

Министерство образования Республики Беларусь Учреждение
образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетейКафедра
информатики
Дисциплина «Операционные системы и среды»

«К ЗАЩИТЕ ДОПУСТИТЬ»
Руководитель курсового проекта
_____.А.В. Давыдчик
_____._____.2023

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проектуна
тему:

**«Дисковые утилиты: проверки, дефрагментация, иное
обслуживание»**

БГУИР КП 1-40 01 01 009 ПЗ

Выполнил студент группы 053504
Щегловская Мария Александровна

(подпись студента)

Курсовой проект представлен на
проверку _____._____.2023

(подпись студента)

Минск 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ИСПОЛЬЗУЕМЫ ТЕОРЕТИЧЕСКИЕ СОСТОВЛЯЮЩИЕ ПРОЕКТА.....	4
НЕОБХОДИМОСТЬ РАЗРАБОТКИ.....	13
ВЫБОР ИНСТРУМЕНТОВ РАЗРАБОТКИ	15
РЕАЛИЗАЦИЯ ПРОГРАММНОГО СРЕДСТВА.....	17

ВВЕДЕНИЕ

Файловая система – это часть операционной системы, назначение которой состоит в том, чтобы обеспечить пользователю удобный интерфейс при работе с данными, хранящимися на диске, и обеспечить совместное использование файлов несколькими пользователями и процессами.

На компьютерах под управлением операционной системой Windows информация на запоминающих устройствах организована файловой системой NTFS. Одним из свойств данной файловой системы является так называемая фрагментация файлов. Она представляет собой процесс разбиения каждого файла на небольшие фрагменты при записи на диск.

По мере использования информации (добавление новой, удаление или перемещение старой) разные блоки одного файла физически могут оказаться записанными не подряд. В результате, считывающее устройство тратит большее количество времени на поиск и чтение блоков одного файла, что приводит к увеличению времени чтения.

Решение данной проблемы заключается в выполнении дефрагментации.

Дефрагментация – процесс обновления и оптимизации логической структуры раздела диска с целью обеспечения хранения файлов в непрерывной последовательности кластеров. После дефрагментации ускоряется чтение и запись файлов, а, следовательно, и работа программ. [4]

Процесс выполнения дефрагментации логического диска можно реализовать как вручную, так и программно, запустив необходимый скрипт.

Целью моей работы является создание в среде разработки Visual Studio Code приложения (системной утилиты), выполняющей дефрагментацию логических дисков.

ИСПОЛЬЗУЕМЫЕ ТЕОРЕТИЧЕСКИЕ СОСТАВЛЯЮЩИЕ ПРОЕКТА

Файловая система - это часть операционной системы, назначение которой состоит в том, чтобы обеспечить пользователю удобный интерфейс при работе с данными, хранящимися на диске, и обеспечить совместное использование файлов несколькими пользователями и процессами. В широком смысле понятие "файловая система" включает:

- совокупность всех файлов на диске,
- наборы структур данных, используемых для управления файлами,

такие, например, как каталоги файлов, дескрипторы файлов, таблицы распределения свободного и занятого пространства на диске,

комплекс системных программных средств, реализующих управление файлами, в частности: создание, уничтожение, чтение, запись, именование, поиск и другие операции над файлами.

Файлы идентифицируются именами. Пользователи дают файлам символьные имена, при этом учитываются ограничения ОС как на используемые символы, так и на длину имени. До недавнего времени эти границы были весьма узкими. Так в популярной файловой системе FAT(от Windows 98 и Windows ME) длина имен ограничивается известной схемой 8.3 (8 символов - собственно имя, 3 символа - расширение имени). Однако пользователю гораздо удобнее работать с длинными именами, так как даже через достаточно большой промежуток времени можно будет вспомнить, что содержит этот файл. Поэтому современные файловые системы, как правило, поддерживают длинные символьные имена файлов. Например, Windows NT в своей новой файловой системе NTFS устанавливает, что имя файла может содержать до 255 символов, не считая завершающего нулевого символа.

Чаще всего в ОС множество файлов могут называться одинаково, при условии, что они будут находиться в разных каталогах, однако в некоторых ОС файловым именам невозможно дать неуникальные имена. В Windows

используется первый случай, то есть для различия файлов с одинаковыми именами между собой каждый файл идентифицируется составным именем, состоящим из имени файла и каталога, где он находится. Что касается Unix систем, то файлам с одинаковыми именами, находящимся в разных каталогах, присваиваются уникальные имена.

