

О. В. КАРАБАЕВА

**СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ.
ФАЙЛОВЫЕ СИСТЕМЫ**

Учебно-методическое пособие

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

О. В. КАРАВАЕВА

СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ. ФАЙЛОВЫЕ СИСТЕМЫ

Учебно-методическое пособие

Киров
2018

УДК 004.45(07)
К21

Допущено к изданию методическим советом факультета автоматизации и вычислительной техники ВятГУ в качестве учебно-методического пособия для студентов направлений 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия» всех профилей подготовки и всех форм обучения

Рецензент

кандидат технических наук,
и.о. заведующего кафедрой АТ ВятГУ
Ю. В. Ланских

Караваяева, О. В.

К21 Системное программное обеспечение. Файловые системы : учебно-методическое пособие / О. В. Караваяева. – Киров : ВятГУ, 2018. – 56 с.

Пособие предназначено для студентов направлений 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия» всех профилей подготовки, всех форм обучения для выполнения лабораторных работ по дисциплине «Системное программное обеспечение».

УДК 004.45(07)

© ВятГУ, 2018

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	5
1.1. Файловые системы	5
1.1.1. Файловая система NTFS	7
1.1.2. Файловая система FAT	9
1.1.3. Структура логического диска с файловой системой FAT	11
2. ВЫПОЛНЕНИЕ ЛАБОРАТОРНОЙ РАБОТЫ	23
2.1. Описание лабораторной установки	23
2.2. Выполнение теста и начального этапа работы	24
2.3. Работа с файлами.....	27
2.3.1. Создание файла	27
2.3.2. Изменение размера файла	36
2.3.3. Создание каталога	37
2.3.4. Копирование, перемещение, переименование файла.....	40
2.3.5. Удаление и восстановление файла	46
2.3.6. Проверка и завершение заданий.....	49
ЗАКЛЮЧЕНИЕ	53
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	54

ВВЕДЕНИЕ

Целью данной лабораторной работы по дисциплине «Системное программное обеспечение» является изучение файловой системы FAT, принципов организации хранения информации на внешних носителях, а также изучение процессов создания, удаления, перемещения и копирования файлов.

Вся работа разделена на теоретическую и практическую часть. Перед тем как приступить к выполнению работы необходимо внимательно прочитать данное учебно-методическое пособие, для того чтобы освежить знания и ответить на вопросы теста, включенного в лабораторную работу.

Практическая часть заключается в прохождении теста и выполнении ряда заданий. Тест ориентирован, в основном, на общие знания по данной теме, но встречаются и вопросы, где необходимо вспомнить конкретные параметры и значения, для этого рекомендуется внимательно читать теорию. Сами задания в лабораторной работе представляют собой действия, которые выполняются операционной системой при работе с файлами.

Лабораторная работа поддерживает возможность сохранения и отката процесса выполнения.

1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1. Файловые системы

Файловая система – логическая организация данных на носителе информации, определяющая логическую и физическую структуру файла, идентификацию и атрибуты файла, а также алгоритмы распределения файлов по пространству носителя и организации файлов в каталоги.

Также файловая система – часть операционной системы, входящая в ядро или представляющая собой отдельный модуль и обеспечивающая запись и чтение и другие операции над файлами на внешних носителях.

Файловая система должна обеспечивать:

- безопасное и надежное хранение данных (т. е. защищенное от несанкционированного использования и различного рода сбоев и ошибок);
- программный интерфейс доступа к файлам;
- организацию файлов в виде иерархии каталогов.

Windows поддерживает несколько файловых систем для различных внешних устройств:

- NTFS – основная файловая система семейства Windows NT;
- FAT (File Allocation Table – таблица размещения файлов) – простая файловая система используемая Windows для устройств флеш памяти, а также для совместимости с другими операционными системами при установке на диски с множественной загрузкой. Основным элементом этой файловой системы является таблица размещения файлов FAT (по имени которой названа вся файловая система), необходимая для определения расположения файла на диске. Существует три варианта FAT, отличающихся разрядностью идентификаторов, указывающих размещение файлов: FAT12, FAT16 и FAT32;
- exFAT (Extended FAT – расширенная FAT) – развитие файловой системы FAT, использующее 64 разрядные идентификаторы. Применяется в основном для устройств флеш-памяти;

- CDFS (CD ROM File System) – файловая система для CD дисков, объединяющая форматы ISO 96601 и Joliet2;

- UDF (Universal Disk Format – универсальный формат дисков) – файловая система для CD и DVD дисков, разработанная для замены ISO 9660.

Для дальнейшего изложения необходимо знать следующие важные понятия: диск, раздел, простые и составные тома, сектор, кластер.

Диск (disk) – устройство внешней памяти, например, жесткий диск или оптический диск (CD, DVD, Blu ray).

Раздел (partition) – непрерывная часть жесткого диска. Диск может содержать несколько разделов.

Том (volume) или логический диск (logical disk) – область внешней памяти, с которой операционная система работает как с единым целым. Тома бывают простые и составные.

Простой том (simple volume) – том, состоящий из одного раздела.

Составной том (multipartition volume) – том, состоящий из нескольких разделов (необязательно на одном диске).

Понятия раздела и простого тома отличаются: во первых, разделы формируются, в основном, только на жестких дисках, а тома создаются и на других устройствах внешней памяти (например на оптических дисках и устройствах флеш памяти), во вторых, понятие «раздел» связано с физическим устройством, а понятие «том» – с логическим представлением внешней памяти.

Сектор (sector) – блок данных фиксированного размера на диске; наименьшая единица информации для диска. Типичный размер сектора для жестких дисков равен 512 байтам, для оптических дисков – 2048 байт. Деление диска на секторы происходит один раз при создании диска в процессе низкоуровневого форматирования и обычно не может быть изменено.

Кластер (cluster) – логический блок данных на диске, включающий один или несколько секторов. Количество секторов, составляющих кластер, обычно кратно степеням двойки. Размер кластера задается операционной системой в

процессе высокоуровневого форматирования, которое может осуществляться многократно.

При записи на диск файл всегда будет занимать целое число кластеров. Например, файл размером 100 байт в файловой системе с размером кластера 4 КБ будет занимать ровно 4 КБ.

Выбор размера кластера связан со следующими соображениями. Малые кластеры позволяют сократить размер фактически неиспользуемого дискового пространства, возникающего за счет размещения файла в целом числе кластеров. Но при этом общее количество кластеров на диске увеличивается и размер служебных структур файловой системы, в которых хранится информация о файлах, возрастает.

Наиболее распространены Windows NT-совместимые файловые системы семейств FAT и NTFS. Файловая система FAT была создана в 70-х годах 1976-1977 годах и использовалась в операционных системах семейств DOS, а затем и Windows. Файловая система NTFS в свою очередь, заменила файловую систему FAT в операционных системах компании Microsoft. Сегодня данные файловые системы все еще широко применяются: FAT для флеш-накопителей, а NTFS на жёстких дисках.

1.1.1. Файловая система NTFS

NTFS (аббревиатура New Technology File System — Файловая Система Новой Технологии) – стандартная файловая система для семейства операционных систем Microsoft Windows NT. Разрабатывалась Microsoft как основная файловая система для серверных версий операционных систем Windows.

Представлена 27 июля 1993 вместе с Windows NT 3.1. NTFS разработана на основе файловой системы HPFS (аббревиатура High Performance File System – Высокопроизводительная Файловая Система), создававшейся Microsoft совместно с IBM для операционной системы OS/2.

В настоящее время NTFS рассматривается в качестве предпочтительной файловой системы как для серверных, так и для клиентских версий Windows.

В NTFS используются 64 разрядные идентификаторы кластеров, поэтому теоретически том NTFS может содержать 264 кластера. Однако текущие реализации в Windows поддерживают только 32 разрядную адресацию кластеров, что при размере кластера максимум 64 КБ позволяет NTFS тому достигать размера до 256 ТБ:

$$2^{32} * 2^{16} \text{ байт} = 2^{48} \text{ байт} = 2^8 * 2^{40} \text{ байт} = 256 \text{ ТБ}.$$

Для томов, больших 4 ГБ, при форматировании Windows предлагает размер кластера по умолчанию 4 КБ.

Некоторые возможности NTFS:

- восстанавливаемость (recoverability) – способность файловой системы возвращаться к работоспособному состоянию после возникновения сбоя. Реализуется такая возможность, во первых, за счет поддержки атомарных транзакций, во вторых, за счет избыточности хранения информации. Атомарная транзакция (atomic transaction) – операция с файловой системой, приводящая к её изменению, которая либо полностью успешно выполняется, либо не выполняется вообще (т.е. в случае сбоя во время атомарной транзакции все изменения откатываются). Избыточность используется при хранении важнейших данных файловой системы, критически необходимых для её корректной работы;

- безопасность (security) – защищенность файлов от несанкционированного доступа; реализуется при помощи модели безопасности Windows;

- шифрование (encryption) – преобразование файла в зашифрованный код, который невозможно прочесть без ключа. Обычные механизмы безопасности, такие как назначение прав доступа пользователей к файлам, не обеспечивают полной защиты информации, например, в случае перемещения диска на другой компьютер. Администратор операционной системы всегда может получить доступ к файлам других пользователей, даже на томе NTFS.

Поэтому в NTFS включена поддержка шифрующей файловой системы EFS (Encrypting File System), которая позволяет легко зашифровывать и расшифровывать файлы;

- поддержка RAID (Redundant Array of Inexpensive (Independent) Disks – массив недорогих (независимых) дисков с избыточностью) – возможность использования для хранения информации нескольких дисков; данные с одного диска автоматически копируются на другие, обеспечивая тем самым повышенную надежность;

- дисковые квоты для пользователей (Per-User Volume Quotas) – возможность выделения для каждого пользователя определенного пространства на диске (квоты); NTFS не позволяет пользователю записывать данные на диск сверх выделенной квоты.

1.1.2. Файловая система FAT

FAT (*FAT16*) – это аббревиатура от *File Allocation Table* (в переводе *Таблица Размещения Файлов*).

Структура FAT была разработана Биллом Гейтсом и Марком МакДональдом в 1977 году. Использовалась в качестве основной файловой системы в операционных системах DOS и Microsoft Windows (до версии Windows ME).

Существует четыре версии FAT – *FAT12*, *FAT16*, *FAT32* и *exFAT*. Они отличаются количеством бит, отведённых для хранения номера кластера.

FAT12 применяется в основном для дискет, *FAT16* – для дисков малого объёма, а новая *exFAT* преимущественно для флэш-накопителей. Максимальный размер кластера, который поддерживается в FAT, составляет 64Кб. (Подробная статья).

FAT16 впервые представлена в ноябре 1987 года. Индекс *16* в названии показывает, что для номера кластера используется 16 бит. Вследствие этого

максимальный объем раздела диска (тома), который может поддерживать эта система, равен 4Гб.

Позже, с развитием технологий и появлением дисков объемом более 4Гб, появилась файловая система *FAT32*. Она использует 32-разрядную адресацию кластеров и появилась вместе с Windows 95 OSR2 в августе 1996 года. *FAT32* ограничена в размере тома в 128Гб. Также эта система может поддерживать длинные имена файлов.

В файловой системе FAT смежные секторы диска объединяются в единицы, называемые кластерами. Количество секторов в кластере равно степени двойки.

Пространство тома FAT32 логически разделено на три смежные области:

- зарезервированная область – содержит служебные структуры, которые принадлежат загрузочной записи раздела (Partition Boot Record – PBR, для отличия от Master Boot Record – главной загрузочной записи диска; также PBR часто некорректно называется загрузочным сектором) и используются при инициализации тома;

- область таблицы FAT, содержащая массив индексных указателей («ячеек»), соответствующих кластерам области данных. Обычно на диске представлено две копии таблицы FAT в целях надежности;

- область данных, где записано собственно содержимое файлов – то есть текст текстовых файлов, кодированное изображение для файлов рисунков, оцифрованный звук для аудиофайлов и т.д. – а также так называемые метаданные – информация относительно имен файлов и папок, их атрибутов, времени создания и изменения, размеров и размещения на диске.

В FAT12 и FAT16 также специально выделяется область корневого каталога. Она имеет фиксированное положение (непосредственно после последнего элемента таблицы FAT) и фиксированный размер, исчисляемый в 32-х байтных элементах, т.е. при описании в Partition Boot Record указывается именно количество 32-х

байтных элементов, каждый из которых описывает какой-либо элемент корневого каталога (будь то файл или другой вложенный каталог).

Если кластер принадлежит файлу, то соответствующая ему ячейка в таблице FAT содержит номер следующего кластера этого же файла. Если ячейка соответствует последнему кластеру файла, то она содержит специальное значение (FFFFh для FAT16). Таким образом выстраивается цепочка кластеров файла. Неиспользуемым кластерам в таблице соответствуют нули. «Плохим» кластерам (которые исключаются из обработки, например, по причине нечитаемости соответствующей области устройства) также соответствует специальный код.

Для практического знакомства с файловыми системами выбрана FAT из-за ее относительной простоты. Далее ее структура будет рассмотрена более подробно.

1.1.3. Структура логического диска с файловой системой FAT

Логический диск с файловой системой FAT можно разделить на системную область и область данных. Системная область логического диска создается и инициализируется при форматировании, а в последующем обновляется при работе с файловой структурой. Область данных логического диска содержит обычные файлы и файлы-каталоги; эти объекты образуют иерархию, подчиненную корневному каталогу. Элемент каталога описывает файловый объект, который может быть либо обычным файлом, либо файлом-каталогом. Область данных, в отличие от системной области, доступна через пользовательский интерфейс операционной системы. Системная область состоит из следующих компонентов (расположенных в логическом адресном пространстве друг за другом):

- загрузочной записи (Boot Record, BR);
- зарезервированных секторов (Reserved Sectors, ResSec);
- таблицы размещения файлов (File Allocation Table, FAT);

– корневого каталога (Root Directory, RDir).

Непосредственно после системной области расположена область данных логического диска (Рисунок 1.1).

