

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт искусственного интеллекта

Кафедра программного обеспечения систем радиоэлектронной аппаратуры

КУРСОВАЯ РАБОТА

по дисциплине: «Технология создания программного продукта»

(наименование дисциплины)

Тема курсовой работы: «ПО уничтожения данных на жестких дисках»

Студент группы: КМБО-02-19, Гурин Михаил Вадимович

(учебная группа, фамилия, имя, отчество студента) (подпись студента)

Руководитель курсовой работы: Абрамов Антон Сергеевич

(Должность, звание, ученая степень) (подпись руководителя)

Работа представлена к защите «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Допущен к защите «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Москва 2022

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc116868407)

[1. ТРЕБОВАНИЯ К СИСТЕМЕ 4](#_Toc116868408)

[1.1. Функциональные требования 4](#_Toc116868409)

[1.2. Нефункциональные требования 5](#_Toc116868410)

[2. ПРОЦЕСС РАЗРАБОТКИ 6](#_Toc116868411)

[2.1. Описание модулей 6](#_Toc116868412)

[2.2. Сценарии тестирования 7](#_Toc116868413)

[3. ОПИСАНИЕ ПРОДУКТА 10](#_Toc116868414)

[4. ВОЗМОЖНОСТИ СИСТЕМЫ 11](#_Toc116868415)

[5. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 12](#_Toc116868416)

[5.1. Требования для запуска 12](#_Toc116868417)

[5.2. Эксплуатация системы 13](#_Toc116868418)

[5.2.1. Использование средств графического интерфейса 13](#_Toc116868419)

[5.2.2. Использование интерфейса командной строки. 22](#_Toc116868420)

[ЗАКЛЮЧЕНИЕ 25](#_Toc116868421)

[СПИСОК ЛИТЕРАТУРЫ 26](#_Toc116868422)

ВВЕДЕНИЕ

Последним, но оттого не менее важным, этапом в жизненном цикле хранения информации является ее уничтожение. Данный процесс может проходить как естественным путем, так и проводиться специально с целью контроля вывода данных из эксплуатации и сохранения их конфиденциальности.

Сам процесс искусственного уничтожения может проводится по-разному. В зависимости от типа, можно выделить два вида стирания, физический – при котором, носитель уничтожается вместе с хранимой на нем информацией, и программный, при котором уничтожаются только данные, но не устройство хранения.

Все подвиды программного уничтожения данных базируются на одном и том же принципе – записи определенного шаблона (паттерна), во все или лишь в некоторые ячейки памяти. Количество циклов перезаписи и вариативность шаблонов при этом зависит от выбранного алгоритма уничтожения, один из самых простых вариантов – одноцикловый проход с записью блоков фиксированного размера, содержащих нулевые байты. Такой вариант пригоден для решения задач низкоуровневого форматирования, когда для формирования структуры разделов может потребоваться полная очистка диска от остаточных следов данных.

1. ТРЕБОВАНИЯ К СИСТЕМЕ
   1. Функциональные требования

* Уничтожение данных на выбранном накопителем, определенном как устройство типа «несъемный диск» с выбранным размером уничтожаемого за один шаг цикла блока и выбранными опциями.
* Вывод доступных для проведения процедуры уничтожения данных носителей информации в табличном виде, с отображением информации об индексе устройства, наименовании фирмы-производителя, модели и ревизии, суммарной емкости в общепринятых единицах измерения информации, а также о размере минимально доступного для стирания блока (сектора).
* Вывод сведений о текущем выбранном для стирания диске таких как: индекс, используемый на носителе информации стиль разделов (MBR или GPT). В случае если не один раздел не выбран, должно выводиться Not Selected и N/A соответственно.
* Для носителей информации, размеченных в системах разделов GPT или MBR отображение списка смонтированных разделов в отдельной таблице. Для каждого раздела должны содержаться сведения об индексе, метке тома и размере в общепринятых единицах измерения информации.
* Для тех версий Windows NT, которые не поддерживают устройства со стилем разделов GPT (2000/XP) обеспечить возможность ручного поиска GPT-разделов на диске, с целью предотвращения несанкционированной потери данных.
* Выбор размера стираемого блока при помощи графического интерфейса, как из стандартного шаблона, так путем ввода произвольного положительного целочисленного значения в байтах.
* Возможность прерывания текущей операции стирания.
* Возможность обновления информации о разделах и носителях информации в ходе выполнения программы в ручном режиме.
* Ведение лога операции в формате текстового файла, содержащего время выполнения операции, время начала, время окончания, статус операции, коды и статусы возникших в ходе выполнения операции, ошибок.
  1. Нефункциональные требования
* Данное программное обеспечение должно работать на персональных компьютерах класса IBM-PC под управлением операционной системы Windows NT не ниже версии 5.0.
* Отображение прогресса выполняемой операции, отзывчивость графического интерфейса во время проведения процедуры стирания.
* Защита от стирания системного раздела, раздела на котором расположен сам продукт, и раздела, на котором хранится лог операции.
* Наличие развитого интерфейса командной строки с возможностью интеграции в состав командных сценариев и скриптов автоматического управления с целью автономной работы без участия оператора.