Каталог – это, с одной стороны, группа файлов, объединенных пользователем исходя из некоторых соображений (например, файлы, содержащие программы игр, или файлы, составляющие один программный пакет), а, с другой стороны, – это файл, содержащий системную информацию о группе файлов, его составляющих. В каталоге содержится список файлов, входящих в него, и устанавливается соответствие между файлами и их характеристиками (атрибутами).

Иерархия каталогов может быть деревом или сетью. Каталоги образуют дерево, если файлу разрешено входить только в один каталог, и сеть – если файл может входить сразу в несколько каталогов.

NTFS – предпочитаемая файловая система в семействе Windows 2003 Server, Windows XP, Windows 2000 и Windows NT. Она была разработана, чтобы удовлетворять требованиям быстродействующих файловых и сетевых серверов, а так же персональных ЭВМ и, при этом, обойти многие из ограничений, ранее сделанных в файловых системах FAT16 и FAT32. Наиболее важными из этих требований следуют ниже:

- Восстанавливаемость данных. NTFS ограничивает возможность порчи данных, организовывая операции ввода-вывода (I/O) при помощи транзакций (групповых операций). Транзакции являются элементарными операциями, который означают, что или вся операция ввода-вывода (I/O) должна завершиться, или ни одно из этих действий не может завершиться. Если что-нибудь прерывает транзакцию, происходящую из-за потери электропитания компьютера или отмены операции ввода-вывода (I/O), NTFS делает все

возможное, чтобы гарантировать, что любое изменение, сделанное в файловой системе (поскольку часть операции ввода- вывода (I/O) была отменена или был сделан откат назад), возвращает файловую систему в ее прежнее состояние, перед тем как операция ввода-вывода (I/O) началась.

- NTFS является также и полностью восстанавливаемой файловой системой. Она предназначена, чтобы восстанавливать последовательность данных на диске после сбоя центрального процессора, системного аварийного отказа или ошибки ввода - вывода (I/O). NTFS дает возможность операционной системе, чтобы восстановиться без использования вами утилит проверки диска. Однако NTFS обеспечивают некоторые дисковые утилиты в случае, если восстановление завершается ошибкой, или искажение данных происходит за пределами управления файловой системы.

- Сохранение устойчивости при сбоях. Методы хранения избыточных данных могут использоваться NTFS, чтобы гарантировать, что, если данные разрушаются на одном физическом диске, то может извлекаться сохраненная копия из зеркала диска. NTFS всегда использует избыточность данных, чтобы защитить внутренние структуры данных, содержащие метаданные, очень важные для целостности тома.

- Защита данных. NTFS выполняет файлы и каталоги как защищенные объекты согласно архитектуре защиты (системы безопасности) объекта Windows. Доступ к объектам файла и каталога в NTFS может быть ограничен конкретным.

пользователям и их группам в среде этой архитектуры. Функции защиты данных для файлов и каталогов не включаются в файловых системах FAT. [3]

Другие улучшенные функции, обеспечиваемые NTFS - перечислены ниже:

- Многочисленные потоки данных. NTFS файлы могут состоять из более, чем одного потока. Дополнительные потоки могут содержать любой вид

данных, хотя обычно это - данные, характеризующие файл или метаданные.

- Имена в Unicode. Unicode – стандартный символьный набор, используемый в NTFS. Он заменяет устаревший однобайтовый символьный набор ASCII. Каждый символ, используемый в каждом главном естественном языке, представлен уникальным двухбайтовым числом в символьном наборе Unicode.

- Усовершенствованная индексация атрибута файла. NTFS включает в себя способность индексировать атрибуты файла в качестве средства расположения и сортировки многочисленных файлов, которые быстро совместно используют похожие данные. В файловых системах FAT32 и FAT16 Вы можете индексировать имена файлов, но не их атрибуты. Также, эти файловые системы не имеет функциональных возможностей, чтобы сортировать индексированные имена файлов FAT32 и FAT16.

