

## Лекции №11-12

### Операционные системы. Логическая структура диска. Обмен данными между приложениями.

#### ВЗАИМОДЕЙСТВИЕ ПОЛЬЗОВАТЕЛЯ С ПК

Компьютер обменивается информацией с человеком посредством набора определенных правил, которые в компьютерной литературе называются интерфейсом. Интерфейс, по определению - это правила взаимодействия операционной системы с пользователями, а также соседних уровней в сети ЭВМ.

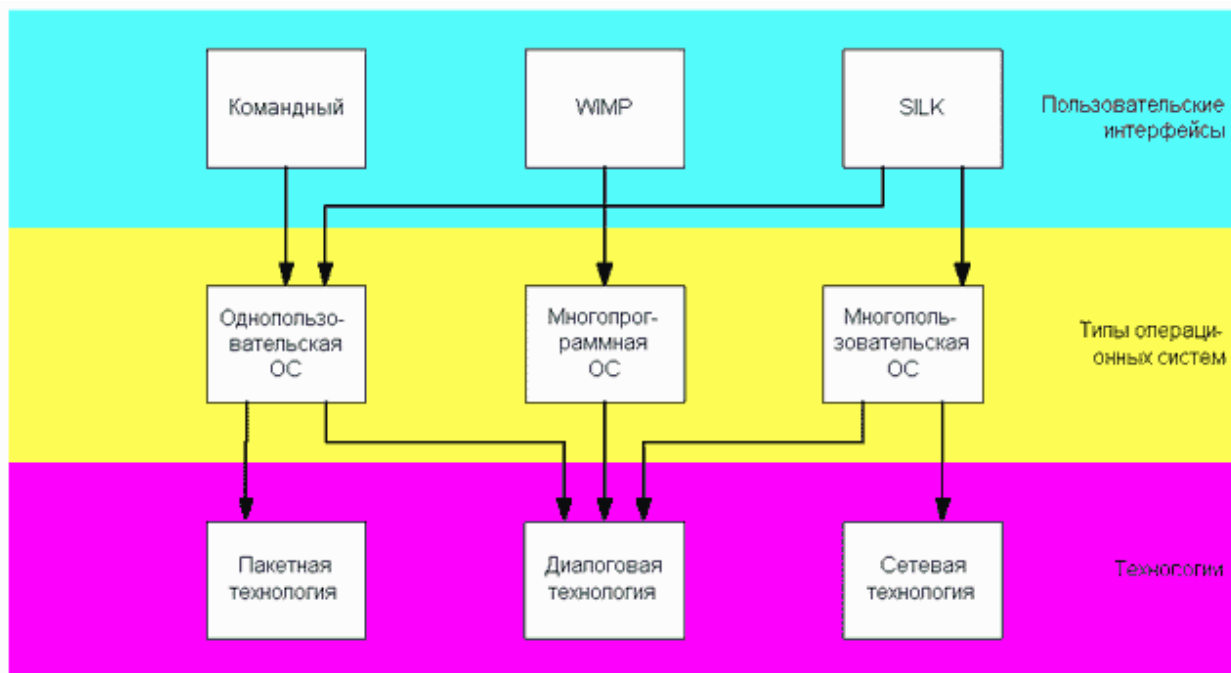


Рис. А.1. Взаимодействие типов операционных систем, пользовательских интерфейсов и технологий их реализации.

#### Типы интерфейсов.

1) *Командный интерфейс.* Командный интерфейс называется так потому, что в этом виде интерфейса человек подает "команды" компьютеру, а компьютер их выполняет и выдает результат человеку. Командный интерфейс реализован в виде пакетной технологии и технологии командной строки.

2) *WIMP - интерфейс* (Window - окно, Image - образ, Menu - меню, Pointer - указатель). Характерной особенностью этого вида интерфейса является то, что диалог с пользователем ведется не с помощью команд, а с помощью графических образов - меню, окон, других элементов. Хотя и в этом интерфейсе подаются команды машине, но это делается "опосредованно", через графические образы. Этот вид интерфейса реализован на двух уровнях технологий: простой графический интерфейс и "чистый" WIMP - интерфейс.

3) *SILK - интерфейс* (Speech - речь, Image - образ, Language - язык, Knowledge - знание). Этот вид интерфейса наиболее приближен к обычной, человеческой форме общения. В рамках этого интерфейса идет обычный "разговор" человека и компьютера. При этом компьютер находит для себя команды, анализируя человеческую речь и находя в ней ключевые фразы. Результат выполнения команд он также преобразует в понятную человеку форму.

