Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования   
«Алтайский государственный техничеcкий университет им. И. И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчет защищен с оценкой \_\_\_\_\_

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Отчет

по лабораторной работе № 9

по дисциплине «Операционные системы»

Студент гр. ПИ-82

Иванков Д.В.

Преподаватель доцент, к. т. н.

Боровцов Е. Г.

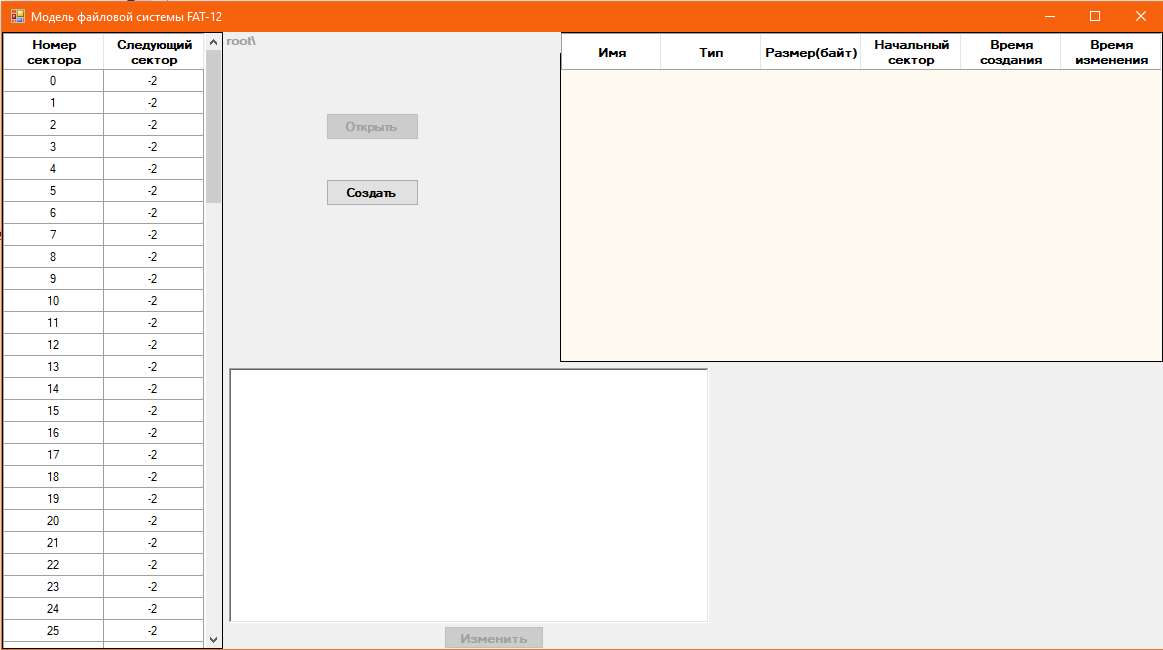
Барнаул 2020

Для моделирования я выбрал файловую систему FAT12, потому что её достаточно легко реализовать.

Программировал я на языке с++ в среде разработки Visual Studio 2013 в интерфейсе программирования windows forms.

**Работа программы**

При запуске программы запускается главное (и единственное) окно:



Далее файлы и каталоги я буду называть просто файлами. Если потребуется, буду уточнять.

Сначала опишу что где расположено:

Слева представлена FAT таблица, у которой в столбце “номер сектора” указан **номер логического сектора,** а в столбце “следующий сектор” указывается следующий сектор в кластере, который занимает какой-либо файл. Если значение во втором столбце равно “-2” – это значит, что сектор свободен. Если значение во втором столбце равно “-1” – это значит, что этот сектор последний в кластере, занимаемым файлом. Остальные значения – указатель на следующий сектор в кластере.



В данном примере сектора 0 – 3 образуют кластер, занимаемый каким-то файлом, причём 3 сектор является последним для кластера. 4 сектор не занят данными.

В FAT12 размер одного сектора составляет 512 байт. Я же в своей программе сократил размер до 128 байт, чтобы не работать с большим количеством данных. Мне кажется, что так даже более наглядно при открытии файла (об этом позже).

По середине окна расположены две кнопки: “Открыть” и “Создать”. Сначала доступна только кнопка “Создать”. Кнопка “Открыть” станет доступна при выборе созданных файлов (чуть позже об этом).

Сверху над кнопками указывается путь, в котором находится пользователь на данный момент. Изначально это корневой каталог.

Корневой каталог расположен в файловой системе после FAT таблиц (одна дублируется, чтобы повысить надёжность файловой системы). В настоящей системе FAT12 она занимает 14 секторов на дискете, однако я в модели выделил всего 2 сектора.

Справа расположена таблица директории. В ней будут располагаться все файлы, находящиеся в текущей директории, а также информация о них в упорядоченном виде: имя, тип, размер в байтах, начальный сектор в таблице FAT, время создания и время изменения.

