

Министерство образования Республики Беларусь  
Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И  
РАДИОЭЛЕКТРОНИКИ  
Факультет компьютерных систем и сетей  
Кафедра электронных вычислительных машин

Дисциплина: Системное программное обеспечение вычислительных машин  
(СПОВМ)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему:

«Virtual CD-ROM»

Студент: гр.350501

Соловцов В. В.

Руководитель:

Яночкин А.Л.

Минск 2015

# **СОДЕРЖАНИЕ**

## **СОДЕРЖАНИЕ**

## **ВВЕДЕНИЕ**

## **1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

### **1.1 Общие сведения**

### **1.2 Назначения и цели программного средства**

### **1.3 Требования к программному средству**

## **2. ОБЗОР ЛИТЕРАТУРЫ**

### **2.1 Драйвер виртуального диска**

#### **2.1.1 Обработка запросов Plug and Play**

#### **2.1.2 Обработка расширенных запросов**

#### **2.1.3 Инициализация драйвера**

## **3. СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ**

### **3.1 Структурная схема программы**

### **3.2 Взаимодействие между формами**

## **4. ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ**

## **ЗАКЛЮЧЕНИЕ**

## **СПИСОК ЛИТЕРАТУРЫ**

## ВВЕДЕНИЕ

CD- и DVD-диски относятся к числу самых популярных сегодня внешних носителей информации, и на них хранятся самые разнообразные данные — начиная от программного обеспечения и заканчивая фото- и музыкальными коллекциями. Однако работать с ними не всегда удобно, и причин тому несколько. CD- и DVD-диски необходимо вставлять в накопитель, и обмен данными с ними производится заметно медленнее, чем с жестким диском, — в итоге, например, запуск программы с компакт-диска потребует заметно больше времени, чем с винчестера. CD- и DVD-диски перестают читаться в случае появления на них сбойных секторов или если их поцарапать, что рано или поздно происходит при активной эксплуатации. Мобильные пользователи сталкиваются с еще одной сложностью: отправляясь в командировку, им приходится брать с собой весь комплект дисков, который может потребоваться в работе, что увеличивает размер багажа.

Избежать подобных проблем можно, скопировав информацию с компакт-дисков на винчестер, но данный вариант подходит далеко не для каждого диска. Например, диски с играми, лицензионными базами данных и т.п. при копировании на винчестер нередко отказываются работать. В таких случаях придется пойти другим путем: создать на компьютере виртуальные CD- и DVD-приводы и поместить в них образы часто используемых дисков. Технически это предполагает прохождение двух этапов. Вначале потребуется создать нужные файлы образов дисков (то есть виртуальные компакт-диски) — это можно сделать с помощью программ для записи и копирования CD-DVD-дисков, и сохранить данные образы на жестком диске. А затем необходимо воспользоваться специальной программой-эмулятором виртуальных накопителей с помощью которой придется создать нужное число виртуальных CD/DVD-приводов и подключить в каждом из них по подготовленному образу диска.

В итоге с виртуальными компакт-дисками можно будет работать точно так же, как и с настоящими. Более того, это намного быстрее и комфортнее. Во-первых, ускорится доступ к информации, поскольку с жесткого диска информация считывается гораздо быстрее, чем с компакт-диска, к тому же для запуска виртуального диска не требуется вставлять диск в накопитель (достаточно щелкнуть по его иконке). Во-вторых, уменьшится вероятность потери ценных данных в результате выхода дисков из строя, ведь CD- и DVD-диски не вставляются в накопитель, а потому их поверхность не изнашивается. Кроме того, открываются дополнительные возможности, например, применение виртуальных компакт-дисков на компьютерах, не имеющих физического CD/DVD-привода.

Исходя из всего этого, было принято решение разработать программное средство, позволяющее осуществлять копирование и виртуализацию оптических дисков.

## **1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

### **1.1 Общие сведения.**

Название: «Виртуальный CD-ROM».

«Виртуальный CD-ROM» – программа для создания виртуальных приводов.

### **1.2 Назначения и цели программного средства.**

Назначения: программа предназначена для виртуализации оптических дисков.

Цели: облегчение и ускорение работы с оптическими дисками, уменьшение количества CD- и DVD-дисков, требующихся для использования как на работе, так и на домашнем ПК.