1. ПРОЦЕСС РАЗРАБОТКИ
   1. Описание модулей

Данное программное обеспечение состоит из двух модулей. Первый модуль представляет собой оконное приложение Windows, в котором реализован сам процесс стирания информация, отображение сведений о дисках и вывод прогресса на экран. Второй модуль – вспомогательная библиотека динамической компоновки, содержащая экспортные функции для работы с таблицей разделов GPT, получения сведений о жесткий диск и записи информации.

Код модуля оконного приложения состоит из трех файлов:

* MainForm.pas содержит в себе описание классов формы, потока стирания и списка описателей, базированного на односвязном списке. Также в этом модуле содержатся описания всех констант, глобальных переменных и вспомогательных функций, необходимых для работы.
* ArgumentParser.pas содержит необходимые классы для реализации интерфейса командной строки – класс представляющий аргумент командной строки как сущность, а также класс парсера входных аргументов в двух исполнениях – регистрозависимом и регистронезависимом.
* HEPE.dpr содержит инициализационный код с защитой от запуска нескольких копий программы.

Код вспомогательной библиотеки динамической компоновки представлен в файле main.cpp, который содержит описание необходимых структур данных и набор экспортных функций для работы с разделами GPT, функции разрешения символьных ссылок и преобразования путей из формата NT в формат Dos, функции получения сведений о носителях информации, таких как серийный номер, название производителя, и т. д., а также вспомогательные функции для работы с файловым указателем и записи буфера с массивом нулевых байтов непосредственно на носитель данных.

Для разработки главного модуля использовалась среда программирования Delphi, язык программирования Object Pascal, для реализации вспомогательной библиотеки – среда Visual Studio, язык программирования С.

* 1. Сценарии тестирования

Для того, чтобы провести нижеизложенные сценарии тестирования, необходимо выполнить следующие шаги:

* Подключить и смонтировать диск доступный для стирания.
* Скопировать файлы программы на целевой компьютер.

В таблицах 1 – 3 представлены сценарии тестирования данного продукта:

Таблица 1. Сценарий тестирования – проверка ручного сканирования дисков GPT.

|  |  |  |
| --- | --- | --- |
| Название | 1). Проверка ручного сканирования дисков GPT | |
| Действие | Ожидаемый результат: | Результат теста:  Пройден  Провален |
| Предусловие |  |  |
| Имеется как минимум один диск, доступный для стирания | В верхней таблице представлен как минимум один диск. | Пройден |
| В системе имеется как минимум один диск, имеющий, систему разделов GPT. |  |  |
| Шаги теста: |  |  |
| Запустить программу | Появится графический интерфейс программы | Пройден |
| Выбрать диск с разметкой GPT из представленного списка | Соответствующий пункт в таблице будет выделен | Пройден |
| Нажать на кнопку “Scan for GPT partitions” | В нижней таблице будет отображен список GPT разделов | Пройден |

Таблица 2. Сценарий тестирования – проверка блокировки раздела.

|  |  |  |
| --- | --- | --- |
| Название | 2). Убедиться, что разделы, смонтированные на целевом диске, блокируются. | |
| Действие | Ожидаемый результат: | Результат теста:  Пройден  Провален |
| Предусловие |  |  |
| Имеется как минимум один диск, доступный для стирания | В верхней таблице представлен как минимум один диск. | Пройден |
| На диске присутствует по крайней мере один смонтированный системой раздел | В нижней таблице представлен как минимум один раздел. | Пройден |
| Шаги теста: |  |  |
| Запустить программу | Появится графический интерфейс программы | Пройден |
| Выбрать диск со смонтированным разделом | В нижней таблице представлен как минимум один смонтированный раздел | Пройден |
| Запустить процедуру стирания, нажав кнопку “Start” | После утвердительного ответа на сообщение программы, процесс стирания начнется | Пройден |
| Во время процедуры стирания, проверить возможность доступа, открыв том в проводнике Windows | Будет выведено сообщение об ошибке на экран, при попытке доступа к заблокированному разделу | Пройден |