**Имя файла** преобразуется в соответствии с требованиями файловой системы FAT12:

* Имя должно состоять из 8 символов (+ 3 символа под расширение. Имя и расширение отделяются точкой). В своей программе я автоматически прописываю всем файлам (не каталогам) расширение TXT
* Если у файла имя занимает меньше 8ми символов, то оставшиеся заполняются пробелами (в моей модели я вместо пробела использую нижнее подчёркивание для наглядности)
* Все имена и расширения хранятся в верхнем регистре
* В имени не могут быть использованы следующие символы: **. " / \ [ ] : ; | = ,**

Два файла с одинаковым именем не могут располагаться в одной директории.

У каталогов могут быть расширения, но я в модели не стал добавлять такую возможность.

**Тип файла** указывает, является ли файл каталогом или нет. В программе данный параметр хранится в логическом виде (0 – файл, 1 - каталог), а в таблицу я вывожу слова file или catalog.

С **размером файла**, думаю, всё понятно. По размеру я определяю количество секторов, занимаемое файлом. Файл занимает сектор полностью, даже если он весит всего 1 байт. Однако если файл (не каталог) пустой (его размер равен 0 байт), то он не занимает места в FAT таблице. В FAT таблице, в самом её начале, выделено 2 неиспользуемых сектора. Как я понял, пустые файлы ссылаются на один из этих секторов, и не занимают места в памяти. В моей модели в этом случае в столбец “начальный сектор” записывается значение “-1”.

Однако каталоги по сути своей тоже весят 0 байт, однако всё равно занимают сектора. Так как каталоги хранят в себе другие файлы, то они должны где-то хранить данные о них точно так же, как это делает корневой каталог. В FAT12 запись о файле хранится в 32 байтах (сюда включается информация о файлах и ссылки на них). Учитывая то, что я сократил размер сектора до 128 байт, можно посчитать количество файлов, записи о которых хранятся в одном секторе, занимаемом каталогом: 128 / 32 = 4. (Про добавление и возможные ошибки при этом действии позже).

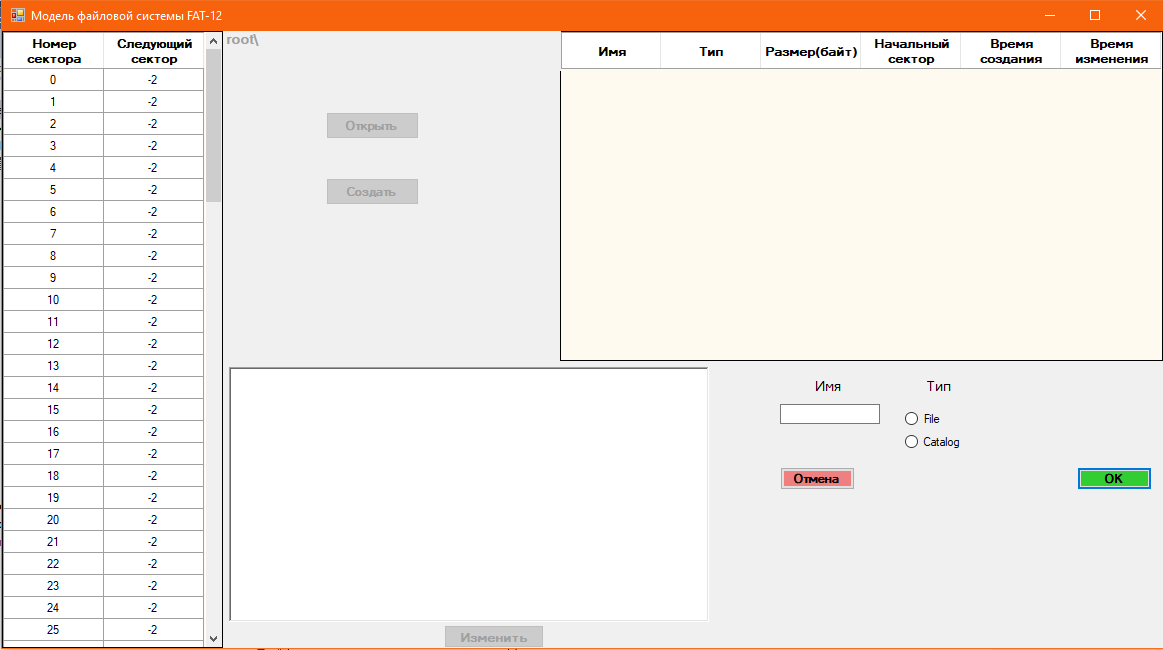
**Начальный сектор** указывает на начало кластера, занимаемого файлом.

С **временем создания и изменения** всё понятно. Время выставляется по системному времени компьютера. При создании файла время создания и время изменения равны.

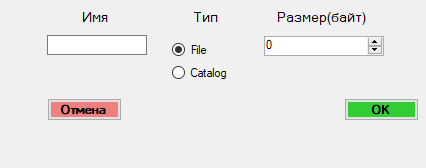
Снизу по середине расположено окошко, в которое будет выводиться содержимое текстовых файлов (в моей модели все файлы (не каталоги) текстовые).