### **1.3 Требования к программному средству**

Данная программа должна уметь создавать несколько активных виртуальных приводов, изменять размер существующих приводов, поддерживать популярные форматы образов (iso, bin) и иметь удобный и понятный интерфейс.

## 2. ОБЗОР ЛИТЕРАТУРЫ

В этом разделе будут рассмотрены основные теоритические сведения, необходимые для создания программы «Виртуальный CD-ROM».

### 2.1 Драйвер виртуального диска

В Windows виртуальные диски реализуются с помощью драйверов режима ядра (kernel mode drivers). Драйвер реализуется как набор процедур, каждая из которых предназначена для реализации отдельного типа обращений к драйверу со стороны диспетчера ввода/вывода. Процедуры, которые необходимо поддерживать драйверу приведены в таблице 1.

Таблица 1

Процедура	Описание
DriverEntry	Выполняется при загрузке драйвера операционной системой. Здесь драйвер регистрирует свои остальные точки входа и выполняет свою общую инициализацию.
Unload	Вызывается при выгрузке драйвера. Здесь необходимо освободить все затребованные ресурсы.
AddDevice	Здесь создаётся объект-устройство, соответствующий полученному уведомлению от менеджера устройств, и выполняется инициализация данных, специфичных для данного устройства.
DispatchPnP	Выполняет обработку специфичных Plug&Play запросов , таких как инициализация устройства, таких как инициализация устройства, остановка, удаление устройства и обрабатывать остальные запроса
DispatchPower	Выполняет обработку запросов по управлению питанием устройства
DispatchSystemControl	Обрабатывает запросы от подсистемы инструментария Windows (WMI)
DispatchCreate, DispatchClose, DispatchRead, DispatchWrite	Обслуживают запросы на чтение запись данных для устройства.

#### 2.1.1 Обработка запросов Plug and Play

В процессе работы диспетчер ввода вывода может динамически управлять состоянием устройства: запускать, останавливать и выгружать. Реализация этих функций драйвером устройства хранения обеспечивает при

обработке специфичных PnP IRP пакетов. В таблице 2 приведены описания IRP пакетов, которые должны поддерживаться.

Таблица 2

IRP_MN_Xxx	Описание
IRP_MN_START_DEVICE	Инициализация устройства с заданными ресурсами
IRP_MN_QUERY_STOP_DEVICE	Проверка осуществимости остановки устройства для перераспределения ресурсов
IRP_MN_STOP_DEVICE	Остановка устройства с потенциальной возможностью перезапуска или удаления из системы
IRP_MN_CANCEL_STOP_DEVICE	Уведомляет, что предыдущий запрос на остановку не получит дальнейшего развития
IRP_MN_QUERY_REMOVE_DEVICE	Проверка осуществимости безопасного удаления устройства
IRP_MN_REMOVE_DEVICE	Выполнить безопасное удаление устройства
IRP_MN_CANCEL_REMOVE_DEVICE	Уведомляет, что предыдущий запрос на удаление не получит дальнейшего развития
IRP_MN_SURPRISE_REMOVAL	Уведомляет, что устройство было удалено без предварительного предупреждения

Применительно к виртуальному диску большая часть этих сообщений не влечет каких-либо дополнительных действий, т.к. у виртуального диска нет дополнительных буферов, данные с которые должны быть записаны на диск при остановке устройства, или поддержки функций управления электропитанием устройства.

### 2.1.2 Обработка расширенных запросов

Для управления самим устройством диспетчер ввода/вывода посылает драйверу пакет с кодом управления вводом/выводом (IOCTL) [1].

Некоторые коды управления:

IOCTL\_DISK\_GET\_PARTITION\_INFO – сообщить о типе, размере и природе раздела диска.

IOCTL\_DISK\_IS\_WRITABLE – проверка можно ли на диск записывать данные

IOCTL\_DISK\_SET\_PARTITION\_INFO – изменить тип раздела

IOCTL\_DISK\_GET\_LENGTH\_INFO – получить длину указанного диска, тома или раздела.

### 2.1.3 Инициализация драйвера

Для инициализация драйвера вызывается следующая процедура (DriverEntry) [2]:

```
NTSTATUS DriverEntry(IN PDRIVER_OBJECT DriverObject, IN  
PUNICODE_STRING RegistryPath)
```

Тип NTSTATUS, соответствующий возвращаемому значению, определяет тип ошибки. Многие функции драйвера возвращают значение этого типа. Если работа проходит успешно, результат принимает значение STATUS\_SUCCESS.