## ОПЕРАЦИОННАЯ СИСТЕМА (ОС)

**ОС – комплекс системных и служебных программных средств.**

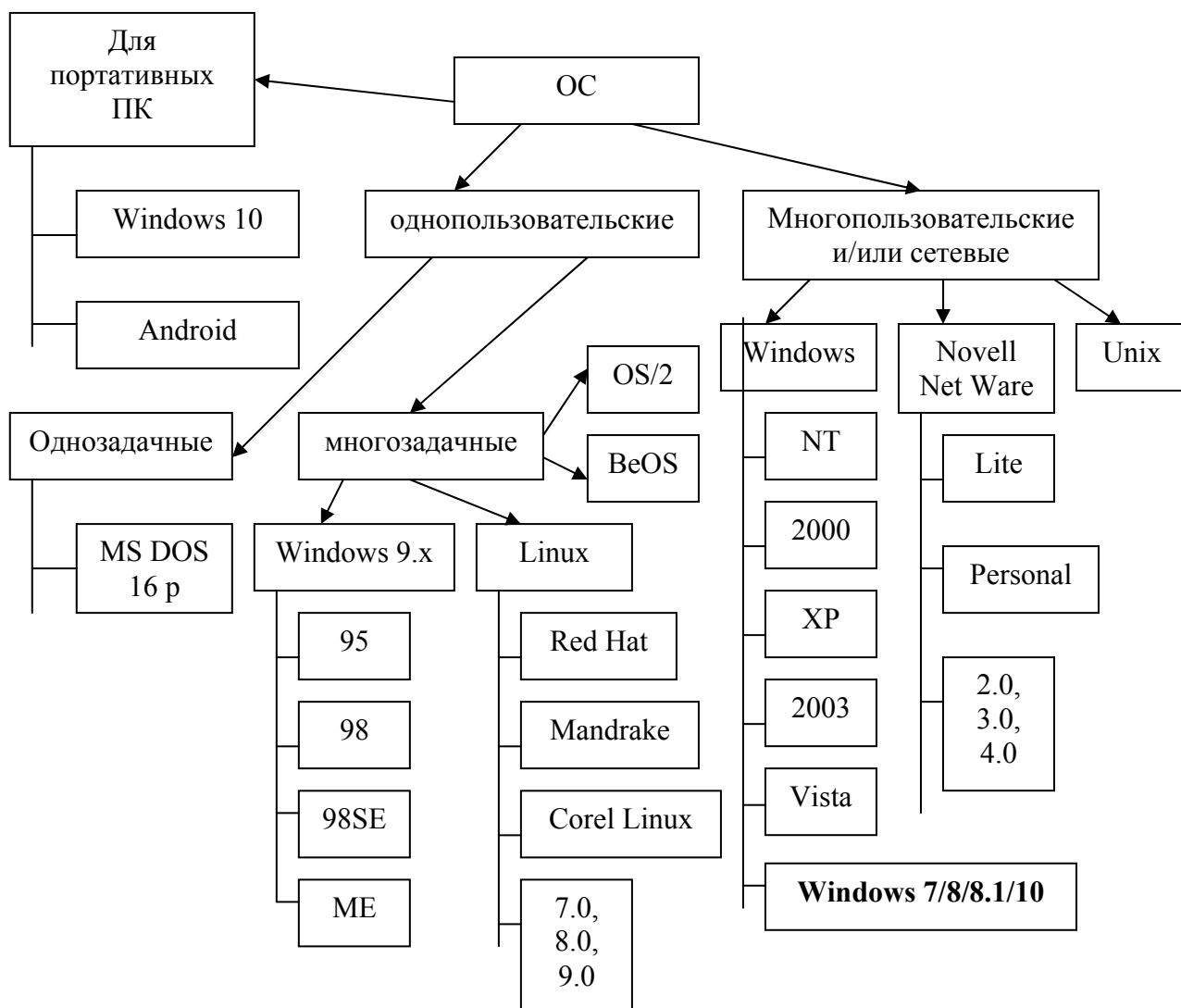
Основная функция всех операционных систем – *посредническая*. Она заключается в обеспечении нескольких видов интерфейсов:

- **интерфейс пользователя** (интерфейс между пользователем и программно-аппаратными средствами ПК);
- **аппаратно-программный интерфейс** (интерфейс между программным и аппаратным обеспечением);
- **программный интерфейс** (интерфейс между различными видами программного обеспечения).

**ОС могут различаться:**

- особенностями реализации внутренних алгоритмов управления основными ресурсами компьютера (процессорами, памятью, устройствами),
- особенностями использованных методов проектирования, типами аппаратных платформ,
- областями использования и многими другими свойствами.

### Классификация операционных систем



Операционные системы могут различаться особенностями реализации внутренних алгоритмов управления основными ресурсами компьютера (процессорами, памятью, устройствами), особенностями использованных методов проектирования, типами аппаратных платформ, областями использования и многими другими свойствами.

В зависимости от **особенностей использованного алгоритма** управления процессором операционные системы делят на:

- многозадачные и однозадачные,
- многопользовательские и однопользовательские,
- на системы, поддерживающие многоплатформенную обработку и не поддерживающие ее,
- на многопроцессорные (асимметричные ОС и симметричные ОС) и однопроцессорные системы.

**По типу аппаратуры** различают операционные системы персональных компьютеров, мини-компьютеров, мэйнфреймов, кластеров и сетей ЭВМ. Среди перечисленных типов компьютеров могут встречаться как однопроцессорные варианты, так и многопроцессорные.

### **Особенности методов построения**

При описании операционной системы часто указываются особенности ее структурной организации и основные концепции, положенные в ее основу.