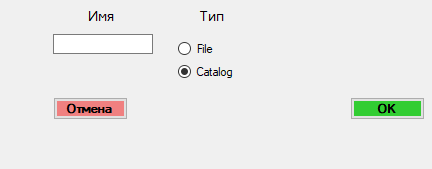
Кнопка “Изменить” недоступна. Она становится доступна, когда пользователь пытается изменить содержимое файла (изменяет текст в окошке, про которое я рассказал чуть выше).

Если пользователь нажмёт на кнопку “Создать”, то появится меню, в котором его попросят ввести название файла, выбрать тип и ввести размер, если пользователь выбрал тип “file” (кнопка “Создать” и таблица директорий станут недоступными). А также появятся кнопки “Отмена” и “ОК”:



На скриншоте видно, когда меню открывается, все поля пустые и не выбран тип файла.





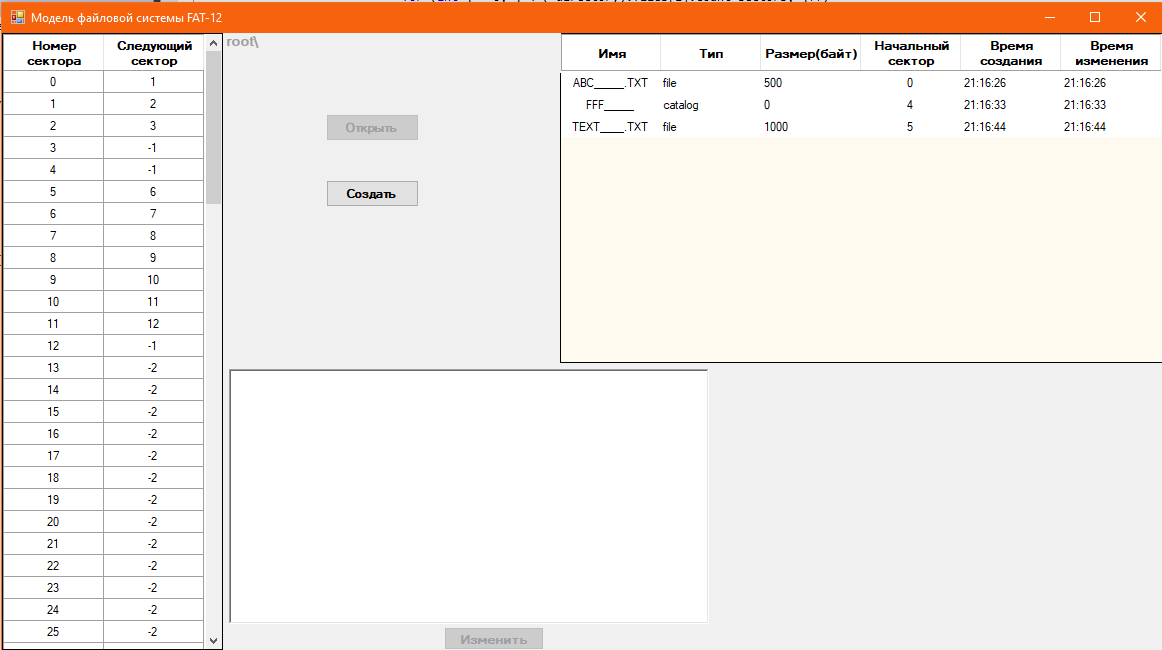
При нажатии на кнопку “Отмена”, это меню пропадает, а кнопка “Создать” и таблица директорий снова становятся активными.

При нажатии на кнопку “ОК” происходит добавление файла с указанными данными, а также проверяется ряд условий, выявляющих ошибки:

* Имя должно состоять хотя бы из одного символа. Этим символом может быть пробел, однако, как я выше указывал, в моей модели происходит преобразование к имени, которое бы хранилось в настоящем FAT12 без расширения длинных имён. Следовательно, если у файла имя занимает меньше 8ми символов, то оставшиеся заполняются пробелами, а пробелы представлены в виде нижнего подчёркивания. В итоге имя файла при введённом пробеле будет состоять из 8 пробелов, то есть введённый пользователем пробел превратится в нижнее подчёркивание. Я пишу об этом, потому что может возникнуть путаница с этими двумя символами. Лучше не использовать символ \_ в имени файла!!!
* Длина имени не должна превышать 8 символов
* Имя не должно содержать недопустимые символы
* Имена в одной директории не должны повторяться
* Не выбран тип файла (как, например, на первом слайде)
* В корневом каталоге не может храниться больше 8 файлов (это ограничение связано с выделением мною для корневого каталога 2 секторов)
* Нехватка памяти для добавляемого файла
* Нехватка места для расширения каталога (возникает, когда в каталог добавляется файл, для записи которого на диск места хватает, а места, для записи о нём (32 байта) в каталоге (не корневом) места не хватает. Следовательно, каталог пытается расшириться. Если для этого не хватает места, то выдаётся данная ошибка)

