МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра вычислительные системы и технологии

Лабораторная работа № 3

Файловая система FAT

ОТЧЕТ

по лабораторной работе

по дисциплине

Принципы и методы  
организации системных программных средств

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Викулова Е.Н.

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сапожников В.О.

19-ИВТ-3

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2022

**План работы**

Подготовительные действия:

Самостоятельно выбрать *hex*-редактор для работы с файловой системой FAT(ex*FAT*), ознакомиться с его функциональными возможностями.

- подготовить носитель с файловой системой *FAT* (USB-флэш-накопитель, внешний диск), или создать FAT раздел на диске.

Используя выбранный *hex*-редактор, уметь выполнять следующие действия:

* работать с диском на физическом (*c,h,s*) и логическом (*LBA*) уровнях;
* находить и читать:

a) *данные загрузочной области*:

* команду перехода к коду загрузки;
* идентификатор системы,
* дескриптор носителя;
* количество байт в секторе;
* количество секторов в кластере;
* количество секторов, зарезервированных для загрузочной записи;
* количество таблиц *FAT*, размер *FAT*;
* количество записей в корневом каталоге;
* общее количество секторов на диске;
* байты сигнатуры загрузочного сектора;
* и др. (например, при работе с *FAT*32, данные актуальные только для *FAT*32).

б) *область FAT*;

в) *данные корневого каталога*:

* + - имя файла или подкаталога,
    - расширение имени,
    - атрибуты файла,
    - время и дату создания файла,
    - размер файла;

г) *данные подкаталогов*;

д) *содержимое файлов*:

* изменять имя, размер и атрибуты файла;
* создавать, удалять и восстанавливать файлы;
* объединять файлы;
* удалять кластеры из файла.

1. Изучить свойство фрагментации дисковой памяти. Выполняя файловые операции добавления, удаления, модификации файлов создать фрагментацию отдельных файлов.

С помощью выбранного *hex*-редактора:

* найти цепочки кластеров, принадлежащих фрагментированным файлам;
* отслеживать изменения при создании новых файлов (каталогов) и модификации файлов (например, при изменении размера файла);
* выявить стратегию выделения дискового пространства, которую

использует операционная система.

1. Изучить причины возникновения ошибок логической структуры диска

и методы их исправления.

С помощью выбранного *hex*-редактора ввести логические некорректности в файловую структуру (потерянные кластеры, пересекающиеся файлы).

Используя утилиты проверки и восстановления (найти и выбрать самостоятельно), попытаться восстановить файловую структуру диска.

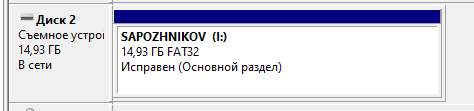
**Выбор редактора**

WinHex редактор от X-Ways Software Technology AG, распространяемый для ОС Windows. Как и все другие HEX-редакторы, WinHex может отображать контрольные суммы или коды программных файлов, которые не под силу обычным текстовым редакторам.

Ко всему прочему, утилита может использоваться для повседневной работы в аварийных ситуациях для проверки и редактирования всех известных и неизвестных видов файлов, восстановления случайно удаленных файлов из Корзины Windows (и не только) или цифровых карт камеры и многое другое.

Поскольку для данной работы необходим Hex редактор с полной функциональностью, а условная бесплатная версия WinHex такой не обладает, то была использована “пиратская” версия, что мы, конечно же, осуждаем.

Для работы была использован Flash накопитель на 16ГБ, отформатированный в FAT32 при помощи встроенной в Windows утилиты управления дисками.



**Логическая структура диска FAT**

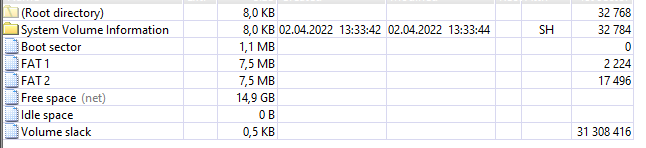
При форматировании диска создается набор системных данных, необходимых для работы с файлами на логическом уровне. Подготовленный носитель файловой системы принято также называть томом (volume).

Логическая структура диска (тома) *FAT* состоит из четырех областей, размеченных статически:

- загрузочная запись (*Boot Record*),

- область таблиц *FAT* (2 копии),

- корневой каталог (*Root Dir*),

- область данных *(Data Area*).

**Загрузочный сектор *Boot Record***

Первые 3 байта занимает команда перехода к коду загрузки, которая обычно EB5890h (наш случай не стал исключением)



Следующие 8 байт занимает идентификатор OEM ID, указывает какая система форматировала раздел.