- Динамическое перераспределение плохих кластеров. Когда операция чтения на томе NTFS, который не защищен от ошибок, встречает испорченные данные в группе секторов, каждый сектор в группе (кластере), отмечается как плохой, и последующие попытки выполнить операции чтения в этом секторе приведут к возвращению ошибки. В том же самом сценарии в файловых системах FAT, сама файловая система не отмечает дефектные секторы - пользователь должен запустить утилиту Chkdsk.exe, чтобы сделать это.

- Жесткие связи и подсоединения. Жесткие связи и подсоединения - это два способа, которыми объекты-хранилища могут быть связаны в среде NTFS.

- Поддержка сжатия и разреженного файла. Тома NTFS поддерживают сжатие файла на базе отдельного файла. Алгоритм сжатия файла, используемый NTFS - это метод сжатия Lempel-Ziv. Это - не имеющий потерь алгоритм сжатия, что означает, что данные, при сжатии и восстановлении из сжатого состояния файла, не теряются, в противоположность алгоритмам сжатия с потерями типа формата файла изображения JPEG, где некоторые данные

теряются каждый раз, когда происходят сжатие и восстановление сжатых данных.

- Системные журналы изменений. NTFS создает и поддерживает системные журналы изменений для каждого тома, которые отслеживают все изменения, сделанные в них.

- Отслеживание распределенных связей. оболочка Windows дает возможность пользователю создать файлы на её рабочем столе, связанные с приложениями, которые постоянно находятся в другом месте на томе. Меню Пуск, которое пользователь может конфигурировать, содержит много образцов этого вида связей. Также и технология связывания и внедрения объекта, или OLE, дает возможность прикладной программе внедрить связи от внешних файлов внутрь файлов, которые они создают и поддерживают. Составные части набора Office 2000: Word, PowerPoint и Excel— являются примерами приложений, которые используют OLE технологию. Проблема возникает в предыдущих случаях тогда, когда файл, будучи связанным (источник связи) перемещается, что делает его недоступным через посредство связи также, называемый как клиент связи. Отслеживание распределенной связи было сначала введено в версии NTFS, поставляемую с Windows 2000, чтобы дать возможность клиентским приложениям проследить за источниками связи, которые переместились. В результате этого, приложения и пользователи, которые создают связи, не должны обслуживать редактирование связи самостоятельно, когда источник ссылки перемещается.

- Шифрование. NTFS обеспечивает Шифрующую файловую систему, или (EFS)

для криптографической защиты файлов и каталогов.

- Поддержка POSIX. Ниже перечислены функциональные возможности POSIX,

которые были введены в Windows 2000:

а) Файлы могут получить доступ в файловых системах NTFS в соответствии с соглашениями об именовании POSIX. Соглашения POSIX допускают имена файлов, которые имеют концевые пробелы, имена файла, которые имеют точки в конце и имена файла, которые являются идентичными, если не считать регистр символов.

б) Разрешаются пересечения, где атрибуты защиты каждого родительского каталога в пути файла или каталога используются в определении, имеет ли конкретный пользователь доступ к нему.

в) Отметки времени "Файл изменил время". г) Жесткие ссылки в POSIX-стиле.

- API дефрагментации. Файл сохраняется на дисковом носителе информации в одном или нескольких кластерах. Кластеры - это элементарный модуль данных, составленных из одного или нескольких секторов. Сектора - это физические модули сохранения данных.

Поскольку файл записывается на диск, файл не может записаться в непрерывных кластерах. Кластеры, состоящие из нескольких несмежных участков, замедляют процесс чтения и записи файла. Удаленные друг от друга на диске кластеры, состоящие из нескольких несмежных участков, обрабатываются медленнее из-за увеличенного времени, которое тратится для перемещения головок чтения/записи жесткого диска к ним. Файл с кластерами, состоящими из нескольких несмежных участков, как говорят - фрагментирован. Чтобы оптимизировать файлы для быстрого доступа, том может дефрагментироваться.

- Дефрагментация - это процесс перемещения кластеров файла на диске, который делает их непрерывными. NTFS не выполняет дефрагментацию, но с версией 5.0 это действие предоставляется для приложений, чтобы выполнять дефрагментацию при помощи вызова API.

- Точки повторной обработки. В среде NTFS, файл или каталог могут

содержать точку повторной обработки, которая является совокупностью определяемых пользователем данных.