Таблица 3. Сценарий тестирования – проверка достоверности результата уничтожения.

|  |  |  |
| --- | --- | --- |
| Название | 3). Убедиться в том, что выбранный диск действительно стирается. | |
| Действие | Ожидаемый результат: | Результат теста:  Пройден  Провален |
| Предусловие |  |  |
| Имеется как минимум один диск, доступный для стирания | В верхней таблице представлен как минимум один диск. | Пройден |
| Шаги теста: |  |  |
| Запустить программу | Появится графический интерфейс программы | Пройден |
| Выбрать диск со смонтированным разделом | В нижней таблице представлен как минимум один смонтированный раздел | Пройден |
| Запустить процедуру стирания, нажав кнопку “Start” | После утвердительного ответа на сообщение программы, процесс стирания начнется | Пройден |
| После завершения процедуры стирания, открыть оснастку управления дисками Windows. | При запуске оснастки будет предложено инициализировать пустой жесткий диск. | Пройден |

1. ОПИСАНИЕ ПРОДУКТА

Программное обеспечение стирания данных на жестких дисках «HEPE» предназначено проведения процедуры одноциклового полного стирания данных с использованием шаблона, состоящего из массива нулевых байтов жестких магнитных дисков, твердотельных накопителей информации и иных устройств, которые в системах семейства Windows NT определяются как логическое устройство типа «несъемный диск».

Данный продукт предназначен в среде выполнения операционных систем семейства Windows NT, обладает средствами как ручной, так и автономной работы, а также поддерживается гибкая настройка параметров, как средствами графического интерфейса, так и аргументами командной строки, что позволяет использовать данную программу в составе скриптов автоматического управления, командных сценариях, и т. д.

1. ВОЗМОЖНОСТИ СИСТЕМЫ

* Вывод сведений о присутствующих в системе носителях информации в табличном виде.
* Вывод сведений о текущем выбранном для стирания носителе информации.
* Вывод сведений о смонтированных разделах, которые расположены на выбранном носителе информации в табличном виде.
* Наличие кнопки ручного сканирования GPT-разделов для тех версий Windows NT, которые не поддерживают стиль разделов GPT.
* Возможность выбора размер стираемого за один шаг цикла блока, как из стандартных значений, так и произвольных в байтах.
* Графический интерфейс выбора дополнительных опций, таких как пропуск ошибок, сохранение лога с диалогом выбора местоположения, закрытие после завершения операции, тихий режим, при котором не выводятся предупреждающие диалоговые окна, выключение компьютера после завершения операции, отключение блокировки и сканирования разделов.
* Вывод прогресса текущей операции при помощи индикатора прогресса.
* Ручное обновление данных о дисках и разделах.
* Завершение работы Windows в ручном режиме.

1. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ
   1. Требования для запуска

Для запуска данного продукта не требуется разрешения никаких внешних дополнительных зависимостей, требуется наличие только трех файлов:

* HEPE.exe – главный исполняемый модуль программы.
* HEPEH.dll – вспомогательная библиотека динамической компоновки, единственная зависимость главного модуля, которая реализована в рамках данной курсовой работы.
* hepeargs.txt – текстовый файл, в котором находится подробное описание ключей командной строки. Содержимое этого файла выводится в стандартном текстовом редакторе Windows, при запуске программы с ключом /help. В случае если этого файла не окажется в директории с программой, при запуске с этим ключом будет выдано сообщение об ошибке.

Внимание: для запуска программы, необходимо, чтобы разрешение экрана было минимум 1024x768, в противном случае при запуске будет выдано сообщение об ошибке. Аргументы командной строки для отключения данной проверки при запуске, а также настройки разрешения экрана описаны в руководстве по использованию интерфейса командной строки далее.

Также, для правильной работы, файлы следует располагать таким образом, чтобы соблюдалось 3 условия:

* Стираемым диском не может являться диск, на котором расположена запущенная в данный момент копия операционной системы.
* Стираемым диском не может являться диск, на котором располагается копия данного программного обеспечения.
* Стираемым диском не может являться диск, на котором расположен файл лога операции.

Важно также отметить, что при определении выполнения данных условий также будут разрешаться все символьные ссылки на пути к программе или файлу лога.

Настоятельно рекомендуется запускать данный продукт из-под среды предустановки Windows PE, запущенной с внешнего носителя информации (флэш-память, компакт-диск, и т. д.). Таким образом можно практически стопроцентно гарантировать выполнение трех вышеизложенных условий.

