**Лекция 11. Управление файловой системой и ее оптимизация. Примеры файловых систем. Пользовательский интерфейс: клавиатура, мышь, монитор.**

 **Цель занятия:** Сформировать знания студентов о файловых системах, их оптимизации, а так же о пользовательском интерфейсе.

**Файловая система (ФС)** является важной частью любой операционной системы, которая отвечает за организацию хранения и доступа к информации на каких-либо носителях. Рассмотрим в качестве примера файловые системы для наиболее распространенных в наше время носителей информации – магнитных дисков. Как известно, информация на жестком диске хранится в секторах (обычно 512 байт) и само устройство может выполнять лишь команды считать/записать информацию в определенный сектор на диске. В отличие от этого файловая система позволяет пользователю оперировать с более удобным для него понятием - файл. Файловая система берет на себя организацию взаимодействия программ с файлами, расположенными на дисках. Для идентификации файлов используются имена. Современные файловые системы предоставляют пользователям возможность давать файлам достаточно длинные мнемонические названия.

Под каталогом в ФС понимается, с одной стороны, группа файлов, объединенных пользователем исходя из некоторых соображений, с  другой стороны каталог - это файл, содержащий системную информацию о группе составляющих его файлов. Файловые системы обычно имеют иерархическую структуру, в которой уровни создаются за счет каталогов, содержащих информацию о файлах и каталогах более низкого уровня.

Рассмотрим более подробно структуру жесткого диска. Базовой единицей жесткого диска является **раздел,**создаваемый во время разметки жесткого диска. Каждый раздел содержит один том, обслуживаемый какой-либо файловой системой и имеющий таблицу оглавления файлов - корневой каталог. Некоторые операционные системы поддерживают создание томов, охватывающих несколько разделов. Жесткий диск может содержать до четырех основных разделов. Это ограничение связано с характером организации данных на жестких дисках IBM-совместимых компьютеров. Многие операционные системы позволяют создавать, так называемый, **расширенный**(extended) раздел, который по аналогии с разделами может разбиваться на несколько логических дисков.

В первом физическом секторе жесткого диска располагается головная запись загрузки и таблица разделов (табл. 1). **Головная запись загрузки**(**master boot record, MBR**) - первая часть данных на жестком диске. Она зарезервирована для программы начальной загрузки BIOS (*ROM Bootstrap routin*e), которая при загрузке с жесткого диска считывает и загружает в память первый физический сектор на активном разделе диска, называемый **загрузочным сектором (Boot Sector)**. Каждая запись в таблице разделов (partition table) содержит начальную позицию и размер раздела на жестком диске, а также информацию о том, первый сектор какого раздела содержит загрузочный сектор.

|  |  |
| --- | --- |
| **Размер (байт)** | **Описание** |
| 446 | Загрузочная запись (MBR) |
| 16 | Запись 1 раздела |
| 16 | Запись 2 раздела |
| 16 | Запись 3 раздела |
| 16 | Запись 4 раздела |
| 2 | Сигнатура 055AAh |

**Табл. 1. Таблица деления диска**

В широком смысле понятие "файловая система" включает:

* совокупность всех файлов на диске,
* наборы служебных структур данных, используемых для управления файлами, такие как, например, каталоги файлов, дескрипторы файлов, таблицы распределения свободного и занятого пространства на диске,
* комплекс системных программных средств, реализующих управление файлами, в частности операции по созданию, уничтожению, чтению, записи, именованию файлов, установке атрибутов и уровней доступа, поиску и т.д.

Различие между файловыми системами заключается, в основном, в способах распределения пространства между файлами на диске и организации на диске служебных областей.

Современные операционные системы стремятся обеспечить пользователя возможностью работать одновременно с несколькими файловыми системами. В этом случае ФС рассматривается как часть подсистемы ввода-вывода. В большинстве операционных систем (Windows 95, NT, OS/2) реализуется механизм переключения файловых систем (File System Switch, FSS), позволяющий поддерживать различные типы ФС. В соответствии с этим подходом информация о файловых системах и файлах разбивается на две части – зависимую от ФС и не зависимую. FSS обеспечивает интерфейс между ядром и файловой системой, транслируя запросы ядра в операции, зависящие от типа файловой системы. При этом ядро имеет представление только о независимой части ФС.