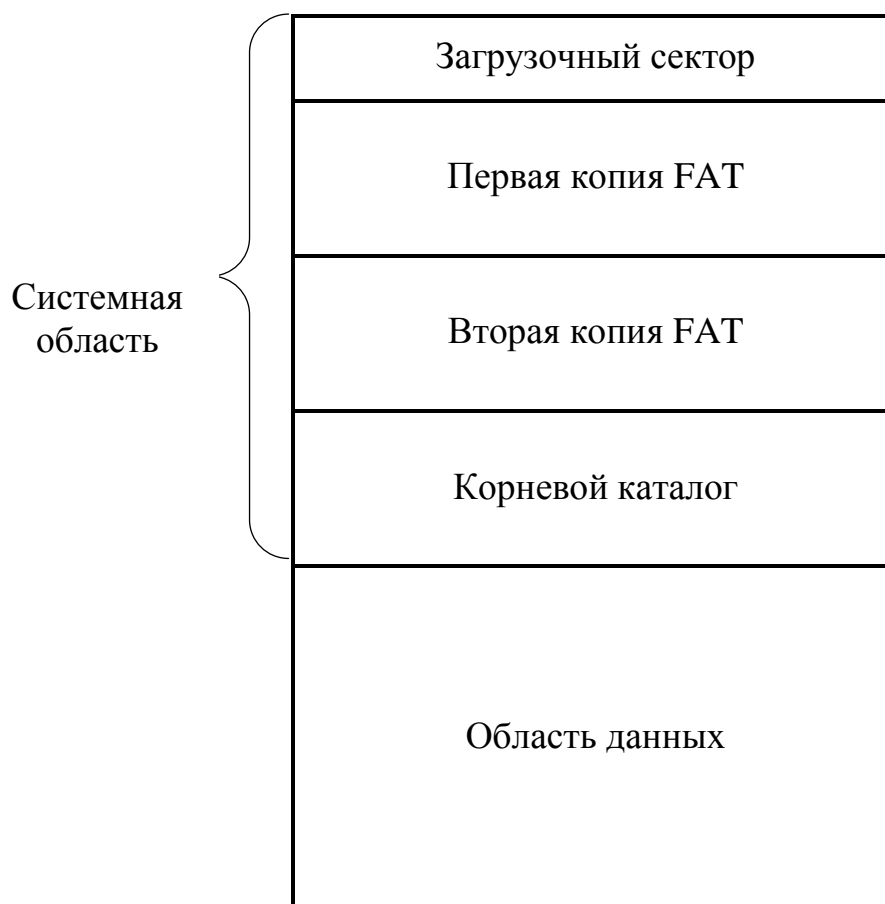


Рис. 1.1. Структура логического диска FAT16

Запись начальной загрузки

Запись начальной загрузки имеют все диски, даже если на них нет операционной системы. Она всегда имеет длину в один сектор и занимает сектор 0. Основное, что в нем содержится, – короткая программа, которая запускает процесс загрузки операционной системы с диска в память. Программа начинается с первых трех байтов сектора. Байты 1 и 0 содержат инструкцию для перехода к исполняемой части программы (обычно `jmp 2b`), а байт 2 содержит `90h` (NOP). Пространство между двумя этими частями

программы содержит параметрическую информацию о диске (таблица 1).
Последние два байта записи содержат индикацию 55А.

Таблица 1

Параметрическая информация о диске с FAT16

Смещение	Длина	Содержание
00h	3 байта	Команда перехода на программу начально загрузки
03h	8 байт	Идентификатор изготовителя
0Bh	1 слово	Размер сектора в байтах
0Dh	1 байт	Размер кластера в секторах
0Eh	1 слово	Количество зарезервированных секторов в начале
10h	1 байт	Количество копий FAT
11h	1 слово	Количество элементов основного каталога
13h	1 слово	Общее количество секторов диска
15h	1 байт	Идентификатор формата
16h	1 слово	Размер FAT в секторах
18h	1 слово	Количество секторов на дорожке
1Ah	1 слово	Количество поверхностей (головок)
1Ch	4 байта	Количество скрытых секторов
27h	4 байта	Серийный номер тома
2Bh	11 байт	Метка тома

FAT32 диски содержат больше зарезервированных секторов, чем диски FAT12 или FAT16. Число зарезервированы секторов обычно 32, но может изменяться (Таблица 2).

Так как в FAT32 блок параметров BIOS (BPB) имеет больший размер, чем стандартный BPB, то блок начальной загрузки на диске занимает более 1 сектора. Кроме того, имеется сектор в зарезервированной области на дисках FAT32, который содержит значения количества свободных секторов и номер кластера, который был распределен последним (Таблица 2). Эти дополнительные поля позволяют системе инициализировать значения без необходимости чтения всей таблицы размещения файлов.

Таблица 2

Параметрическая информация о диске с FAT32

Диапазон	Описание
0-2	Ассемблерная команда перехода к загрузочному сектору
3-10	Имя OEM в кодировке ASCII
11-12	Количество байтов в секторе
13-13	Количество секторов в кластере
14-15	Размер зарезервированной области в секторах
16-16	Количество копий FAT
17-18	Максимальное количество файлов в корневом каталоге
19-20	16-разрядное количество секторов в файловой системе
21-21	Тип носителя
22-23	16-разрядный размер каждой копии FAT
24-25	Количество секторов в дорожке
26-27	Количество головок
28-31	Количество секторов перед началом раздела

Диапазон	Описание
32-35	32-разрядное количество секторов в файловой системе
36-39	32-разрядный размер одной копии FAT (в секторах)
40-41	Режим обновления нескольких структур FAT
42-43	Основной и дополнительный номер версии

Таблица 3

Структура данных сектора FSINFO диска с FAT16

Диапазон	Описание
0-3	Сигнатура (0x41615252)
4-483	Не используется
484-487	Сигнатура (0x6147272)
488-491	Количество свободный кластеров
492-495	Следующий свободный кластер
496-507	Не используется
508-511	Сигнатура (0xAA550000)

Таблица распределения дискового пространства

Таблица распределения дискового пространства (File Allocation table – FAT) содержит информацию о расположении файлов, свободном пространстве на дисках и неисправных блоках, а также идентификатор формата диска. Для каждого файла поддерживается цепочка элементов FAT, каждый из которых указывает область фиксированной длины, занимаемой файлом на диске.

В каталоге, содержащем файл, есть указатель к началу цепочки. При удалении файла элементы и соответствующие области диска освобождаются и могут быть использованы для другого файла. Основное преимущество этой организации заключается в возможности не только последовательного, но и прямого доступа к данным файла. Недостатком является постепенное фрагментирование диска при активной работе с файловой системой (создание и удаление файлов) – файлы не занимают непрерывные области, фрагментирование значительно снижает быстродействие. Поскольку FAT содержит очень важную для надежности данных информацию, то на диске хранится множественное количество копий. Каждый элемент FAT (за исключением первых двух) соответствует определенному кластеру диска.

Кластер – это группа стандартных секторов размером 512 байт. Группировка секторов необходима, чтобы уменьшить размер FAT. Однако большие кластеры, используемые на фиксированном диске, напрасно расходуют дисковое пространство при записи маленьких файлов. Первые два элемента FAT (3 или 4 байта) используются в качестве идентификатора формата диска. Значащим является только первый байт, остальные всегда содержат FFh.

Поскольку первые два элемента FAT (0 и 1) используются идентификатором формата диска, то нумерация фактических элементов начинается с двух. Нумерацию кластеров также принято начинать с двух. Каждый элемент FAT содержит число, которое идентифицирует соответствующий кластер или как свободный, или включенный в файл, или неиспользуемый, или зарезервированный для специальных целей.

Длина элементов FAT 12, 16 или 32 бита. Использование 12-битовых элементов FAT преследует цель максимально уменьшить размер FAT и использовалось на дискетах. В данном случае значения двух соседних элементов сохраняются в трех последовательных байтах. Это усложняет

прослеживание цепочки элементов, но работу с FAT обычно берет на себя операционная система, которая осуществляет необходимые вычисления.

Для преобразования номера кластера в номер сектора нужно выполнить следующее:

- из номера кластера вычитается 2;
- результат умножается на количество секторов в одном кластере;
- к полученному результату добавляется количество секторов, занимаемых системными областями.

В том случае, когда количество секторов больше 4080, используется FAT с 16-битовыми элементами. Специальные значения элементов FAT, такие как дефектный блок, последний элемент цепочки FAT и т.д. являются логическим расширением значений 12-битового варианта – просто добавляется шестнадцатеричное F в начале. При поиске следующего элемента в цепочке FAT сложные вычисления уже не нужны, так как элементы имеют длину 2 байта.

Следует обратить внимание, что старшие 4 бита 32-разрядных значений в FAT32 таблице размещения файлов зарезервированы и не являются частью номера кластера. Приложения, которые непосредственно читают FAT32 таблицу размещения файлов, должны маскировать эти биты и сохранять их при записи новых значений.

Пример.

В элементе каталога для файла FILEN.TXT указано

- номер первого кластера, распределенного файлу равен 1B4h;
- длина файла 1928 байт;
- размер кластера – 2 сектора.

В этом случае цепочка кластеров для этого файла будет выглядеть так, как показано на рисунке 1.2.

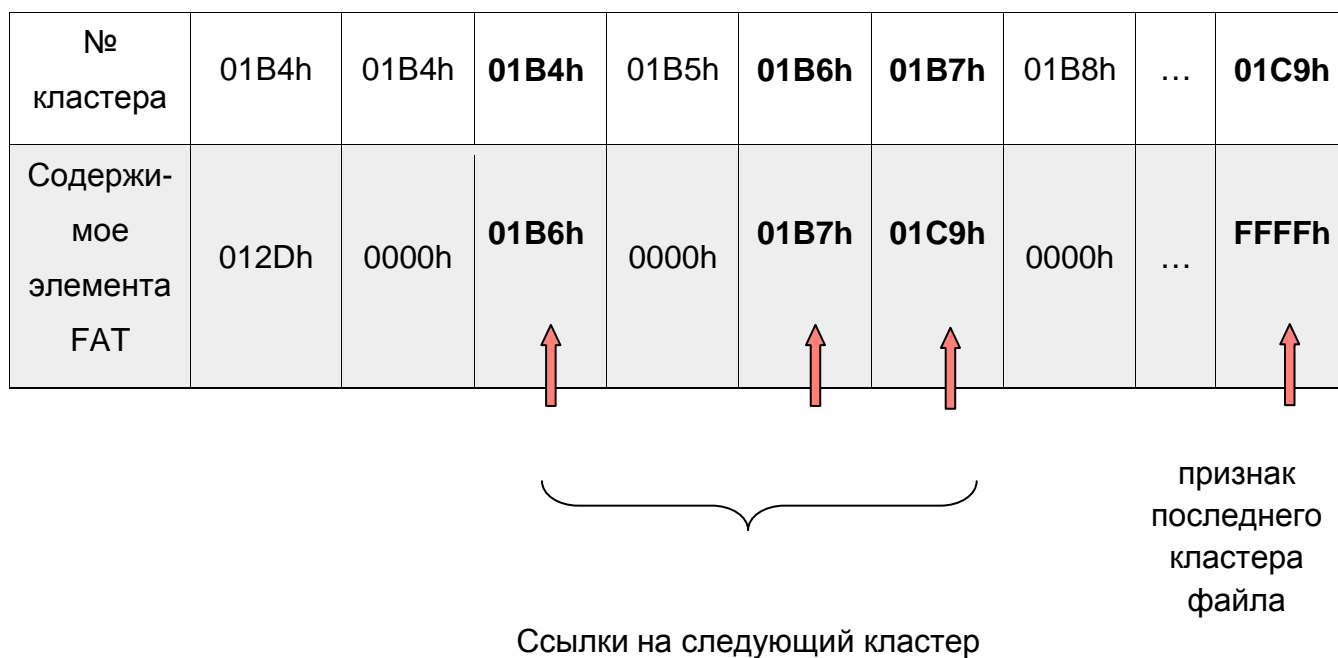


Рис. 1.2. Структура логического диска FAT16

Система выделяет файлу кластеры, для которых в соответствующих элементах таблицы FAT находятся нули. При этом, свободные кластеры ищутся каждый раз с начала таблицы FAT.

Каталоги и файлы

Основной (корневой) каталог – это начало древовидной файловой структуры диска. Как и все остальные каталоги, создаваемые на диске, он содержит информацию о файлах, которые ему принадлежат. Наиболее существенная часть этой информации – имя файла и указатель к началу соответствующей цепочки элементов FAT. После формирования диска корневой каталог либо пуст, либо содержит системный файлы. Пользователь строит файловую структуру диска, добавляя к основному каталогу файлы или новые каталоги. С точки зрения файловой организации каталоги тоже являются файлами. Исключением является корневой каталог, который находится на определенном месте на диске, имеет фиксированную длину и который нельзя удалить.

Корневой каталог диска FAT32 не имеет фиксированного расположения, как на дисках с FAT12 и FAT16. На дисках с FAT32, корневой каталог –

обычная цепочка кластеров. Элемент в структуре BPB FAT32 содержит номер первого кластера в корневом каталоге, что позволяет корневому каталогу расти так, как это может быть необходимо в процессе работы.

Для каждого файла на диске имеется один элемент в определенном каталоге. Один элемент корневого каталога выделяется для метки диска. Для каждого каталога имеется элемент в его родительском каталоге. Кроме того, каждый каталог, за исключением корневого, содержит по одному элементу со специальными именами «.» и «..». Эти элементы указывает начало цепочки в FAT для текущего каталога и для его родительского каталога.

При создании файла файловая система ОС выделяет один элемент в таблице каталога, куда записываются свойства файла, определяет количество кластеров (секторов), необходимых для размещения файла. Затем файловая система в FAT таблице находит первый свободный кластер и записывает его номер в элемент каталога, а в ячейку FAT таблицы, соответствующую первому свободному кластеру записывает номер следующего свободного кластера (сектора) и т.д. пока не будет определен последний, необходимый для размещения кластер, в который ОС поставит метку FFFh (последний) и только после этого ОС разместит данные в области данных по выбранным секторам.

Каждый элемент каталога имеет длину 32 байта и состоит из восьми полей (таблица 2). Для Fat32 элемент каталога отличается длиной номера начального кластера, поэтому структура его элементов имеет несколько другой вид и представлена в таблице 3.

Таблица 2

Элементы каталога FAT12 и FAT16

Смещение	Число байтов	Содержание
00h	8	Имя файла в кодах ASCII
08h	3	Расширение имени файла в кодах ASCII
0Bh	1	Байт атрибутов файла
0Ch	10	Зарезервировано
16h	2	Время создания или последней модификации
18h	2	Дата создания или последней модификации
1Ah	2	Номер начального кластера файла
1Ch	4	Фактическая длина файла в байтах

Первый байт поля имени используется для обозначения трех специальных случаев:

1) значение 00H в первом байте показывает, что этот элемент каталога никогда не был использован. Так как каталог заполняется последовательно, это означает, что и следующие за ним элементы не были использованы;

2) при удалении файла DOS записывает E5H в первом байте соответствующего элемента каталога. Все остальные баты элемента остаются без изменений. Элемент становится свободным, но после него могут быть другие активные элементы. При удалении файла DOS освобождает и все элементы соответствующей цепочки в FAT таблице. Но сам файл продолжает существовать на диске, что позволяет соответствующим программам восстановить удаленный файл, если элемент каталог или кластеры файла в области данных не используется другим файлом;

3) значение 2EH (символ «.») в первом байте показывает, что этот элемент служит для описания самого каталога. Если второй байт содержит 2EH, элемент описывает родительский каталог («..»).