* 1. Эксплуатация системы

В зависимости от поставленных задач, возможно использование двух разных типов взаимодействия с программой:

* С помощью средств графического интерфейса.
* С помощью интерфейса командной строки.

Особенности использования двух типов взаимодействия изложены ниже.

* + 1. Использование средств графического интерфейса

Все взаимодействие с графическим интерфейсом происходит через главное окно программы. Интерфейс главного окна программы представлен на рисунке 1:

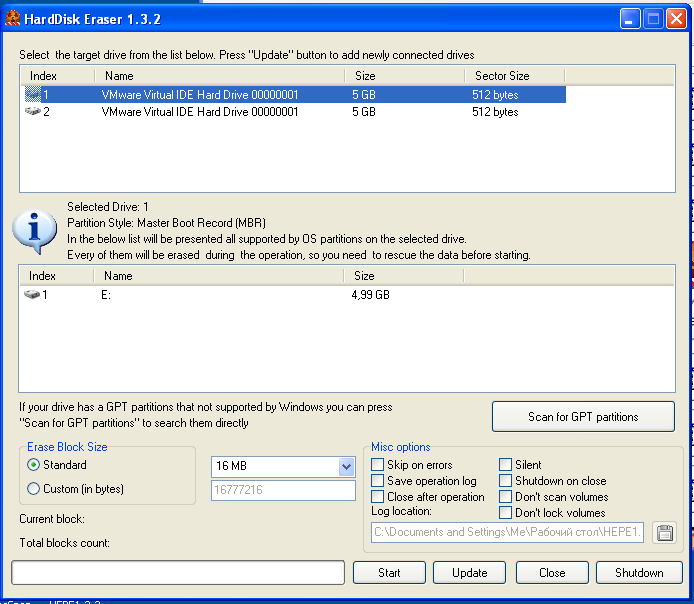


Рисунок 1. Главное окно программы

В верхней таблице представлен список дисков доступных для стирания: индекс, производитель, модель, ревизия, емкость и размер сектора. Ниже написана информация о текущем выбранном диске, такая как индекс и стиль таблицы разделов, расположенной на диске. В случае если при запуске программы не будет найдено ни одного доступного для стирания диска, при запуске будет выдано предупреждение, и информационный знак сменится на предупреждающий (Рисунки 2, 3).

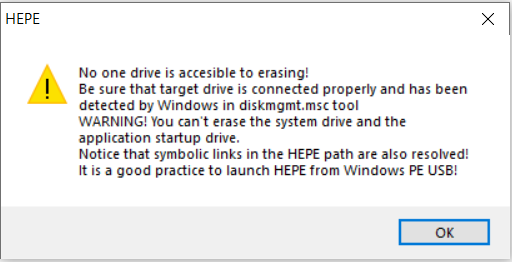


Рисунок 2. Предупреждение программы

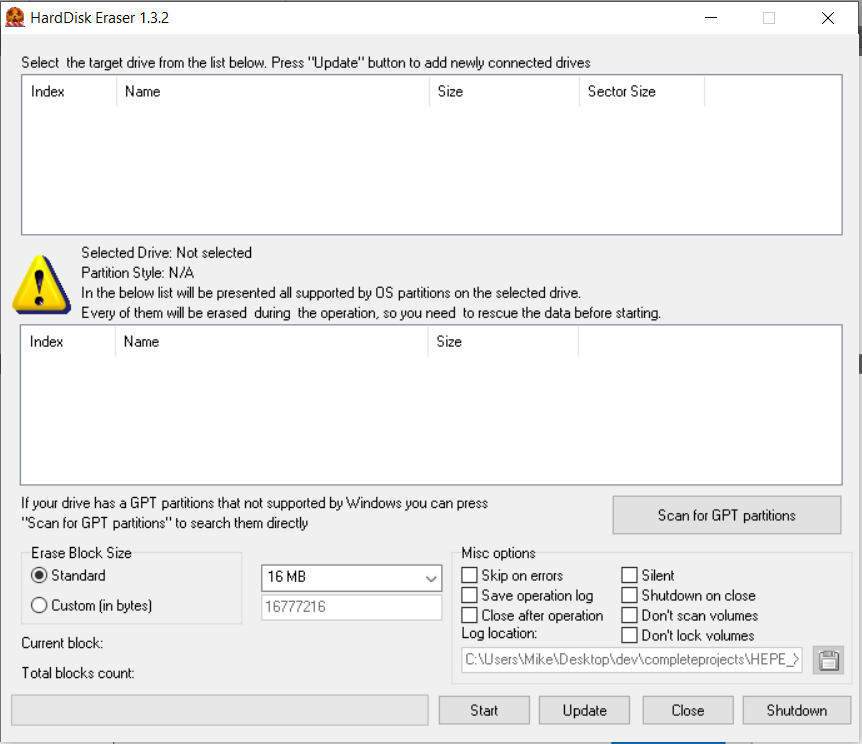


Рисунок 3. Интерфейс программы, в случае если ни одного доступного для стирания диска не было найдено