К таким базовым концепциям относятся:

- Способы построения ядра системы - монолитное ядро или микроядерный подход. Большинство ОС использует монолитное ядро, которое компонуется как одна программа, работающая в привилегированном режиме и использующая быстрые переходы с одной процедуры на другую, не требующие переключения из привилегированного режима в пользовательский и наоборот. Альтернативой является построение ОС на базе микроядра, работающего также в привилегированном режиме и выполняющего только минимум функций по управлению аппаратурой, в то время как функции ОС более высокого уровня выполняют специализированные компоненты ОС - серверы, работающие в пользовательском режиме. При таком построении ОС работает более медленно, так как часто выполняются переходы между привилегированным режимом и пользовательским, зато система получается более гибкой - ее функции можно наращивать, модифицировать или сужать, добавляя, модифицируя или исключая серверы пользовательского режима. Кроме того, серверы хорошо защищены друг от друга, как и любые пользовательские процессы.
- Построение ОС на базе объектно-ориентированного подхода дает возможность использовать все его достоинства, хорошо зарекомендовавшие себя на уровне приложений, внутри операционной системы, а именно: аккумуляцию удачных решений в форме стандартных объектов, возможность создания новых объектов на базе имеющихся с помощью механизма наследования, хорошую защиту данных за счет их инкапсуляции во внутренние структуры объекта, что делает данные недоступными для несанкционированного использования извне, структурированность системы, состоящей из набора хорошо определенных объектов.
- Наличие нескольких прикладных сред дает возможность в рамках одной ОС одновременно выполнять приложения, разработанные для нескольких ОС. Многие современные операционные системы поддерживают одновременно прикладные среды MS-DOS, Windows, UNIX (POSIX), OS/2 или хотя бы некоторого подмножества из этого популярного набора. Концепция множественных прикладных сред наиболее просто реализуется в ОС на базе микроядра, над

которым работают различные серверы, часть которых реализуют прикладную среду той или иной операционной системы.

- Распределенная организация операционной системы позволяет упростить работу пользователей и программистов в сетевых средах. В распределенной ОС реализованы механизмы, которые дают возможность пользователю представлять и воспринимать сеть в виде традиционного однопроцессорного компьютера. Характерными признаками распределенной организации ОС являются: наличие единой справочной службы разделяемых ресурсов, единой службы времени, использование механизма вызова удаленных процедур (RPC) для прозрачного распределения программных процедур по машинам, многократной обработки, позволяющей распараллеливать вычисления в рамках одной задачи и выполнять эту задачу сразу на нескольких компьютерах сети, а также наличие других распределенных служб.

### Логическое хранение и кодирование информации

Для обеспечения наиболее оптимальной производительности и работы накопителя как запоминающего устройства, а также, для улучшения программного интерфейса, накопители не используются системами в первичном виде, а в них, на основе физически присутствующих структур - дорожек и секторов, используется логическая структура хранения и доступа к информации. Ее тип и характеристики зависят от используемой операционной системы, и называется она - файловой системой. В настоящее время имеется достаточно много типов различных файловых систем, практически столько же, сколько и различных операционных систем, однако, все они основывают свои логические структуры данных на нескольких первичных логических структурах.

Каждый диск содержит следующие области:

- главную загрузочную запись - **Master Boot Record (MBR)** которая, в свою очередь, содержит загрузочную запись - загрузочный сектор (**Boot Record**), содержащий некоторую идентификационную информацию, параметры диска, а также программу начальной загрузки операционной системы;
- таблицу разбиения диска (**Partition Table**), содержащую информацию о разбиении одного физического диска на несколько (в частном случае, один) логических дисков; содержащая 4 записи - элементы логических разделов – **Partitions**;
- две копии (одна из них - резервная) таблицы размещения файлов (**File Allocation Table - FAT**), содержащей информацию о размещении на диске каждого файла;
- корневой каталог (**Root Directory**);
- собственно данные - каталоги и файлы.

