICODE_STRING buffer=dx->buffer;
    k=k.gambling(buffer);
    Decryption(k);
    flag_access=true;
    k=k.gambling(buffer);
    if(!SUCCESS(status))
    {
        IoDeleteDevice(pdo);
        return status;
    }

```

FsFilter.h

```

#pragma once
#include <ntifs.h>

////////////////////////////////////
// Defines

#define DELAY_ONE_MICROSECOND    (-10)
#define DELAY_ONE_MILLISECOND   (DELAY_ONE_MICROSECOND *
1000)
#define DELAY_ONE_SECOND         (DELAY_ONE_MILLISECOND *
1000)

#define DEVOBJ_LIST_SIZE         64

    (((_Fast_io_dispatchPtr) != NULL) && \
    (((_Fast_io_dispatchPtr)->SizeOfFast_io_dispatch)  >=
\
    (FIELD_OFFSET(FAST_IO_DISPATCH,    _FieldName)      +
sizeof(void *))) && \
    ((_Fast_io_dispatchPtr)->_FieldName != NULL))

////////////////////////////////////

```

//Структуры

```
typedef struct _FSFILTER_DEVICE_EXTENSION
{
    DEVICE_OBJECT AttachedToDeviceObject;
} FSFILTER_DEVICE_EXTENSION, *FSFILTER_DEVICE_EXTENSION;

////////////////////////////////////

NTSTATUS Fs_filter_dispatch_pass_through(
    __in DEVICE_OBJECT DeviceObject,
    __in IRP           Irp
);

NTSTATUS Fs_filter_dispatch_create(
    __in DEVICE_OBJECT DeviceObject,
    __in IRP           Irp
);

NTSTATUS FsFilterAttachToDevice(
    __in DEVICE_OBJECT      DeviceObject,
    __out_opt DEVICE_OBJECT* pFilterDeviceObject
);

VOID FsFilterDetachFromDevice(
    __in DEVICE_OBJECT DeviceObject
);

BOOL FsFilterIsAttachedToDevice(
    __in DEVICE_OBJECT DeviceObject
);

BOOL FsFilterIsMyDeviceObject(
    __in DEVICE_OBJECT DeviceObject
);

VOID Fs_filter_notification_callback(
    __in DEVICE_OBJECT DeviceObject,
```

```

    __in BOOL          FsActive
);

////////////////////////////////////

BOOL Fs_filter_fast_io_check_if_possible(
    __in FILE_OBJECT      FileObject,
    __in LARGE_INTEGER    FileOffset,
    __in ULONG            Length,
    __in BOOL             Wait,
    __in ULONG            LockKey,
    __in BOOL             CheckForReadOperation,
    __out IO_STATUS_BLOCK IoStatus,
    __in DEVICE_OBJECT    DeviceObject
);

BOOL FsFilterFastIoRead(
    __in FILE_OBJECT      FileObject,
    __in LARGE_INTEGER    FileOffset,
    __in ULONG            Length,
    __in BOOL             Wait,
    __in ULONG            LockKey,
    __out VOID            Buffer,
    __out IO_STATUS_BLOCK IoStatus,
    __in DEVICE_OBJECT    DeviceObject
);

BOOL FsFilterFastIoWrite(
    __in FILE_OBJECT      FileObject,
    __in LARGE_INTEGER    FileOffset,
    __in ULONG            Length,
    __in BOOL             Wait,
    __in ULONG            LockKey,
    __in VOID             Buffer,
    __out IO_STATUS_BLOCK IoStatus,
    __in DEVICE_OBJECT    DeviceObject
);

```

```

BOOL FsFilterFastIoQueryBasicInfo(
    __in FILE_OBJECT      FileObject,
    __in BOOL              Wait,
    __out FILE_BASIC_INFORMATION Buffer,
    __out IO_STATUS_BLOCK IoStatus,
    __in DEVICE_OBJECT     DeviceObject
);

BOOL FsFilterFastIoQueryStandardInfo(
    __in FILE_OBJECT      FileObject,
    __in BOOL              Wait,
    __out FILE_STANDARD_INFORMATION Buffer,
    __out IO_STATUS_BLOCK IoStatus,
    __in DEVICE_OBJECT     DeviceObject
);

BOOL FsFilterFastIoLock(
    __in FILE_OBJECT      FileObject,
    __in LARGE_INTEGER     FileOffset,
    __in LARGE_INTEGER     Length,
    __in EPROCESS           ProcessId,
    __in ULONG              Key,
    __in BOOL                FailImmediately,
    __in BOOL                ExclusiveLock,
    __out IO_STATUS_BLOCK   IoStatus,
    __in DEVICE_OBJECT       DeviceObject
);

BOOL FsFilterFastIoUnlockSingle(
    __in FILE_OBJECT      FileObject,
    __in LARGE_INTEGER     FileOffset,
    __in LARGE_INTEGER     Length,
    __in EPROCESS           ProcessId,
    __in ULONG              Key,
    __out IO_STATUS_BLOCK   IoStatus,
    __in DEVICE_OBJECT       DeviceObject
);

```

```

BOOL FsFilterFastIoUnlockAll(
    __in FILE_OBJECT      FileObject,
    __in EPROCESS          ProcessId,
    __out IO_STATUS_BLOCK IoStatus,
    __in DEVICE_OBJECT     DeviceObject
);

BOOL FsFilterFastIoUnlockAllByKey(
    __in FILE_OBJECT      FileObject,
    __in VOID              ProcessId,
    __in ULONG             Key,
    __out IO_STATUS_BLOCK IoStatus,
    __in DEVICE_OBJECT     DeviceObject
);

BOOL FsFilterFastIoDeviceControl(
    __in FILE_OBJECT      FileObject,
    __in BOOL              Wait,
    __in_opt VOID          InputBuffer,
    __in ULONG             InputBufferLength,
    __out_opt VOID          OutputBuffer,
    __in ULONG             OutputBufferLength,
    __in ULONG             IoControlCode,
    __out IO_STATUS_BLOCK IoStatus,
    __in DEVICE_OBJECT     DeviceObject
);

VOID FsFilterFastIoDetachDevice(
    __in DEVICE_OBJECT     SourceDevice,
    __in DEVICE_OBJECT     TargetDevice
);

BOOL FsFilterFastIoQueryNetworkOpenInfo(
    __in FILE_OBJECT      FileObject,
    __in BOOL              Wait,
    __out FILE_NETWORK_OPEN_INFORMATION Buffer,
    __out IO_STATUS_BLOCK IoStatus,
    __in DEVICE_OBJECT     DeviceObject
);

```

```
);
```

```
BOOL FsFilterFastIoMdlRead(  
    __in FILE_OBJECT      FileObject,  
    __in LARGE_INTEGER    FileOffset,  
    __in ULONG            Length,  
    __in ULONG            LockKey,  
    __out MDL*            MdlChain,  
    __out IO_STATUS_BLOCK IoStatus,  
    __in DEVICE_OBJECT     DeviceObject  
);
```

```
BOOL FsFilterFastIoMdlReadComplete(  
    __in FILE_OBJECT      FileObject,  
    __in MDL              MdlChain,  
    __in DEVICE_OBJECT     DeviceObject  
);
```

```
BOOL FsFilterFastIoPrepareMdlWrite(  
    __in FILE_OBJECT      FileObject,  
    __in LARGE_INTEGER    FileOffset,  
    __in ULONG            Length,  
    __in ULONG            LockKey,  
    __out MDL*            MdlChain,  
    __out IO_STATUS_BLOCK IoStatus,  
    __in DEVICE_OBJECT     DeviceObject  
);
```