### 3. СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ

Данный проект включает в себя непосредственно драйвер виртуального диска и приложение, осуществляющее связь между пользователем и драйвером. Драйвер обеспечивает создание виртуального диска. Драйвер получает IRP (I/O request packet) запросы, обрабатывает их и возвращает результат. Приложение обслуживает запросы от ОС к виртуальным дискам. Структурная схема программы приведена ниже.

#### 3.1 Структурная схема программы

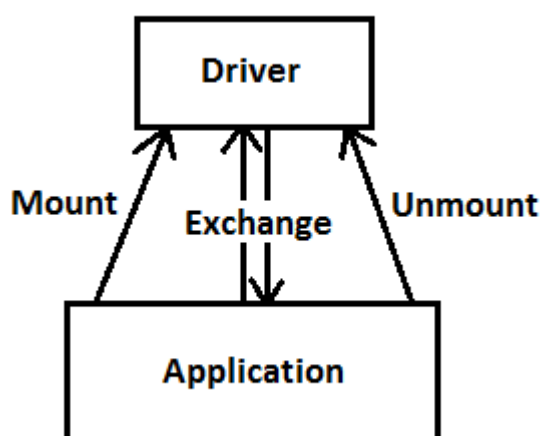


Рис. 3.1 – Структурная схема программы

#### 3.2 Взаимодействие между формами

В программе реализовано 4 формы: MainForm, AboutForm, AddForm, ExtendSizeForm. Взаимодействие между формами представлено на рис 3.2

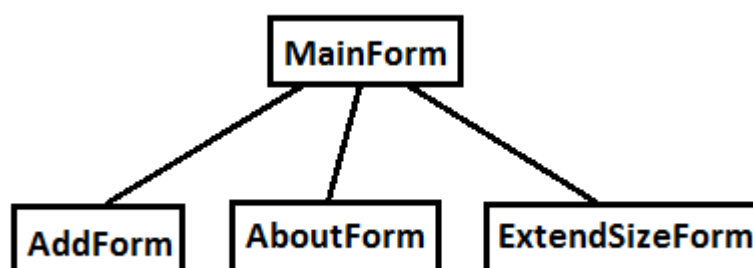


Рис. 3.2 – Взаимодействие между формами

MainForm – главная форма приложения. Из этой формы можно перейти в форму добавления нового диска (AddForm), форму изменения размера существующего диска (ExtendSizeForm) и форму «О программе» (AboutForm).

## 4. ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ

Данный раздел включает в себя описание основных методов программы и их листинги.

```
NTSTATUS DriverEntry(IN PDRIVER_OBJECT DriverObject, IN  
PUNICODE_STRING RegistryPath);
```

Где: `DriverObject` - указатель на объект-драйвер, соответствующий загружаемому драйверу; `RegistryPath` - указатель на строку в формате Unicode с именем ключа реестра, соответствующего загружаемому драйверу. Возвращаемое значение имеет тип `NTSTATUS`. Если возвращается успешный статус завершения, диспетчер ввода/вывода немедленно позволяет производить обработку запросов к объектам-устройствам, созданным драйвером. Во всех остальных случаях драйвер не загружается в память, и запросы к нему не передаются. `DriverEntry` определяет аппаратное обеспечение, которое драйвер будет контролировать. Это аппаратное обеспечение выделяется драйверу, то есть помечается как находящееся под управлением данного драйвера. В данном методе выполняется вызов `IoCreateDevice` для создания объекта устройства для каждого физического или логического устройства под управлением данного драйвера, в процессе которого инициализируется структура расширения устройства для каждого созданного объекта устройства. В данном методе необходимо сообщить системе остальные процедуры обработки пакетов `IRP`.

Выгрузка драйвера состоит из двух частей: удаление объекта устройства и выгрузка самого драйвера.



## **ЗАКЛЮЧЕНИЕ**

## СПИСОК ЛИТЕРАТУРЫ

1. [https://msdn.microsoft.com/en-us/library/windows/desktop/aa363979\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa363979(v=vs.85).aspx)
2. <https://msdn.microsoft.com/en-us/library/windows/hardware/ff544113%28v=vs.85%29.aspx>