Следующие 2 байта указывают количество байт в секторе. Обычно 512



Действительно, 200h = 512d

Затем 1 байт занимает кол-во секторов в кластере. Данное число должно быть равно степеням двойки (1, 2, 4, 8, 16, 32, 64). В нашем случае оно равно 10h = 16d



Следующие 2 байта занимает значение кол-ва секторов, зарезервированных для загрузочной записи(записей).



1 байт на кол-во структур в FAT разделе, обычно 2 (как и в нашем случае)



2 байта – кол-во 32-байтовых записей каталога в корневом каталоге разделов FAT12 и FAT16. В разделах FAT32 должно быть равно 0 (так и есть)



2байта – число, указывающее общее кол-во секторов в разделах, содержащих менее 65536 секторов. FAT32 должно быть равно 0.



Для FAT 32 данное значение указано в поле по смещению 20h и занимает 4 байта



1 байт-дескриптор носителя. Для всех несменных носителей обычно F8h (как в нашем случае), для большинства сменных F0h



2 байта – число, указывающее общее кол-во секторов, занятых разделами FAT12, FAT16. В FAT32 данной число равно 0.



Количество секторов, занятых разделами FAT указано по смещение 24h и занимает 4 байта



Нулевой сектор загрузочной записи раздела FAT32 завершается байтами сигнатуры, значение которых должно быть равно 55Ааh.



*Дополнительно:*

2 байта –кол-во секторов на дорожке для прерывания 13h, на жестких дисках обычно 63.

3Fh = 63d

Следующие 2 байта – число головок для прерывания 13h. На жёстких дисках обычно 255

Действительно, FFh = 255d

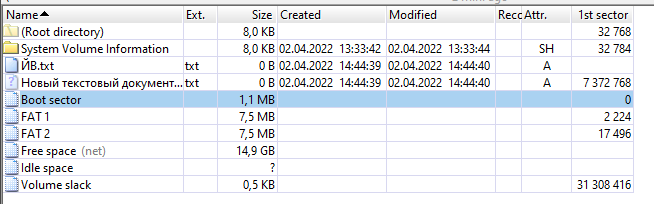
По смещение 28h, находится информация характерная только для FAT32. Биты 0-3. Номер активного раздела FAT с отсчетом от нуля. Действителен только в том случае, если зеркальное отражение отключено (бит 7 =1). Биты 4-6. Зарезервированные. Бит 7 – “0” указывает, что раздел FAT зеркально отражен, “1” – что раздел FAT активен. Биты 8-15 зарезервированы.



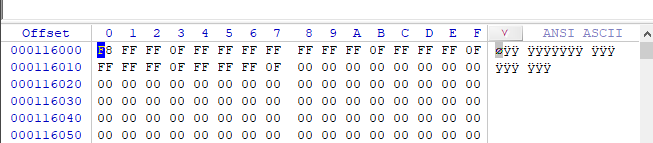
В нашем случае все биты равны нулям, раздел FAT отражен.

**Разделы FAT**

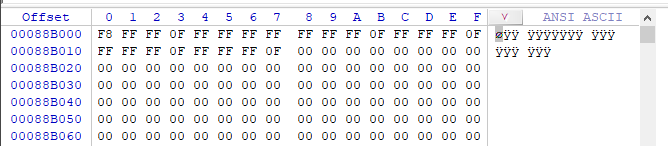
WinHex обладает удобным меню навигации по диску, которое позволяет сместится на разделы FAT



FAT1:



FAT2:

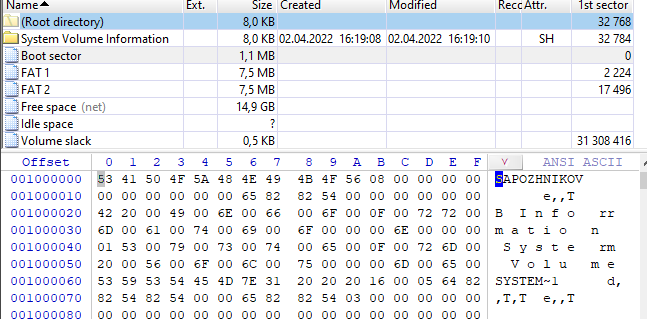


Теперь посчитаем смещение вручную. Кол-во зарезервированных секторов равно 8B0 ()h = 2224d. Тогда FAT1 находится по смещению 2224 \* 512 = 1138688 = 11600h, куда нас и перенаправил WinHex.