Файловая система представляет многоуровневую структуру, на верхнем уровне которой располагается так называемый переключатель файловых систем (в Windows, такой переключатель называется устанавливаемым диспетчером файловой системы - installable filesystem manager, IFS). Он обеспечивает интерфейс между приложением и конкретной файловой системой, к которой обращается приложение. Переключатель файловых систем преобразует запросы к файлам в формат, воспринимаемый следующим уровнем - уровнем драйверов файловых систем. Для выполнения своих функций драйверы файловых систем обращаются к драйверам конкретных устройств хранения информации.

Клиент-серверные приложения предъявляют повышенные требования к производительности файловых систем. Современные файловые системы должны обеспечивать эффективный доступ к файлам, поддержку носителей данных достаточно большого объема, защиту от несанкционированного доступа к данным и сохранение целостности данных. Под целостностью данных подразумевается способность ФС обеспечивать отсутствие ошибок и нарушений согласованности в данных, а также восстанавливать поврежденные данные.

**FAT**

Файловая система FAT (File Allocation Table) была разработана Биллом Гейтсом и Марком МакДональдом в 1977 году и первоначально использовалась в операционной системе 86-DOS. Чтобы добиться переносимости программ из операционной системы CP/M в 86-DOS, в ней были сохранены ранее принятые ограничения на имена файлов. В дальнейшем 86-DOS была приобретена Microsoft и стала основой для ОС MS-DOS 1.0, выпущенной в августе 1981 года. FAT была предназначена для работы с гибкими дисками размером менее 1 Мбайта, и вначале не предусматривала поддержки жестких дисков. В настоящее время FAT поддерживает файлы и разделы размеров до 2 Гбайт.

В FAT применяются следующие соглашения по именам файлов:

* + имя должно начинаться с буквы или цифры и может содержать любой символ ASCII, за исключением пробела и символов "/\[]:;|=,^\*?
	+ Длина имени не превышает 8 символов, за ним следует точка и необязательное расширение длиной до 3 символов.
	+ регистр символов в именах файлов не различается и не сохраняется.

Структура раздела FAT изображена на рисунке 2. В блоке параметров BIOS содержится необходимая BIOS информация о физических характеристиках жесткого диска. Файловая система FAT не может контролировать отдельно каждый сектор, поэтому она объединяет смежные сектора в **кластеры** (**clusters**). Таким образом, уменьшается общее количество единиц хранения, за которыми должна следить файловая система. Размер кластера в FAT является степенью двойки и определяется размером тома при форматировании диска (табл. 2). Кластер представляет собой минимальное пространство, которое может занимать файл. Это приводит к тому, что часть пространства диска расходуется впустую. В состав операционной системы входят различные утилиты (DoubleSpace, DriveSpace), предназначенные для уплотнения данных на диске.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Загрузочный секторБлок параметров BIOS (BPB) | FAT | FAT(копия) | Корневой каталог | Область файлов |

**Рис. 2**

Свое название FAT получила от одноименной таблицы размещения файлов. В таблице размещения файлов хранится информация о кластерах логического диска. Каждому кластеру в FAT соответствует отдельная запись, которая показывает, свободен ли он, занят ли данными файла, или помечен как сбойный (испорченный). Если кластер занят под файл, то в соответствующей записи в таблице размещения файлов указывается адрес кластера, содержащего следующую часть файла. Из-за этого FAT называют файловой системой со связанными списками. Оригинальная версия FAT, разработанная для DOS 1.00, использовала 12-битную таблицу размещения файлов и поддерживала разделы объемом до 16 Мб (в DOS можно создать не более двух разделов FAT). Для поддержки жестких дисков размером более 32 Мб разрядность FAT была увеличена до 16 бит, а размер кластера - до 64 секторов (32 Кб). Так как каждому кластеру может быть присвоен уникальный 16-разрядный номер, то FAT поддерживает максимально 216, или 65536 кластеров на одном томе.