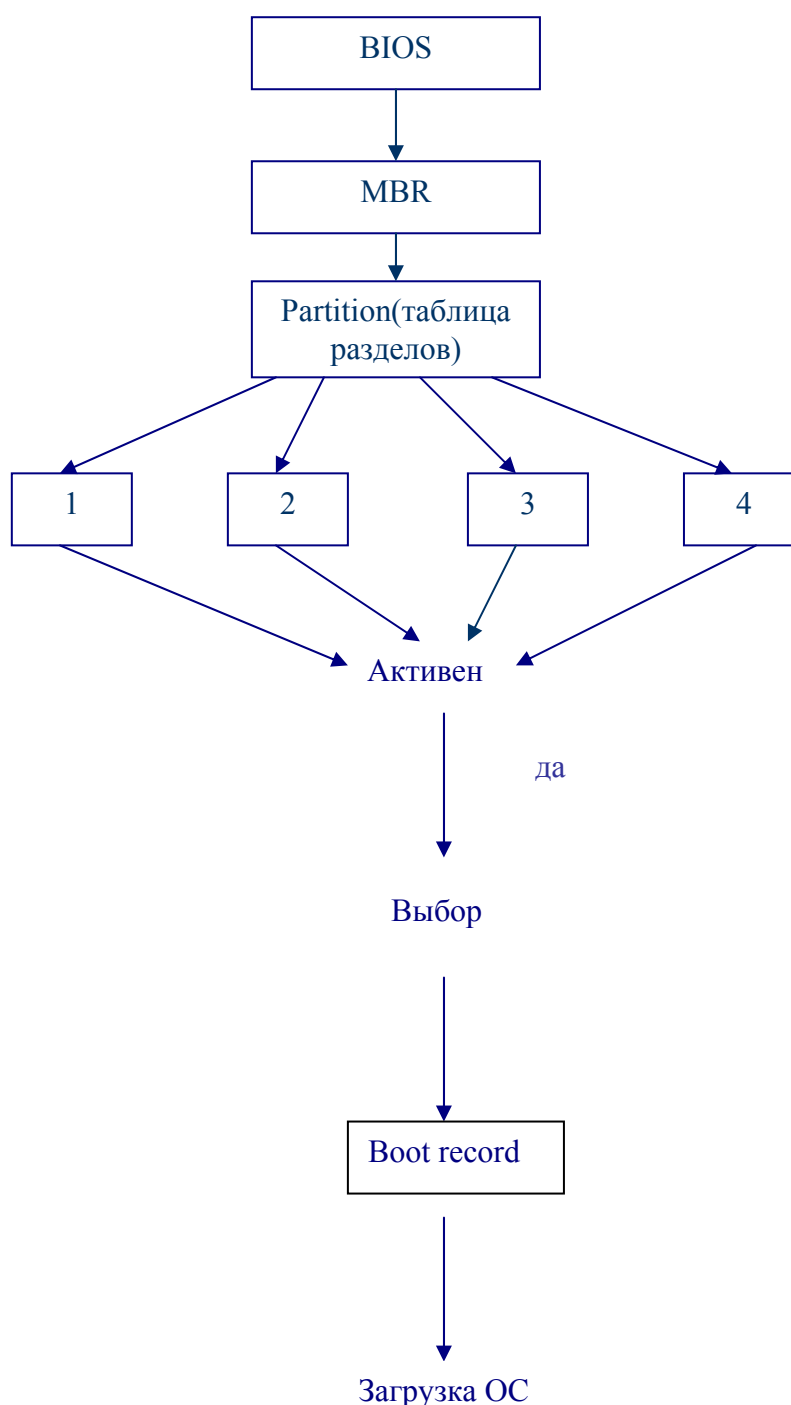
Первый сектор жесткого диска (сектор 1, дорожка 0, головка 0) содержит главную загрузочную запись **MBR**. Структура **MBR**. представлена в таблице

Название записи в MBR	Длина, байт
Загрузочная запись – Boot Record	446
Элемент таблицы разделов 1 – Partition 1	16
Элемент таблицы разделов 2 – Partition 2	16
Элемент таблицы разделов 3 – Partition 3	16
Элемент таблицы разделов 4 – Partition 4	16
Сигнатура окончания Partition Table	2

Все разделы могут содержать **загрузчик операционной системы**(Boot Record), который располагается, как правило, в первом секторе и занимает один сектор. В этом секторе располагаются структуры - записи, имеющие отношение лишь к конкретной операционной системе и, следовательно, они могут отличаться для разных разделов и версий операционных систем. Загрузчик операционной системы представляет собой маленькую программу, осуществляющую считывание в память начального кода операционной системы во время ее старта. Загрузочная запись считывается в оперативную память главной загрузочной записью, после чего ей передается управление. Загрузочная запись и выполняет загрузку операционной системы.

**Загрузка операционной системы с жесткого диска - двухступенчатый процесс.**

Вначале модули инициализации BIOS считывают главную загрузочную запись в память и, ей передается управление. Главная загрузочная запись просматривает таблицу разделов и находит активный раздел. Если активных разделов несколько, на консоль выводится сообщение о необходимости выбора активного раздела для продолжения загрузки.



После того как активный раздел найден, главная загрузочная запись считывает самый первый сектор раздела в оперативную память. Этот сектор содержит загрузочную запись, которой главная загрузочная запись и передает управление. Загрузочная запись активного раздела выполняет загрузку операционной системы, находящейся в активном разделе.

Такой двухступенчатый метод загрузки операционной системы необходим по той причине, что способ загрузки зависит от самой операционной системы, поэтому каждая операционная система имеет свой собственный загрузчик. Фиксированным является только расположение загрузочной записи - самый первый сектор активного раздела.

Для жестких дисков типичной является ситуация, когда имеется четыре записи в таблице разделов и соответственно четыре раздела. DOS-ориентированные ОС используют только два из них, остальные резервируются на случай параллельного использования других операционных систем. Благодаря наличию MBR на одном физическом жестком носителе может располагаться несколько файловых систем различного типа различных операционных систем. Логические разделы тоже имеют некоторую иерархическую структуру в зависимости от типа и вида ОС и ее файловой системы. Активным может быть только один раздел.