- Каталоги как точки повторной обработки тома. Точки повторной обработки тома - это каталоги в томе, которые приложение может использовать, чтобы "собрать" другой том, то есть установить его для использования в месте, которое задает пользователь. Другими словами, Вы можете использовать точку повторной обработки тома как шлюз к тому. Когда том установлен в точке повторной обработки тома, пользователи и приложения могут видеть монтируемый том с помощью точки повторной обработки тома или имени диска. Например, при помощи установленной точки монтирования тома, пользователь может видеть диск D как "C:\mnt\Ddrive" так же как "D:".

Используя точки повторной обработки тома, Вы можете объединить в одну логическую файловую систему в корне различные файловые системы такие как NTFS, 16-разрядная файловая система FAT, файловых система ISO-9660 для диска CD-ROM и так далее. Ни пользователи, ни приложения не нуждаются в информации о томе, на котором находится конкретный файл. Вся информация, которой они должны определить местонахождение заданного файла - полный путь к нему. Тома могут быть перестроены, заменены или разделены на многие тома без пользователей или приложений, которые должны изменить назначения.

Дефрагментация — процесс обновления и оптимизации логической структуры раздела диска с целью обеспечения хранения файлов в непрерывной последовательности кластеров. После дефрагментации ускоряется чтение и запись файлов, а следовательно и работа программ, ввиду того, что последовательные операции чтения и записи выполняются быстрее случайных обращений (например, для жесткого диска при этом не требуется перемещение головки). Другое определение дефрагментации: перераспределение файлов на диске, при котором они располагаются в

непрерывных областях.

Длинные файлы занимают несколько кластеров. Если запись производится на незаполненный диск, то кластеры, принадлежащие одному файлу, записываются подряд. Если диск переполнен, на нём может не быть цельной области, достаточной для размещения файла. Тем не менее, файл все-таки запишется, если на диске много мелких областей, суммарный размер которых достаточен для записи. В этом случае файл записывается в виде нескольких фрагментов.

Процесс разбиения файла на небольшие фрагменты при записи на диск называется фрагментацией. Если на диске много фрагментированных файлов, скорость чтения носителя уменьшается, поскольку поиск кластеров, в которых хранятся файлы, на жёстких дисках требует времени. На флеш-памяти, например, время поиска не зависит от расположения секторов, и практически равно нулю, поэтому для них дефрагментация не требуется.

Некоторое ПО требует, чтобы определённые файлы в обязательном порядке хранились в последовательно расположенных секторах (например, встроенный эмулятор CD- ROM в приводе Zalman VE-200 предъявляет такое требование к файлам образов). Даже если в такой привод будет установлен твердотельный накопитель, очевидно, дефрагментация ему всё-таки понадобится.

Дефрагментация чаще всего используется для таких файловых систем, как File Allocation Table для MS-DOS и Microsoft Windows, так как в программах для работы с ними обычно не предусмотрено никаких средств для предотвращения фрагментации, и она появляется даже на почти пустом диске и небольшой нагрузке.

Помимо замедления компьютера в работе с файловыми операциями (таких как чтение и запись), фрагментация файлов негативно сказывается на «здоровье» жёсткого диска, так как заставляет постоянно перемещаться

позиционирующие головки диска, которые осуществляют чтение и запись данных. Для устранения проблемы фрагментации существуют программы - дефрагментаторы, принцип работы которых заключается в «сборе» каждого файла из его фрагментов. Общим недостатком таких программ является их медленная работа — процесс дефрагментации обычно занимает очень много времени (до нескольких часов).

Несмотря на то, что среди программ такого рода имеются довольно популярные и известные, хорошую дефрагментацию можно провести своими силами. Для этого достаточно создать на диске новый раздел с файловой системой, после чего просто скопировать на этот логический диск те файлы, которые на нём будут храниться. Во время такого копирования на пустой диск, запись данных осуществляется последовательно, без фрагментации. Кроме того, сам процесс записи файлов занимает в несколько раз меньше времени, чем дефрагментация с помощью программ.

Знание файловой структуры и принципов работы жёсткого диска помогает сильно уменьшить фрагментацию файлов благодаря правильной организации работы пользователя. Эта организация заключается в том, что на чистый диск сначала записываются те файлы, которые предполагается хранить долго: любимые фильмы, архивы, музыку. Те же файлы, которые не планируется долго хранить, лучше записывать в конец диска: их удаление не вызовет дробления других файлов.