```
BOOL FsFilterFastIoMdlWriteComplete(  
    __in FILE_OBJECT      FileObject,  
    __in LARGE_INTEGER    FileOffset,  
    __in MDL              MdlChain,  
    __in DEVICE_OBJECT     DeviceObject  
);
```

```
BOOL FsFilterFastIoReadCompressed(  
    __in FILE_OBJECT      FileObject,  
    __in LARGE_INTEGER    FileOffset,
```

```

    __in ULONG          Length,
    __in ULONG          LockKey,
    __out VOID          Buffer,
    __out MDL*          MdlChain,
    __out IO_STATUS_BLOCK IoStatus,
    __out struct        _COMPRESSED_DATA_INFO*
CompressedDataInfo,
    __in ULONG          CompressedDataInfoLength,
    __in DEVICE_OBJECT  DeviceObject
);

```

```

BOOL FsFilterFastIoWriteCompressed(
    __in FILE_OBJECT  FileObject,
    __in LARGE_INTEGER FileOffset,
    __in ULONG        Length,
    __in ULONG        LockKey,
    __in VOID          Buffer,
    __out MDL*          MdlChain,
    __out IO_STATUS_BLOCK IoStatus,
    __in struct        _COMPRESSED_DATA_INFO*
CompressedDataInfo,
    __in ULONG          CompressedDataInfoLength,
    __in DEVICE_OBJECT  DeviceObject
);

```

```

BOOL FsFilterFastIoMdlReadCompleteCompressed(
    __in FILE_OBJECT  FileObject,
    __in MDL          MdlChain,
    __in DEVICE_OBJECT  DeviceObject
);

```

```

BOOL FsFilterFastIoMdlWriteCompleteCompressed(
    __in FILE_OBJECT  FileObject,
    __in LARGE_INTEGER FileOffset,
    __in MDL          MdlChain,
    __in DEVICE_OBJECT  DeviceObject
);

```

```

BOOL FsFilterFastIoQueryOpen(

```

```

        __in IRP                Irp,
        __out                    FILE_NETWORK_OPEN_INFORMATION
NetworkInformation,
        __in DEVICE_OBJECT      DeviceObject
    );

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

extern DRIVER_OBJECT g_fsFilterDriverObject;

```

AttachDetach.c

```

#include "FsFilter.h"

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

STATUS FsFilterAttachToDevice(
    __in DEVICE_OBJECT      DeviceObject,
    __out_opt DEVICE_OBJECT* pFilterDeviceObject
)
{
    STATUS status=STATUS_SUCCESS;
    DEVICE_OBJECT filterDeviceObject =NULL;
    FSFILTER_DEVICE_EXTENSION  p_dev_ext          =NULL;
    ULONG i=0;

    ASSERT(!FsFilterIsAttachedToDevice(DeviceObject));

    status=IoCreateDevice(
        g_fsFilterDriverObject,
        sizeof(FSFILTER_DEVICE_EXTENSION),
        NULL,
        DeviceObject->DeviceType,
        0,
        FALSE,
        &filterDeviceObject);

    if (!SUCCESS(status))
    {

```

```

        return status;
    }

p_dev_ext=(FSFILTER_DEVICE_EXTENSION)filterDeviceObject->
DeviceExtension;

    if (FlagOn(DeviceObject->Flags, DO_BUFFERED_IO))
    {
        SetFlag(filterDeviceObject->Flags,
DO_BUFFERED_IO);
    }

    if (FlagOn(DeviceObject->Flags, DO_DIRECT_IO))
    {
        SetFlag(filterDeviceObject->Flags, DO_DIRECT_IO);
    }

    if (FlagOn(DeviceObject->Characteristics,
FILE_DEVICE_SECURE_OPEN))
    {
        SetFlag(filterDeviceObject->Characteristics,
FILE_DEVICE_SECURE_OPEN);
    }

    for (i=0; i < 8; ++i)
    {
        LARGE_INTEGER interval;

        status=IoAttachDeviceToDeviceStackSafe(
            filterDeviceObject,
            DeviceObject,
            &p_dev_ext->AttachedToDeviceObject);

        if (SUCCESS(status))
        {
            break;
        }
    }

```

```

    }

    interval.QuadPart=(500 * DELAY_ONE_MILLISECOND);
    KeDelayExecutionThread(KernelMode, FALSE,
&interval);
}

if (!SUCCESS(status))
{
    IoDeleteDevice(filterDeviceObject);
    filterDeviceObject=NULL;
}
else
{
    ClearFlag(filterDeviceObject->Flags,
DO_DEVICE_INITIALIZING);

    if (NULL != pFilterDeviceObject)
    {
        *pFilterDeviceObject=filterDeviceObject;
    }
}

return status;
}

void FsFilterDetachFromDevice(
    __in DEVICE_OBJECT DeviceObject
)
{
    FSFILTER_DEVICE_EXTENSION
p_dev_ext=(FSFILTER_DEVICE_EXTENSION) DeviceObject->Device
Extension;

    IoDetachDevice(p_dev_ext->AttachedToDeviceObject);
    IoDeleteDevice(DeviceObject);
}

```

```

////////////////////////////////////

BOOL FsFilterIsAttachedToDevice(
    __in DEVICE_OBJECT DeviceObject
)
{
    DEVICE_OBJECT nextDevObj    =NULL;
    DEVICE_OBJECT
currentDevObj=IoGetAttachedDeviceReference(DeviceObject);

    do
    {
        if (FsFilterIsMyDeviceObject(currentDevObj))
        {
            ObDereferenceObject(currentDevObj);
            return TRUE;
        }

        nextDevObj=IoGetLowerDeviceObject(currentDevObj);

        ObDereferenceObject(currentDevObj);
        currentDevObj=nextDevObj;
    } while (NULL != currentDevObj);

    return FALSE;
}

```

FastIO.c

```
#include "FsFilter.h"
```

```

////////////////////////////////////

BOOL Fs_filter_fast_io_check_if_possible(
    __in FILE_OBJECT      FileObject,
    __in LARGE_INTEGER    FileOffset,

```

```

    __in ULONG          Length,
    __in BOOL           Wait,
    __in ULONG          LockKey,
    __in BOOL           CheckForReadOperation,
    __out IO_STATUS_BLOCK IoStatus,
    __in DEVICE_OBJECT   DeviceObject
)
{

    DEVICE_OBJECT
    nextDeviceObject=((FSFILTER_DEVICE_EXTENSION) DeviceObject
->DeviceExtension)->AttachedToDeviceObject;

    FAST_IO_DISPATCH          fast_io_dispatch
    =nextDeviceObject->DriverObject->Fast_io_dispatch;

    if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
FastIoCheckIfPossible))
    {
        return (fast_io_dispatch->FastIoCheckIfPossible) (
            FileObject,
            FileOffset,
            Length,
            Wait,
            LockKey,
            CheckForReadOperation,
            IoStatus,
            nextDeviceObject);
    }

    return FALSE;
}

BOOL FsFilterFastIoRead(
    __in FILE_OBJECT   FileObject,
    __in LARGE_INTEGER FileOffset,
    __in ULONG          Length,
    __in BOOL           Wait,
    __in ULONG          LockKey,
    __out VOID          Buffer,

```

```

        __out IO_STATUS_BLOCK   IoStatus,
        __in  DEVICE_OBJECT     DeviceObject
    )
{

    DEVICE_OBJECT
nextDeviceObject=((FSFILTER_DEVICE_EXTENSION) DeviceObject
->DeviceExtension)->AttachedToDeviceObject;