Таблица 3

Элементы каталога FAT32

Смещение	Размер	Значение
0	8	Имя файла в кодах ASCII
8	3	Расширение имени файла в кодах ASCII
11	1	Атрибут
12	8	Зарезервировано
20	2	Номер начального кластера (старшие разряды)
22	2	Время создания или последней модификации
24	2	Дата создания или последней модификации
26	2	Номер начального кластера (младшие разряды)
28	4	Фактическая длина файла в байтах

Для файлов, которым не выделено места, и для метки тома, поле номера кластера содержит 0000H. Байты атрибута определяют каждый файл как системный, скрытый и т.д. Значение битов байта атрибута приведены в таблице 4.

Таблица 4

Назначение битов байта атрибутов

7 6 5 4 3 2 1 0	ЗНАЧЕНИЕ	НАЗНАЧЕНИЕ
. 1	1h	защищенный
. 1 .	2h	скрытый
. 1 . .	4h	системный
. . . . 1 . . .	8h	метка тома
. . . 1	10h	каталог
. . 1	20h	архивный
. 0	40h	не используется
0	80h	не используется

Когда программа отправляет запрос к операционной системе с требованием предоставить ей содержимое какого-либо файла, ОС просматривает его запись в каталоге, чтобы найти первый кластер этого файла. Затем она обращается к элементу FAT для данного кластера, чтобы найти следующий кластер в цепочке. Повторяя этот процесс, пока не обнаружит последний кластер файла, таким образом ОС определяет, какие кластеры принадлежат данному файлу. Таким образом, система может предоставить программе любую часть запрошенного файла.

На рисунке 1.3 показана схема чтения файла FILE6.TXT.

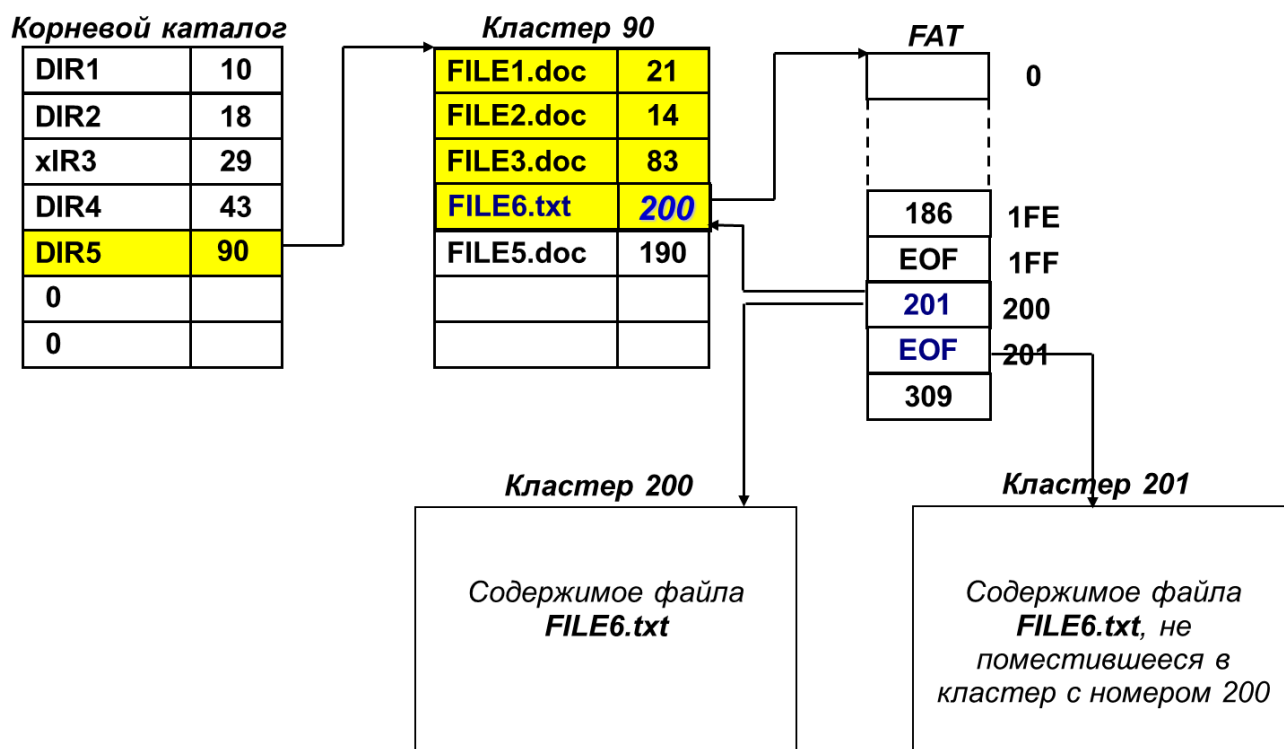


Рис. 1.3. Чтение файла

Благодаря простоте организации файловой системы FAT для обычного USB накопителя, если на ней не хранятся файлы размером более 4 Гб, FAT32 будет самым лучшим выбором, а флеш-накопитель будет прочитан практически где угодно.

2. ВЫПОЛНЕНИЕ ЛАБОРАТОРНОЙ РАБОТЫ

2.1. Описание лабораторной установки

Основное меню программы (Рисунок. 2) состоит из четырех пунктов:

1. Задание:

1.1. Начать выполнение работы – открывает диалог выбора варианта задания для лабораторной работы или сбрасывает весь прогресс выполнения лабораторной включая результаты прохождения теста;

1.2. Пройти тест – пункт меню открывающий окно для прохождения теста;

1.3. Задание по варианту – открывает окно с заданиями выбранного варианта.

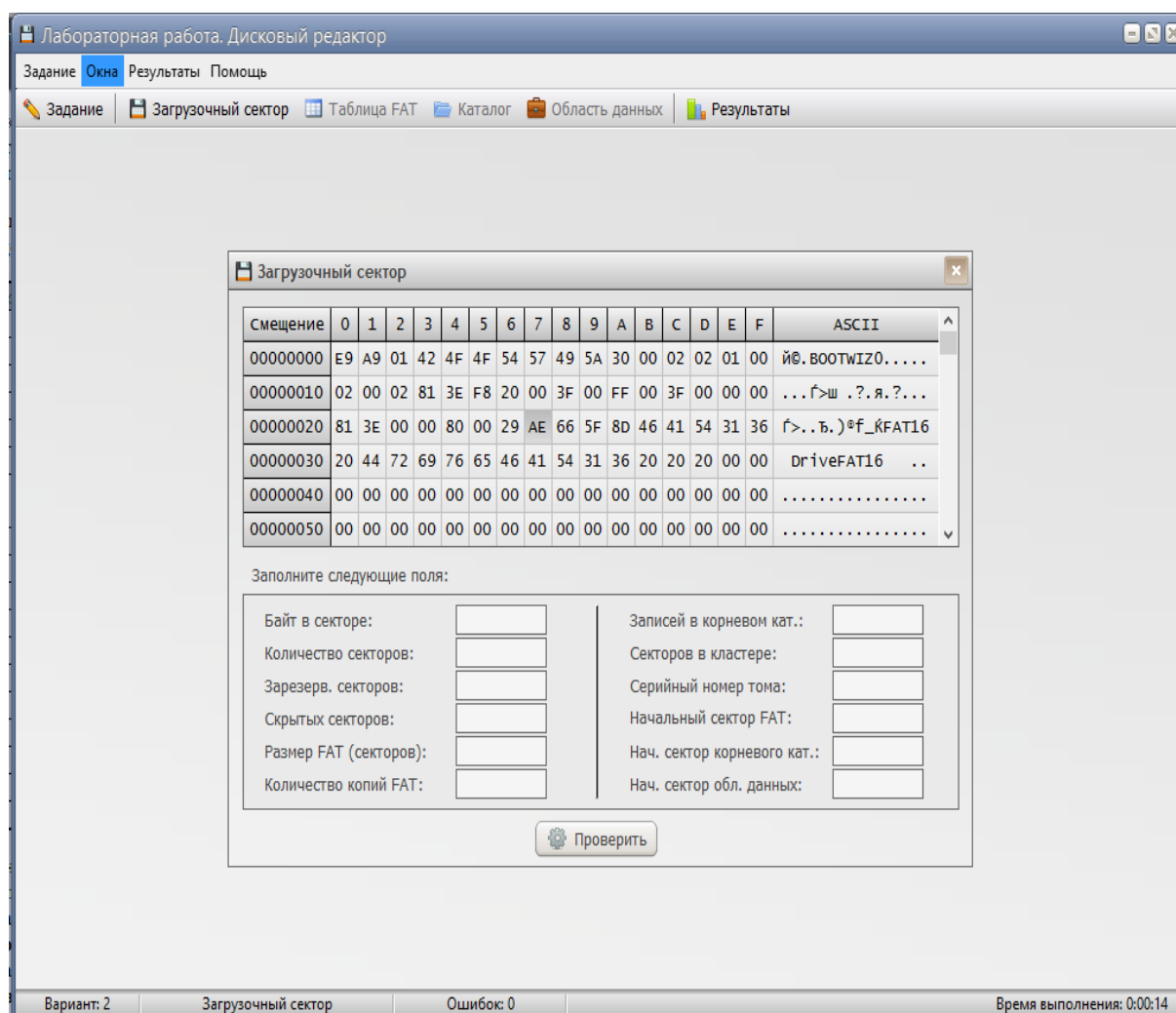


Рис. 2.1. Основное окно программы

2. Окна:

- 2.1. Загрузочный сектор – открывает окно просмотра и редактирования загрузочного сектора;
- 2.2. Таблица FAT – открывает окно просмотра и редактирования таблицы FAT;
- 2.3. Каталог – открывает окно просмотра и редактирования каталогов;
- 2.4. Область данных – открывает окно просмотра и редактирования области данных.

3. Результаты:

- 3.1. Окно результатов – выводит результаты по каждому заданию, такие как время выполнения, балл, количество ошибок;
- 3.2. Загрузить файлы результатов – загружает файл результатов;
- 3.3. Сохранить результаты – сохраняет прогресс выполнения лабораторной работы;
- 3.4. Восстановление сессии

4. Помощь:

- 4.1. Вызов справки F1 – открывает файл справки;
- 4.2. О программе – выводит информацию о программе.

2.2. Выполнение теста и начального этапа работы

Рекомендуется начать выполнение работы с теста. На прохождение теста дается пять минут. В тесте десять вопросов. На каждый вопрос возможен один или несколько вариантов ответа. Будьте внимательны, если есть возможность выбрать несколько вариантов, это значит, что правильных ответов как минимум два. Настоятельно рекомендуется прочитать теорию, так как некоторые вопросы достаточно сложные.

Помимо теста, перед выполнением самих заданий необходимо заполнить поля с параметрами системы. Размещение параметров в загрузочном секторе, приведено на рисунке 3 и соответствует информации в таблице 1:

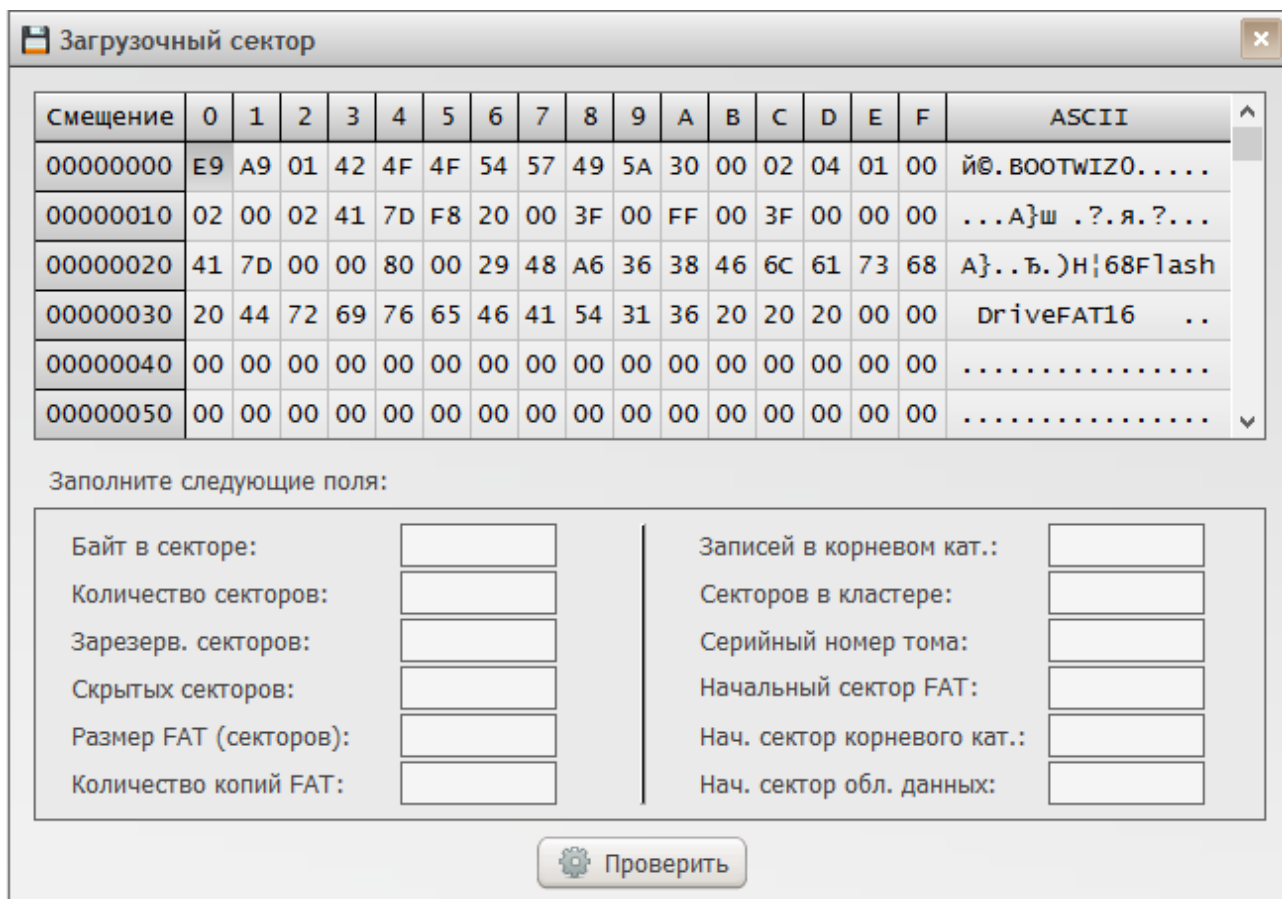


Рис. 2.2. Окно загрузочного сектора

В отображенном загрузочном секторе все параметры представлены в шестнадцатеричной системе счисления, но в поля они должны быть занесены в десятичной системе счисления. Кроме того, в компьютерах, использующих процессоры Intel, все данные хранятся так, что младший байт находится по младшему адресу, то есть сначала записывают последний (младший) байт, а потом записывают первый (старший) байт. В данном случае это означает, что для перевода в десятичную систему счисления, байты чисел из загрузочного сектора требуется считывать с начиная с последнего к первому. Например, запись 3F000000 в таблице соответствует числу 0000003F.