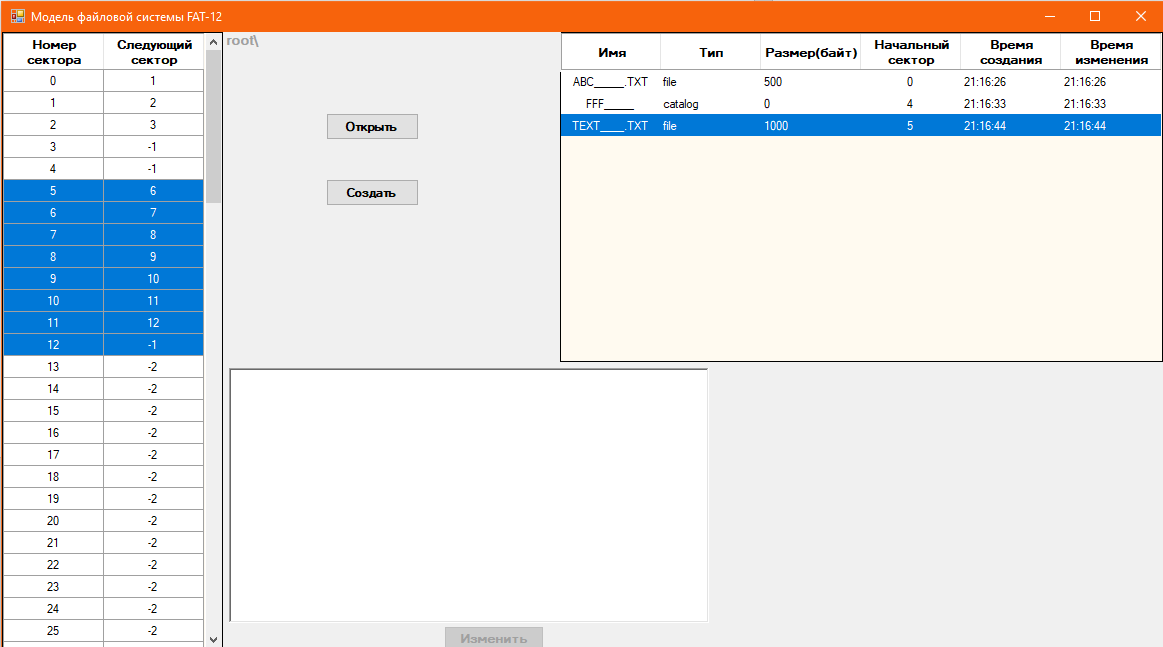
Все эти ошибки отменяют добавление файла.

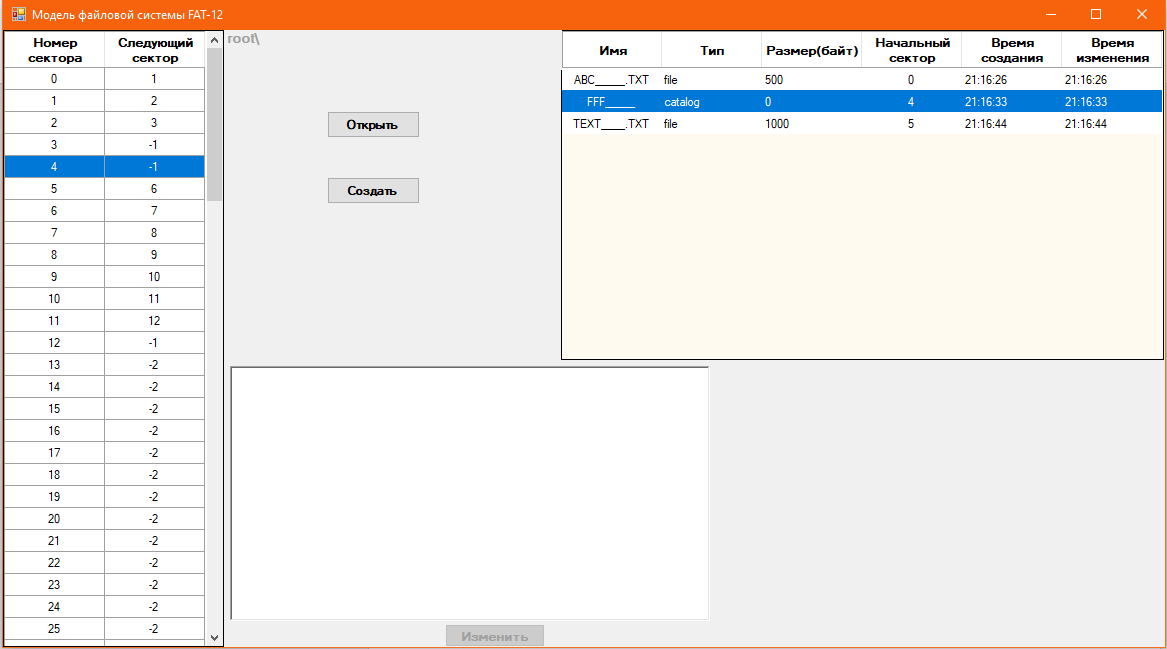
Пока пользователю доступно меню для добавления файла, ему недоступно ничто другое. Чтобы выйти нужно нажать кнопку “Отмена”.