|  |  |  |
| --- | --- | --- |
| **Размер раздела** | **Размер кластера** | **Тип FAT** |
| < 16 Мб | 4 Кб | FAT12 |
| 16 Мб – 127 Мб | 2 Кб | FAT16 |
| 128 Мб – 255 Мб | 4 Кб | FAT16 |
| 256 Мб – 511 Мб | 8 Кб | FAT16 |
| 512 Мб – 1023 Мб | 16 Кб | FAT16 |
| 1 Гб – 2 Гб | 32 Кб | FAT16 |

**Табл. 2**

Поскольку загрузочная запись слишком мала для хранения алгоритма поиска системных файлов на диске, то системные файлы должны находиться в определенном месте, чтобы загрузочная запись могла их найти. Фиксированное положение системных файлов в начале области данных накладывает жесткое ограничение на размеры корневого каталога и таблицы размещения файлов. Вследствие этого общее число файлов и подкаталогов в корневом каталоге на диске FAT ограничено 512.

Каждому файлу и подкаталогу в FAT соответствует 32-байтный элемент каталога (directory entry), содержащий имя файла, его атрибуты (архивный, скрытый, системный и “только для чтения”), дату и время создания (или внесения в него последних изменений), а также прочую информацию (табл. 3).

|  |  |
| --- | --- |
| **Содержание** | **Размер (байт)** |
| Имя файла | 8 |
| Расширение | 3 |
| Байт атрибутов | 1 |
| Зарезервировано | 10 |
| Время | 2 |
| Дата | 2 |
| Номер начального кластера с данными | 2 |
| Размер файла | 4 |

**Табл. 3. Элемент каталога**

Файловая система FAT всегда заполняет свободное место на диске последовательно от начала к концу. При создании нового файла или увеличении уже существующего она ищет самый первый свободный кластер в таблице размещения файлов. Если в процессе работы одни файлы были удалены, а другие изменились в размере, то появляющиеся в результате пустые кластеры будут рассеяны по диску. Если кластеры, содержащие данные файла, расположены не подряд, то файл оказывается **фрагментированным**. Сильно фрагментированные файлы значительно снижают эффективность работы, так как головки чтения/записи при поиске очередной записи файла должны будут перемещаться от одной области диска к другой. В состав операционных систем, поддерживающих FAT, обычно входят специальные утилиты дефрагментации диска, предназначенные повысить производительность файловых операций.

Еще один недостаток FAT заключается в том, что ее производительность сильно зависит от количества файлов, хранящихся в одном каталоге. При большом количестве файлов (около тысячи), выполнение операции считывания списка файлов в каталоге может занять несколько минут. Это обусловлено тем, что в FAT каталог имеет линейную неупорядоченную структуру, и имена файлов в каталогах идут в порядке их создания. В результате, чем больше в каталоге записей, тем медленнее работают программы, так как при поиске файла требуется просмотреть последовательно все записи в каталоге.

Поскольку FAT изначально проектировалась для однопользовательской операционной системы DOS, то она не предусматривает хранения такой информации, как сведения о владельце или полномочия доступа к файлу/каталогу.

FAT является наиболее распространенной файловой системой и ее в той или иной степени поддерживают большинство современных ОС. Благодаря своей универсальности FAT может применяться на томах, с которыми работают разные операционные системы.

Хотя нет никаких препятствий использовать при форматировании дискет любую другую файловую систему, большинство ОС для совместимости используют FAT. Отчасти это можно объяснить тем, что простая структура FAT требует меньше места для хранения служебных данных, чем остальные системы. Преимущества других файловых систем становятся заметны только при использовании их на носителях объемом более 100 Мб.

Надо отметить, что FAT - простая файловая система, не предотвращающая порчи файлов из-за ненормального завершения работы компьютера. В состав операционных систем, поддерживающих FAT, входят специальные утилиты проверяющие структуру и корректирующие несоответствия в файловой системе.

