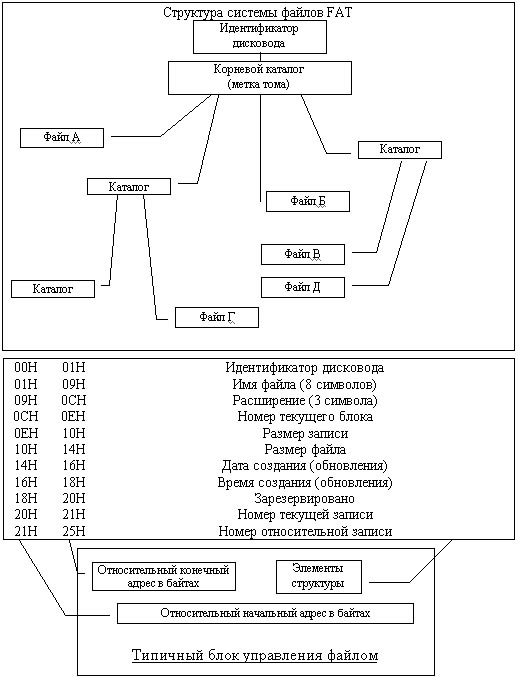
# Модель файловой системы FAT

ОДЕССКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ ХОЛОДА  
Институт информационных технологий  
кафедра информационных систем  
  
  
кУРСОВАЯ РАБОТА ПО ДИСЦИПЛИНЕ  
“тЕОРИЯ ОПЕРАЦИОННЫХ СИСТЕМ”  
  
“модель файловой системы FAT”  
  
  
Выполнил студент 322Б группы Холод Денис.  
Руководитель Чмырь И.А.  
  
  
  
  
  
  
ОДЕССА  
1997  
Содержание:

|  |  |
| --- | --- |
| Цель курсовой работы | 3 |
| Идея программы | 3 |
| О системе FAT | 4 |
| Структура системы файлов FAT | 5,6 |
| О реализации задания | 7 |
| О примененных уровнях абстракции | 8 |
| Об интерфейсе | 9 |
| Исходный текст: Макросы, вызываемые событиями | 10 |
| Исходный текст: Базовые события | 13 |
| Исходный текст: Макросы нижнего уровня абстракции | 15 |
| Список литературы, примененной при подготовке курсовой работы | 17 |

Цель:  
        Разработка программы, моделирующей организацию файловой системы FAT.  
Идея программы:  
        Программа позвовяет экспериментально исследовать логическую структуру файловой системы FAT, состоящую из трех областей: FAT (File Allocation Table), корневой каталог и область файлов. Имеется панель описания операций с файлами, находящимися на диске, а именно: добавить файл, удалить файл, перезаписать файл (с изменением размера). Кроме того, предоставлена возможность графического отображения соответствия файлов секторам на диске.  
  
Файловая система FAT:  
        Так называемая файловая система FAT использовалась во всех версиях МСДОС и в первых двух выпусках OS/2 (версии 1.0 и 1.1). Каждый логический том имеет собственный FAT, который выполняет две функции: содержит информацию распределения для каждого файла в томе в форме списка связей модулей распределения (кластеров) и указывает, какие модули распределения свободны.   
        Когда FAT был изобретен, это было превосходное решение для управления дисковым пространством, главным образом потому что гибкие диски, на которых он использовался, редко были размером более, чем несколько Mb. FAT был достаточно мал, чтобы находиться в памяти постоянно, позволял обеспечивать очень быстрый произвольный доступ к любой части любого файла. Когда FAT был применен на жестких дисках, он стал слишком большим для резидентного нахождения в памяти и ухудшал производительность системы. Кроме того, так как информация относительно свободного дискового пространства рассредотачивалась "поперек" большого количества секторов FAT, он был непрактичен при распределении файлового пространства, и фрагментация файлов оказалась препятствием высокой эффективности.   
        Кроме того, использование относительно больших кластеров на жестких дисках привело к большому количеству неиспользуемых участков, так как в среднем для каждого файла половина кластера была потрачена впустую.  
        Ограничения FAT на наименование файлов и каталогов унаследованы из CP/M. Когда Paterson создавал 86DOS, одной из его первых целей было облегчить импорт из CP/M в его новую операционную систему. А следовательно принятые в CP/M ограничения на имена файлов и расширений перенеслись в 86DOS.  
        В течение нескольких лет Microsoft и IBM сделали попытку продлить жизнь файловой системы FAT благодаря снятию ограничений на размеры тома, улучшению cтратегий распределения, кэширования имен пути, и перемещению таблиц и буферов в расширенную память. Но они могут расцениваться только как временные меры, потому что файловая система просто не подходила к большим устройствам произвольного доступа.   
  