Первый раздел жесткого диска называется главным разделом (**Primary Partition**), а второй расширенным (**Extended Partition**). Главный раздел всегда должен присутствовать на диске, с него происходит загрузка ОС. Расширенного же раздела может не быть, он создается лишь в том случае, когда необходимо получить более одного логического устройства на физическом диске. Логический раздел размещает в себе такие структуры файловой системы как логические диски или устройства, или тома (оформленные как подразделы), загрузчик операционной системы, таблицы распределения файлов, области пользовательских данных в которых размещаются записи о каталогах и файлах и данные файлов. По своей структуре логические подразделы или диски схожи с разделами. Основным отличием является то, что их число может быть более четырех, а последний элемент каждого показывает является ли он последним логическим подразделом раздела, или указывает на следующий элемент таблицы логических устройств или подразделов. Таблица подразделов строится только на расширенной таблице разделов, каждый ее элемент соответствует логическому устройству с односимвольным именем D:, E: и т.д.. Главная таблица разделов содержит только одно логическое устройство – диск C:. Таблица подразделов создается при создании расширенной таблицы разделов, а число элементов таблицы подразделов определяется пользователем. При определении числа логических устройств пользователь определяет и долю дискового пространства расширенного раздела, отводимую каждому логическому устройству – задает объем логических дисков. В дальнейшем, число и объем логических устройств не может быть изменено без потери данных, расположенных на перераспределяемых логических устройствах.

На основе разделов в MS-DOS и Windows ориентированных ОС создается дальнейшая структура. FAT находится вслед за загрузочным сектором, и в ней описано физическое размещение всех файлов на диске. За FAT следует ее копия. В таблице размещения файлов столько записей, сколько кластеров на диске.

Основной единицей хранения информации является кластер (cluster) - группа секторов. Весь диск разбивается операционной системой на участки одинакового размера, называемые кластерами. Размеры кластера всегда будут кратными степени числа 2 - 2, 4, 8, 16 и, наконец, 32 Кбайт (4, 8, 16, 32 и 64 сектора). Для каждого кластера FAT имеет свою индивидуальную ячейку, в которой хранится информация об использовании данного кластера. Другими словами, таблица размещения файлов - это массив, содержащий информацию о кластерах. Размер этого массива определяется общим количеством кластеров на логическом диске. Все свободные кластеры помечены в ней нулями. Если файл занимает несколько кластеров, то эти кластеры связаны в список. Для связанных в

список кластеров элементы таблицы FAT содержат номера следующих используемых данным файлом кластеров. Конец списка отмечен в таблице специальным значением. Номер первого кластера, распределенного файлу, хранится в элементе каталога, описывающего данный файл.

Итак, FAT - массив информации об использовании кластеров диска, содержит односвязные списки кластеров, распределенных файлам.

Существуют три формата FAT - 12-битовый, 16-битовый и 32-битовый. Эти форматы используют, соответственно, 12, 16 и 32 битов для хранения информации об одном кластере диска.

**12-битовый формат** удобен для дискет с небольшим количеством секторов - вся таблица размещения файлов помещается целиком в одном секторе.  $2^{12}=4096$  кластеров

При использовании **FAT 16-битового формата** объем дискового раздела в формате не может превышать 2 Гбайт.  $2^{16}=65536$  кластеров

**32-битовый формат** FAT описывает  $2^{32}= 4294967296$  кластеров. Использование этой системы теоретически позволяет работать с разделами от 512 Мбайт до 2 Тбайт. В отличие от **FAT16**, **FAT32** не имеет фиксированного размера корневого каталога, теперь там можно хранить сколько угодно файлов и каталогов.

Структура FAT. Первый байт FAT называется "**Описатель среды**" (Media Descriptor) или байт **ID** идентификации FAT. Следующие 5 байтов для 12-битового формата или 7 байтов для 16-битового формат всегда содержат значение **0ffh**.

Остальная часть FAT состоит из 12-битовых или 16-битовых ячеек, каждая ячейка соответствует одному кластеру диска. Эти ячейки могут содержать следующие значения:

FAT12	FAT16	Что означает
000h	0000h	Свободный кластер
ff0h - ff6h	fff0h - fff6h	Зарезервированный кластер
ff7h	fff7h	Плохой кластер
ff8h - fffh	fff8h - ffffh	Последний кластер в списке
002h - fefh	0002h - ffefh	Номер следующего кластера в списке

Последний кластер каждого файла используется не полностью. Неиспользованная часть кластера называется **хвостом**. Чем больше размер кластера, тем больше потерь из-за неполного их использования. Объем неиспользованной памяти можно оценить, если умножить половину размера кластера на диске на общее количество дисковых файлов.

Файловая система DOS имеет древовидную структуру. В корневом каталоге располагаются 32-байтовые элементы, которые содержат информацию о файлах и других каталогах. Корневой каталог располагается за второй копией FAT и состоит из 32-битных записей, каждая из которых содержит информацию о файле или подкаталоге. Корневой каталог занимает непрерывную область фиксированного размера. Размер корневого каталога задается при форматировании и определяет максимальное количество файлов и каталогов, которые могут быть описаны в корневом каталоге. Номер начального кластера в записи каталога является отправной точкой для отслеживания файла по цепочке номеров в ячейках FAT.