**HPFS**

Высокопроизводительная файловая система HPFS (High Performance File System) была представлена фирмой IBM в 1989 году вместе с операционной системой OS/2 1.20. Файловая система HPFS также поддерживалась ОС Windows NT до версии 3.51 включительно. По производительности эта ФС существенно опережает FAT. HPFS позволяет использовать жесткие диски объемом до 2 Терабайт (первоначально до 4 Гбайт). Кроме того, она поддерживает разделы диска размером до 512 Гб и позволяет использовать имена файлов длиной до 255 символов (на каждый символ при этом отводится 2 байта). В HPFS по сравнению с FAT уменьшено время доступа к файлам в больших каталогах.

HPFS распределяет пространство на диске не кластерами как в FAT, а физическими секторами по 512 байт, что не позволяет ее использовать на жестких дисках, имеющих другой размер сектора. Эти секторы принято называть блоками. Чтобы уменьшить фрагментацию диска, при распределении пространства под файл HPFS стремится, по возможности, размещать файлы в последовательных смежных секторах. Фрагмент файла, располагающийся в смежных секторах, называется **экстентом**.

Для нумерации единиц распределения пространства диска HPFS использует 32 разряда, что дает 232, или более 4 миллиардов номеров. Однако HPFS использует числа со знаком, что сокращает число возможных номеров блоков до 2 миллиардов. Помимо стандартных атрибутов файла, HPFS поддерживает **расширенные атрибуты**файла (Extended Attributes, EA), которые могут содержать до 64 Кб различных дополнительных сведений о файле.

Диск HPFS имеет следующие три базовые структуры (рис. 3): загрузочный блок (**BootBlock**), дополнительный блок (**SuperBlock**) и резервный блок (**SpareBlock**).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Загрузочный блок | Дополнительный блок | Резервный блок | Группа 1 | Битовая карта группы 1 | Битовая карта группы 2 | Группа 2 | Группа 3 | Битовая карта группы 3 | Битовая карта группы 4 | Группа 4 |

**Рис. 3. Дисковый раздел HPFS**

***Загрузочный блок*** в HPFS аналогичен загрузочному блоку в FAT. Он располагается в секторах с 0 по 15 и занимает на диске 8 Кб. Системные файлы, также как и в FAT, располагаются в корневом каталоге, но при этом физически могут находиться в любом месте на диске.

В 16 секторе размещается ***дополнительный блок***, содержащий указатель на **список блоков битовых карт**(bitmap block list). В этом списке перечислены все блоки на диске, в которых расположены битовые карты, используемые для обнаружения свободных секторов. Также в дополнительном блоке хранится указатель на список дефектных блоков (bad block list), указатель на группу каталогов (directory band), указатель на файловый узел корневого каталога и дата последней проверки диска. Файловый узел (fnode) – это структура диска HPFS, которая содержит информацию о расположении файла и о его расширенных атрибутах.

В следующем секторе находится ***резервный блок***, содержащий карту аварийного замещения (hotfix map), указатель на список свободных запасных блоков (directory emergency free block list) и ряд системных флагов. Резервный блок обеспечивает высокую отказоустойчивость HPFS и позволяет восстанавливать поврежденные данные на диске.

Остальное пространство диска разделено на **группы**(band) хранения данных. Каждая группа занимает 8 Мб и имеет свою собственную битовую карту свободного пространства, которая похожа на таблицу размещения файлов FAT. Каждому сектору группы соответствует один бит к ее битовой карте, показывающий занят ли соответствующий сектор. Битовые карты двух групп располагаются на диске рядом, также как располагаются и сами группы. Это дает возможность непрерывно разместить на жестком диске файл размером до 16 Мб.

Одна из групп данных размером 8 Мб, расположенная в середине жесткого диска и называемая **группой каталогов,**хранит информацию о каталогах диска. В ней наряду с остальными каталогами располагается и корневой каталог. Расположение группы каталогов в центре диска значительно сокращает время позиционирования головок чтения/записи.