Начальный сектор FAT всегда единице. Начальный сектор корневого каталога размещается после всех системных секторов, куда входят зарезервированные, скрытые сектора и сектора, занятые под таблицу FAT.

Начальный сектор корневого каталога можно рассчитать по следующей формуле

$$\begin{aligned} \text{№}_{\text{нач.сек.}} &= \text{количество зарезервированных секторов} + \\ &+ \text{размер FAT} * \text{количество копий FAT} + 1. \end{aligned}$$

Для заданного примера получается $32 * 2 + 1 + 1 = 66$

Начальный сектор области данных располагается после корневого каталога, значит:

Начальный сектор области данных = начальный сектор корневого каталога + сектора, занятые корневым каталогом.

Поскольку известно количество элементов корневого каталога и размер одного элемента тоже (для тех, кто читал теорию), сектора, занятые корневым каталогом сосчитать не составляет труда.

Пример заполнения полей представлен на рисунке 2.3.

Смещение	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
00000000	E9	A9	01	42	4F	4F	54	57	49	5A	30	00	02	04	01	00	й@. BOOTWIZ0.....
00000010	02	00	02	41	7D	F8	20	00	3F	00	FF	00	3F	00	00	00	...A}ш .?.я.?....
00000020	41	7D	00	00	80	00	29	48	A6	36	38	46	6C	61	73	68	A}. .Ъ.)H!68Flash
00000030	20	44	72	69	76	65	46	41	54	31	36	20	20	20	00	00	DriveFAT16 ..
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Заполните следующие поля: Задание выполнено ✓

Байт в секторе:	512	Записей в корневом кат.:	512
Количество секторов:	32065	Секторов в кластере:	4
Зарезерв. секторов:	1	Серийный номер тома:	943105608
Скрытых секторов:	63	Начальный сектор FAT:	1
Размер FAT (секторов):	32	Нач. сектор корневого кат.:	65
Количество копий FAT:	2	Нач. сектор обл. данных:	97

Рис. 2.3. Заполнение данных о системе

Таким образом, первое задание лабораторной работы выполнено, получена и расшифрована параметрическая информация о логическом диске, с которым предстоит работать далее. Полученная информация будет использоваться при выполнении любых операций с файлами.

2.3. Работа с файлами

2.3.1. Создание файла

При записи на диск нового файла выполняется следующая процедура (в данной лабораторной установке процедура несколько упрощена).

В заданном каталоге ищется первый свободный элемент, в котором нужно заполнить данные, описывающие создаваемый файл. Чтобы узнать номер начального кластера для файла, надо найти в таблице FAT первый свободный ее элемент.

По данным таблицы формата диска, расположенной в его boot-секторе, и известному размеру файла в байтах вычисляется количество кластеров, необходимых для размещения файла.

Начиная с начального элемента FAT, определяется цепочка свободных элементов (это элементы FAT, содержащие «0»), расположенных в порядке возрастания их номеров. При этом в каждый элемент, содержащий «0» записывается номер следующего элемента цепочки, а в последний элемент записывается код FFFFh (EOF), означающий, что данный сектор будет последним для данного файла.

Файл последовательно записывается в кластеры рабочей области диска, номера которых соответствуют номерам элементов FAT, включенных в цепочку.

Например, при записи файла размером 1234 байта на диск формата 512 байтов в секторе и 1 сектор в кластере, этому файлу будет выделено три кластера с номерами 10, 11 и 16. При этом последний (16-й) кластер будет занят лишь частично.

На рисунке 2.4 показан пример задания для создания файла.

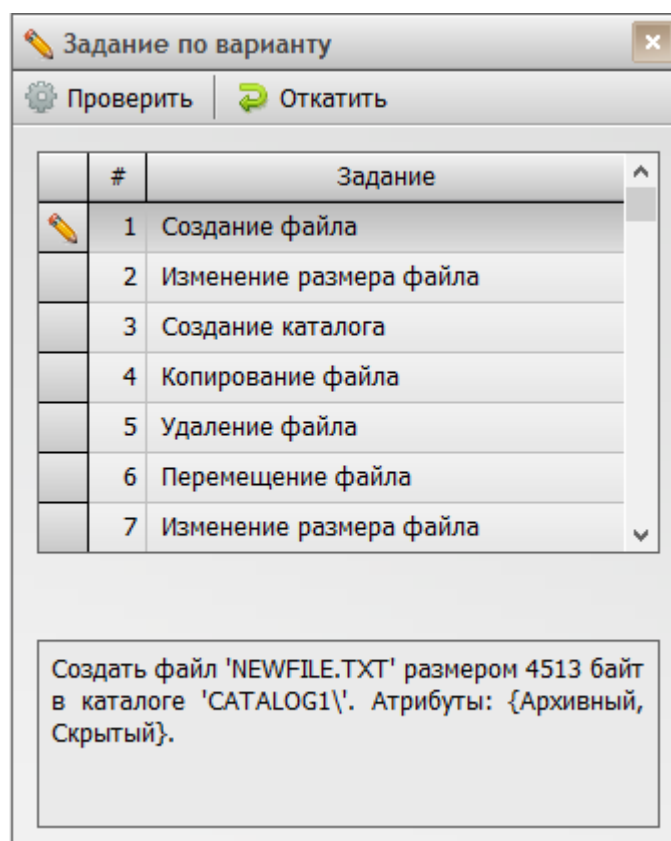


Рис. 2.4. Задание по созданию файла

Для создания файла нужно выполнить следующие действия:

1. В окне «Каталог» найти заданный каталог и перейти в него (Рисунок 2.5).

Сектор 65							
Flash Dr	ive	0	06.03.11	21:55:30	0000	--T---	
CATALOG1		0	06.03.11	22:29:46	0002	-D----	
WMIAPRPL	H	1087	03.03.11	03:07:30	000C	A-----	
_ONG5	TXT	1381	22.06.10	15:21:56	000D	A-----	
SONG8	TXT	1713	19.09.10	20:18:00	000E	A-----	

Рис. 2.5. Выбор нужного каталога

2. В этом каталоге найти свободный дескриптор, выделить его и нажать кнопку «Изменить» (Рисунок 2.6).

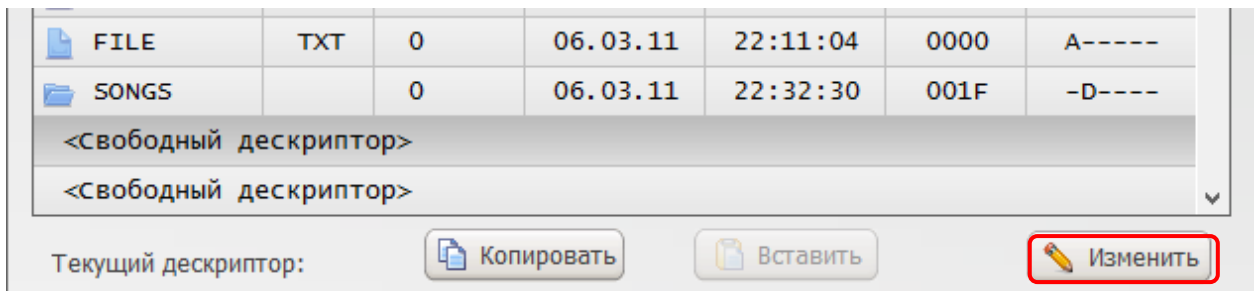


Рис. 2.6. Изменение дескриптора каталога

ВАЖНО! Может случиться ситуация, когда в каталоге нет свободных дескрипторов. На рисунке 2.7 изображен пример такой ситуации.

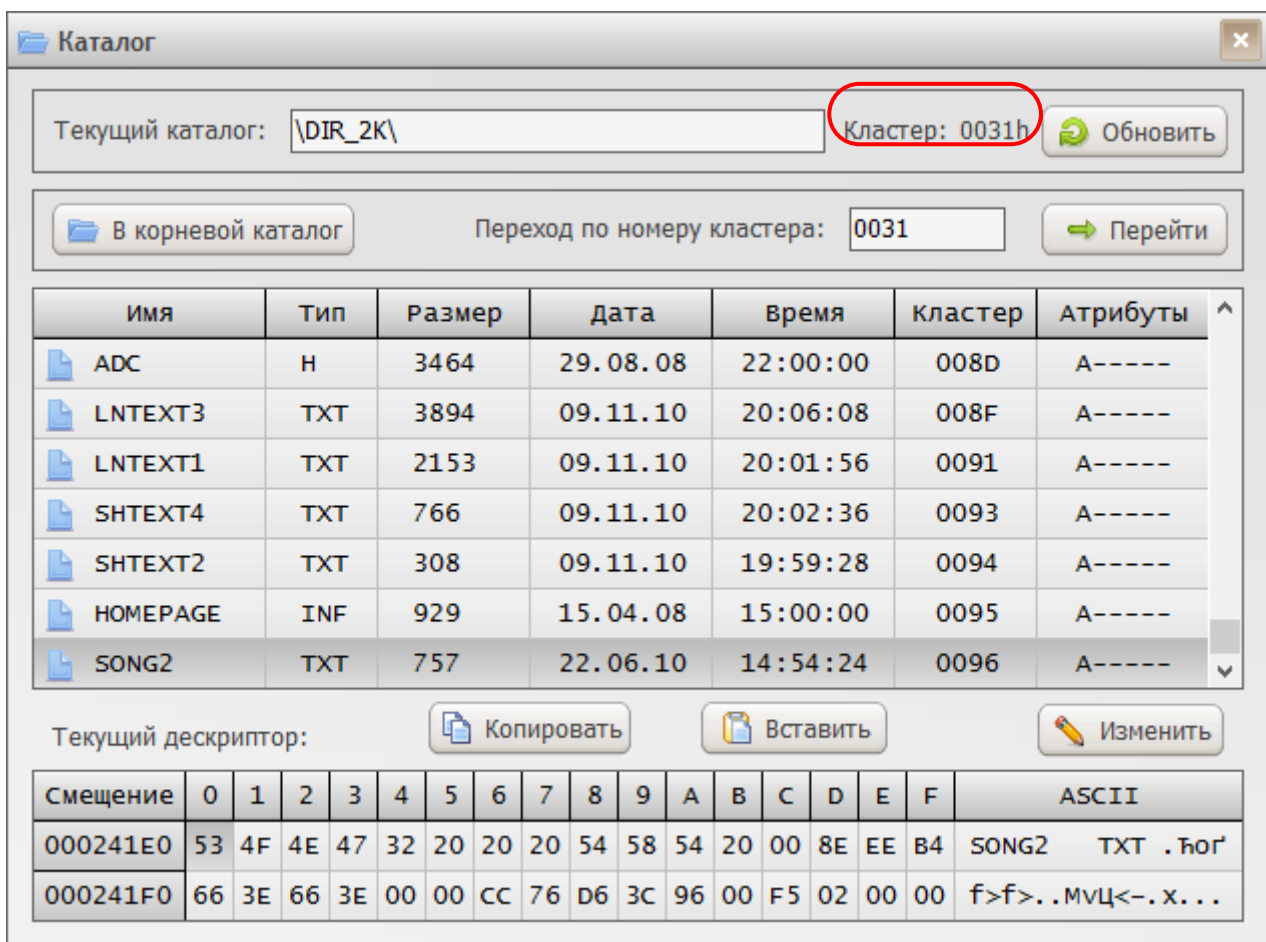


Рис. 2.7. В каталоге нет свободных дескрипторов

Для того, чтобы увеличить размер каталога, надо найти в таблице FAT элемент, описывающий данный каталог (его номер выделен на рисунке 2.7), и добавить в цепочку еще один элемент, описывающий свободный (содержит номер 0000) кластер, как показано на рисунке 2.8, т.е. в ячейку с номером 0031

записать ссылку на свободный элемент (в примере это 0056), а в выбранную ячейку записать символ конца файла.

Таблица FAT

кластер	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0030	<EOF>	0056	<EOF>	<EOF>	0035	<EOF>	0037	0038
0038	0039	<EOF>	003B	003C	003D	<EOF>	003F	0040
0040	0041	0042	<EOF>	0044	0045	<EOF>	0047	0048
0048	<EOF>	004A	004B	<EOF>	004D	<EOF>	004F	<EOF>
0050	0051	<EOF>	0053	<EOF>	<EOF>	<EOF>	<EOF>	<EOF>
0058	0000	<EOF>	005B	<EOF>	005D	<EOF>	005F	<EOF>
0060	<EOF>	<EOF>	<EOF>	<EOF>	<EOF>	<EOF>	<EOF>	<EOF>
0068	<EOF>	<EOF>	<EOF>	<EOF>	<EOF>	<EOF>	<EOF>	<EOF>
0070	0000	<EOF>	<EOF>	<EOF>	<EOF>	<EOF>	<EOF>	<EOF>
0078	<EOF>	<EOF>	<EOF>	0000	<EOF>	<EOF>	<EOF>	<EOF>

Изменить Кластер 0031: 0056

Рис. 2.8. Добавление кластера каталогу

Затем необходимо очистить область данных выбранного кластера. Для этого надо перейти в окно «Область данных» (Рисунок 2.9). В правой части окна расположен список кластеров с данными, в которых можно выполнять различные операции. В этот список надо добавить номер нового кластера и нажать кнопку «Очистить кластеры», таким образом, содержимое кластера будет очищено, и в каталоге появятся свободные дескрипторы (Рисунок 2.10).