НЕОБХОДИМОСТЬ РАЗРАБОТКИ

У большинства пользователей ПК в качестве устройства долговременного хранения информации используется стандартный HDD (англ. — Hard Disk Drive), но недавно на рынке появился новый вид «жесткого диска» — SSD (англ. Solid State Drive). SSD представляет собой принципиально иной тип накопителя. Он построен на тех же технологиях, что используются во флеш-памяти, и аналогичен флеш-накопителям по организации ячеек.

Существует множество способов увеличить работоспособность и быстродействие ПК. Один из таких действий является дефрагментация.

Дефрагментация жёстких дисков (HDD) и твердотельных накопителей (SSD). В процессе дефрагментации происходит сбор и перенос на близкое расстояние друг от друга частей файлов. Помимо этого, все данные перемещаются в начало жесткого диска, а свободное место располагается в его конце, что ускоряет доступ к данным.

Фрагментация возникает по мере сохранения, изменения или удаления файлов тома. Исходный файл и последующие изменения, внесенные в него, часто сохраняются в разных местах тома. При этом меняется не расположение самого файла в ОС Windows, а только участки тома, в которых сохранены биты информации, образующие файл. По истечению времени файл и том фрагментируются, что приводит к замедлению работы компьютера, которому требуется обращаться к разным местам тома для открытия одного файла.

Дефрагментация позволит быстрее загружать программы и файлы. Дефрагментация помогает повысить быстродействие в ресурсоемких трехмерных играх, то есть ускоряет загрузку.

графических данных; повышается скорость записи на диск, например, при установке новых программ или после удаления ранее установленных.

Основным недостатком твердотельных накопителей информации (SSD) по сравнению с HDD является их ограниченный срок службы, так как SSD-накопитель представляет собой множество ячеек флеш-памяти и каждая такая ячейка имеет свой ресурс. HDD в отличие от SSD может прослужить от 4 до 10 лет.

Накопители SSD являются дорогостоящими элементами ПК.

Следующий критерий — скорость. Скорость работы SSD превосходит HDD в несколько раз. SSD в процессе работы не нагревается, не шумит и уровень энергопотребления в отличие от HDD ниже.

Современные операционные системы и программы не готовы к работе с SSD накопителями. Если проанализировать работу ПО Windows, то выяснится,

что операционная система создает сотни тысяч временных файлов во время своей работы. Например, браузеры постоянно кэшируют данные из Интернета, что позволяет получать быстрый доступ к регулярно посещаемым Интернет – ресурсам, но опять же нагружает SSD–накопитель. Microsoft Word при создании текстового документа производит автосохранение, что означает дополнительный цикл перезаписи на накопителе информации.

Производимые действия являются десятками тысяч циклов перезаписи ежегодно, что сказывается прямым образом на ресурсе SSD–накопителя.

Для того, чтобы износ ячеек SSD–устройства был равномерным, производители устройств устанавливают в него специальную программу, которая по определенному алгоритму распределяет ячейки для записи или перезаписи информации, которая позволяет снизить износ ячеек.

Дефрагментация не поможет увеличить ресурс SSD–накопителя, так как при дефрагментации происходит перемещение фрагментов файлов, которые записывались на уже фрагментированный диск. Одноразовая дефрагментация позволит частично переместить фрагменты файлов и освободить ячейки, в которых эти фрагменты хранятся продолжительное время. В дальнейшем дефрагментация будет задействовать только участки SSD–накопителя, в которых происходит постоянная перезапись ячеек, то есть те области, в которые сохраняются временные файлы, что приведет только к большему износу ячеек.

Не следует выполнять дефрагментацию на твердотельных дисках. После дефрагментации SSD изменений в скорости работы ПК заметно не будет. В HDD, например, используются механические детали (головка, движущаяся по поверхности магнитного диска и считывающая

данные). В SSD– накопителе доступ к любой ячейке памяти осуществляется одинаково быстро и никак не зависит от взаимного расположения этих данных.

Дефрагментация в HDD необходима, так как после её выполнения, файлы открываются быстрее, производительность ПК увеличивается. Однако, как выяснилось, на твердотельных накопителях дефрагментацию выполнять не требуется. Этот процесс изнашивает накопитель, сокращая срок его службы. Более того, если на вашем компьютере установлен SSD, то стоит отключить процесс автоматической дефрагментации и не выполнять её в дальнейшем.