В отличие от линейной структуры FAT, структура каталога в HPFS представляет собой сбалансированное дерево (так называемое B-дерево) с записями, расположенными в алфавитном порядке. Сбалансированное дерево состоит из корневого (root block) и оконечных блоков (leaf block). Блоки занимают 4 последовательных сектора и в среднем могут содержать 40 записей. Каждая запись корневого блока указывает на один из оконечных блоков (если только в каталоге не меньше 40 файлов); в свою очередь, каждая запись в оконечном блоке указывает на файловый узел файла или на оконечный блок следующего уровня. Таким образом, двухуровневая структура может содержать 40 оконечных блоков по 40 записей в каждом и описывать до 1600 файлов. При поиске файловая система HPFS просматривает только необходимые ветви дерева.

Файловый узел имеет размер 512 байт и всегда по возможности располагается непосредственно перед первым блоком своего файла. Каждый файл и каталог диска HPFS имеет свой файловый узел. Информация, хранящаяся в файловом узле, включает в себя расширенные атрибуты файла, если они достаточно малы, чтобы поместится в один сектор диска, и сокращенное имя файла в формате 8.3. Если расширенные атрибуты не помещаются в файловый узел, то в него записывается указатель на них. Положение файла на диске описывается в файловом узле двумя 32-битными числами. Первое из чисел представляет собой указатель на первый блок файла, а второе - длину экстента. Если же файл фрагментирован, то его размещение описывается дополнительными парами 32-битных чисел. В файловом узле можно хранить информацию максимум о 8 экстентах файла. Если файл имеет больше число экстентов, то в его файловый узел записывается указатель на блок размещения (allocation block), который может содержать до 40 указателей на экстенты или на другие блоки размещения. Таким образом, двухуровневая структура блоков размещения может хранить информацию о 480 (12\*40) секторах, что теоретически, позволяет работать с файлами размером до 7.68 Гб (12\*40\*16 Мб).

**VFAT**

Файловая система VFAT (Virtual FAT), реализованная в Windows NT 3.5, Windows 95 (DOS 7.0), - это файловая система FAT, включающая поддержку длинных имен файлов (Long File Name, LFN) в кодировке UNICODE (каждый символ имени кодируется 2 байтами). VFAT использует ту же самую схему распределения дискового пространства, что и файловая система FAT, поэтому размер кластера определяется величиной раздела.

В VFAT ослаблены ограничения, устанавливаемые соглашениями по именам файлов FAT:

* + имя может быть длиной до 255 символов.
	+ в имя можно включать несколько пробелов и точек, однако, текст после последней точки рассматривается как расширение.
	+ регистр символов в именах не различается, **но**сохраняется.

Основной задачей при разработке VFAT была необходимость корректной работы старых программ, не поддерживающих длинные имена файлов. Как правило, прикладные программы для доступа к файлам используют функции ОС. Если у элемента каталога установить “нереальную” комбинацию битов атрибутов: “только для чтения”, “скрытый”, “системный”, “метка тома” – то любые файловые функции старых версий DOS и Windows не заметят такого элемента каталога. В итоге для каждого файла и подкаталога в VFAT хранится два имени: длинное и короткое в формате 8.3 для совместимости со старыми программами. Длинные имена (LFN) хранятся в специальных записях каталога, байт атрибутов, у которых равен 0Fh. Для любого файла или подкаталога непосредственно перед единственной записью каталога с его именем в формате 8.3 находится группа из одной или нескольких записей, представляющих длинное имя. Каждая такая запись содержит часть длинного имени файла не более 13 символов, из всех таких записей ОС составляет полное имя файла. Поскольку одно длинное имя файла может занимать до 21 записи, а корневой каталог FAT ограничен 512 записями, желательно ограничить использование длинных имен в корневом каталоге.

|  |  |
| --- | --- |
| **Содержание** | **Размер (байт)** |
| Порядок следования | 1 |
| Первые пять символов LFN | 10 |
| Байт атрибутов (0Fh) | 1 |
| Указатель типа (всегда 0) | 1 |
| Контрольная сумма части имени | 1 |
| Следующие шесть символов LFN | 12 |
| Номер начального кластера (всегда 0) | 2 |
| Следующие два символа LFN | 4 |