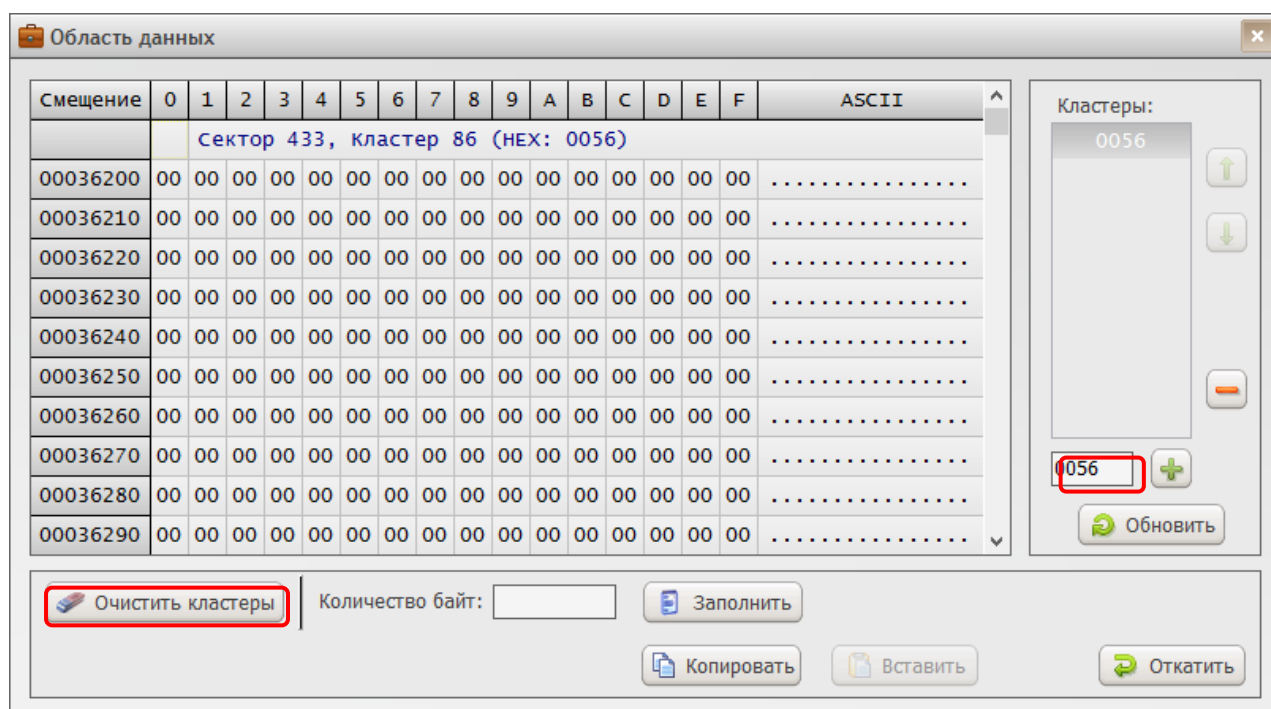


Рис. 2.9. Очистка кластера

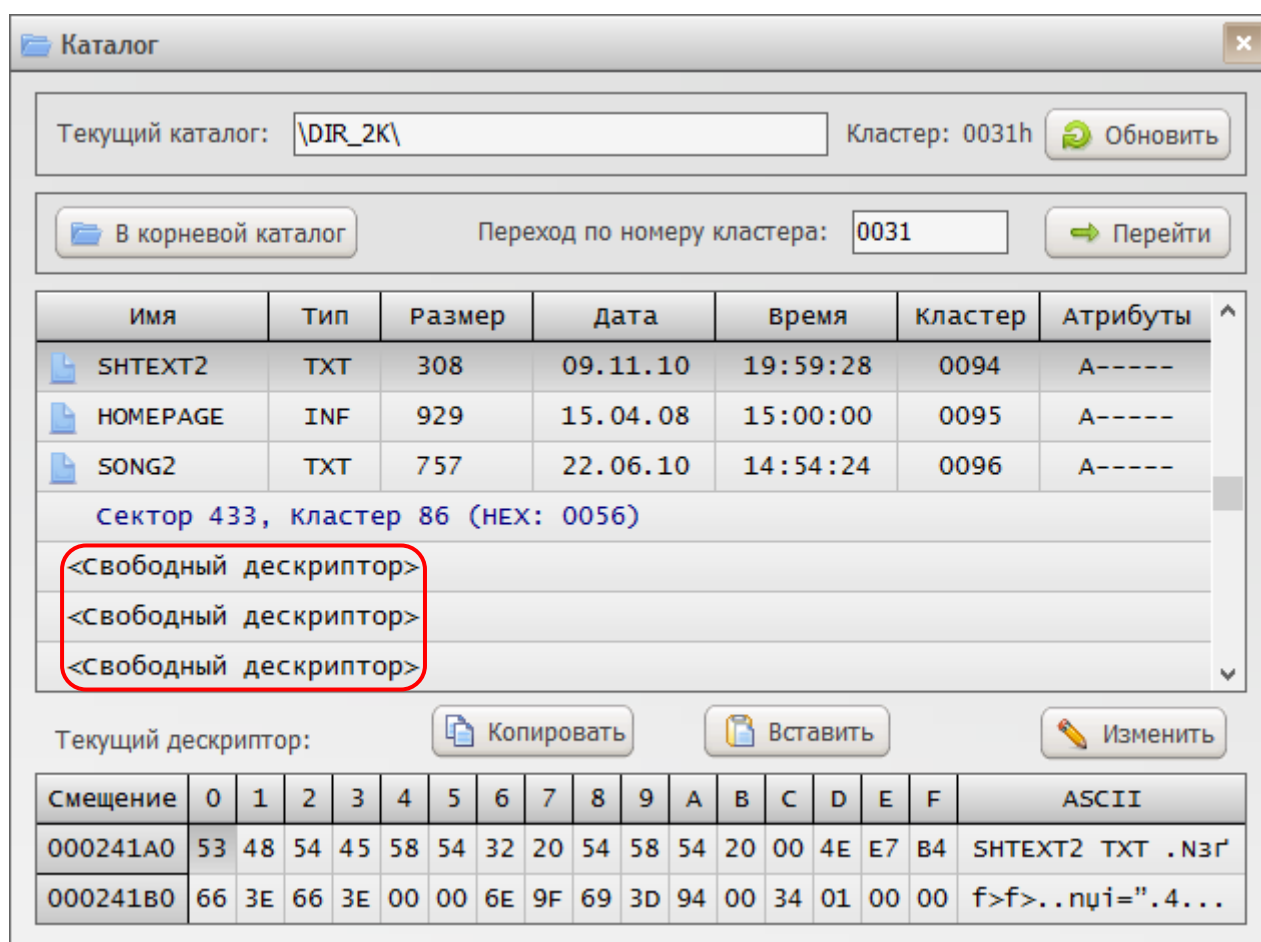


Рис. 2.10. Свободные дескрипторы в каталоге

ВАЖНО! Если надо создать файл с размером 0 т.е. пустой, в поле «Кластер» записывается значение 0000 и шаги 4 и 5 не выполняются. Пример заполнения файлового дескриптора для пустого файла показан на рисунке 2.11.

Редактирование дескриптора каталога

Имя: SHTEXT2 Расширение: TXT Атрибуты (HEX): 20

Размер: 0 Кластер (HEX): 0000

☐ Пометить как удаленный ☐ Очистить дескриптор

#	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
00	53	48	54	45	58	54	32	20	54	58	54	20	00	4E	E7	B4	SHTEXT2 TXT .N3Г
10	66	3E	66	3E	00	00	EA	8B	83	4A	00	00	00	00	00	00	f>f>..к<ѓJ.....

Отменить Сохранить

Рис. 2.11. Дескриптор пустого файла

В появившемся окне редактора дескриптора каталога надо заполнить поля, описывающие файл. Для внесения изменений надо убрать галочку в поле «Очистить дескриптор».

Поля «Имя», «Размер», «Расширение» заполняются в соответствии с заданием. Для заполнения поля «Кластер» необходимо перейти в окно «Таблица FAT» и найти в таблице первый элемент, описывающий свободный кластер. Если кликнуть на этот элемент, то внизу окна появится его номер (Рисунок 2.12). Этот номер и следует записать в поле «Кластер». При заполнении поля «Кластер» необходимо помнить, что размерность его WORD (2 байта), и запись A3, к примеру, программа не примет. Необходимо писать 00A3.

Таблица FAT								
кластер	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
	Сектор 1							
0000			<EOF>	0004	0005	<EOF>	0007	<EOF>
0008	0009	<EOF>	<EOF>	<EOF>	<EOF>	0000	<EOF>	<EOF>
0010	0011	<EOF>	<EOF>	0000	<EOF>	<EOF>	<EOF>	<EOF>
0018	0000	<EOF>	001B	001C	001D	001E	<EOF>	<EOF>
0020	<EOF>	0000	<EOF>	<EOF>	<EOF>	0026	0027	<EOF>
0028	0000	0000	0000	0000	002D	002E	<EOF>	<EOF>
0030	<EOF>	<EOF>	<EOF>	<EOF>	0035	<EOF>	0037	0038
0038	0039	<EOF>	003B	003C	003D	<EOF>	003F	0040
0040	0041	0042	<EOF>	0044	0045	<EOF>	0047	0048

Кластер 000D:

0000

Рис. 2.12. Поиск первого элемента, описывающего свободный кластер

В поле «Атрибуты» заносится байт, сформированный в соответствии с таблицей 5. Для этого необходимо просуммировать значения из таблицы, соответствующие атрибутам создаваемого файла.

Например, надо задать атрибуты файла «защищенный», «скрытый» и «архивный». Им соответствуют значения 1h, 2h и 20h соответственно. Затем надо сложить эти значения:

$$1h + 2h + 20h = 23h$$

Таким образом байт атрибутов файла с атрибутами «защищенный», «скрытый» и «архивный» имеет значение 23h.

На рисунке 2.13 показан пример заполненного дескриптора каталога.

Редактирование дескриптора каталога

Имя: Расширение: Атрибуты (HEX):

Размер: Кластер (HEX):

☐ Пометить как удаленный ☐ Очистить дескриптор

#	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
00	4E	45	57	46	49	4C	45	20	54	58	54	22	00	00	00	00	NEWFILE TXT"....
10	00	00	00	00	00	00	00	00	00	00	0D	00	A1	11	00	00ÿ...

Рис. 2.13. Заполнение дескриптора каталога

После заполнения нажать кнопку «Сохранить».

Далее надо создать цепочку FAT для нового файла. Сначала надо рассчитать сколько кластеров займет файл, для этого делим размер файла на количество байт в кластере (размер кластера = размер сектора * количество секторов в кластере) и округлить до целого в большую сторону.

Например, при кластере размером 2048 байт, файл размером 4513 байт займет: $4513 / 2048 = 2,2$ кластера. После округления получаем 3 кластера.

Затем переходим в окно «Таблица FAT». В находим ранее выбранный элемент, описывающий первый кластер файла. Если файл занимает 1 кластер, изменяем значение первого элемента на FFFF (конец файла). Если файл занимает несколько кластеров, ищем следующий свободный элемент таблицы, и его номер записываем в первый элемент цепочки. Далее по цепочке в каждый элемент записываем номер следующего свободного. В последний записываем FFFF (конец файла). Таким образом получается цепочка FAT в которой каждый элемент содержит ссылку на следующий элемент цепочки. Пример цепочки FAT представлен на рисунке 2.14.

кластер	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
	Сектор 1							
0000			<EOF>	0004	0005	<EOF>	0007	<EOF>
0008	0009	<EOF>	<EOF>	<EOF>	<EOF>	0013	<EOF>	<EOF>
0010	0011	<EOF>	<EOF>	0018	<EOF>	<EOF>	<EOF>	<EOF>
0018	<EOF>	<EOF>	001B	001C	001D	001E	<EOF>	<EOF>
0020	<EOF>	0000	<EOF>	<EOF>	<EOF>	0026	0027	<EOF>
0028	0000	0000	0000	0000	002D	002E	<EOF>	<EOF>
0030	<EOF>	<EOF>	<EOF>	<EOF>	0035	<EOF>	0037	0038
0038	0039	<EOF>	003B	003C	003D	<EOF>	003F	0040
0040	0041	0042	<EOF>	0044	0045	<EOF>	0047	0048

Рис. 2.14. Создание цепочки FAT

В данном случае файл помещается в 3 кластера, в элементе, описывающем первый кластер (000D) записан номер второго (0013), во втором номер третьего (0018), а в третьем символ конца файла FFFF.

Затем необходимо заполнить область данных файла. Для этого надо перейти в окно «Область данных» (Рисунок 2.15). Справа в этом окне расположен список кластеров, с которыми производятся действия. В этот

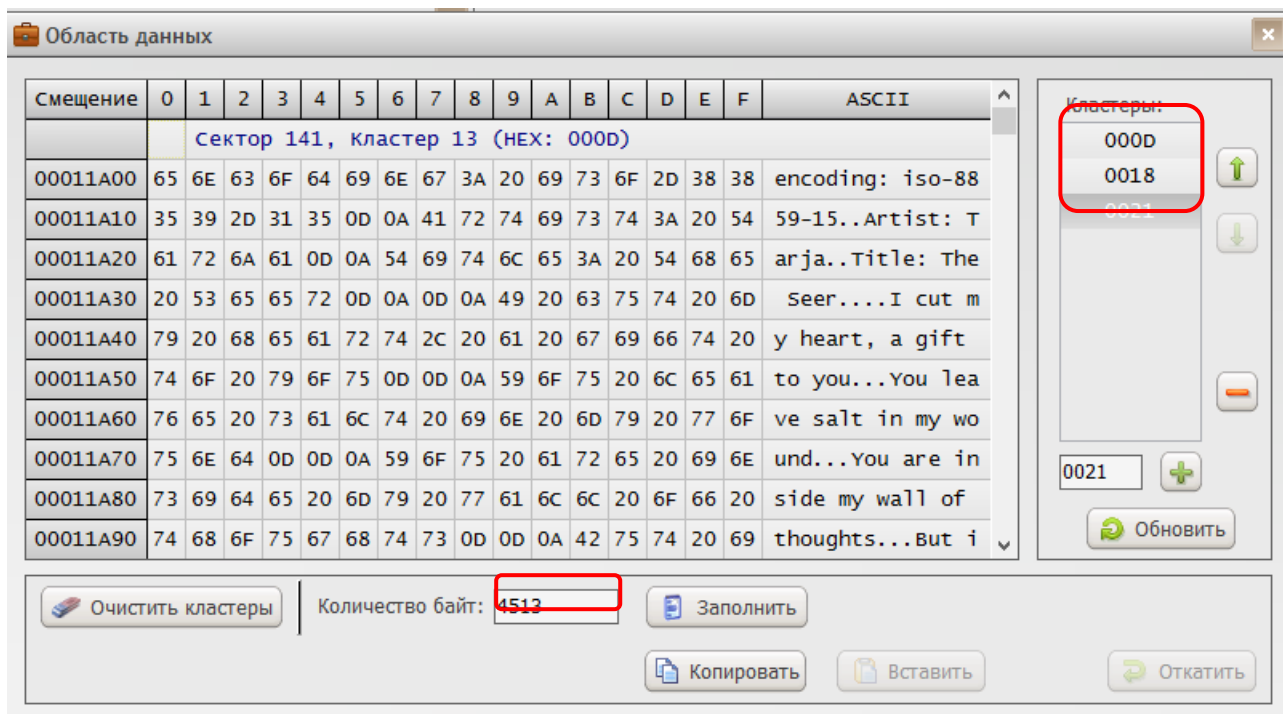


Рис. 2.15. Заполнение области данных файла

список нужно занести всю цепочку номеров кластеров, выделенных файлу, начиная с первого. Затем все эти кластеры очищаются кнопкой «Очистить кластеры».

Далее надо заполнить кластеры данными. Для этого вводится размер файла в поле «Количество байт» и нажимается «Заполнить».

На этом выполнение первого задания закончено.