После корневого каталога на логическом диске находится область файлов и подкаталогов корневого каталога. Область данных разбита на кластеры, причем нумерация кластеров начинается с числа 2. Кластеру с номером 2 соответствуют первые сектора области данных. Любой каталог содержит 32-байтовые элементы - дескрипторы, описывающие файлы и другие каталоги. **Формат дескриптора:**



Размер	Содержимое
8	Имя файла или каталога, выровненное на левую границу и дополненное пробелами.
3	Расширение имени файла, выровненное на левую границу и дополненное пробелами.
1	Атрибуты файла.
10	Зарезервировано.
2	Время создания файла или время его последней модификации.
2	Дата создания файла или дата его последней модификации.
2	Номер первого кластера, распределенного файлу.
4	Размер файла в байтах.

Под значение размера файла в записи каталога отведено 4 байта. Следовательно, размер файла в MS-DOS 4 Гб.

В любом каталоге, кроме корневого, два первых дескриптора имеют специальное назначение. Первый дескриптор содержит в поле имени строку:

" . "

Этот дескриптор указывает на содержащий его каталог. Т.е. каталог имеет ссылку сам на себя.

Второй специальный дескриптор содержит в поле имени строку:

" . . "

Этот дескриптор указывает на каталог более высокого уровня.

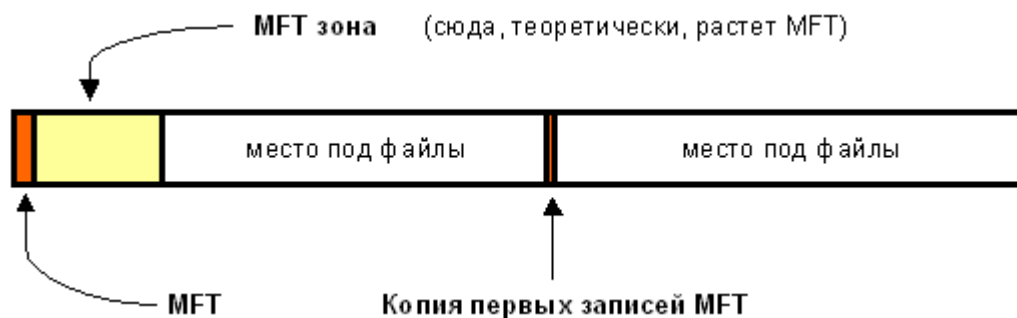
Таким образом, в древовидной структуре каталогов имеются ссылки как в прямом, так и в обратном направлении. Эти ссылки можно использовать для проверки сохранности структуры каталогов файловой системы.

Для других ОС, например, UNIX - использование разделов происходит иначе. Как правило, их может быть более четырех, все они равноправны и одинаково могут быть загрузочными, содержат собственные файловые системы на основе i-узлов. Такие файловые системы являются тегами и не имеют таблиц распределения порций информации. Дисковое пространство распределяется посекторно, что дает максимально возможное использование пространства раздела, но несколько снижает производительность. Весь раздел разбивается на иерархически связанную цепочку узлов разного уровня, которым соответствует некоторое количество секторов. На основе узлов строится понятие файлов и каталогов, и в таких системах файлы и каталоги действительно не различаются, т.к. каталог является файлом, содержащим структуру узлов. Один раздел отводится для дискового свопа и имеет упрощенную структуру, т.к. никогда не содержит файлов и каталогов.

Операционные системы Microsoft семейства Windows NT (2000, XP, 2003) и Vista, Windows 7/8/8.1/10 используют файловую систему NTFS (NT File System) – одну из самых сложных и удачных из существующих на данный момент файловых систем. Максимальный размер раздела NTFS в данный момент ограничен лишь размерами жестких дисков. NTFS делит все полезное место на кластеры, поддерживает почти любые размеры кластеров – от 512 байт до 64 Кбайт, неким стандартом же считается кластер размером 4 Кбайт. Диск NTFS условно делится на две части. Первые 12% диска отводятся под так называемую MFT зону – пространство, в которое растет метафайл MFT (об этом ниже). Запись каких-либо данных в эту область невозможна. MFT-зона всегда держится



пустой – это делается для того, чтобы самый главный, служебный файл (MFT) не фрагментировался при своем росте. Остальные 88% диска представляют собой обычное пространство для хранения файлов.



Файловая система NTFS представляет собой выдающееся достижение структуризации: каждый элемент системы представляет собой файл – даже служебная информация. Самый главный файл на NTFS называется MFT, или Master File Table – общая таблица файлов. Именно он размещается в MFT зоне и представляет собой централизованный каталог всех остальных файлов диска, и, как ни парадоксально, себя самого. MFT поделен на записи фиксированного размера (обычно 1 Кбайт), и каждая запись соответствует какому-либо файлу (в общем смысле этого слова). Первые 16 файлов несут служебный характер и недоступны операционной системе – они называются метафайлами, причем самый первый метафайл – сам MFT. Каждый из них отвечает за какой-либо аспект работы системы. Метафайлы находятся в корневом каталоге NTFS диска – они начинаются с символа имени "\$".

Имя файла в NTFS представляет собой: обязательный элемент - запись в MFT и опциональный элемент - потоки данных файла (streams). Каждый файл на NTFS, в общем-то, имеет несколько абстрактное строение - у него нет как таковых данных, а есть потоки. Один из потоков и носит привычный нам смысл - данные файла. Но большинство атрибутов файла - тоже потоки!