**Табл. 4. Элемент каталога для длинного имени**

Короткое имя генерируется файловой системой автоматически в формате 8.3. Для создания коротких имен (псевдонимов) файлов используется следующий алгоритм:

1. Из длинного имени удалить все символы не допустимые в именах FAT. Удалить точки в конце и начале имени. После этого удалить все точки, находящиеся внутри имени кроме последней.
2. Обрезать строку, расположенную перед точкой, до 6 символов и добавить в ее конец "~1". Обрезать строку за точкой до 3 символов.
3. Полученные буквы преобразовать в прописные. Если сгенерированное имя совпадает с уже существующим, то увеличить число в строке "~1".

Данный алгоритм зависит от версии операционной системы и в будущих версиях может меняться

Редактирование файлов программами, не поддерживающими длинные имена файлов, может приводить к потере длинных имен. Windows обнаруживает подобные элементы каталога, так как их контрольная сумма не соответствует больше тому, что записано в последующей записи каталога в формате 8.3. Однако такие записи не удаляются системой автоматически, они занимают дисковое пространство, до тех пор, пока вы не запустите программу ScanDisk, входящую в состав операционной системы. Большинство старых дисковых утилит воспримут записи, соответствующие длинным именам, как ошибки логической структуры диска. Попытки использовать данные утилиты, в лучшем случае приведет к потере длинных имен, а в худшем - к потере информации на диске.

**Заключение**

Развитие файловых систем персональных компьютеров определялось двумя факторами - появлением новых стандартов на носители информации и ростом требований к характеристикам файловой системы со стороны прикладных программ (разграничение уровней доступа, поддержка длинных имен файлов в формате UNICODE). Первоначально, для файловых систем первостепенное значение имело увеличение скорости доступа к данным и минимизация объема хранимой служебной информации. Впоследствии с появлением более быстрых жестких дисков и увеличением их объемов, на первый план вышло требование надежности хранения информации, которое привело к необходимости избыточного хранения данных.

Эволюция файловой системы была напрямую связана с развитием технологий реляционных баз данных. Файловая система использовала последние достижения, разработанные для применения в СУБД: механизмы транзакций, защиты данных, систему самовосстановления в результате сбоя.

Развитие файловых систем привело к изменению самого понятия "файл" от первоначального толкования как упорядоченная последовательность логических записей, до понятия файла, как объекта, имеющего набор характеризующих его атрибутов (включая имя файла, его псевдоним, время создания и собственно данные), реализованного в NTFS.

За свою 20 летнюю историю файловая система прошла путь от простой системы, взявшей на себя функции управления файлами, до системы, представляющей собой полноценную СУБД, обладающую встроенным механизмом протоколирования и восстановления данных.

В отличие от попыток ввести стандарт на протокол, описывающий правила доступа к удаленным файловым системам (CIFS, NFS), не стоит ожидать появления подобного стандарта, описывающего файловые системы для жестких дисков. Это можно объяснить тем, что файловая система жестких дисков все еще продолжает оставаться одной из главных частей операционной системы, влияющей на ее производительность. Поэтому каждый производитель операционных систем будет стремиться использовать файловую систему, "родную" для его ОС.

Дальнейшая эволюция файловых систем пойдет по пути совершенствования механизмов хранения данных, оптимизации хранения мультимедийных данных, использования новых технологий, применяемых в базах данных (возможность полнотекстового поиска, сортировка файлов по различным атрибутам).

**Контрольные вопросы:**

* + 1. *Что такое файловая система?*
		2. *Каково основное назначение файловой системы?*
		3. *Какие файловые системы вы знаете?*
		4. *Охарактеризуйте файловую систему FAT.*
		5. *Охарактеризуйте файловую систему NTFS*
		6. *Охарактеризуйте файловую систему HPFS*
		7. *Охарактеризуйте файловую систему VFAT*