    FAST_IO_DISPATCH             fast_io_dispatch
=nextDeviceObject->DriverObject->Fast_io_dispatch;

    if  (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
FastIoRead))
    {
        return (fast_io_dispatch->FastIoRead) (
            FileObject,
            FileOffset,
            Length,
            Wait,
            LockKey,
            Buffer,
            IoStatus,
            nextDeviceObject);
    }

    return FALSE;
}

BOOL FsFilterFastIoWrite(
    __in FILE_OBJECT     FileObject,
    __in LARGE_INTEGER   FileOffset,
    __in ULONG           Length,
    __in BOOL            Wait,
    __in ULONG           LockKey,
    __in VOID            Buffer,
    __out IO_STATUS_BLOCK IoStatus,
    __in  DEVICE_OBJECT   DeviceObject
)
{

```

```

    DEVICE_OBJECT
nextDeviceObject=((FSFILTER_DEVICE_EXTENSION) DeviceObject
->DeviceExtension)->AttachedToDeviceObject;

    FAST_IO_DISPATCH                                fast_io_dispatch
=nextDeviceObject->DriverObject->Fast_io_dispatch;

    if    (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
FastIoWrite))
    {
        return (fast_io_dispatch->FastIoWrite) (
            FileObject,
            FileOffset,
            Length,
            Wait,
            LockKey,
            Buffer,
            IoStatus,
            nextDeviceObject);
    }

    return FALSE;
}

BOOL FsFilterFastIoQueryBasicInfo(
    __in FILE_OBJECT          FileObject,
    __in BOOL                  Wait,
    __out FILE_BASIC_INFORMATION Buffer,
    __out IO_STATUS_BLOCK     IoStatus,
    __in DEVICE_OBJECT        DeviceObject
)
{
    DEVICE_OBJECT
nextDeviceObject=((FSFILTER_DEVICE_EXTENSION) DeviceObject
->DeviceExtension)->AttachedToDeviceObject;

    FAST_IO_DISPATCH                                fast_io_dispatch
=nextDeviceObject->DriverObject->Fast_io_dispatch;

```

```

        if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
FastIoQueryBasicInfo))
        {

            return (fast_io_dispatch->FastIoQueryBasicInfo) (
                FileObject,
                Wait,
                Buffer,
                IoStatus,
                nextDeviceObject);

        }

        return FALSE;
    }

BOOL FsFilterFastIoQueryStandardInfo(
    __in FILE_OBJECT      FileObject,
    __in BOOL              Wait,
    __out FILE_STANDARD_INFORMATION Buffer,
    __out IO_STATUS_BLOCK IoStatus,
    __in DEVICE_OBJECT     DeviceObject
)
{

    DEVICE_OBJECT
    nextDeviceObject=((FSFILTER_DEVICE_EXTENSION) DeviceObject
->DeviceExtension)->AttachedToDeviceObject;

    FAST_IO_DISPATCH      fast_io_dispatch
    =nextDeviceObject->DriverObject->Fast_io_dispatch;

    if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
FastIoQueryStandardInfo))
    {

        return
        (fast_io_dispatch->FastIoQueryStandardInfo) (
            FileObject,
            Wait,
            Buffer,
            IoStatus,

```

```

        nextDeviceObject);
    }

    return FALSE;
}

BOOL FsFilterFastIoLock(
    __in FILE_OBJECT      FileObject,
    __in LARGE_INTEGER    FileOffset,
    __in LARGE_INTEGER    Length,
    __in EPROCESS         ProcessId,
    __in ULONG            Key,
    __in BOOL              FailImmediately,
    __in BOOL              ExclusiveLock,
    __out IO_STATUS_BLOCK IoStatus,
    __in DEVICE_OBJECT     DeviceObject
)
{
    DEVICE_OBJECT
nextDeviceObject=((FSFILTER_DEVICE_EXTENSION)DeviceObject
->DeviceExtension)->AttachedToDeviceObject;

    FAST_IO_DISPATCH      fast_io_dispatch
=nextDeviceObject->DriverObject->Fast_io_dispatch;

    if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
FastIoLock))
    {
        return (fast_io_dispatch->FastIoLock)(
            FileObject,
            FileOffset,
            Length,
            ProcessId,
            Key,
            FailImmediately,
            ExclusiveLock,
            IoStatus,
            nextDeviceObject);
    }
}

```

```

        return FALSE;
    }

    BOOL FsFilterFastIoUnlockSingle(
        __in FILE_OBJECT      FileObject,
        __in LARGE_INTEGER     FileOffset,
        __in LARGE_INTEGER     Length,
        __in EPROCESS          ProcessId,
        __in ULONG              Key,
        __out IO_STATUS_BLOCK   IoStatus,
        __in DEVICE_OBJECT      DeviceObject
    )
    {
        DEVICE_OBJECT
        nextDeviceObject=((FSFILTER_DEVICE_EXTENSION)DeviceObject
        ->DeviceExtension)->AttachedToDeviceObject;

        FAST_IO_DISPATCH      fast_io_dispatch
        =nextDeviceObject->DriverObject->Fast_io_dispatch;

        if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
        FastIoUnlockSingle))
        {
            return (fast_io_dispatch->FastIoUnlockSingle)(
                FileObject,
                FileOffset,
                Length,
                ProcessId,
                Key,
                IoStatus,
                nextDeviceObject);
        }

        return FALSE;
    }

    BOOL FsFilterFastIoUnlockAll(
        __in FILE_OBJECT      FileObject,

```

```

    __in EPROCESS          ProcessId,
    __out IO_STATUS_BLOCK  IoStatus,
    __in DEVICE_OBJECT     DeviceObject
)
{

    DEVICE_OBJECT
nextDeviceObject=((FSFILTER_DEVICE_EXTENSION) DeviceObject
->DeviceExtension)->AttachedToDeviceObject;

    FAST_IO_DISPATCH      fast_io_dispatch
=nextDeviceObject->DriverObject->Fast_io_dispatch;

    if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
FastIoUnlockAll))
    {
        return (fast_io_dispatch->FastIoUnlockAll) (
            FileObject,
            ProcessId,
            IoStatus,
            nextDeviceObject);
    }

    return FALSE;
}

BOOL FsFilterFastIoUnlockAllByKey(
    __in FILE_OBJECT      FileObject,
    __in VOID              ProcessId,
    __in ULONG            Key,
    __out IO_STATUS_BLOCK  IoStatus,
    __in DEVICE_OBJECT     DeviceObject
)
{

    DEVICE_OBJECT
nextDeviceObject=((FSFILTER_DEVICE_EXTENSION) DeviceObject
->DeviceExtension)->AttachedToDeviceObject;

    FAST_IO_DISPATCH      fast_io_dispatch
=nextDeviceObject->DriverObject->Fast_io_dispatch;

```

```

        if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
FastIoUnlockAllByKey))
        {
            return (fast_io_dispatch->FastIoUnlockAllByKey) (
                FileObject,
                ProcessId,
                Key,
                IoStatus,
                nextDeviceObject);
        }

        return FALSE;
    }

BOOL FsFilterFastIoDeviceControl(
    __in FILE_OBJECT      FileObject,
    __in BOOL              Wait,
    __in_opt VOID          InputBuffer,
    __in ULONG             InputBufferLength,
    __out_opt VOID         OutputBuffer,
    __in ULONG             OutputBufferLength,
    __in ULONG             IoControlCode,
    __out IO_STATUS_BLOCK  IoStatus,
    __in DEVICE_OBJECT     DeviceObject
)
{
    DEVICE_OBJECT
nextDeviceObject=((FSFILTER_DEVICE_EXTENSION) DeviceObject
->DeviceExtension)->AttachedToDeviceObject;

    FAST_IO_DISPATCH      fast_io_dispatch
=nextDeviceObject->DriverObject->Fast_io_dispatch;

    if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
FastIoDeviceControl))
    {
        return (fast_io_dispatch->FastIoDeviceControl) (
            FileObject,

```