В нижней таблице будет отображена информация о всех смонтированных разделов, которые расположены на выбранном диске, если таковые есть, а именно: индекс, буква и метка тома, а также его размер (Рисунок 1).

Для ручного поиска GPT разделов предусмотрена кнопка «Scan for GPT partitions». В случае если на диске действительно присутствует таблица разделов типа GPT, то в нижней таблице будет отображена информация аналогичная вышеуказанной, за исключением того, что перед индексом будет проставлена метка N/A, а также вместо метки будет указан уникальный идентификатор (GUID) раздела (Рисунок 4).

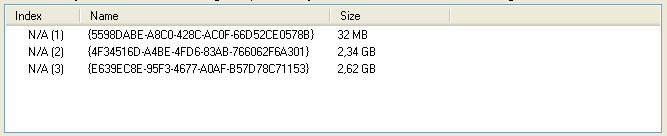


Рисунок 4. Результат ручного поиска GPT разделов

В случае если раздел не содержит таблицу разделов типа GPT, или диск не выбран, будет выдано предупреждающее сообщение (Рисунки 5-6).



Рисунок 5. Сообщение о том, что выбранный раздел, не имеет таблицу разделов типа GPT

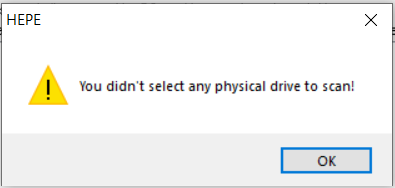


Рисунок 6. Сообщение о том, что раздел для сканирования не выбран

В группе «Erase block size» можно выбрать размер блока, который будет стираться за один раз. Размер может быть выбран как из списка стандартных значений, в котором описан диапазон блоков от 4 килобайт до 128 мегабайт, так и указан произвольно, в байтах (Рисунок 1). По умолчанию размер блока равняется 16 мегабайтам.

Внимание: размер блока обязательно должен быть кратен размеру сектора диска и не равняться нулю. В случае если это требование не будет выполнено, при начале операции будет показано предупреждающее сообщение, и операция будет прервана (Рисунок 7). Также, размер блоков больше 16 мегабайт поддерживается не во всех версиях Windows.

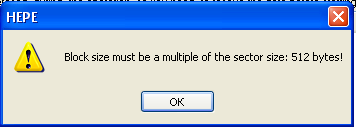


Рисунок 7. Предупреждающее сообщение о том, что размер блока не кратен размеру сектора.

В группе «Misc. Options» представлен список дополнительных опций, которые могут быть активированы до начала операции:

* «Skip on errors» - сообщения об ошибках ввода-вывода и ошибках открытия разделов для блокировки с кодом 2 (путь не найден), произошедших непосредственно в ходе операции выводиться не будут. Такие ошибки не являются критическими, и не влияют на ход проведения процедуры.
* «Save operation log» - позволяет сохранить лог операции в текстовый файл, местоположение которого определяется пользователем. По умолчанию находится в директории с программой, и имеет имя «hepe<версия>.log». Внимание: лог не должен находиться на диске, который будет стираться. Операция не может быть начата, пока данное условие не выполнится пользователем. Местоположение лога может быть выбрано по нажатии кнопки с пиктограммой дискеты в нижней части группы, текущий выбранный путь будет указан в текстовом поле рядом. Файл лога содержит сведения о времени начала операции, заблокированных на период проведения операции разделах, общем размере диска, размере стираемого блока, общем количестве блоков, которое равно частному от общего размера и размера блока, названии диска, индексах успешно уничтоженных блоков, сведения об ошибках ввода-вывода, критических ошибках, статусе завершения операции и количестве затраченного времени.
* «Close after operation» - программа будет закрыта автоматически после завершения операции.
* «Silent» - программа не будет выдавать предупреждение при начале операции, а также не будет выводить сообщения о критических ошибках, произошедших в ходе выполнения операции.
* «Shutdown on close» - Windows автоматически завершит работу после окончания операции.
* «Don’t scan volumes» - не сканировать диски на наличие смонтированных разделов, также отключает возможность ручного сканирования GPT разделов и активирует опцию «Don’t lock volumes».
* «Don’t lock volumes» - не блокирует смонтированные разделы на период проведения операции. Внимание: активация данного флага может привести к несанкционированной потере данных, вследствие чего настоятельно не рекомендуется активация двух последних опций.