Каталог на NTFS представляет собой специфический файл, хранящий ссылки на другие файлы и каталоги, создавая иерархическое строение данных на диске. Файл каталога поделен на блоки, каждый из которых содержит имя файла, базовые атрибуты и ссылку на элемент MFT, который уже предоставляет полную информацию об элементе каталога. Внутренняя структура каталога представляет собой бинарное дерево.

В данной таблице сведены воедино все существенные плюсы и минусы распространенных в наше время систем, таких как FAT32, FAT и NTFS. В настоящее время большинство пользователей делают выбор между Windows XP и Windows Vista.

	FAT	FAT32	NTFS
Системы, её поддерживающие	DOS, Windows9X, NT всех версий	Windows98, NT всех версий	NT4, NT5, 2000, XP, 2003, Vista, Windows 7/8/8.1/10
Максимальный размер тома	2 Гбайт	практически неограничен	практически неограничен
Макс. число файлов на томе	примерно 65 тысяч	практически неограничено	практически неограничено
Имя файла	с поддержкой длинных имен - 255 символов, системный набор символов	с поддержкой длинных имен - 255 символов, системный набор	255 символов, любые символы любых алфавитов (65 тысяч разных начертаний)

		СИМВОЛОВ	
Возможные атрибуты файла	Базовый набор	Базовый набор	всё, что придет в голову производителям программного обеспечения
Безопасность	нет	нет	да (начиная с NT5.0 встроена возможность физически шифровать данные)
Сжатие	нет	нет	да
Квотирование	нет	нет	да
Устойчивость к сбоям	средняя (система слишком проста и поэтому ломаться особо нечему :))	плохая (средства оптимизации по скорости привели к появлению слабых по надежности мест)	полная - автоматическое восстановление системы при любых сбоях (не считая физические ошибки записи, когда пишется одно, а на самом деле записывается другое)
Экономичность	минимальная (огромные размеры кластеров на больших дисках)	улучшена за счет уменьшения размеров кластеров	максимальна. Очень эффективная и разнообразная система хранения данных
Быстродействие	высокое для малого числа файлов, но быстро уменьшается с появлением большого количества файлов в каталогах. результат - для слабо заполненных дисков - максимальное, для заполненных - плохое	полностью аналогично FAT, но на дисках большого размера (десятки гигабайт) начинаются серьезные проблемы с общей организацией данных	система не очень эффективна для малых и простых разделов (до 1 Гбайт), но работа с огромными массивами данных и внушительными каталогами организована как нельзя более эффективно и очень сильно превосходит по скорости другие системы

**Организацию файловой системы на CD-ROM описывает стандарт ISO-9660** (опубликован в 1988 г. под заголовком "Information processing - Volume and file structure of CD-ROM for information interchange" организацией International Organization for Standards - ISO), идентичный стандарту BS-7061 (опубликован в 1989г. British Standards Institution), и являющийся развитием более раннего де-факто стандарта HighSierra (называющегося в оригинале "28 May 1986 Working Paper for Information Processing - Volume and File Structure of Compact Read Only Optical Discs for Information Interchange", и являющегося некоей бумагой, присланной для обсуждения в ISO консорциумом производителей компьютерного оборудования).

Возможные символы в именах файлов по ISO9660:

ABCDEFGHIJKLMNOPQRSTUVWXYZ (ASCII HEX 41 to 5A)  
0123456789 (ASCII HEX 30 to 39)  
\_ (UNDERSCORE - ASCII HEX 5F)

. (FULL STOP - ASCII HEX 2E)  
; (SEMI COLON - ASCII HEX 3B)

Имена файлов могут иметь две части: NAME и EXTENSION. Имя должно содержать как минимум один символ, в поле NAME или EXTENSION. Поля NAME и EXTENSION разделяются символом FULL STOP, и допустим только один FULL STOP в имени файла. Также можно использовать в конце имени файла символ SEMI COLON, после которого должен идти номер в диапазоне от 1 до 32767, трактуемый как номер версии файла. SEMI COLON и номер версии обычно не показываются в списке директории, но могут использоваться для доступа к файлам с одинаковыми именами и разными версиями. Строго говоря, SEMI COLON и номер версии ОБЯЗАТЕЛЬНЫ по стандарту, но на практике многие ISO-9660 форматтеры не добавляют их к именам файлов.

Level 1 этого стандарта включает форматы файловых систем MS-DOS и HFS (Apple Macintosh), со следующими ограничениями на имена файлов:

- a. Поле NAME не может содержать более 8 символов
- b. Поле EXTENSION не может содержать более 30 символов
- c. В сумме оба поля не могут содержать более 30 символов (это число не включает символы FULL STOP, SEMI COLON и номер версии, так что полная длина имени файла может достигать 37 символов)

Level 2 описывает файловую систему с длинными именами, ограничения такие:

- a. Поле NAME не может содержать более 30 символов
- b. Поле EXTENSION не может содержать более 30 символов
- c. В сумме оба поля не могут содержать более 30 символов (это число не включает символы FULL STOP, SEMI COLON и номер версии, так что полная длина имени файла может достигать 37 символов)

Возможные символы в именах директорий по ISO9660:

ABCDEFGHIJKLMNOPQRSTUVWXYZ (ASCII HEX 41 to 5A)  
0123456789 (ASCII HEX 30 to 39)

Имена директорий могут содержать только поле NAME, и не могут быть длиннее 31 символа в Level 2, и не длиннее 8 символов в Level 1. Допустимый уровень вложенности директорий до 32 в Level 2 и до 8 в Level 1 (что кстати означает, что допустимы лишь 31 или 7 уровней поддиректорий в терминах MS-DOS).