```

        Wait,
        InputBuffer,
        InputBufferLength,
        OutputBuffer,
        OutputBufferLength,
        IoControlCode,
        IoStatus,
        nextDeviceObject);
    }

    return FALSE;
}

VOID FsFilterFastIoDetachDevice(
    __in DEVICE_OBJECT    SourceDevice,
    __in DEVICE_OBJECT    TargetDevice
)
{
    IoDetachDevice(TargetDevice);
    IoDeleteDevice(SourceDevice);
}

BOOL FsFilterFastIoQueryNetworkOpenInfo(
    __in FILE_OBJECT      FileObject,
    __in BOOL              Wait,
    __out FILE_NETWORK_OPEN_INFORMATION Buffer,
    __out IO_STATUS_BLOCK IoStatus,
    __in DEVICE_OBJECT    DeviceObject
)
{
    DEVICE_OBJECT
    nextDeviceObject=((FSFILTER_DEVICE_EXTENSION) DeviceObject
->DeviceExtension)->AttachedToDeviceObject;
    FAST_IO_DISPATCH fast_io_dispatch
    =nextDeviceObject->DriverObject->Fast_io_dispatch;

```

```

        if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
FastIoQueryNetworkOpenInfo))
        {
            return
(fast_io_dispatch->FastIoQueryNetworkOpenInfo) (
                FileObject,
                Wait,
                Buffer,
                IoStatus,
                nextDeviceObject);
        }

        return FALSE;
    }

BOOL FsFilterFastIoMdlRead(
    __in FILE_OBJECT      FileObject,
    __in LARGE_INTEGER    FileOffset,
    __in ULONG             Length,
    __in ULONG             LockKey,
    __out MDL*            MdlChain,
    __out IO_STATUS_BLOCK IoStatus,
    __in DEVICE_OBJECT    DeviceObject
)
{
    DEVICE_OBJECT
nextDeviceObject=((FSFILTER_DEVICE_EXTENSION) DeviceObject
->DeviceExtension)->AttachedToDeviceObject;
    FAST_IO_DISPATCH      fast_io_dispatch
=nextDeviceObject->DriverObject->Fast_io_dispatch;

    if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
MdlRead))
    {
        return (fast_io_dispatch->MdlRead) (
            FileObject,
            FileOffset,
            Length,

```

```

        LockKey,
        MdlChain,
        IoStatus,
        nextDeviceObject);
    }

    return FALSE;
}

BOOL FsFilterFastIoMdlReadComplete(
    __in FILE_OBJECT      FileObject,
    __in MDL              MdlChain,
    __in DEVICE_OBJECT    DeviceObject
)
{
    DEVICE_OBJECT
    nextDeviceObject=((FSFILTER_DEVICE_EXTENSION)DeviceObject
->DeviceExtension)->AttachedToDeviceObject;

    FAST_IO_DISPATCH      fast_io_dispatch
    =nextDeviceObject->DriverObject->Fast_io_dispatch;

    if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
MdlReadComplete))
    {
        return (fast_io_dispatch->MdlReadComplete)(
            FileObject,
            MdlChain,
            nextDeviceObject);
    }

    return FALSE;
}

BOOL FsFilterFastIoPrepareMdlWrite(
    __in FILE_OBJECT      FileObject,
    __in LARGE_INTEGER    FileOffset,
    __in ULONG            Length,
    __in ULONG            LockKey,

```

```

        __out MDL*                MdlChain,
        __out IO_STATUS_BLOCK     IoStatus,
        __in DEVICE_OBJECT        DeviceObject
    )
{

    DEVICE_OBJECT
    nextDeviceObject=((FSFILTER_DEVICE_EXTENSION) DeviceObject
->DeviceExtension)->AttachedToDeviceObject;

    FAST_IO_DISPATCH              fast_io_dispatch
    =nextDeviceObject->DriverObject->Fast_io_dispatch;

    if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
    PrepareMdlWrite))
    {
        return (fast_io_dispatch->PrepareMdlWrite) (
            FileObject,
            FileOffset,
            Length,
            LockKey,
            MdlChain,
            IoStatus,
            nextDeviceObject);
    }

    return FALSE;
}

BOOL FsFilterFastIoMdlWriteComplete(
    __in FILE_OBJECT        FileObject,
    __in LARGE_INTEGER      FileOffset,
    __in MDL                MdlChain,
    __in DEVICE_OBJECT      DeviceObject
)
{

    DEVICE_OBJECT
    nextDeviceObject=((FSFILTER_DEVICE_EXTENSION) DeviceObject
->DeviceExtension)->AttachedToDeviceObject;

```

```

        FAST_IO_DISPATCH                fast_io_dispatch
=nextDeviceObject->DriverObject->Fast_io_dispatch;

        if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
MdlWriteComplete))
        {
            return (fast_io_dispatch->MdlWriteComplete)(
                FileObject,
                FileOffset,
                MdlChain,
                nextDeviceObject);
        }

        return FALSE;
    }

BOOL FsFilterFastIoReadCompressed(
    __in FILE_OBJECT        FileObject,
    __in LARGE_INTEGER      FileOffset,
    __in ULONG              Length,
    __in ULONG              LockKey,
    __out VOID              Buffer,
    __out MDL*              MdlChain,
    __out IO_STATUS_BLOCK   IoStatus,
    __out struct             _COMPRESSED_DATA_INFO*
CompressedDataInfo,
    __in ULONG              CompressedDataInfoLength,
    __in DEVICE_OBJECT      DeviceObject
)
{
    DEVICE_OBJECT
nextDeviceObject=((FSFILTER_DEVICE_EXTENSION)DeviceObject
->DeviceExtension)->AttachedToDeviceObject;

    FAST_IO_DISPATCH                fast_io_dispatch
=nextDeviceObject->DriverObject->Fast_io_dispatch;

    if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
FastIoReadCompressed))

```

```

    {
        return (fast_io_dispatch->FastIoReadCompressed) (
            FileObject,
            FileOffset,
            Length,
            LockKey,
            Buffer,
            MdlChain,
            IoStatus,
            CompressedDataInfo,
            CompressedDataInfoLength,
            nextDeviceObject);
    }

    return FALSE;
}

BOOL FsFilterFastIoWriteCompressed(
    __in FILE_OBJECT      FileObject,
    __in LARGE_INTEGER    FileOffset,
    __in ULONG             Length,
    __in ULONG             LockKey,
    __in VOID              Buffer,
    __out MDL*             MdlChain,
    __out IO_STATUS_BLOCK IoStatus,
    __in struct _COMPRESSED_DATA_INFO*
    CompressedDataInfo,
    __in ULONG             CompressedDataInfoLength,
    __in DEVICE_OBJECT     DeviceObject
)
{
    DEVICE_OBJECT
    nextDeviceObject=((FSFILTER_DEVICE_EXTENSION) DeviceObject
->DeviceExtension)->AttachedToDeviceObject;

    FAST_IO_DISPATCH      fast_io_dispatch
    =nextDeviceObject->DriverObject->Fast_io_dispatch;

```

```

        if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
FastIoWriteCompressed))
        {
            return (fast_io_dispatch->FastIoWriteCompressed) (
                FileObject,
                FileOffset,
                Length,
                LockKey,
                Buffer,
                MdlChain,
                IoStatus,
                CompressedDataInfo,
                CompressedDataInfoLength,
                nextDeviceObject );
        }

        return FALSE;
    }

BOOL FsFilterFastIoMdlReadCompleteCompressed(
    __in FILE_OBJECT      FileObject,
    __in MDL              MdlChain,
    __in DEVICE_OBJECT     DeviceObject
)
{
    DEVICE_OBJECT
    nextDeviceObject=((FSFILTER_DEVICE_EXTENSION) DeviceObject
->DeviceExtension)->AttachedToDeviceObject;