  
  
О реализации задания  
        Модель файловой системы представляет собой рабочий лист Microsoft Excel, на котором в виде ячеек представлены три логические структуры файловой системы:  
  
Каталог файлов  
Таблица распределения  
Область файлов  
  
        На рабочем листе также расположены две панели управления:  
  
Операции с файлами  
добавить  
удалить  
перезаписать  
Визуализация FAT  
показать файл  
убрать стрелки  
  
        Исходя из контекста вопроса параллельное выполнение процессов или потоков (MultiThreading) реализовывать нет необходимости. Учитывая эти факторы, а также необходимость разработки “дружественного” интерейса с применением графических примитивов, было решено использовать язык программирования VisualBasic (VBA) от фирмы Microsoft и электронную таблицу Excel 5.0 фирмы Microsoft.  
        Так как модель является типичной управляемой событиями системой, то алгоритм ее работы достаточно прост, а точнее представляет собой ряд алгоритмов, которые пересекаются только в среднем уровне абстракции. (см. ниже) События вызываются пользователем посредством воздействия на органы управления.   
  
О примененных уровнях абстракции  
        Все возможные события были тщательно проанализированы автором и разделены на три категории, или на три уровня абстракции:   
События, вызываемые внешними воздействиями. Таковыми являются все события, вызываемые пользователем посредством вызова пунктов из групп операций с файлами.  
События фундаментальной природы, т.е. базовые события, порождаемые предудущими. Их значительно меньше, и они никак не вникают в суть породивших из внешних событий.  
События низкого уровня абстракции, которые являются искуственно синтезированными из предыдущих двух групп событиями и имеют целью понизить сложность системы за счет абстрагирования от непринципиальных тонкостей и усложнений в системе, диктуемых контекстом. Таковыми, например, являются события обработки неправильных ситуаций.  
Об интерфейсе  
        Интерфейс является интуитивно понятным и не навязчивым, всегда, насколько это возможно, является “демократичным” по отношению к пользователю. В программах, написанных таким стилем, очень четко прослеживается причинно-следственная связь, хорошо знакомая любому человеку и являющаяся неотъемлемой частью образного мышления. Это делат программу весьма простой в использовании, а алгоритм работы простым в понимании. Вся необходимая для работы информация представлена в диалогах, о неправильных ситуциях программа сообщает пользователю и это единственный (и законный) случай нарушения свободы действий пользователя.  
  
        Далее приводится исходный текст программы с обширными комментариями, в которых делается упор не на особенности языка программирования и конкретной реализации, а на представление составляющих файловой системы как объектов, а процедур и функций как методов, применяемых к этим объектам.  
  
        Макросы, вызываемые событиями  
Public Type FileID 'Тип, описывающий файл: имя,размер и точка входа в FAT  
Name As String  
Size As Integer  
First As Integer  
End Type  
  
Sub PressAddFile() 'Макрос, вызываемый кнопкой "Добавить Файл"  
DialogSheets("Add").EditBoxes("Name").Text = "" 'подготовка диалога "Добавление файла"  
DialogSheets("Add").EditBoxes("Size").Text = "" 'очистка полей ввода  
Sheets("Add").Show 'Вызов диалога "Добавление файла"  
With DialogSheets("Add") 'Проверка на правильность введенных данных  
If (.EditBoxes("Name").Text = "") Or (.EditBoxes("Size").Text = "") Or (.EditBoxes("Size").Text = "0") Then Exit Sub  
End With  
Dim NewFile As FileID 'Экземпляр переменной с описанием создаваемого файла  
With DialogSheets("Add")  
NewFile.Name = .EditBoxes("Name").Text  
NewFile.Size = .EditBoxes("Size").Text  
End With  
Call AddFile(NewFile) 'Вызов процедуры добавления файла  
Refresh 'обновили графическое изображение размещения файлов  
End Sub  
  