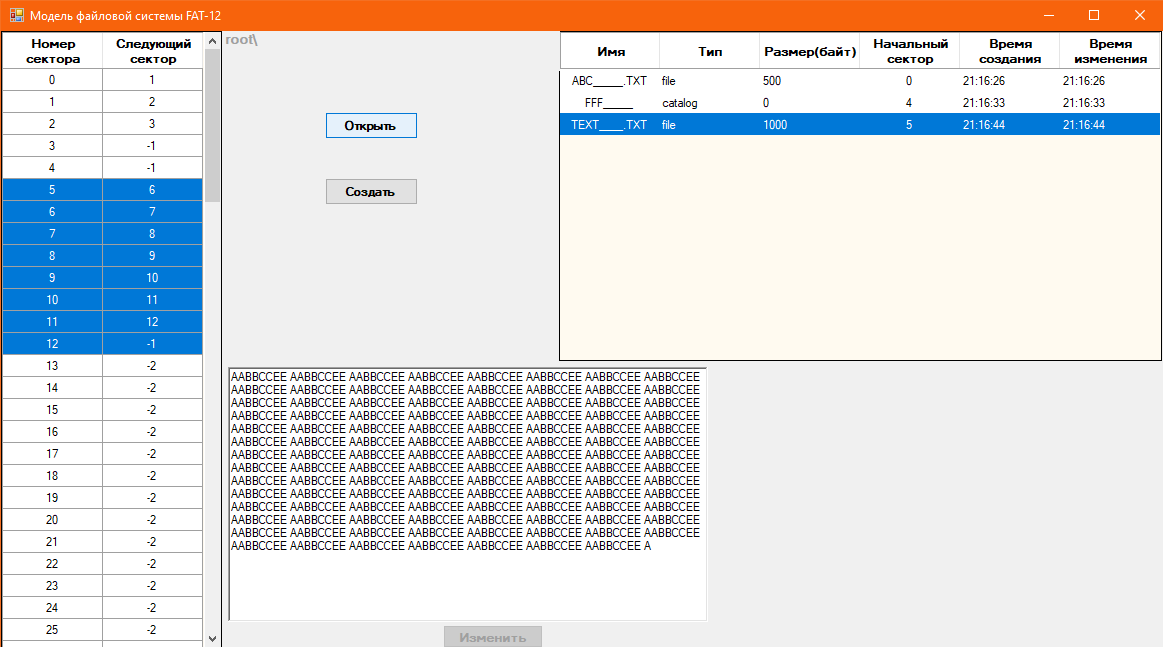
Мы создали два файла, разных размеров, и каталог.

При нажатии на любую запись, в FATтаблице отобразится кластер, занимаемый файлом, и станет активной кнопка “Открыть”:



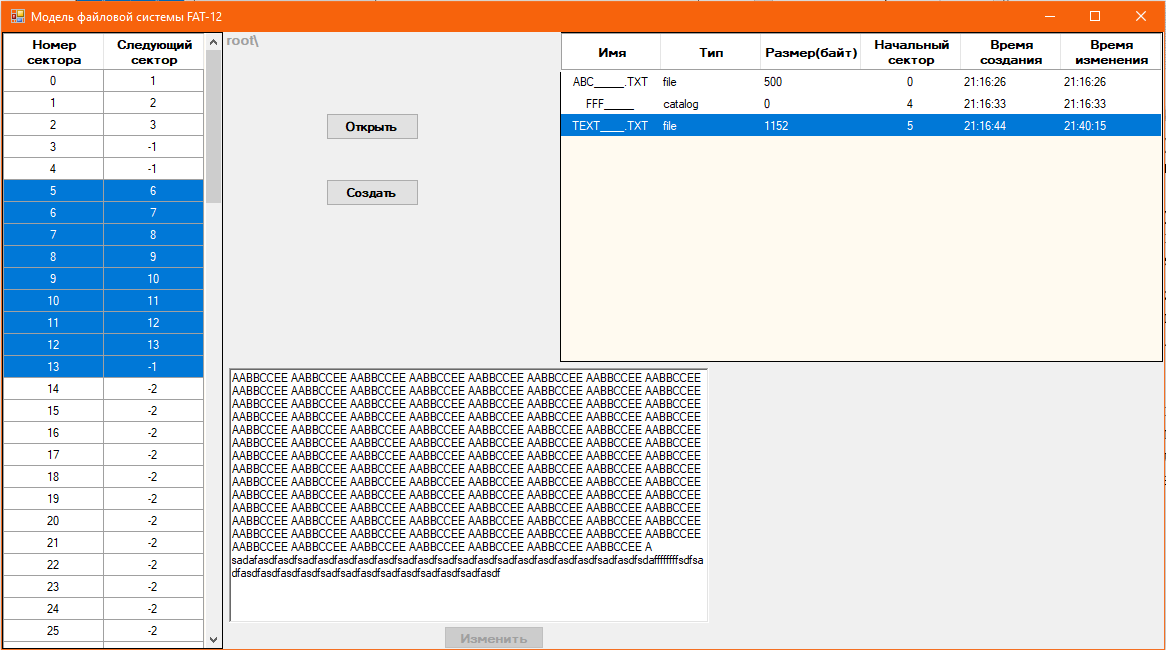


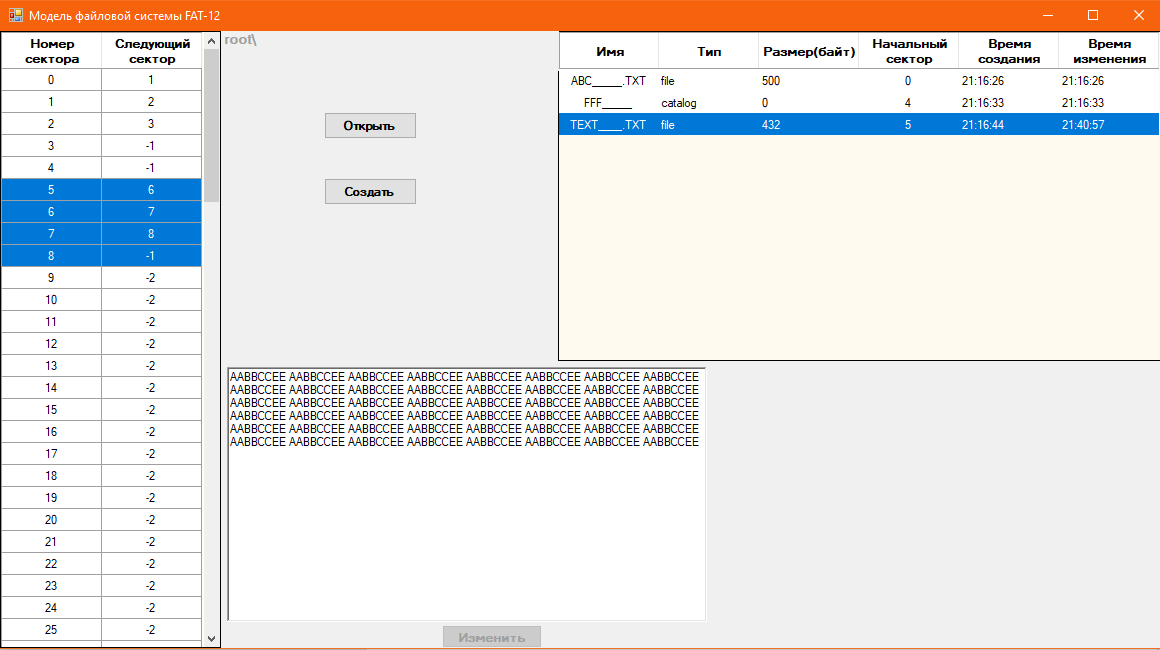
Чтобы открыть файл, нужно при выбранном файле нажать кнопку “Открыть”. Если открывать текстовый файл, то его содержимое выпишется в окошечко снизу по середине (я заполняю файлы одинаковыми записями, пока не достигну их размера).