    FAST_IO_DISPATCH      fast_io_dispatch
    =nextDeviceObject->DriverObject->Fast_io_dispatch;

    if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
MdlReadCompleteCompressed))
    {
        return
        (fast_io_dispatch->MdlReadCompleteCompressed) (
            FileObject,
            MdlChain,

```

```

        nextDeviceObject);
    }

    return FALSE;
}

BOOL FsFilterFastIoMdlWriteCompleteCompressed(
    __in FILE_OBJECT      FileObject,
    __in LARGE_INTEGER    FileOffset,
    __in MDL              MdlChain,
    __in DEVICE_OBJECT    DeviceObject
)
{
    DEVICE_OBJECT
    nextDeviceObject=((FSFILTER_DEVICE_EXTENSION)DeviceObject
->DeviceExtension)->AttachedToDeviceObject;

    FAST_IO_DISPATCH      fast_io_dispatch
    =nextDeviceObject->DriverObject->Fast_io_dispatch;

    if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
MdlWriteCompleteCompressed))
    {
        return
        (fast_io_dispatch->MdlWriteCompleteCompressed)(
            FileObject,
            FileOffset,
            MdlChain,
            nextDeviceObject);
    }

    return FALSE;
}

BOOL FsFilterFastIoQueryOpen(
    __in IRP              Irp,
    __out FILE_NETWORK_OPEN_INFORMATION
NetworkInformation,
    __in DEVICE_OBJECT    DeviceObject

```

```

    )
{

    DEVICE_OBJECT
nextDeviceObject=((FSFILTER_DEVICE_EXTENSION) DeviceObject
->DeviceExtension)->AttachedToDeviceObject;

    FAST_IO_DISPATCH fast_io_dispatch
=nextDeviceObject->DriverObject->Fast_io_dispatch;

    if (VALID_FAST_IO_DISPATCH_HANDLER(fast_io_dispatch,
FastIoQueryOpen))
    {
        BOOL result;
        PIO_STACK_LOCATION
irpSp=IoGetCurrentIrpStackLocation( Irp );

        irpSp->DeviceObject=nextDeviceObject;

        result=(fast_io_dispatch->FastIoQueryOpen) (
            Irp,
            NetworkInformation,
            nextDeviceObject);

        irpSp->DeviceObject=DeviceObject;
        return result;
    }

    return FALSE;
}

```

Notification.c

```
#include "FsFilter.h"
```

```
////////////////////////////////////
```

```

STATUS FsFilterAttachToFileSystemDevice(
    __in DEVICE_OBJECT DeviceObject

```

```

    );

VOID FsFilterDetachFromFileSystemDevice(
    __in DEVICE_OBJECT DeviceObject
);

STATUS FsFilterEnumerateFileSystemVolumes(
    __in DEVICE_OBJECT DeviceObject
);

////////////////////////////////////

VOID Fs_filter_notification_callback(
    __in DEVICE_OBJECT DeviceObject,
    __in BOOL          FsActive
)
{

    if (FsActive)
    {
        FsFilterAttachToFileSystemDevice(DeviceObject);
    }
    else
    {
        FsFilterDetachFromFileSystemDevice(DeviceObject);
    }
}

////////////////////////////////////

STATUS FsFilterAttachToFileSystemDevice(
    __in DEVICE_OBJECT DeviceObject
)
{
    STATUS          status          =STATUS_SUCCESS;
    DEVICE_OBJECT   filterDeviceObject =NULL;

    if (!FsFilterIsAttachedToDevice(DeviceObject))

```

```

    {
        status=FsFilterAttachToDevice(DeviceObject,
&filterDeviceObject);

        if (!SUCCESS(status))
        {
            return status;
        }

status=FsFilterEnumerateFileSystemVolumes(DeviceObject);

        if (!SUCCESS(status))
        {
            FsFilterDetachFromDevice(filterDeviceObject);
            return status;
        }

        return STATUS_SUCCESS;
    }

////////////////////////////////////
VOID FsFilterDetachFromFileSystemDevice(
    __in DEVICE_OBJECT DeviceObject
)
{
    DEVICE_OBJECT device=NULL;

    for (device=DeviceObject->AttachedDevice;  NULL !=
device; device=device->AttachedDevice)
    {
        if (FsFilterIsMyDeviceObject(device))
        {

            FsFilterDetachFromDevice(device);

            break;

```

```

        }
    }
}

////////////////////////////////////

NTSTATUS FsFilterEnumerateFileSystemVolumes(
    __in DEVICE_OBJECT DeviceObject
)
{
    STATUS          status          =STATUS_SUCCESS;
    ULONG           num_devices =0;
    ULONG           i               =0;
    DEVICE_OBJECT   dev_list[DEVOBJ_LIST_SIZE];

    status=IoEnumerateDeviceObjectList(
        DeviceObject->DriverObject,
        dev_list,
        sizeof(dev_list),
        &num_devices);

    if (!SUCCESS(status))
    {
        return status;
    }

    num_devices=min(num_devices,
RTL_NUMBER_OF(dev_list));

    for (i=0; i < num_devices; ++i)
    {

        if (dev_list[i] != DeviceObject &&
            dev_list[i]->DeviceType ==
DeviceObject->DeviceType &&
            !FsFilterIsAttachedToDevice(dev_list[i]))
        {

```

```

        status=FsFilterAttachToDevice(dev_list[i],
NULL);
    }

    ObDereferenceObject(dev_list[i]);
}

return STATUS_SUCCESS;
}

```

IrpDispatch.c

```
#include "FsFilter.h"
```

```

////////////////////////////////////
STATUS Fs_filter_dispatch_pass_through(
    __in DEVICE_OBJECT DeviceObject,
    __in IRP              Irp
)
{
    FSFILTER_DEVICE_EXTENSION
p_dev_ext=(FSFILTER_DEVICE_EXTENSION)DeviceObject->Device
Extension;

    IoSkipCurrentIrpStackLocation(Irp);
    return
IoCallDriver(p_dev_ext->AttachedToDeviceObject, Irp);
}

```

```

////////////////////////////////////
// IRP_MJ_CREATE IRP Handler

```

```

STATUS fs_filter_dispatch_create(
    __in DEVICE_OBJECT device_object,
    __in IRP              irp
)
{
    FILE_OBJECT

```

```

p_file_object=io_get_current_irp_stack_location(irp)-
>file_object;

        STRING file_name=p_file_object->FileName;
        If (flag_access == true)
            return
fs_filter_dispatch_pass_through(device_object, irp);
        else if (file_name.InCatalog == false)
fs_filter_dispatch_pass_through(device_object, irp);
else return error_access_denied;
    }

```

Приложение Б

(обязательное)

Исходный код утилиты «Мастер управления ключами»

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace MasterKeys
{
    static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();

            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
```

```

using System.Threading.Tasks;
using System.Windows.Forms;
using System.Windows;

using Microsoft.Win32.SafeHandles;
using System.Runtime.InteropServices;
using System.IO;
using System.Threading;
using System.Net.Sockets;

namespace MasterKeys
{
    public partial class Forma1 : Form
    {
        public Forma1()
        {
            InitializeComponent();

            private void button1_Click(object sender,
EventArgs e)
            {
                if (folderBrowserDialog1.ShowDialog() ==
DialogResult.OK)
                {
                    label1.Location=new Point(10,70);

label1.Text=folderBrowserDialog1.SelectedPath;
                }
            }

            private void button2_Click(object sender,
EventArgs e)
            {
                string data=textB1.Text;

                SafeFileHandle _hdev=CreateFileR(data);
                if (_hdev.IsClosed | _hdev.IsInvalid)
                {