Sub PressDeleteFile() 'в основном рабочем листе нажата кнопка Удалить Файл  
temp = 4  
With DialogSheets("Delete") 'подготавливаем к работе диалог Delete  
.ListBoxes("Name").RemoveAllItems  
While Sheets("Sheet").Cells(temp; 2) <> "" 'заполняем список имеющимися в каталоге файлами  
.ListBoxes("Name").AddItem Text:=Worksheets("Sheet").Cells(temp; 2).Value; Index:=temp - 3  
temp = temp + 1  
Wend  
.Show 'показываем диалог Delete  
If .ListBoxes("Name") = 0 Then Exit Sub  
Dim File As FileID 'переменная для идентификатора удаляемого файла.Передается процедуре DeleteFile  
File.Name = Sheets("Sheet").Cells(.ListBoxes("Name") + 3; 2)  
File.Size = Sheets("Sheet").Cells(.ListBoxes("Name") + 3; 3)  
File.First = Sheets("Sheet").Cells(.ListBoxes("Name") + 3; 4)  
Call DeleteFile(File) 'Вызываем процедуру удаления файла  
Refresh 'обновили графическое изображение размещения файлов  
End With  
End Sub  
  
Sub PressRemakeFile() 'нажата кнопка Изменить\_размеры\_файла  
'по этому поводу подготавливаем к работе диалог Remake  
temp = 4  
With DialogSheets("Remake")  
.ListBoxes("Name").RemoveAllItems  
.EditBoxes("Size").Text = ""  
While Sheets("Sheet").Cells(temp; 2) <> "" 'заполняем список имеющимися в каталоге файлами  
.ListBoxes("Name").AddItem Text:=Worksheets("Sheet").Cells(temp; 2).Value; Index:=temp - 3  
temp = temp + 1  
Wend  
.Show 'показываем диалог Remake  
'при нажатии кнопки OK в диалоге запустится макрос DialogRemakePressOK  
End With  
End Sub  
  
Sub DialogRemakePressName() 'в диалоге Перезапись выбран файл из списка  
With DialogSheets("Remake") 'и поэтому в диалоге Перезапись обновляем поле размера файла в соотв. с выбранным файлом  
.EditBoxes("Size").Text = Sheets("Sheet").Cells(3 + .ListBoxes("Name").ListIndex; 3).Value  
End With  
End Sub  
  
Sub DialogRemakePressOK() 'в диалоге изменение размеров файла нажата кнопка OK  
With DialogSheets("Remake")  
.Hide 'убрали с экрана диалог  
If .ListBoxes("Name").ListIndex = 0 Then Exit Sub  
Dim File As FileID 'Экземпляр типа FileID - идентификатор файла  
File.Name = Sheets("Sheet").Cells(3 + .ListBoxes("Name").ListIndex; 2).Text  
File.Size = Sheets("Sheet").Cells(3 + .ListBoxes("Name").ListIndex; 3).Value  
File.First = Sheets("Sheet").Cells(3 + .ListBoxes("Name").ListIndex; 4).Value  
'проверка на наличие изменений  
If .EditBoxes("Size").Text = File.Size Or .EditBoxes("Size").Text = "0" Then Exit Sub  
'проверка на помещаемость  
If .EditBoxes("Size").Text > (FreeSize + ((File.Size - 1) 8 + 1) \* 8) Then  
temp = MsgBox("Файл " & File.Name & " размером " & .EditBoxes("Size").Text & " не может быть размещен"; vbExclamation; "Перезапись файла")  
Exit Sub  
End If 'перезапись: удаление, а затем запись файла, но с другим размером  
Call DeleteFile(File) 'удалили  
File.Size = .EditBoxes("Size").Text 'изменили размер  
Call AddFile(File) 'записали  
Refresh 'обновили таблицу распределения файлов  
End With  
End Sub  
  