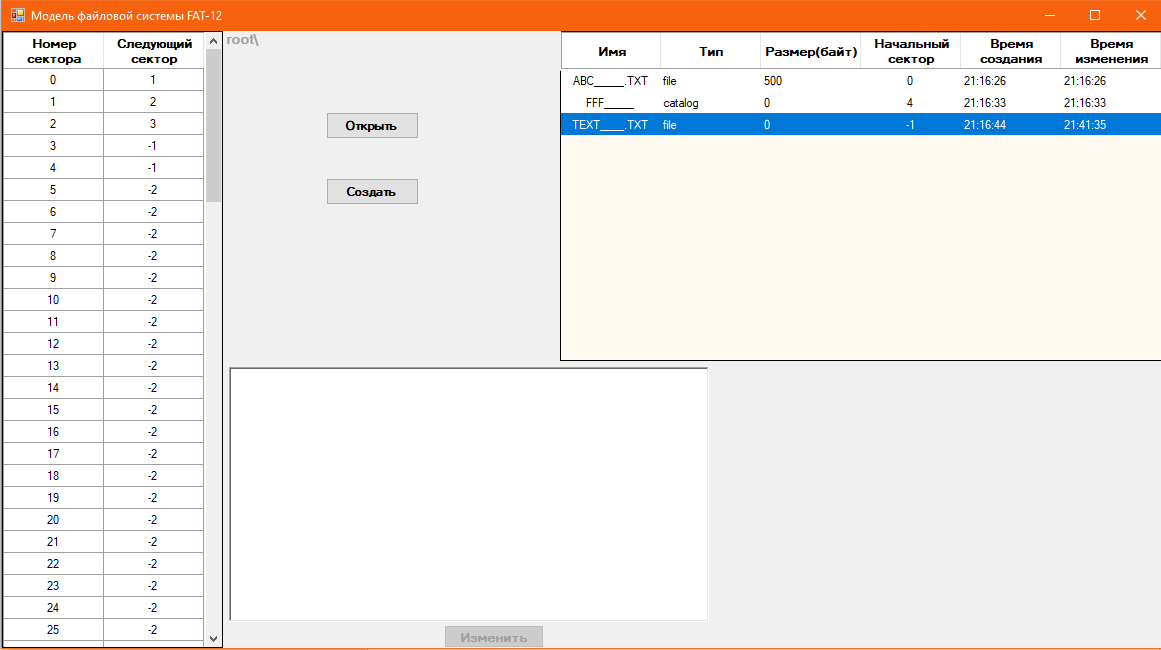


Если изменить содержимое файла (удалить символы в окошечке или дописать новые), то кнопка “Изменить” станет активной. Если после открытия файла переключиться на другой (выбрать другой файл в таблице директории), то окошечко станет пустым, а кнопка “Изменить станет недоступной”. Если при этом были внесены какие-то изменения, но не нажата кнопка “Изменить”, то содержимое файла не меняется.

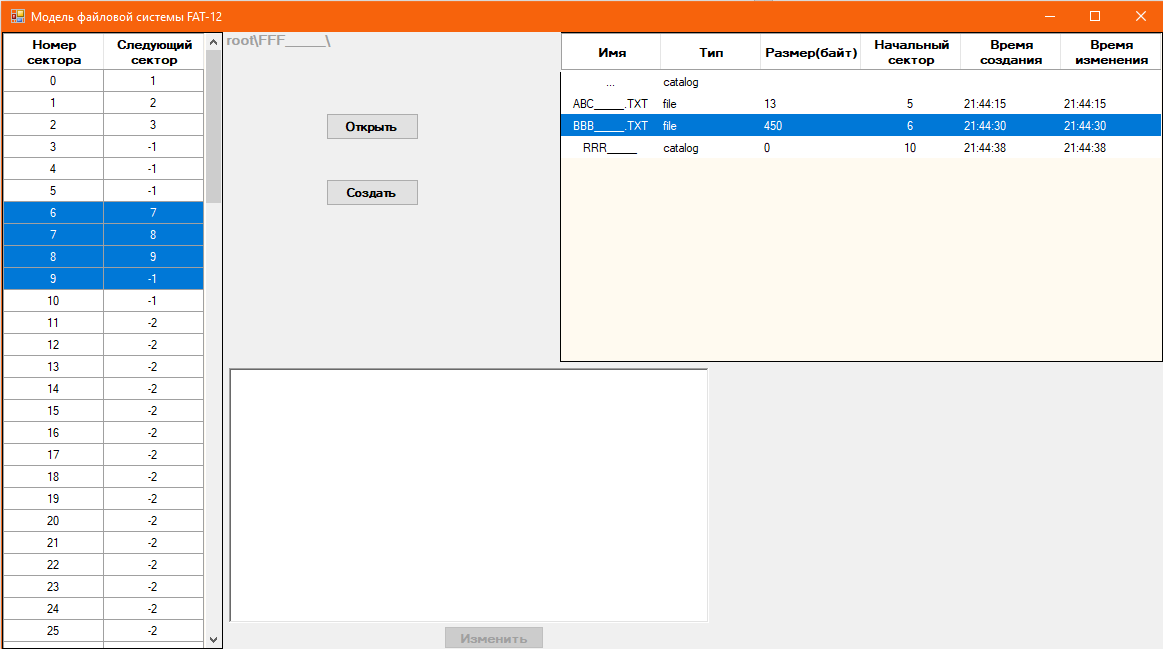
При нажатии на кнопку изменить, содержимое файла, время изменения и размер файла изменяются. При этом, если требуется ещё сектора, для хранения новой информации в файле, то они добавляется к кластеру, занимаемому файлом. И наоборот, если файлу теперь хватает меньшего количества секторов, то ненужные сектора освобождаются. Если же из файла удаляются все данные, то меняется и значение начального сектора (как я описывал выше, пустые файлы ссылаются на неиспользуемые файловой системой сектора вначале FAT таблицы. Я в своей модели в этом случае в столбец “начальный сектор” записываю значение “-1”).