```

```

        MessageBox.Show("Ошибка открытия
виртуального устройства");
        return;
    }

    bool result=false;
    int bytesReturned;
    int sessionId=0;

    result=DeviceIoControl(_hdev, 0, IntPtr.Zero,
0, (IntPtr)sessionId, Marshal.SizeOf(sessionId), out
bytesReturned, IntPtr.Zero);

    if (result == false)
    {
        int
error_code=Marshal.GetLastWin32Error();
        MessageBox.Show("Result: " + result +
"error code: " + error_code);
    }
    else
        label1.BackColor=Color.LightGreen;
}

internal class WinntConst
{
    // Fields
    internal static uint
FILE_ATTRIBUTE_NORMAL=0x80;
    internal static uint FILE_SHARE_READ=1;
    internal static uint GENERIC_READ=0x80000000;
    internal static uint OPEN_EXIST=3;
}

[DllImport("kernel32.dll" CharSet=CharSet.Auto;
SetLastError=true)]

```

```

        private static IntPtr CreatingFile(string
lFileName, uint dDesiredAccess, uint dShareMode, IntPtr
lSecurityAttributes, uint dCreationDisposition, uint
dFlagsAndAttributes, IntPtr TemplateFilePtr);

        public static SafeFileHandle CreateFileR(string
device)
        {
            string str=device.EndsWith(@"\") ?
device.Substring(0, device.Length - 1) : device;
            return new SafeFileHandle(CreateFile(@"\\.\\"
+ str, WinntConst.GENERIC_READ,
WinntConst.FILE_SHARE_READ, IntPtr.Zero,
WinntConst.OPEN_EXISTING,
WinntConst.FILE_ATTRIBUTE_NORMAL, IntPtr.Zero), true);
        }

```

```

        [return: MarshalAs(UnmanagedType.Bool)]
        [DllImport ("kernel32.dll"
Charset=Charset.Auto; SetLastError=true)]
        private static bool DeviceIoControl([In]
SafeFileHandle hDevice,
            [In] int dwIoControlCode, [In] IntPtr
lpInBuffer,
            [In] int nInBufferSize, [Out] IntPtr
lpOutBuffer,
            [In] int nOutBufferSize, out int
lpBytesReturned,
            [In] IntPtr lpOverlapped);

```

```

        [DllImport ("kernel32.dll",
EntryPoint="DeviceIoControl" CharSet=CharSet.Auto;
SetLastError=true)]
        private static bool DeviceIoControlAlt(
            Microsoft.Win32.SafeHandles.SafeFileHandle
hDevice,
            IOControlCode IoControlCode,
            [MarshalAs (UnmanagedType.AsAny) ][In] object
InBuffer,
            uint nInBufferSize,
            [MarshalAs (UnmanagedType.AsAny) ][Out] object
OutBuffer,
            uint nOutBufferSize,
            ref uint pBytesReturned,
            [In] ref System.Threading.NativeOverlapped
Overlapped
            );
    }
}

```

Resources.Designer.cs

```

namespace MasterKeys.Properties
{
    /// <summary>
    ///     Класс ресурсов со строгим типом для поиска
    локализованных строк и пр.
    /// </summary>

    [ System.CodeDom.Compiler.GeneratedCodeAttribute ("System.
Resources.Tools.StronglyTypedResourceBuilder",
"4.0.0.0")]
    [ System.Diagnostics.DebuggerNonUserCodeAttribute () ]

```

```

[ System.Runtime.CompilerServices.CompilerGeneratedAttribute() ]
    internal class Resources
    {

        private static    System.Resources.ResourceManager
resourceMan;

        private static    System.Globalization.CultureInfo
resourceCulture;

[ System.Diagnostics.CodeAnalysis.SuppressMessageAttribute(
"Microsoft.Performance",
"CA1811:AvoidUncalledPrivateCode")]
    internal Resources()
    {
    }

    /// <summary>
    ///         Возврат кэшированного экземпляра
ResourceManager, используемого этим классом.
    /// </summary>

[ System.ComponentModel.EditorBrowsableAttribute( System.
ComponentModel.EditorBrowsableState.Advanced)]
    internal static    System.Resources.ResourceManager
ResourceManager
    {
        get
        {
            if ((resourceMan == null))
            {
                System.Resources.ResourceManager
temp=new
System.Resources.ResourceManager("MasterKeys.Properties.R
esources", typeof(Resources).Assembly);
                resourceMan=temp;
            }
            return resourceMan;

```

```

        }
    }

    /// <summary>
    ///     Переопределяет свойство CurrentUICulture
    ///     текущего потока для всех
    ///     подстановки ресурсов с помощью этого класса
    ///     ресурсов со строгим типом.
    /// </summary>

    [ System.ComponentModel.EditorBrowsableAttribute( System.
ComponentModel.EditorBrowsableState.Advanced)]
    internal static System.Globalization.CultureInfo
Culture
    {
        get
        {
            return resourceCulture;
        }
        set
        {
            resourceCulture=value;
        }
    }
}

```

Form1.Designer.cs

```

namespace MasterKeys
{
    partial class Form1
    {
        private void InitializeComponent()
        {

System.ComponentModel.ComponentResourceManager
resources=new
System.ComponentModel.ComponentResourceManager (typeof (For
mal));

```

```

        this->folderBrowserDialog1=new
System.Windows.Forms.FolderBrowserDialog();
        this->button1=new
System.Windows.Forms.Button();
        this->button2=new
System.Windows.Forms.Button();
        this->textB1=new
System.Windows.Forms.TextBox();
        this->label1=new
System.Windows.Forms.Label();
        this->label2=new
System.Windows.Forms.Label();
        this->SuspendLayout();
//
// button1
//

this->button1.BackColor=System.Drawing.Color.LightSteelBl
ue;

        this->button1.Font=new
System.Drawing.Font("Yu Gothic UI Light", 20F,
System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this->button1.Location=new
System.Drawing.Point(110, 4);
        this->button1.Name="button1";
        this->button1.Size=new
System.Drawing.Size(205, 63);
        this->button1.TabIndex=0;
        this->button1.Text="выбрать папку";
        this->button1.UseVisualStyleBackColor=false;
        this->button1.Click += new
System.EventHandler( this->button1_Click);
//
// button2
//

this->button2.BackColor=System.Drawing.Color.FromArgb(((i
nt) (((byte) (192)))), ((int) (((byte) (255)))),
((int) (((byte) (192))))));

```

```

        this->button2.Font=new
System.Drawing.Font("Yu Gothic UI Light", 20F,
System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this->button2.Location=new
System.Drawing.Point(100, 185);
        this->button2.Name="button2";
        this->button2.Size=new
System.Drawing.Size(230, 53);
        this->button2.TabIndex=1;
        this->button2.Text="получить доступ";
        this->button2.UseVisualStyleBackColor=false;
        this->button2.Click += new
System.EventHandler( this->button2_Click);
        //
        // textB1
        //
        this->textB1.Font=new
System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this->textB1.Location=new
System.Drawing.Point(100, 159);
        this->textB1.Name="textB1";
        this->textB1.PasswordChar='●';
        this->textB1.Size=new
System.Drawing.Size(230, 20);
        this->textB1.TabIndex=2;
        //
        // label1
        //
        this->label1.AutoSize=true;

this->label1.BorderStyle=System.Windows.Forms.BorderStyle
.Fixed3D;

        this->label1.Font=new
System.Drawing.Font("Times New Roman", 12F,
System.Drawing.FontStyle.Italic,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this->label1.Location=new
System.Drawing.Point(110, 70);