Sub Visualisation() 'визуализация файла  
temp = 4  
With DialogSheets("Visualisation") 'подготавливаем к работе диалог Visualisation  
.ListBoxes("Name").RemoveAllItems  
While Sheets("Sheet").Cells(temp; 2) <> "" 'заполняем список имеющимися в каталоге файлами  
.ListBoxes("Name").AddItem Text:=Worksheets("Sheet").Cells(temp; 2).Value; Index:=temp - 3  
temp = temp + 1  
Wend  
.Show 'показываем диалог Visualisation  
If .ListBoxes("Name") = 0 Then Exit Sub 'выбрали ли что-нибудь ?  
Dim NumberFile As Integer  
NumberFile = .ListBoxes("Name").ListIndex 'номер по каталогу визуализируемого файла  
Sheets("Sheet").Cells(NumberFile + 3; 2).ShowDependents 'показываем линии, проходящие от ячейки каталога с именем визуализируемого файла до всех занятых им ячеек области файлов  
End With  
End Sub  
Базовые события  
Const ColorOfPaper = 33 'цвет фона области файлов  
Const ColorUsedPartOfFAT = 2 'цвет занятой части области файлов  
  
Sub AddFile(NewFile As FileID) 'Процедурa добавления файла  
'Проверка наличия присутствия в достаточном для создания файла количестве свободного пространства  
If NewFile.Size > FreeSize Then  
temp = MsgBox("Файл " + NewFile.Name + " не может быть размещен из-за нехватки свободного места."; vbExclamation; "Процесс создания файла")  
Exit Sub  
End If  
count = NewFile.Size 'счетчик уже записанной части файла  
NewFile.First = NextFreeCellFAT ' задание точки входа в FAT для данного файла  
Dim PreviousCellFAT As Integer 'Номер последней модифицировавшейся ячейки FAT  
PreviousCellFAT = NextFreeCellFAT  
Call ToFAT(PreviousCellFAT; 0) 'Записали в нее ноль-признак того, что это последний (пока) кластер данного файла  
count = count - 8 'как минимум 8 байт файла уже размещены  
While count > 0 'пока весь файл не разместился  
Call ToFAT(PreviousCellFAT; NextFreeCellFAT) 'в последнюю модифицировавшуюся ячейку FAT вносим указатель на следующую свободную ячейку  
PreviousCellFAT = NextFreeCellFAT  
Call ToFAT(PreviousCellFAT; 0) 'Записали в нее ноль-признак того, что это последний (пока) кластер данного файла  
count = count - 8 'как минимум 8 байт файла уже размещены  
Wend  
Call AddFileToCatalog(NewFile) 'добавили файл в каталог  
End Sub  
  
Sub DeleteFile(File As FileID) 'процедура удаляет заданный файл без разговоров  
Call DeleteCellFromFAT(File.First)  
Call DeleteFileFromCatalog(File.Name)  
End Sub  
  
Sub Refresh() 'обновление изображения области файлов  
With Sheets("Sheet")  
.Range("F6:U13").Interior.ColorIndex = ColorOfPaper 'окрасили область файлов в цвет фона  
.Range("F6:U13").Value = "" 'очистили все ячейки области файлов  
.Range("F6:U13").NumberFormat = "0" 'формат числа для области файлов-целые числа  
.ClearArrows 'убрали все стрелки  
Dim PointerToFile As String  
NumberFile = 1  
While .Cells(NumberFile + 3; 2) <> "" 'последовательно просматриваем область файлов. Для каждого файлавыполняется следующая процедура:  
NumberCellFAT = .Cells(NumberFile + 3; 4) 'точка входа в FAT N-ного файла  
PointerToFile = "=R" & NumberFile + 3 & "C2" 'указатель на ячейку каталога с именем N-ного файла  
Relation = (.Cells(NumberFile + 3; 3) - 1) Mod 8 'смещение. Являет собой размер неполностью заполненного кластера  
While .Cells(3; NumberCellFAT + 5) <> 0 'просмотр FAT до признака конца цепочки  
.Range(Cells(6; NumberCellFAT + 5); Cells(6 + 7; NumberCellFAT + 5)).Interior.ColorIndex = ColorUsedPartOfFAT + NumberFile 'выделение цветом очередного непоследнего кластера N-ного файла  
.Range(Cells(6; NumberCellFAT + 5); Cells(6 + 7; NumberCellFAT + 5)).Font.ColorIndex = ColorUsedPartOfFAT + NumberFile  
.Range(Cells(6; NumberCellFAT + 5); Cells(6 + 7; NumberCellFAT + 5)).Formula = PointerToFile 'каждая ячейка этого кластера теперь указывает на ячейку с именем файла в каталоге  
NumberCellFAT = .Cells(3; NumberCellFAT + 5) 'взяли следующую ячейку FAT из цепочки  
Wend 'теперь обработка последнего кластера N-ного файла. Отличается тем, что он может быть занят не полностью  
.Range(Cells(6; NumberCellFAT + 5); Cells(6 + Relation; NumberCellFAT + 5)).Interior.ColorIndex = ColorUsedPartOfFAT + NumberFile  
.Range(Cells(6; NumberCellFAT + 5); Cells(6 + Relation; NumberCellFAT + 5)).Font.ColorIndex = ColorUsedPartOfFAT + NumberFile  
.Range(Cells(6; NumberCellFAT + 5); Cells(6 + Relation; NumberCellFAT + 5)).Formula = PointerToFile  
  