2.3.2. Изменение размера файла

В следующем задании нужно изменить размер файла (Рисунок 2.16). Для изменения размера файла необходимо выполнить следующие действия:

1. Найти нужный файл в каталоге и нажать кнопку «Изменить»
2. Если старый размер файла 0 т.е. файл пустой, необходимо проделать такие же действия как при создании файла, начиная с 3 шага.
3. Если файл был не пустой, необходимо пересчитать количество занятых файлом кластеров и перестроить его цепочку FAT, т.е. удалить из нее лишние элементы или добавить недостающие.
4. Если количество кластеров не изменилось, цепочку FAT изменять не надо, а сразу менять область данных.
5. Далее необходимо заново заполнить область данных файла т.е. занести номера кластеров в список, очистить их, а затем заполнить.

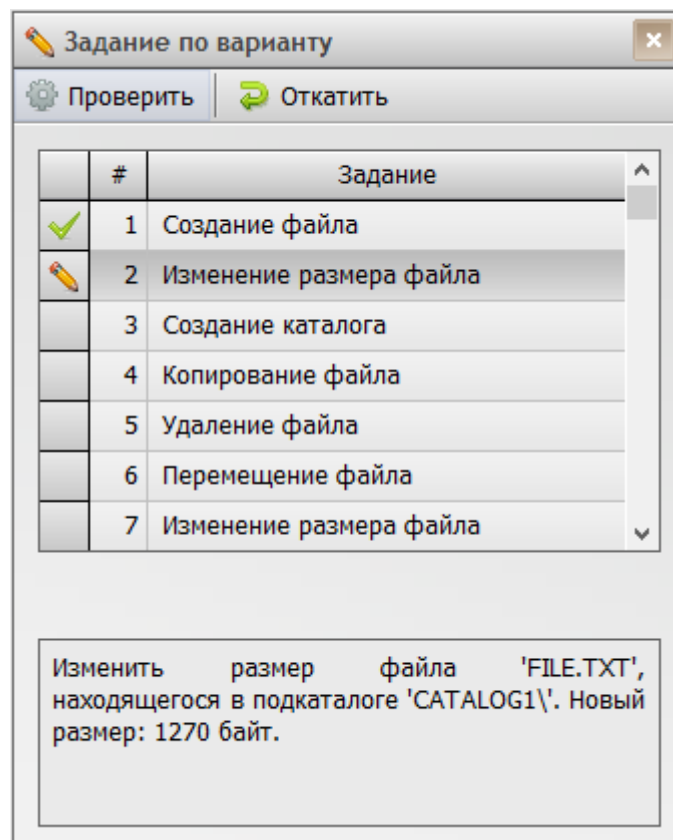


Рис. 2.16. Задание по изменению размера файла

После завершения данного задания можно переходить к выполнению следующего (Рисунок 2.17).

2.3.3. Создание каталога

Алгоритм создания каталога похож на алгоритм создания файла, с некоторыми отличиями:

1. Найти в нужном каталоге свободный дескриптор и нажать «Изменить».
2. Затем надо заполнить дескриптор каталога. В отличие от файла, каталог не имеет расширения, имеет размер 0 и атрибут «Каталог» (10h) вдобавок к обозначенным в задании. Также, в отличие от файла с размером 0, для каталога надо выделять реальный элемент в таблице FAT.

На рисунке 2.18 показан пример заполнения дескриптора каталога.

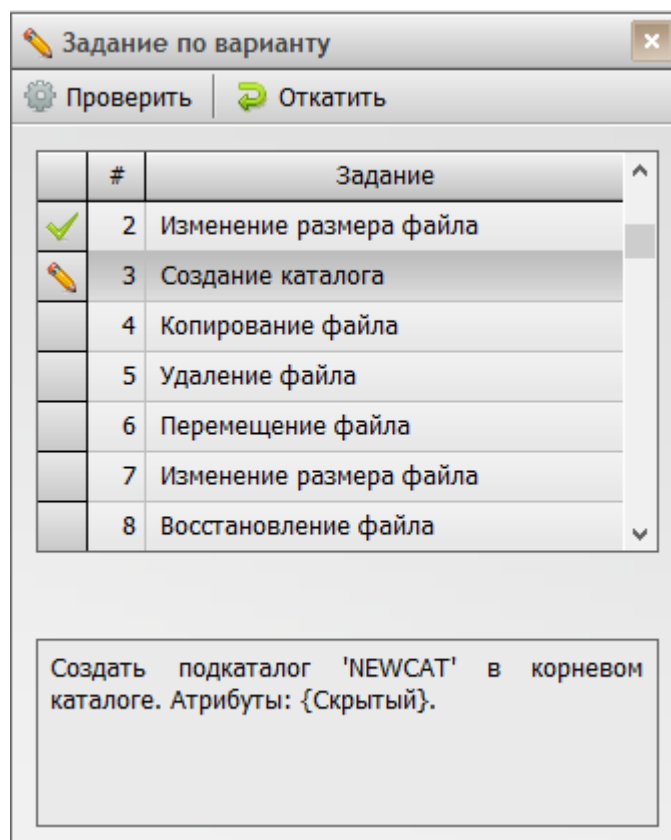


Рис. 2.17. Задание по созданию каталога

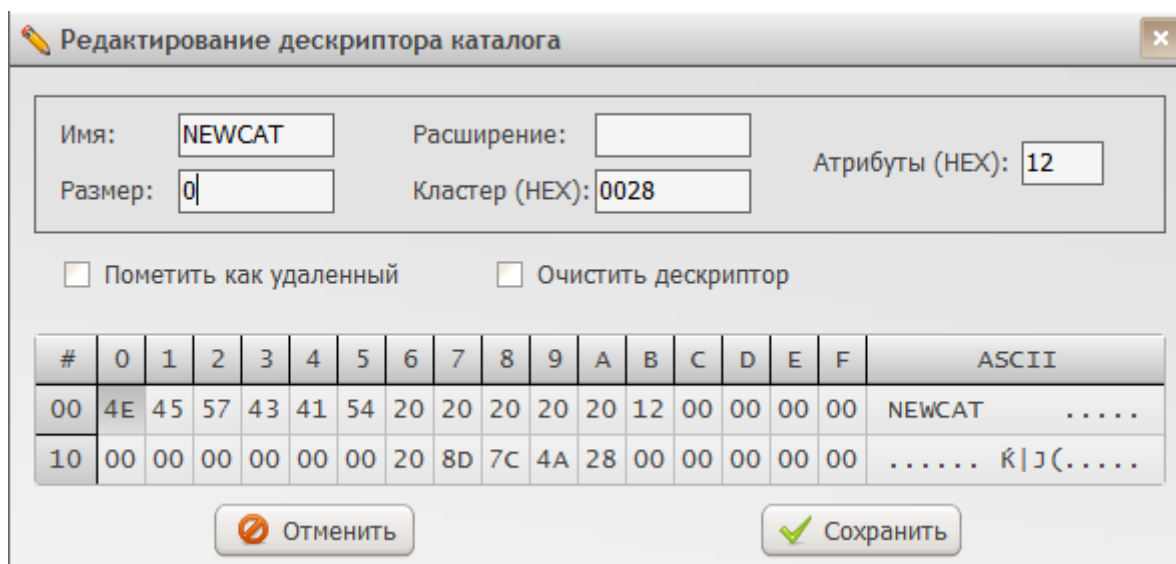


Рис. 2.18. Заполненный дескриптор каталога

3. Через окно «Область данных», очищаем кластер, предназначенный под каталог.

4. Так как каталог занимает только один кластер, то в таблице FAT ставим в соответствующий элемент метку «Конец файла».

5. Теперь необходимо создать подкаталоги «..» и «.» в созданном каталоге (Рисунок 2.19). Каталог «..» – это ссылка на родительский каталог, а «.» – ссылка на сам каталог.

Имя	Тип	Размер	Дата	Время	Кластер	Атрибуты
Сектор 249, кластер 40 (HEX: 0028)						
.		0	03.04.17	16:32:28	0028	-D--H-
..		0	06.03.11	22:29:46	0000	-D----

Рис. 2.19. Дескрипторы каталогов «.» и «..»

6. В новом каталоге выбираем первый свободный дескриптор и создаем элемент «.», он почти полностью совпадает с дескриптором описываемого каталога, отличается только именем (Рисунок 2.20). Можно просто скопировать дескриптор созданного по заданию каталога, и вставить в первый дескриптор внутри него.

#	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
00	2E	20	20	20	20	20	20	20	20	20	12	00	00	00	00	00	.
10	00	00	00	00	00	00	0E	84	83	4A	28	00	00	00	00	00,fJ(.....

Рис. 2.20. Дескриптор текущего каталога «.»

7. Выбираем следующий свободный дескриптор и создаем «..». Так как он описывает родительский каталог, дескриптор должен совпадать с дескриптором родительского каталога, опять же, кроме имени (Рисунок 2.21). Можно

скопировать дескриптор каталога, в котором по заданию создавался новый, и вставить во второй дескриптор внутри нового каталога.

Редактирование дескриптора каталога

Имя: Расширение: Атрибуты (HEX):

Размер: Кластер (HEX):

☐ Пометить как удаленный ☐ Очистить дескриптор

#	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
00	2E	2E	20	20	20	20	20	20	20	20	20	10	00	0A	B6	B3if
10	66	3E	66	3E	00	00	B7	B3	66	3E	00	00	00	00	00	00	f>f>...if>.....

Рис. 2.21. Дескриптор каталога «..»

2.3.4. Копирование, перемещение, переименование файла

Задание по варианту

#	Задание
9	Создание каталога
10	Удаление файла
11	Изменение размера файла
12	Копирование файла
13	Удаление файла
14	Перемещение файла
15	Восстановление файла

Произвести копирование файла 'SEARCH.INI', находящегося в подкаталоге 'DIR_2K\', в подкаталог 'DIR_2K\DIRECT\'. Изменить имя файла на 'SRCH_02.INI'.

Рис. 2.22. Задание по копированию файла

Обратите внимание, что при копировании файла, может потребоваться переименовать новый файл.

При переименовании файла изменяются только значения полей «Имя» (и, если необходимо, «Расширение») соответствующего дескриптора каталога.

Процедура копирования файла реализуется последовательным выполнением рассмотренных выше процедур чтения, записи и переименования.

1. Находим в указанном каталоге необходимый файл (Рисунок 2.23).
2. Выделяем его, нажимаем «Копировать».
3. Вставляем скопированный дескриптор на место свободного дескриптора, указанного в задании каталога.

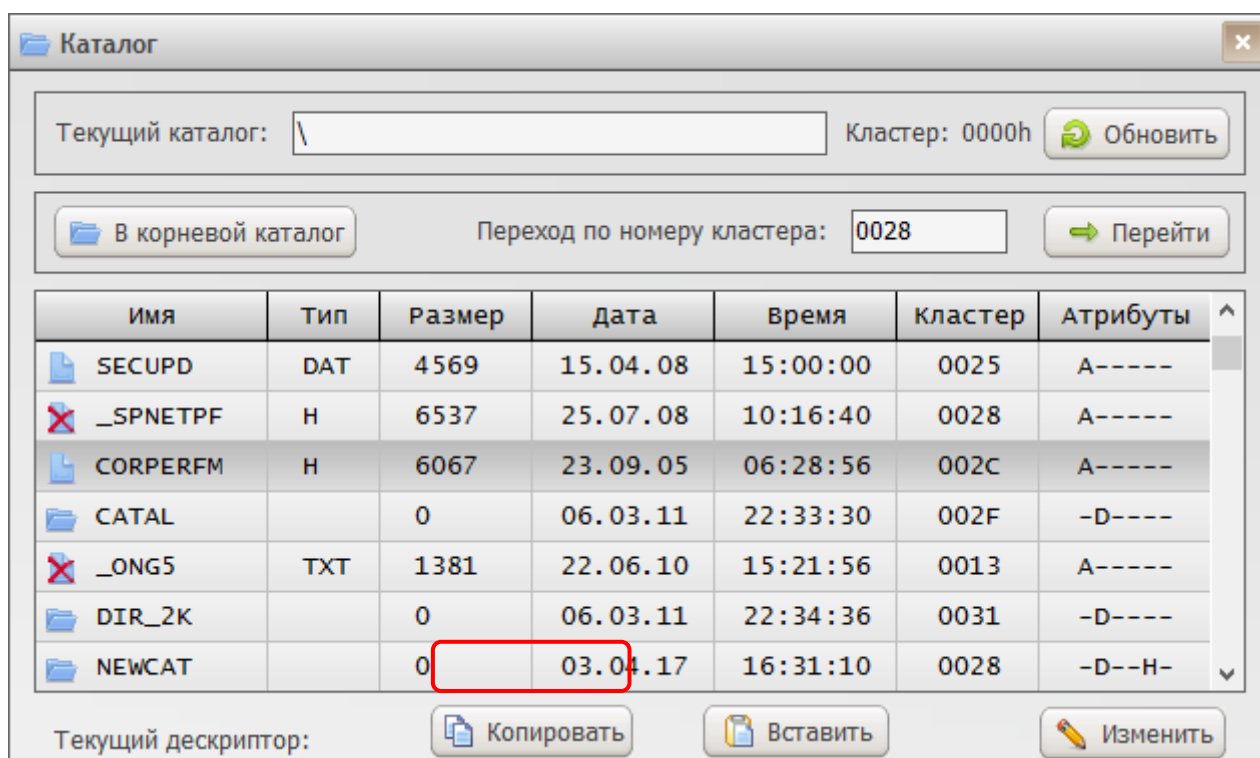


Рис. 2.23. Дескриптор копируемого файла

4. Заходим в «Редактирование дескриптора», и изменяем «Кластер», предварительно выбрав свободный элемент по FAT, при необходимости можно изменить и другие параметры (например, имя) (Рисунок 2.24).

Редактирование дескриптора каталога

Имя: Расширение: Атрибуты (HEX):

Размер: Кластер (HEX):

☐ Пометить как удаленный ☐ Очистить дескриптор

#	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
00	43	4F	52	5F	50	45	52	46	48	20	20	20	00	9A	1D	B4	COR_PERFH . ђ. ђ
10	66	3E	66	3E	00	00	38	86	83	4A	29	00	B3	17	00	00	f>f>..8†ŕŕ).i...

Рис. 2.24. Дескриптор нового файла

5. Считаем по FAT исходного файла (или по размеру файла), сколько кластеров занимает копируемый файл, выделяем столько же, начиная с отведенного в 4 шаге элемента.

6. Запоминаем размер файла, и через окно «Область данных» производим копирование содержимого из кластеров исходного файла, в кластеры нового. Для этого надо сначала занести в список цепочку кластеров исходного файла, ввести его размер в поле «Количество байт» и нажать кнопку «Копировать». Затем надо очистить список и занести туда цепочку кластеров нового файла и нажать кнопку «Вставить» (Рисунок 2.25).