Для того, чтобы выключить компьютер нажмите на кнопку «Shutdown» в нижней части окна (Рисунок 1). Данная опция может пригодиться в среде предустановки Windows PE, где нет прямой возможности для выключения.

Для того, чтобы закрыть программу, нажмите на кнопку «Close» в нижней части окна. Данная опция может пригодиться в том случае, если программа была запущена в полноэкранном режиме.

Для того, чтобы обновить и актуализировать информацию о доступных дисках, нажмите кнопку «Update» в нижней части экрана. В случае если в процессе обновления не будет обнаружено ни одного доступного для стирания жесткого диска, будет выдано предупреждающее сообщение, и информационный знак сменится на предупреждающий (Рисунки 2, 3).

Для того, чтобы начать операцию стирания, нажмите кнопку «Start» в нижней части экрана. Для того, чтобы процесс начался, необходимо, чтобы были выполнены следующие условия:

* Выбран диск для стирания.
* Указан размер стираемого блока, кратный размеру сектора.

В случае если не выполнено первое условие, при попытке начать операцию, пользователю будет показано предупреждающее сообщение, показанное на Рисунке 8.

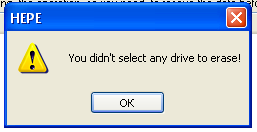


Рисунок 8. Предупреждение

В случае, если не выполнено второе условие, будет показано сообщение, которое представлено на Рисунке 7.

Если все условия выполнены и опция «close after operation» не активирована, перед началом будет показано предупреждение о том, что все данные будут уничтожены, и что операцию отменить невозможно, на которое необходимо ответить утвердительно, нажав кнопку «Да» (Рисунок 9).

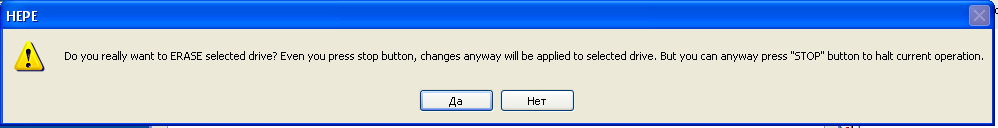


Рисунок 9. Предупреждение перед началом операции

После начала процедуры стирания, большая часть элементов управления окажется заблокирована, появиться кнопка «Stop» для прерывания операции, в графе Current Block будет отображен текущий стираемый блок, в графе Total Blocks Count будет отображаться общее число блоков, которое равно частному от размера диска и размера сектора. Также прогресс будет выведен на индикатор в нижней части экрана (Рисунок 10).

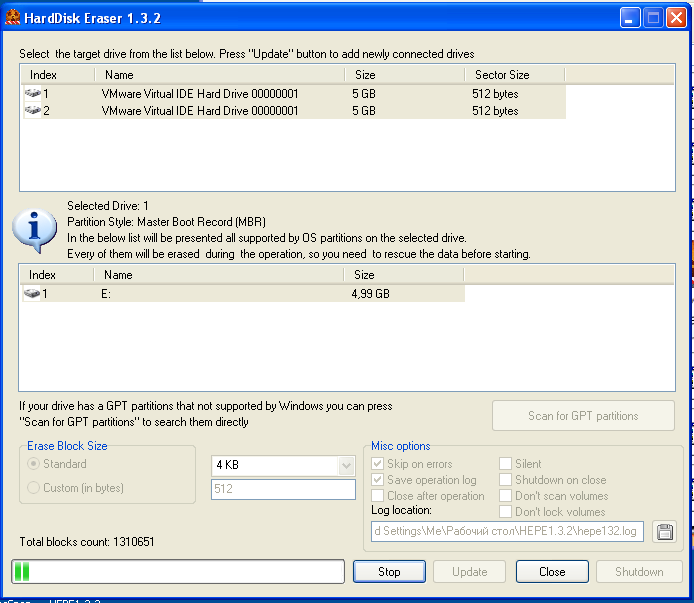


Рисунок 10. Главное окно программы во время проведения процедуры стирания жесткого диска

Внимание: во время проведения операции программа не может быть закрыта. При нажатии на кнопку «Close» или пиктограмму «Закрыть» в верхнем правом углу экрана, будет показано соответствующее предупреждение (Рисунок 11).