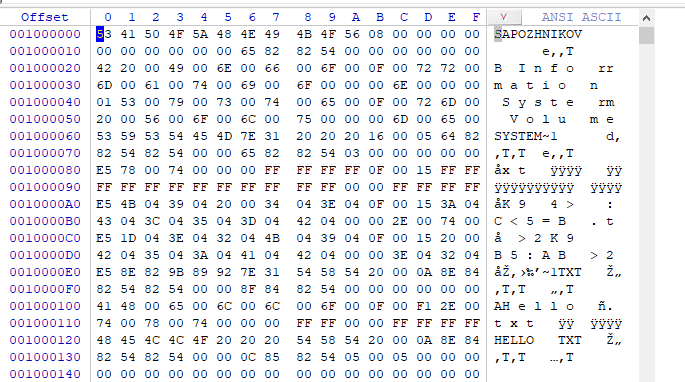
Аналогично для FAT2: 17496\*512=8957952d=88B000h

**Корневой каталог**

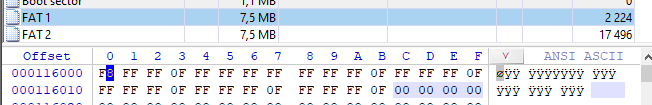
Корневой каталог находит по смещению: 1138688 + 15272 \* 512 \* 2 = =16777216 = 1000000h



Создадим файл Hello.txt с текстом Hello. В корневом каталоге добавилась запись:



Так же появилась запись и в таблице FAT:



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Смещение | Значение | | Расшифровка | | Описание |
| 000 | 48454C4C4А202020 | | “HELLO ” | | Имя файла |
| 008 | 545854 | | “TXT” | | Расширение файла |
| 00B | 20 | | Архивный | | Атрибуты файла |
| 00С | Зарезервировано | | | | |
| 1A | 0500 | 5 | | Начальный кластер | |
| 1С | 05 | 5 байт | | Размер файла в байтах | |

Значения 5ого кластера:

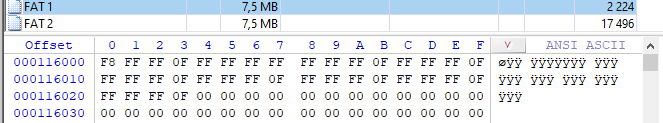


**Работа с файловой системой**

Добавление подкаталогов и файлов.

Создадим каталог Sapozhnikov и поместим в него jack.txt с некоторым содержимым

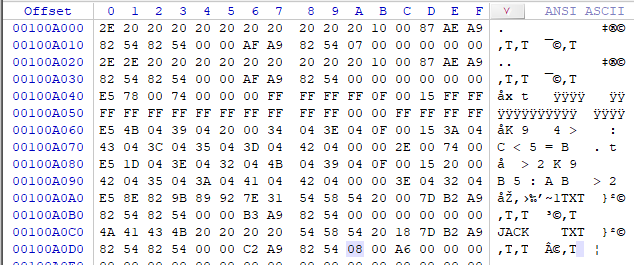
Изменение FAT:



Главный каталог: текстовый файл Hello.txt, подкаталог Sapozhnikov. Причем в каталоге видно начальный кластер 07



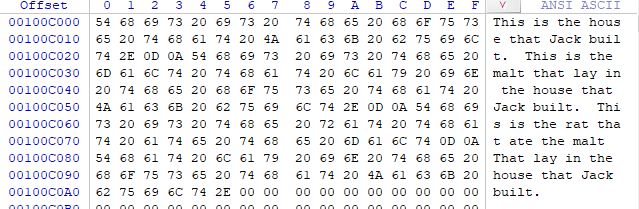
Содержимое подкаталога Sapozhnikov(мы сейчас в 7 секторе, кстати) и текстовый файл jacm.txt



Причем у файла видно начальный кластер с содержимым – 08

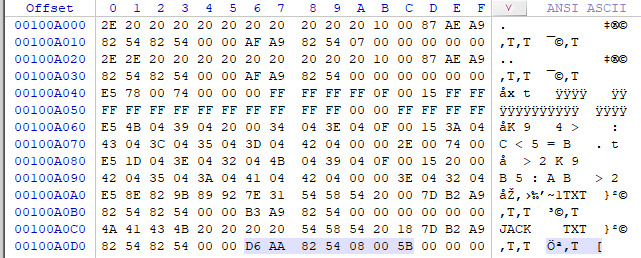


Содержимое файла jack.txt (кластер 8)

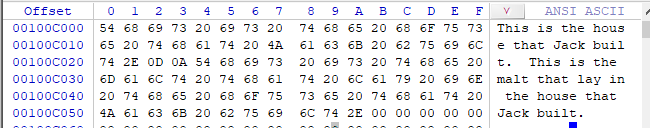


Изменим содержимое файла, удалив 2 последние строки.