ВЫБОР ИНСТРУМЕНТОВ РАЗРАБОТКИ

Для разработки и реализации проекта был выбран высокоуровневый язык программирования Python.

Основное преимущество языка Python заключается в удобочитаемости, ясности и более высоком качестве, отличающими его от других инструментов в мире языков сценариев. Программный код на языке Python читается легче, а значит, многократное его использование и обслуживание выполняется гораздо проще, чем использование программного кода на других языках сценариев. Единообразие оформления программного кода на языке Python облегчает его понимание даже для тех, кто не участвовал в его создании. Кроме того, Python поддерживает самые современные механизмы многократного использования программного кода, каким является объектно-ориентированное программирование (ООП).

По сравнению с компилирующими или строго типизированными языками, такими как C, C++ и Java, Python во много раз повышает производительность труда разработчика. Объем программного кода на языке Python обычно составляет треть или даже пятую часть эквивалентного программного кода на языке C++ или Java. Это означает меньший объем ввода с клавиатуры, меньшее количество времени на отладку и меньший объем трудозатрат на сопровождение. Кроме того, программы на языке Python запускаются сразу же, минуя длительные этапы компиляции и связывания, необходимые в некоторых других языках программирования, что еще больше увеличивает производительность труда программиста.

В составе Python поставляется большое число собранных и переносимых функциональных возможностей, известных как стандартная библиотека. Эта библиотека предоставляет массу возможностей, востребованных в прикладных программах, начиная от поиска текста по шаблону и заканчивая сетевыми функциями. Кроме того, Python допускает расширение как за счет ваших собственных библиотек, так и за счет библиотек, созданных сторонними разработчиками. Из числа сторонних разработок можно назвать инструменты создания веб-сайтов, программирование математических вычислений, доступ к последовательному порту, разработку игровых программ и многое другое. Например, расширение NumPy позиционируется как свободный и более мощный эквивалент системы программирования математических вычислений Matlab.

Сценарии Python легко могут взаимодействовать с другими частями приложения благодаря различным механизмам интеграции. Эта интеграция

позволяет использовать Python для настройки и расширения функциональных возможностей программных продуктов. На сегодняшний день программный код на языке Python имеет возможность вызывать функции из библиотек на языке C/C++, сам вызываться из программ, написанных на языке C/C++, интегрироваться с программными компонентами на языке Java, взаимодействовать с такими платформами, как COM и .NET, и производить обмен данными через последовательный порт или по сети с помощью таких протоколов, как SOAP, XML-RPC и CORBA. Python – не обособленный инструмент.

Как и многие редакторы кода, VS Code использует распространенную схему расположения основных элементов – слева обозреватель файлов, справа редактор кода. В дополнение к этому есть специфичные элементы, обеспечивающие навигацию и выполнение специальных команд.

VS Code работает с файлами и папками в которых находятся проекты. В простейшем случае вы можете открыть файл на редактирование просто выполнив команду `./code index.html`. Более интересным случаем является открытие папки. VS Code сам определяет тип проекта в зависимости от содержимого папки. Например, если в папке находятся файлы `package.json`, `project.json`, `tsconfig.json` или файлы `.sln` и `.proj` для Visual Studio ASP.NET 5.0 то VS Code включает много новых функций которые обеспечивают IntelliSense, подсказки, навигацию по коду, выполнение команд и многое другое.

Операционные системы можно классифицировать по нескольким признакам. По назначению выделяют системы общего назначения и специализированные операционные системы. Последние используются в специализированной вычислительной технике, например, бытовой технике, автомобилях, спецвычислителях военного применения. По количеству одновременно работающих пользователей, операционные системы можно разделить на однопользовательские и многопользовательские. По количеству одновременно работающих программ операционные системы делят на однозадачные и многозадачные.

На сегодняшний день наиболее популярными являются операционные системы семейства Windows, которые являются проприетарным (коммерческим) продуктом корпорации Microsoft.

РЕАЛИЗАЦИЯ ПРОГРАММНОГО СРЕДСТВА

Программа дефрагментации создана в среде разработки Visual Studio Code на языке программирования Python. Реализована данная программа в качестве консольного приложения.

// Программа на github с скриншотами успешной дефрагментации