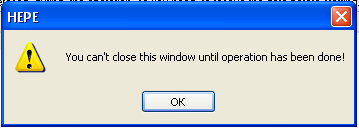


Рисунок 11. Предупреждение

Для того, чтобы отменить операцию, необходимо нажать на кнопку «Stop» в нижней части окна. На появившееся предупреждение необходимо ответить утвердительно (Рисунок 12).

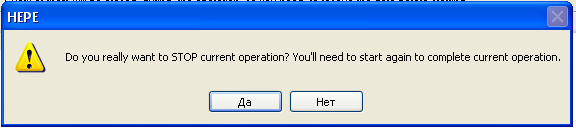


Рисунок 12. Предупреждение

После того, как операция завершится, пользователю будет показано сообщение о завершении операции (Рисунок 13).



Рисунок 13. Сообщение об успешном завершении операции

Если перед началом была активирована опция «Save operation log», то после завершения операции, выбранном местоположении будет создан файл лога. Пример такого файла приведен на Рисунке 14:

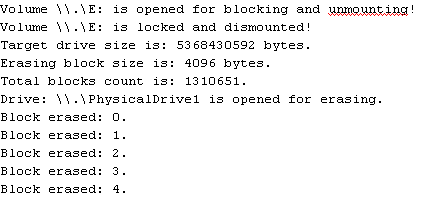


Рисунок 14. Фрагмент файла-лога.

* + 1. Использование интерфейса командной строки.

Для того, чтобы настраивать параметры программы через интерфейс командной строки, и была возможность интеграции в состав командных сценариев, приложением поддерживается система аргументов командной строки. Каждый аргумент должен начинаться со знаков / или -, ключи без знаков являются недопустимыми. Значения аргументов задаются либо через знак =, идущий сразу после аргумента, либо через одиночный пробел. Порядок аргументов может быть произвольным. Список доступных для использования ключей командной строки представлен ниже:

* autostart – программа автоматически начнет процедуру стирания после запуска. Данный параметр может использоваться только в том случае когда указан актуальный индекс доступного для стирания жесткого диска с помощью ключа driveindex.
* autoclose – программа будет автоматически закрыта после завершения операции, действует аналогично флагу «Close after operation» графического интерфейса.
* driveindex – указывает на целочисленное положительное значение актуального индекса доступного для стирания жесткого диска. Данный параметр необходим для использования флага autostart.
* ebstype – тип размерности стираемого блока. Допустимы два значения: standard – для выбора из списка стандартных значений, и custom для указания произвольного целочисленного положительного значения блока, размер которого должен быть кратен размеру сектора целевого диска.
* ebsstandardsize – индекс стандартного размера стираемого блока, доступные значения от 0 до 15 – каждый из номеров соответствует размеру, представленному в выпадающем списке в графическом интерфейсе программы. Любой другой индекс, а также отрицательные значения не могут быть использованы. Этот флаг обрабатывается только в том случае, если указан параметр ebstype со значением standard.
* ebscustomsize – указывает произвольный размер стираемого блока в байтах, который должен быть положительным целочисленным значением, которое кратно размеру сектора выбранного диска. Данный флаг обрабатывается только, когда указан параметр ebstype со значением custom.
* silent – при активации этого флага, все предупреждающие сообщения, сообщения о критических ошибках и сообщение о статусе завершения операции отображаться не будет.
* skiponerrors – при активации этого флага, все предупреждающие сообщения об ошибках ввода-вывода, возникших во время проведения операции, а также сканирования дисков, появляться не будут.
* saveoperationlog – при включении этого флага, программа будет сохранять лог проведения операции. По умолчанию расположением лога является директория, в которой расположена программа.
* saveoperationloglocation – этот параметр позволяет задать произвольное расположение файла-лога. Значение должно ссылаться на существующую директорию и не должно располагаться на диске, предназначенном для стирания. Также не разрешаются пустые значения.
* help – отображает файл справки на экране (hepeargs.txt). Если этот файл не был найден в директории с программой, то будет выведено сообщение об ошибке на экран.
* dontcheckres – программа не проводит проверку разрешения экрана перед запуском. По умолчанию, перед запуском проверяется разрешение экрана, требуется чтобы ширина была больше 700 пикселей, высота – больше 650 пикселей. Если это условие не выполняется, при запуске будет показано сообщение об ошибке и работа программы будет завершена. Данный флаг не доступен для использования из графического интерфейса.
* updatetominres – при активации этого флага, программа попытается увеличить разрешение экрана до минимально разрешенного (1024 x 768), если оно не будет соответствовать минимально разрешенному. Данный флаг не доступен для использования из графического интерфейса.
* upscaleto32bits – при активации этого флага, глубина цвета будет увеличена до 32 бит автоматически, в случае если флаг updatetominres активирован, и увеличение разрешения экрана действительно будет проводиться. Данный флаг не доступен для использования из графического интерфейса.
* topmost – главное окно программы будет показано поверх всех окон. Данный флаг не доступен для использования из графического интерфейса.
* fullscreen – главное окно программы будет отображено в полноэкранном режиме. Данный флаг не доступен для использования из графического интерфейса.