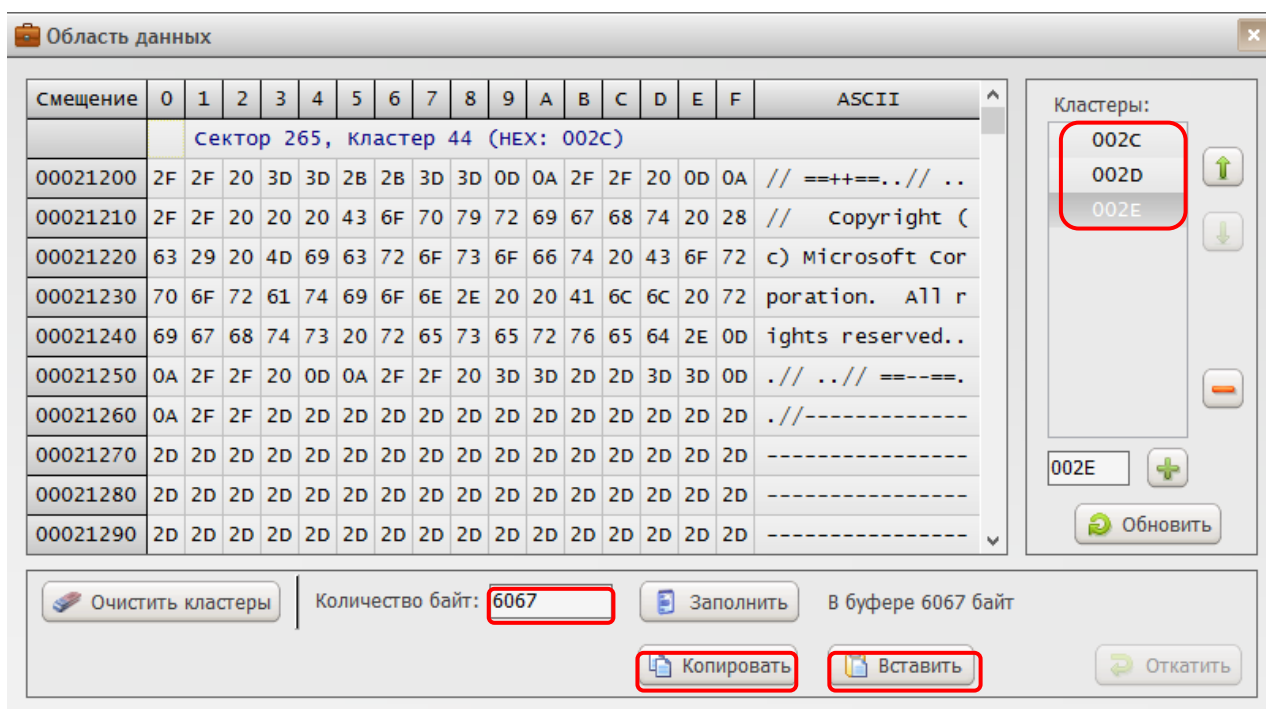


Рис. 2.25. Копирование данных исходного файла

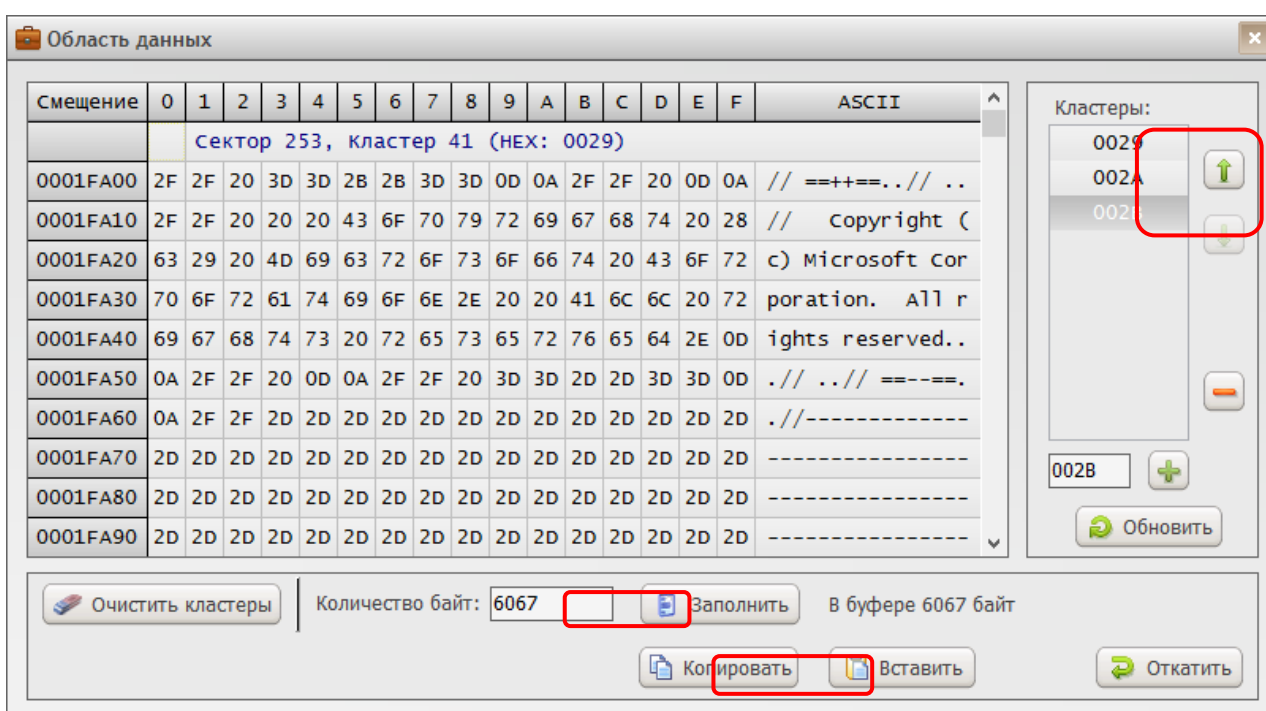


Рис. 2.26. Заполнение данных нового файла

Процедура перемещения файла между двумя каталогами одного логического диска сводится к перемещению соответствующего дескриптора.

Если же файл перемещается на другой логический диск, то он «удаляется» с одного диска и «копируется» на другой.

Алгоритм перемещения файла:

1. Находим в указанном каталоге дескриптор необходимого файла, выбираем его и нажимаем кнопку «Копировать» (Рисунок 2.27).

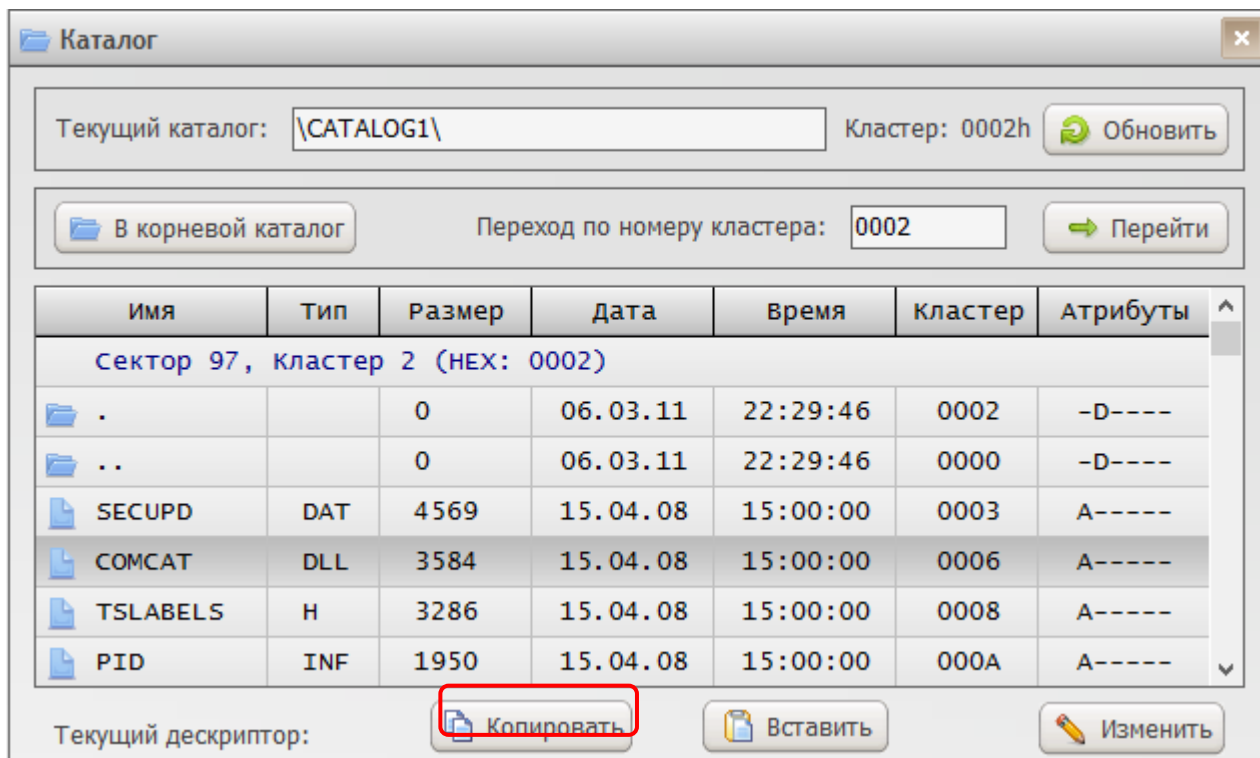


Рис. 2.27. Дескриптор перемещаемого файла

2. Вставляем скопированный дескриптор на место свободного дескриптора в нужном каталоге (Рисунок 2.28).

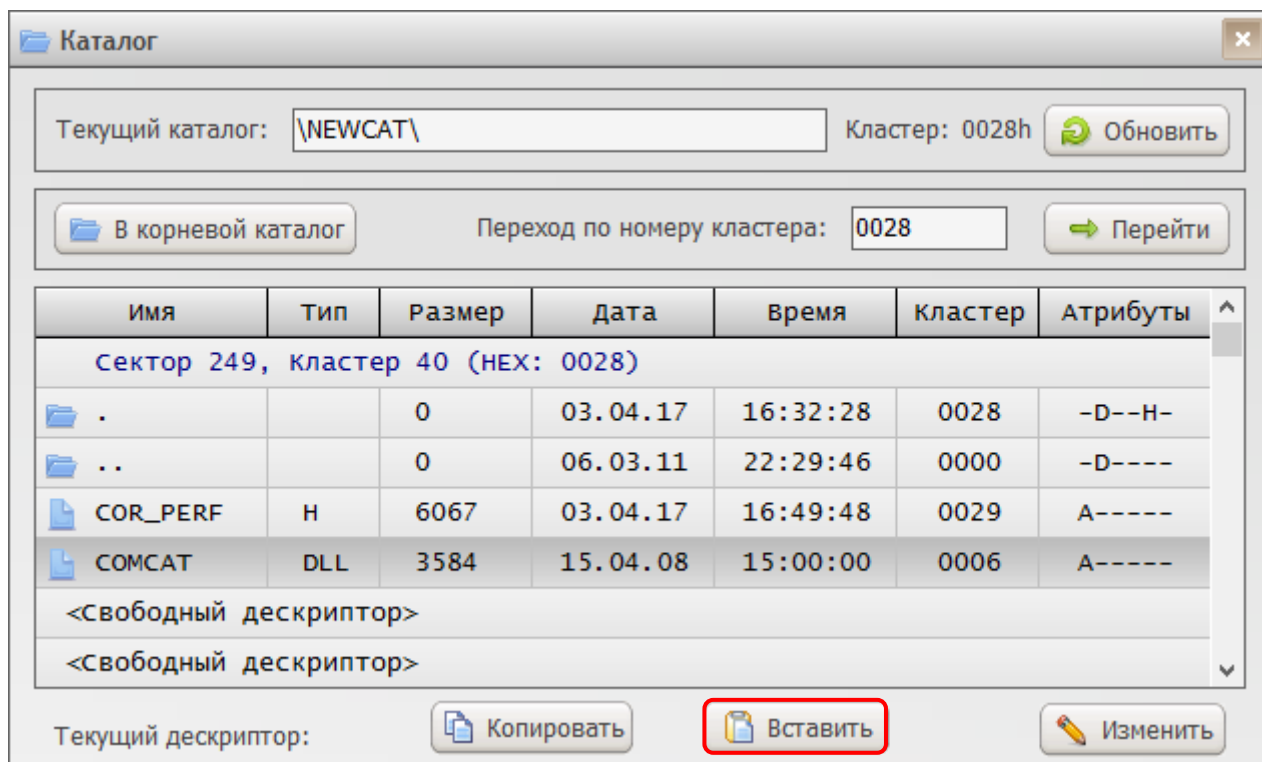


Рис. 2.28. Дескриптор нового файла

3. Удаляем старый дескриптор в исходном каталоге, поставив отметку «Пометить как удаленный» в окне редактирования дескриптора (Рисунок 2.29). При этом первый символ имени файла изменится на символ «_» (E5h) – признак удаленного файла.

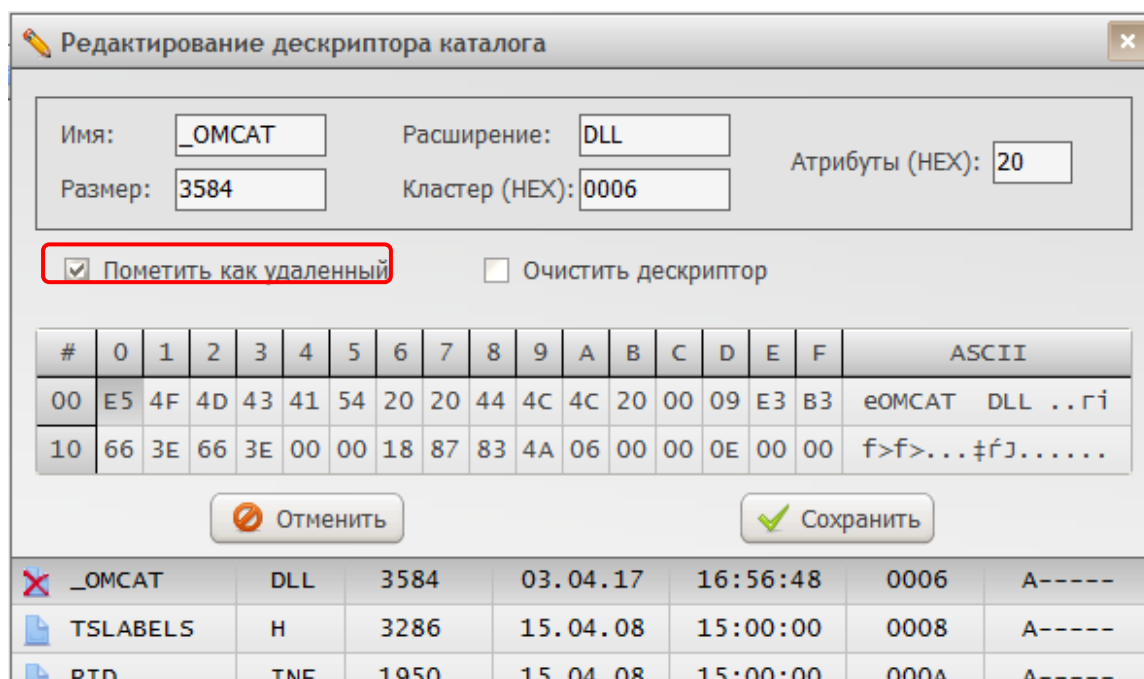


Рис. 2.29. Удаление старого дескриптора

2.3.5. Удаление и восстановление файла

Для удаления файла:

1. Находим необходимый файл в каталоге.
2. В редакторе дескриптора файла помечаем его как удаленный (Рисунок 2.30).