Изменился размер файла и дата (Кластер 7, подкаталог Sapozhnikov)

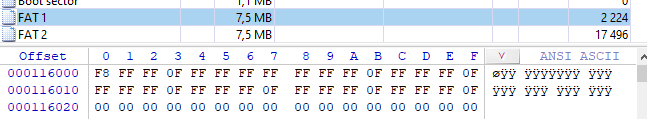


И так же изменилось само содержимое файла.



Теперь удалим текстовый файл стандартным способом, не через WinHex

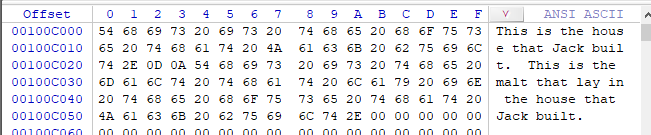
Изменилась цепочка кластеров в FAT



В начале имени файла добавился байт состояния – E5, т.е. файл удален

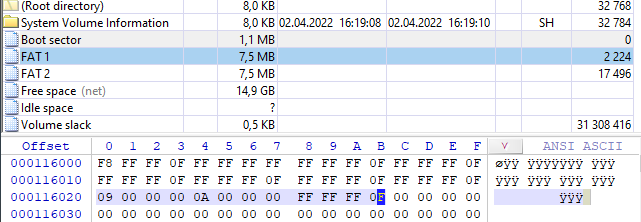


При этом сами данные (8 кластер) остались:

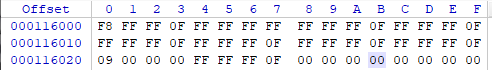


**Повреждение и восстановление файлов.**

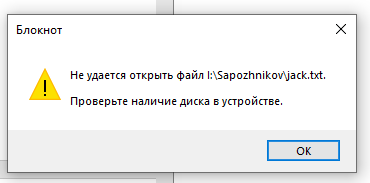
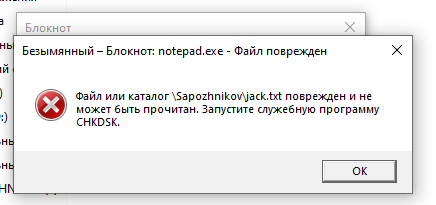
Дополним файл до 18.6Кб, чтобы он занимал несколько кластеров. Число занятый кластеров в таблице FAT изменилось, добавилось еще 3 кластера.



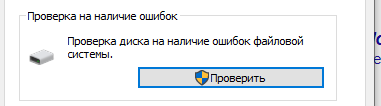
Стираем последний кластер в цепочке. Размер цепочки станет 2 вместо 3.

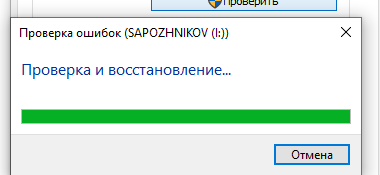


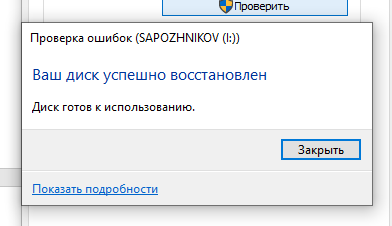
Теперь данный файл распознается как поврежденный



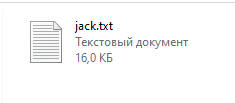
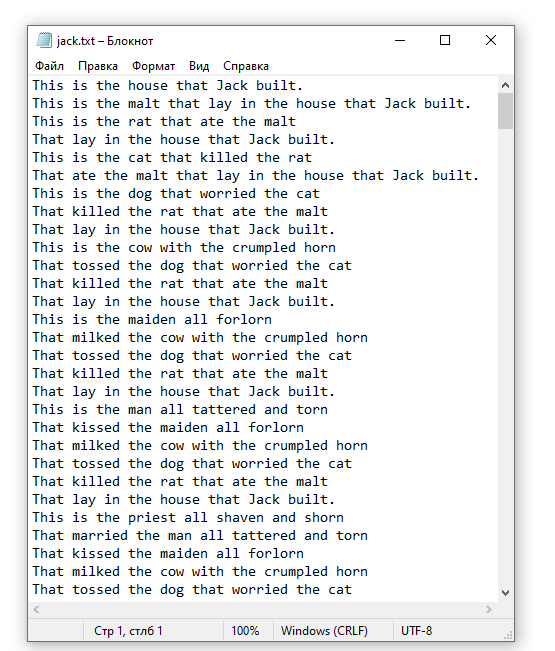
Запустим встроенную в Windows утилиту поиска и справления ошибок файловой системы.