Остальные доступные для использования флаги описаны в файле справки, который может быть выведен на экран при запуске программы c ключом help.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной курсовой работы удалось успешно решить все поставленные задачи, а именно:

* Были сформированы требования к программному продукту.
* Были разработаны сценарии тестирования согласно установленным требованиям.
* Были разработаны все требуемые программные модули.
* Работоспособность написанного программного кода была проверена с помощью тестовых сценариев.
* Было написано подробное руководство пользователя.

Ввиду вышесказанного, был реализован полный жизненный цикл продукта.

Исходные тексты и файлы проектов моделей, а также данный документ загружен на платформу GitHub по адресу:

https://github.com/RaymondRedwood/HEPE

СПИСОК ЛИТЕРАТУРЫ

1. Гецци К., Джазаейри М., Мандриоли Д. Основы инженерии программного обеспечения. 2-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2015. – 832 с.: ил.

2. Орлов С. Технологии разработки программного обеспечения. Разработка сложных программных систем. Учебное пособие. СПб: Питер, 2016. 480 с, ил.

3. Брауде Э.Дж. Технология разработки программного обеспечения (2014)

4. Благодатских В.А. Стандартизация разработки программных средств: учеб. пособие /В.А. Благодатских, В.А. Волнин, К.Ф. Поскакалов; под ред. О.С. Разумова. — М. : Финансы и статистика, 2016. — 288 с : ил.

5. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. СПб: Символ-Плюс, 2016. – 304 с., ил.

6. Д. Кознов. Введение в программную инженерию: Учебный курс. М.: Интуит, 2016.

7. Абрамян, Михаил Visual C# на примерах (+ CD-ROM) / Михаил Абрамян. - М.: БХВ-Петербург, 2017. - 496 c.

8. Агуров, Павел C#. Сборник рецептов (+CD-ROM) / Павел Агуров. - М.: БХВ-Петербург, 2017. - 432 c.

9. Бишоп, Дж. С# в кратком изложении / Дж. Бишоп, Н. Хорспул. - М.: Бином. Лаборатория знаний, 2014. - 472 c.

10. Васильев, Алексей C#. Объектно-ориентированное программирование / Алексей Васильев. - М.: Питер, 2013\4. - 320 c.

11. Зиборов, В. В. Visual C# 2013 на примерах / В.В. Зиборов. - М.: БХВ-Петербург, 2015. - 480 c.

12. Зиборов, Виктор Visual C# 2015 на примерах / Виктор Зиборов. - М.: "БХВ-Петербург", 2016. - 432 c.

13. Касаткин, А. И. Профессиональное программирование на языке си. Управление ресурсами / А.И. Касаткин. - М.: Высшая школа, 2015. - 432 c.

14. Культин, Н. Microsoft Visual C# в задачах и примерах (+ CD-ROM) / Н. Культин. - М.: БХВ-Петербург, 2018. - 314 c.

15. Подбельский, В. В. Язык С#. Базовый курс / В.В. Подбельский. - М.: Финансы и статистика, 2018. - 408 c.

16. Рендольф, Ник Visual Studio 2010 для профессионалов / Ник Рендольф и др. - М.: Диалектика, 2018. - 584 c.

17. Скит, Джон C# для профессионалов. Тонкости программирования / Джон Скит. - М.: Вильямс, 2014. - 608 c.

18. Троелсен, Эндрю Язык программирования C# 5.0 и платформа .NET 4.5 / Эндрю Троелсен. - М.: Вильямс, 2015. - 633 c.

19. Фленов, Михаил Библия C# / Михаил Фленов. - М.: БХВ-Петербург, 2014. - 560 c.