#	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
00	E5	4F	4E	47	39	20	20	20	54	58	54	20	00	0E	12	B4	eONG9 TXT ...г'
10	66	3E	66	3E	00	00	A8	88	06	3D	24	00	48	00	00	00	f>f>..Ё€.=\$.н...

Рис. 2.30. Удаление дескриптора

3. Обнуляем цепочку FAT. Для этого в таблице FAT надо найти элемент, описывающий начальный кластер файла, номер которого указан в его дескрипторе (в данном случае 0024). Если в нем записан символ конца файла (FFFF), очищаем его и на этом удаление файла завершается (Рисунок 2.31). Если элемент содержит номер следующего элемента цепочки, запоминаем этот номер и обнуляем кластер, далее находим следующий элемент и поступаем с ним как с предыдущим. И так пока не встретится элемент, содержащий символ конца файла.

Таблица FAT								
кластер	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
	Сектор 1							
0000			<EOF>	0004	0005	<EOF>	0007	<EOF>
0008	0009	<EOF>	<EOF>	<EOF>	<EOF>	0013	<EOF>	<EOF>
0010	0011	<EOF>	<EOF>	0018	<EOF>	<EOF>	<EOF>	<EOF>
0018	<EOF>	<EOF>	001B	001C	001D	001E	<EOF>	<EOF>
0020	<EOF>	<EOF>	<EOF>	<EOF>	0000	0026	0027	<EOF>
0028	<EOF>	002A	002B	<EOF>	002D	002E	<EOF>	<EOF>
0030	<EOF>	<EOF>	<EOF>	<EOF>	0035	<EOF>	0037	0038
0038	0039	<EOF>	003B	003C	003D	<EOF>	003F	0040
0040	0041	0042	<EOF>	0044	0045	<EOF>	0047	0048

Кластер 0024:

Рис. 2.31. Освобождение элементов цепочки FAT

	#	Задание
	5	Удаление файла
	6	Перемещение файла
	7	Изменение размера файла
	8	Восстановление файла
	9	Создание каталога
	10	Удаление файла
	11	Изменение размера файла

Восстановить файл 'SONG9.TXT', находящийся в подкаталоге 'CATALOG1\SONGS\'

Рис. 2.32. Задание по восстановлению файла

Для восстановления файла необходимо проделать операции обратные удалению файла:

1. Находим удаленный файл, в имени которого вместо первой буквы имени стоит символ «_» (так как он заменяется на E5h), восстанавливаем первую букву имени по заданию (Рисунок 2.33).

Редктирование дескриптора каталога

Имя: SONG9 Расширение: TXT Атрибуты (HEX): 20

Размер: 72 Кластер (HEX): 0024

☐ Пометить как удаленный ☐ Очистить дескриптор

#	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
00	53	4F	4E	47	39	20	20	20	54	58	54	20	00	0E	12	B4	S O N G 9 T X T . . . ' r
10	66	3E	66	3E	00	00	C5	86	83	4A	24	00	48	00	00	00	f > f > . . E f r J \$. H . . .

X ONG9 TXT 72 03.04.17 16:54:10 0024 A-----

Рис. 2.33. Восстановление дескриптора

2. Находим начало цепочки FAT файла (он указан в дескрипторе), и восстанавливаем ее, предварительно подсчитав количество кластеров (Рисунок 2.34).

Таблица FAT								
кластер	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
	Сектор 1							
0000			<EOF>	0004	0005	<EOF>	0007	<EOF>
0008	0009	<EOF>	<EOF>	<EOF>	<EOF>	0013	<EOF>	<EOF>
0010	0011	<EOF>	<EOF>	0018	<EOF>	<EOF>	<EOF>	<EOF>
0018	<EOF>	<EOF>	001B	001C	001D	001E	<EOF>	<EOF>
0020	<EOF>	<EOF>	<EOF>	<EOF>	<EOF>	0026	0027	<EOF>
0028	<EOF>	002A	002B	<EOF>	002D	002E	<EOF>	<EOF>
0030	<EOF>	<EOF>	<EOF>	<EOF>	0035	<EOF>	0037	0038
0038	0039	<EOF>	003B	003C	003D	<EOF>	003F	0040
0040	0041	0042	<EOF>	0044	0045	<EOF>	0047	0048

Кластер 0024:

Рис. 2.34. Восстановление цепочки FAT

2.3.6. Проверка и завершение заданий

Если вы считаете, что выполнили задание, то для его проверки надо нажать кнопку «Проверить» в окне задания (Рисунок 2.35).

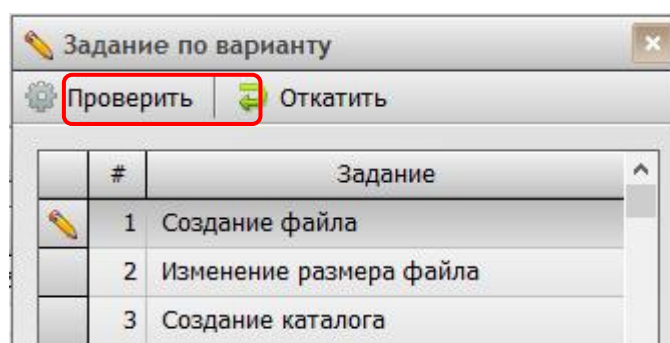


Рис. 2.35. Проверка выполнения задания

Если задание выполнено не до конца, или были допущены ошибки, появится окно со списком неверно заполненных полей или окон (Рисунок 2.36).

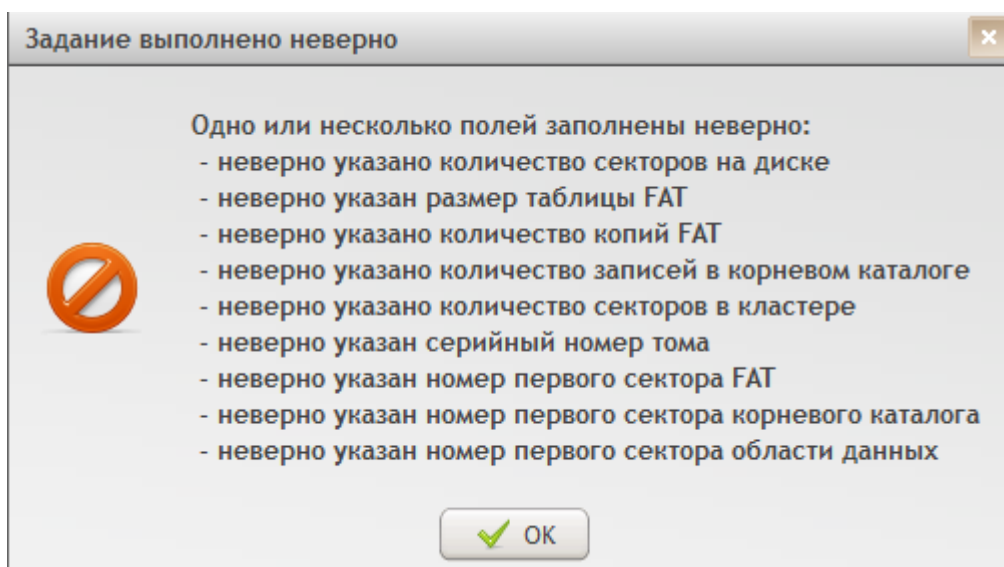


Рис. 2.36. Список ошибок

Если задание выполнено без ошибок, или все ошибки были исправлены, появится окно «Задание выполнено» с количеством допущенных при выполнении задания ошибок (Рисунок 2.37). Количество ошибок запоминается и учитывается при расчёте результата.

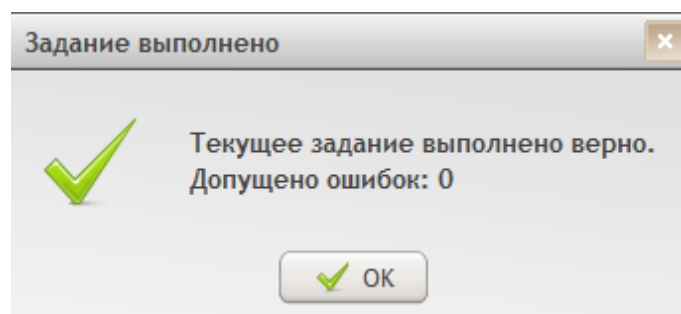


Рис. 2.37. Задание выполнено

Свое продвижение по лабораторной работе можно отслеживать по окну заданий (их количество может отличаться в разных вариантах). После выполнения последнего задания появится окно оповещающее о завершении выполнения лабораторной работы (Рисунок 2.38).

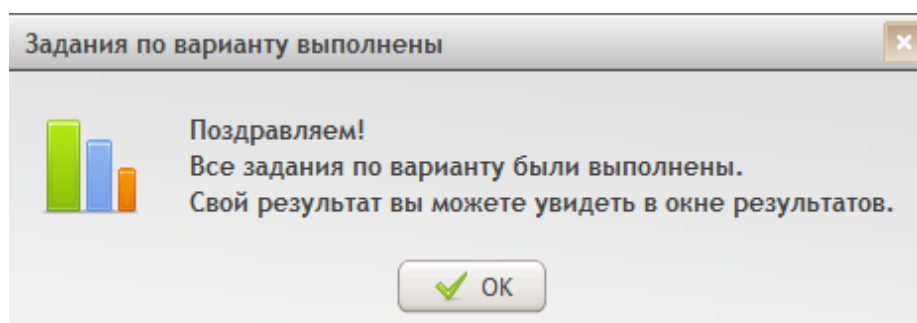


Рис. 2.38. Все задания выполнены

После нажатия кнопки «ОК» появляется окно результатов (Рисунок 2.39).

Результаты				
ignatov a a			Группа: ivt 41	
Начало выполнения: 28.03.2017 / 14:22:03			Вариант: 1	
	Задание	Время	Ошибок	Балл
✓	- Копирование файла	0:12:50	0	100
✓	- Удаление файла	0:05:33	0	100
✓	- Перемещение файла	0:04:22	1	91
✓	- Изменение размера файла	0:01:38	1	91
✓	- Восстановление файла	0:02:47	0	100
✓	- Создание каталога	0:04:27	0	100
✓	- Удаление файла	0:00:46	0	100
✓	- Изменение размера файла	0:00:56	0	100
✓	- Копирование файла	0:10:04	3	74
✓	- Удаление файла	0:00:40	0	100
✓	- Перемещение файла	0:00:42	0	100
✓	- Восстановление файла	0:00:48	0	100
Итоговое время: 4:23:57		Допущено ошибок: 31		
Выполнено заданий: 17		Результирующий балл: 89		

Рис. 2.39. Окно результатов

В окне результатов можно увидеть время выполнения каждого задания, количество допущенных при этом ошибок и соответствующий балл за выполнение задания. В нижней части окна указано итоговое время выполнения

работы, количество выполненных заданий, общее количество ошибок и результирующий балл за выполнение работы.

После завершения выполнения лабораторной работы можно выполнить тест, если он еще не был выполнен или был выполнен unsuccessfully.

Если все задания успешно выполнены, также успешно пройден тест, необходимо показать результат преподавателю и защитить лабораторную работу.

ЗАКЛЮЧЕНИЕ

Для того, чтобы компьютерная система могла нормально работать с информацией, размещенной на жестком диске, винчестер должен иметь четкую логическую структуру. Только так операционная система будет «знать», что и где «лежит». Наиболее удобной для обучения является файловая система FAT16 её легко осмыслить. А зная идею, уже не сложно изучать другие файловые системы – FAT32, NTFS, и т.п.

В результате выполнения лабораторной работы студенты познакомились с работой файловой системы FAT 16 «изнутри» - посмотрели, каким образом файловая подсистема управляет выполнением различных операций с файлами, такими как элементарные операции создания и удаления файлов и каталогов, изменение размеров каталогов, восстановление удаленных файлов, что происходит в файловой системе, когда изменяется размер файла или его содержимое.

Зная принцип организации файловой системы, можно разработать свой драйвер или файловый менеджер на любом вычислительном устройстве.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Лекции по операционным системам [Электронный ресурс]. – Режим доступа: <http://baumanki.net/lectures/10-informatika-i-programmirovanie/334-lekcii-po-operacionnym-sistemam>. – 24.03.2018.
2. Файловые системы и восстановление данных [Электронный ресурс]. – Режим доступа : <http://computermaster.ru/articles/filesystem.html>. – 24.03.2018.
3. Файловые системы NTFS и FAT32 [Электронный ресурс] – Режим доступа : <https://www.windxp.com.ru/sistem2.htm>. – 4.04.2018.
4. Введение во внутреннее устройство Windows. Лекция 11: Файловая система NTFS [Электронный ресурс] // НОУ ИНТУИТ. Национальный открытый университет. – Режим доступа : <https://www.intuit.ru/studies/courses/10471/1078/lecture/16586>. – 15.08.2018.
5. Описание файловой системы FAT16 [Электронный ресурс] // Записки инженера. Доступным языком заметки по IT технологиям. – Режим доступа: <http://s-engineer.ru/opisanie-fajlovoj-sistemy-fat16/>. – 15.08.2018.
6. Что такое FAT-таблица и кластер [Электронный ресурс]. – Режим доступа: http://on-line-teaching.com/IBM-computer/31_FAT_32.html. – 15.08.2018.
7. Тычков, М. Еще раз о FAT 32 [Электронный ресурс] М. Тычков // Whatis.ru. В мире с компьютером. – Режим доступа: <http://whatis.ru/hard/hdd10.shtml>. – 15.08.2018.

Учебное издание

Караваева Ольга Владимировна

СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ. ФАЙЛОВЫЕ СИСТЕМЫ

Учебно-методическое пособие

Авторская редакция

Технический редактор А. Е. Свинина

Подписано в печать 15.08.2018. Печать цифровая. Бумага для офисной техники.
Усл. печ. л. 3,45. Тираж 5 экз. Заказ № 5283.

Федеральное государственное бюджетное образовательное учреждение высшего
образования «Вятский государственный университет».

610000, г. Киров, ул. Московская, 36, тел.: (8332) 74-25-63, <http://vyatsu.ru>