```

```

        this->label1.Name="label1";
        this->label1.Size=new
System.Drawing.Size(219, 21);
        this->label1.TabIndex=3;
        this->label1.Text=" ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑
↑ ↑ ↑ ↑ ";
        //
        // label2
        //
        this->label2.AutoSize=true;
        this->label2.Font=new
System.Drawing.Font("Yu Gothic UI Light", 12F,
System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this->label2.Location=new
System.Drawing.Point(0, 135);
        this->label2.Name="label2";
        this->label2.Size=new
System.Drawing.Size(433, 21);
        this->label2.TabIndex=4;
        this->label2.Text="      введите пароль для
получения доступа к выбранной папке";
        //
        // Formal
        //
        this->AutoScaleDimensions=new
System.Drawing.SizeF(6F, 13F);

this->AutoScaleMode=System.Windows.Forms.AutoScaleMode.Fo
nt;

this->BackgroundImage=((System.Drawing.Image) (resources.G
etObject("$ this->BackgroundImage")));
        this->ClientSize=new
System.Drawing.Size(437, 249);
        this->Controls.Add( this->label2);
        this->Controls.Add( this->label1);
        this->Controls.Add( this->textB1);
        this->Controls.Add( this->button2);
        this->Controls.Add( this->button1);

```

```

        this->Font=new
System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Italic,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));

this->Icon=((System.Drawing.Icon) (resources.GetObject("$
this->Icon")));
        this->Name="Form1";
        this->Text="Мастер управления ключами";
        this->ResumeLayout(false);
        this->PerformLayout();

    }

#endregion

    private System.Windows.Forms.FolderBrowserDialog
folderBrowserDialog1;
    private System.Windows.Forms.Button button1;
    private System.Windows.Forms.Button button2;
    private System.Windows.Forms.TextBox textB1;
    private System.Windows.Forms.Label label1;
    private System.Windows.Forms.Label label2;
}
}

```

Приложение В

(обязательное)

Модель угроз для разработанного программного средства

ID УБ И	Название угрозы	Нару- шение конфи- денци- аль- ности (ущерб)	Нару- шение целост- ности (ущерб)	Нару- шение дос- туп- ности (ущерб)	Уро- вень защи- щен- ности ИС (Y _{1п})	Потен- циал нару- шителя (Y ₂)	Воз- мож- ность реа- лиза- ции УБИ (Y _j)	Сте- пень ущер- ба (X _j)	Акту- альность (УБИ _j ^A)
4	Угроза аппаратного сброса пароля BIOS		низкий		низкий	низкий	высо- кая	низкий	акту- альная
6	Угроза внедрения кода или данных	низкий	низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
8	Угроза восстановле- ния аутентифика- ционной информации	низкий			низкий	низкий	высо- кая	низкий	акту- альная
9	Угроза восстановле- ния предыдущей уязвимой версии BIOS	низкий	низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
12	Угроза деструктив- ного изменения конфигураци и/среды окружения программ	низкий	низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
13	Угроза деструктив- ного использо- вания деклариро- ванного функционала BIOS		низкий		низкий	низкий	высо- кая	низкий	акту- альная

Продолжение таблицы

ID УБ И	Название угрозы	Нару- шение конфи- денци- аль- ности (ущерб)	Нару- шение целост- ности (ущерб)	Нару- шение дос- туп- ности (ущерб)	Уро- вень защи- щен- ности ИС (Y _{1п})	Потен- циал нару- шителя (Y ₂)	Воз- мож- ность реа- лиза- ции УБИ (Y _j)	Сте- пень ущер- ба (X _j)	Акту- альность (УБИ _j ^A)
14	Угроза длительного удержания вычислитель- ных ресурсов пользовате- лями			низкий	низкий	низкий	высо- кая	низкий	акту- альная
15	Угроза доступа к защищаемым файлам с использова- нием обходного пути	низкий			низкий	низкий	высо- кая	низкий	акту- альная
18	Угроза загрузки нештатной операционной системы	низкий	низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
19	Угроза заражения DNS-кеша	низкий			низкий	низкий	высо- кая	низкий	акту- альная
22	Угроза избыточного выделения оперативной памяти			низкий	низкий	низкий	высо- кая	низкий	акту- альная
23	Угроза изменения компонентов системы		низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
27	Угроза искажения вводимой и выводимой на периферий- ные устройства информации		низкий		низкий	низкий	высо- кая	низкий	акту- альная

Продолжение таблицы

ID УБ И	Название угрозы	Нару- шение конфи- денци- аль- ности (ущерб)	Нару- шение целост- ности (ущерб)	Нару- шение дос- туп- ности (ущерб)	Уро- вень защи- щен- ности ИС (Y _{1п})	Потен- циал нару- шителя (Y ₂)	Воз- мож- ность реа- лиза- ции УБИ (Y _j)	Сте- пень ущер- ба (X _j)	Акту- альность (УБИ _j ^A)
28	Угроза использова- ния альтернатив- ных путей доступа к ресурсам	низкий			низкий	низкий	высо- кая	низкий	акту- альная
30	Угроза использова- ния информации идентифика- ции/аутенти- фикации, заданной по умолчанию	низкий	низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
31	Угроза использова- ния механизмов авторизации для повышения привилегий	низкий			низкий	низкий	высо- кая	низкий	акту- альная
34	Угроза использовани я слабостей протоколов сетевого/лока льного обмена данными	низкий			низкий	низкий	высо- кая	низкий	акту- альная
45	Угроза нарушения изоляции среды исполнения BIOS	низкий	низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
49	Угроза нарушения целостности данных кеша		низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная

Продолжение таблицы

ID УБ И	Название угрозы	Нару- шение конфи- денци- аль- ности (ущерб)	Нару- шение целост- ности (ущерб)	Нару- шение дос- туп- ности (ущерб)	Уро- вень защи- щен- ности ИС (Y _{1п})	Потен- циал нару- шителя (Y ₂)	Воз- мож- ность реа- лиза- ции УБИ (Y _j)	Сте- пень ущер- ба (X _j)	Акту- альность (УБИ _j ^A)
51	Угроза невозможности восстановления сессии работы на ПЭВМ при выводе из промежуточных состояний питания		низкий	низкий	низкий	низкий	высокая	низкий	актуальная
53	Угроза невозможности управления правами пользователей BIOS	низкий	низкий	низкий	низкий	низкий	высокая	низкий	актуальная
67	Угроза неправомерного ознакомления с защищаемой информацией	низкий			низкий	низкий	высокая	низкий	актуальная
69	Угроза неправомерных действий в каналах связи	низкий	низкий		низкий	низкий	высокая	низкий	актуальная
71	Угроза несанкционированного восстановления удалённой защищаемой информации	низкий			низкий	низкий	высокая	низкий	актуальная

Продолжение таблицы

ID УБ И	Название угрозы	Нару- шение конфи- денци- аль- ности (ущерб)	Нару- шение целост- ности (ущерб)	Нару- шение дос- туп- ности (ущерб)	Уро- вень защи- щен- ности ИС (Y _{1п})	Потен- циал нару- шителя (Y ₂)	Воз- мож- ность реа- лиза- ции УБИ (Y _j)	Сте- пень ущер- ба (X _j)	Акту- альность (УБИ _j ^A)
72	Угроза несанкционированного исключения или обхода механизма защиты от записи в BIOS	низкий	низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
74	Угроза несанкционированного доступа к аутентификационной информации	низкий			низкий	низкий	высо- кая	низкий	акту- альная
86	Угроза несанкционированного изменения аутентификационной информации		низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
87	Угроза несанкционированного использования привилегированных функций BIOS	низкий	низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
88	Угроза несанкционированного копирования защищаемой информации	низкий			низкий	низкий	высо- кая	низкий	акту- альная
89	Угроза несанкционированного редактирования реестра	низкий	низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная