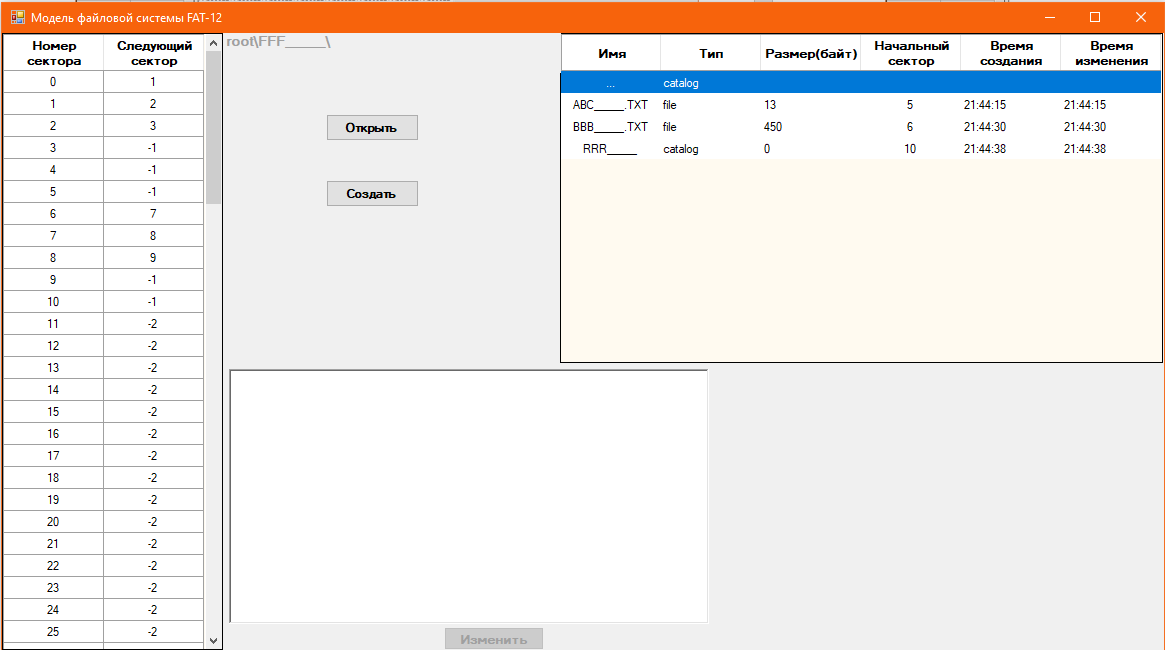




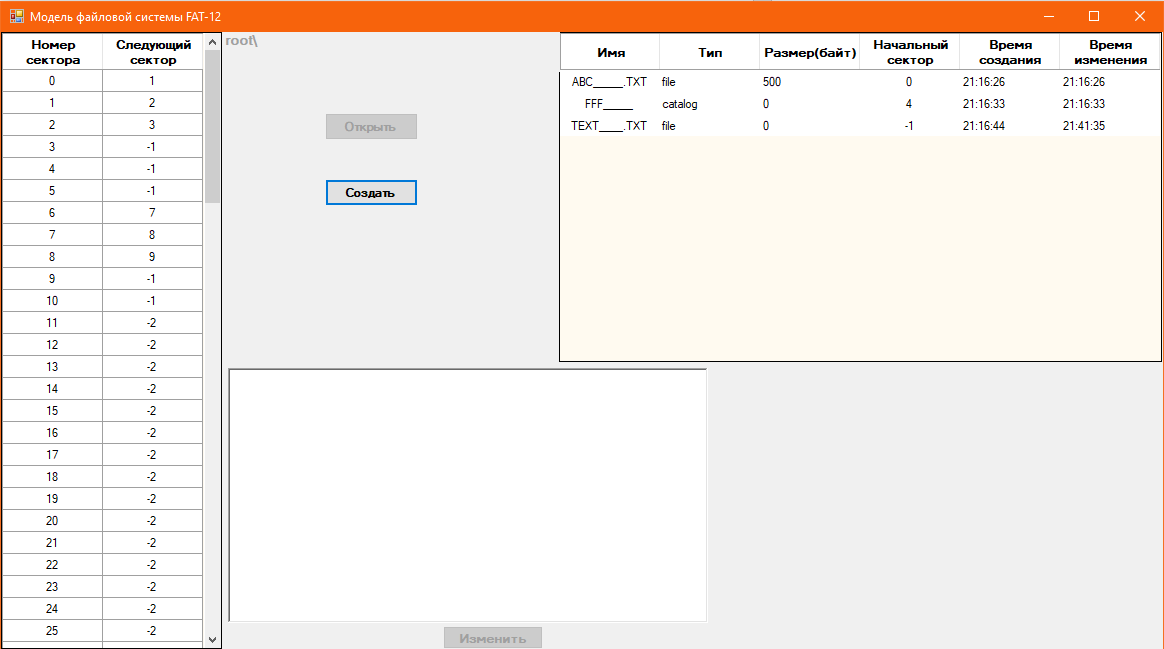


Если же открывать не файл, а каталог, то пользователь переходит в эту директорию (в путь дописывается название каталога, в который перешёл пользователь). В ней от так же может добавлять файлы, каталоги, и делать всё, что описывалось ранее. Для того, чтобы вернуться обратно, я сделал специальную строчку в таблице директории с названием “…”. Таким приёмом пользуются многие файловые менеджеры. Естественно эта строка является искусственной и не занимает места в FAT таблице:





Если открыть эту строку, то мы вернёмся в предыдущую директорию (путь поменяется).



В корневом каталоге этой строки естественно нет.

Забыл упомянуть, что файл может располагаться в памяти в разбросанном порядке, не обязательно хранить файл в памяти в последовательно идущих секторах: 