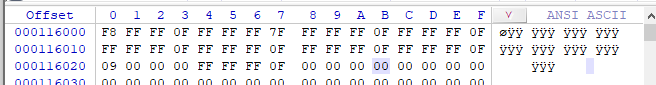




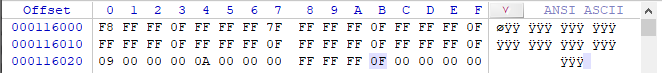
Диск был восстановлен, файл открывается, но его размер уменьшился. Файл был урезан ровно до того, где мы урезали кластер.

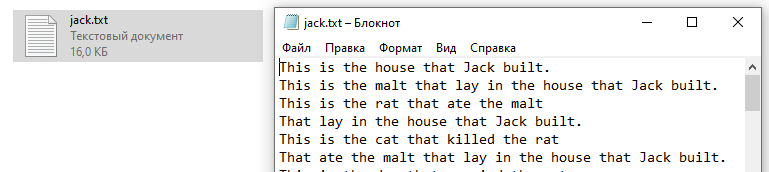


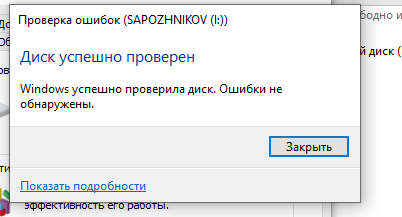
“Хвост” файла был попросту отрезан.



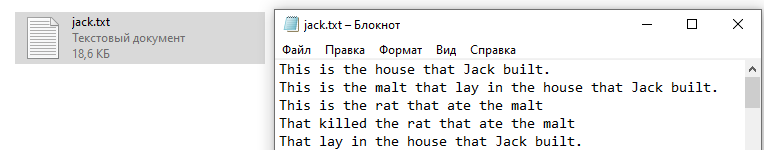
Попытаемся вернуть прежний размер файла восстановив последовательность разделов.







К очень большому удивлению, размер файла и его данные полностью восстановились и нам не пришлось вручную переписывать размер файла.



**Фрагментация файлов**

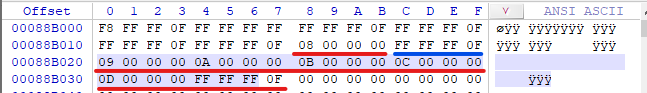
Устроим фрагментацию. Сначала для удобства отформатируем диск



Добавим 2 файла: main.txt и temp.txt



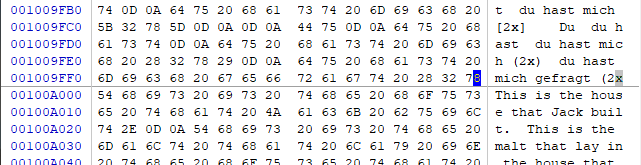
А теперь допишем в temp.txt больше данных.

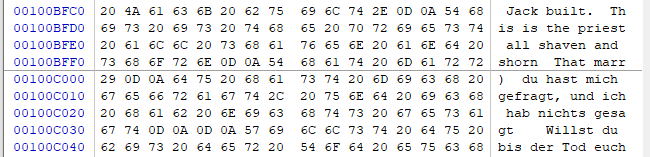


Красный – temp.txt

Синий – main.txt

Сама фрагментация на границах каталогов. Видно, что текст песни группы Rammstein – Du Hast прерывается стихотворением про Дом, который построил Джек.





**Дефрагментация.**

Твердотельные накопители и флешки на основе различных видов флеш-памяти в дефрагментации файловых систем не нуждаются. Более того, дефрагментация, в некоторой степени, вредит флеш-памяти, так как последняя имеет намного более ограниченное количество циклов записи/перезаписи, чем накопители на магнитных дисках при должном использовании.

Сама идея дефрагментации состоит в перераспределении фрагментов файлов и логических структур файловых систем на дисках для обеспечения последовательности кластеров.

**Вывод**

Файловая система FAT32 хорошо подходит для различных носителей. Она имеет высокую скорость работы и достаточно эффективна при работе с файлами небольшого размера. Также она обеспечивает более низкий износ дисков, вследствие меньшего количества передвижений головок чтения/записи. Однако она не ведёт журналирования и поэтому имеет низкую защиту от сбоев.

Для работы с данными в исходном виде достаточно простого Hex-редактора, который позволяет просматривать данные в hex и ascii форматах. WinHex имеет дополнительные преимущества в виде меню навигации по секторам и разделам диска, однако данное ПО является платным, а использование “пиратских” версий мы осуждаем.