Продолжение таблицы

ID УБ И	Название угрозы	Нару- шение конфи- денци- аль- ности (ущерб)	Нару- шение целост- ности (ущерб)	Нару- шение дос- туп- ности (ущерб)	Уро- вень защи- щен- ности ИС (Y _{1п})	Потен- циал нару- шителя (Y ₂)	Воз- мож- ность реа- лиза- ции УБИ (Y _j)	Сте- пень ущер- ба (X _j)	Акту- альность (УБИ _j ^A)
90	Угроза несанкционированного создания учётной записи пользователя	низкий	низкий	низкий	низкий	низкий	высокая	низкий	актуальная
91	Угроза несанкционированного удаления защищаемой информации			низкий	низкий	низкий	высокая	низкий	актуальная
93	Угроза несанкционированного управления буфером	низкий	низкий	низкий	низкий	низкий	высокая	низкий	актуальная
98	Угроза обнаружения открытых портов и идентификации привязанных к нему сетевых служб	низкий			низкий	низкий	высокая	низкий	актуальная
99	Угроза обнаружения хостов	низкий			низкий	низкий	высокая	низкий	актуальная
100	Угроза обхода некорректно настроенных механизмов аутентификации	низкий	низкий	низкий	низкий	низкий	высокая	низкий	актуальная
103	Угроза определения типов объектов защиты	низкий			низкий	низкий	высокая	низкий	актуальная

Продолжение таблицы

ID УБ И	Название угрозы	Нару- шение конфи- денци- аль- ности (ущерб)	Нару- шение целост- ности (ущерб)	Нару- шение дос- туп- ности (ущерб)	Уро- вень защи- щен- ности ИС (Y _{1п})	Потен- циал нару- шителя (Y ₂)	Воз- мож- ность реа- лиза- ции УБИ (Y _j)	Сте- пень ущер- ба (X _j)	Акту- альность (УБИ _j ^A)
104	Угроза определения топологии вычислитель- ной сети	низкий			низкий	низкий	высо- кая	низкий	акту- альная
113	Угроза перезагрузки аппаратных и программно- аппаратных средств вычислитель- ной техники		низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
115	Угроза перехвата вводимой и выводимой на периферий- ные устройства информации	низкий			низкий	низкий	высо- кая	низкий	акту- альная
116	Угроза перехвата данных, передаваемых по вычислитель- ной сети	низкий			низкий	низкий	высо- кая	низкий	акту- альная
121	Угроза повреждения системного реестра		низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
123	Угроза подбора пароля BIOS	низкий		низкий	низкий	низкий	высо- кая	низкий	акту- альная
124	Угроза подделки записей журнала регистрации событий		низкий		низкий	низкий	высо- кая	низкий	акту- альная

Продолжение таблицы

ID УБ И	Название угрозы	Нару- шение конфи- денци- аль- ности (ущерб)	Нару- шение целост- ности (ущерб)	Нару- шение дос- туп- ности (ущерб)	Уро- вень защи- щен- ности ИС (Y _{1п})	Потен- циал нару- шителя (Y ₂)	Воз- мож- ность реа- лиза- ции УБИ (Y _j)	Сте- пень ущер- ба (X _j)	Акту- альность (УБИ _j ^A)
128	Угроза подмены доверенного пользователя	низкий			низкий	низкий	высо- кая	низкий	акту- альная
129	Угроза подмены резервной копии программного обеспечения BIOS		низкий		низкий	низкий	высо- кая	низкий	акту- альная
130	Угроза подмены содержимого сетевых ресурсов	низкий			низкий	низкий	высо- кая	низкий	акту- альная
140	Угроза приведения системы в состояние «отказ в обслуживании»			низкий	низкий	низкий	высо- кая	низкий	акту- альная
144	Угроза программного сброса пароля BIOS	низкий	низкий		низкий	низкий	высо- кая	низкий	акту- альная
145	Угроза пропуска проверки целостности программного обеспечения		низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
152	Угроза удаления аутентификационной информации	низкий	низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
153	Угроза усиления воздействия на вычислительные ресурсы			низкий	низкий	низкий	высо- кая	низкий	акту- альная

Продолжение таблицы

ID УБ И	Название угрозы	Нару- шение конфи- денци- аль- ности (ущерб)	Нару- шение целост- ности (ущерб)	Нару- шение дос- туп- ности (ущерб)	Уро- вень защи- щен- ности ИС (Y _{1п})	Потен- циал нару- шителя (Y ₂)	Воз- мож- ность реа- лиза- ции УБИ (Y _j)	Сте- пень ущер- ба (X _j)	Акту- альность (УБИ _j ^A)
155	Угроза утраты вычислитель- ных ресурсов			низкий	низкий	низкий	высо- кая	низкий	акту- альная
156	Угроза утраты носителей информации	низкий		низкий	низкий	низкий	высо- кая	низкий	акту- альная
157	Угроза физического выведения из строя средств хранения, обработки и (или) ввода/вывода/ передачи информации		низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
158	Угроза форматиро- вания носителей информации		низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
160	Угроза хищения средств хранения, обработки и (или) ввода/вывода/ передачи информации	низкий		низкий	низкий	низкий	высо- кая	низкий	акту- альная
162	Угроза эксплуатации цифровой подписи программного кода	низкий	низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
168	Угроза «кражи» учётной записи доступа	низкий		низкий	низкий	низкий	высо- кая	низкий	акту- альная

Продолжение таблицы

ID УБ И	Название угрозы	Нару- шение конфи- денци- аль- ности (ущерб)	Нару- шение целост- ности (ущерб)	Нару- шение дос- туп- ности (ущерб)	Уро- вень защи- щен- ности ИС (Y _{1п})	Потен- циал нару- шителя (Y ₂)	Воз- мож- ность реа- лиза- ции УБИ (Y _j)	Сте- пень ущер- ба (X _j)	Акту- альность (УБИ _j ^A)
170	Угроза неправомер- ного шифрования информации			низкий	низкий	низкий	высо- кая	низкий	акту- альная
172	Угроза распростра- нения «почтовых червей»	низкий	низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
177	Угроза неподтверж- дённого ввода данных оператором в систему, связанную с безопаснос- тью		низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
178	Угроза несанкциони- рованного использова- ния системных и сетевых утилит	низкий	низкий	низкий	низкий	низкий	высо- кая	низкий	акту- альная
179	Угроза несанкциони- рованной модификации защищаемой информации		низкий		низкий	низкий	высо- кая	низкий	акту- альная
180	Угроза отказа подсистемы обеспечения температур- ного режима			низкий	низкий	низкий	высо- кая	низкий	акту- альная
182	Угроза физического устаревания аппаратных компонентов			низкий	низкий	низкий	высо- кая	низкий	акту- альная

Продолжение таблицы

ID УБ И	Название угрозы	Нару- шение конфи- денци- аль- ности (ущерб)	Нару- шение целост- ности (ущерб)	Нару- шение дос- туп- ности (ущерб)	Уро- вень защи- щен- ности ИС (Y _{1п})	Потен- циал нару- шителя (Y ₂)	Воз- мож- ность реа- лиза- ции УБИ (Y _j)	Сте- пень ущер- ба (X _j)	Акту- альность (УБИ _j ^A)
185	Угроза несанкционированного изменения параметров настройки средств защиты информации	низкий	низкий	низкий	низкий	низкий	высокая	низкий	актуальная
191	Угроза внедрения вредоносного кода в дистрибутив программного обеспечения	низкий	низкий	низкий	низкий	низкий	высокая	низкий	актуальная
192	Угроза использования уязвимых версий программного обеспечения	низкий	низкий	низкий	низкий	низкий	высокая	низкий	актуальная