NumberFile = NumberFile + 1 'работаем со следующим файлом  
Wend  
End With  
End Sub  
  
Макросы нижнего уровня абстракции  
Function FreeSize() As Integer 'Функция для определения количества свободного места в области файлов  
FreeSize = 0  
temp = 6  
While temp < 22 'просмотр области FAT  
If Sheets("Sheet").Cells(3; temp).Value = "" Then \_  
FreeSize = FreeSize + 8 'если ячейка FAT пуста, то своб.места в ней 8 байт  
temp = temp + 1  
Wend  
End Function  
  
Function NextFreeCellFAT() As Integer 'Функция возвращает номер первого найденного свободного элемента FAT  
NextFreeCellFAT = 1  
While NextFreeCellFAT < 17  
If Sheets("Sheet").Cells(3; NextFreeCellFAT + 5).Value = "" Then Exit Function  
NextFreeCellFAT = NextFreeCellFAT + 1  
Wend  
End Function  
  
Sub AddFileToCatalog(File As FileID) 'добавление файла в область каталога  
temp = 4  
With Sheets("Sheet")  
While .Cells(temp; 2) <> ""  
temp = temp + 1 'поиск свободного места в каталоге файлов  
Wend 'кто дочитает до этого места-бутылка шампанского !  
.Cells(temp; 2) = File.Name 'Запись в каталог информации о файле  
.Cells(temp; 3) = File.Size  
.Cells(temp; 4) = File.First  
End With  
End Sub  
  
Sub DeleteFileFromCatalog(NameDeletedFile As String) 'удаление информации о файле из каталога  
Position = 4  
While Sheets("Sheet").Cells(Position; 2).Text <> NameDeletedFile  
Position = Position + 1  
Wend  
For temp = Position To 16 + 3 'следующие за удаляемым файлом ячейки сдвигаются, \_  
затирая запись об удаляемом файле  
Sheets("sheet").Range(Cells(temp; 2); Cells(temp; 4)).Value = \_  
Sheets("sheet").Range(Cells(temp + 1; 2); Cells(temp + 1; 4)).Value  
Next  
End Sub  
  
Sub ToFAT(NumberCell As Integer; Value As Integer) 'Процедура записи в заданную ячейку FAT заданного значения  
Sheets("Sheet").Cells(3; NumberCell + 5).Value = Value  
End Sub  
  
Sub DeleteCellFromFAT(StartCell As Integer) 'рекурсивная функция удаления цепочки из FAT \_  
Вызывается с номером точки входа  
' MsgBox ("DeleteCellFromFAT, StartCell=" & StartCell)  
If Sheets("Sheet").Cells(3; 5 + StartCell).Value = 0 Then 'если в рассматриваемой точке - 0, \_  
то это значит, что она является указателем конца файла  
Sheets("Sheet").Cells(3; 5 + StartCell) = "" 'и поэтому очищаем эту ячейку FAT  
Else 'в противном случае это указатель на следующую ячейку FAT - вызываем эту же процедуру  
DeleteCellFromFAT (Sheets("sheet").Cells(3; 5 + StartCell).Value) 'но с новым номером ячейки FAT  
Sheets("sheet").Cells(3; 5 + StartCell) = "" 'и затем ее очищаем  
End If  
End Sub  
  
  
  
Список литературы, примененной при подготовке курсовой работы.



|  |  |
| --- | --- |
| 1. | Microsoft Systems Journal, Sept 1989. Получен из Computer Library Periodicals, Jan 1990, Doc #14753 |
| 2. | Профессиональная работа в MS-DOS, Р.Данкан, Мир, 1993 |
| 3. | Excel 5.0 для профессионалов, Н.Николь, Р. Альбрехт, Москва, “Эком”, 1996 |