## Технологии Windows. Обмен данными между приложениями

Существует два метода обмена данными между приложениями Windows:

- DDE (Dynamic Data Exchange) – динамический обмен данными;
- OLE (Object Linking and Embedding) – связывание и внедрение объектов.

### 1. DDE

Первая версия технологии обмена данными. Представляет собой протокол передачи сообщений и создает статическую связь. Отвечает за обмен информацией через буфер обмена (копировать, вырезать, вставить).

### 2. OLE (OLE1, OLE2)

Дальнейшее развитие технологии DDE. Предоставляет функции по связыванию и внедрению объектов.

Объект – данные, перемещаемые из одного приложения в другое с сохранением исходного формата.

OLE позволяет создать объект в одном из W-приложений и затем вставить его в другой файл. Объекты, создаваемые в разных приложениях, могут быть объединены в один составной документ, который сохраняет связи со всеми исходными приложениями.

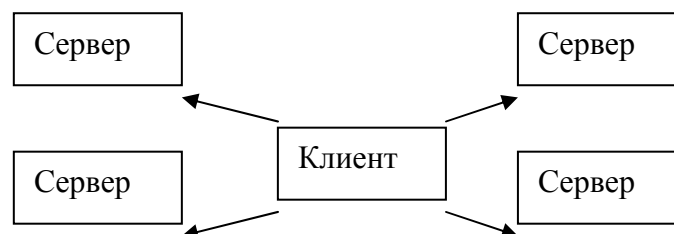
Составной документ – документ, содержащий один или несколько объектов.

OLE позволяет централизовать работу в пределах одного приложения и в одном документе, присоединяя и привязывая объекты из других приложений. Составной документ управляется каким-нибудь одним приложением, а связи обеспечивают пути к другим приложениям, где могут создаваться или редактироваться объекты.

Для обмена данными используется технология «клиент-сервер».

Клиент – приложение, в котором создан составной документ (документ-клиент), содержащий объекты.

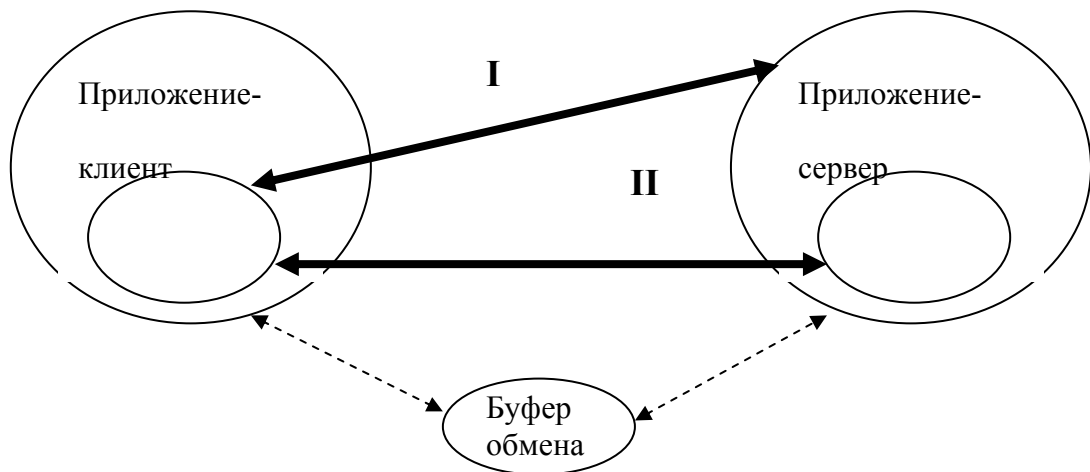
Сервер – приложение, которое клиент вызывает для работы с объектом.



Объекты бывают двух типов:

- Связанные – тогда они существуют в отдельных файлах;
- Внедренные – находятся внутри основного файла.

Соответственно существует два типа связей:



**I** – связь между документом-клиентом и приложением-сервером.

Обеспечивает документу-клиенту возможность знать, каким приложением был создан объект. Это – погруженный объект. Он содержит всю информацию из буфера обмена, которая использовалась для создания этого объекта.

Реализуется путем выделения объекта в приложении-сервере и выполнения команд из меню **Правка: Копировать** или **Вырезать**. Данные помещаются в буфер обмена, при этом сохраняется код, который действует как указатель к приложению-серверу. В приложении-клиенте выполняется команда из меню **Правка: Вставить**

Погруженный объект содержит связь с приложением-сервером, которая позволяет запустить его и отредактировать объект (двойным щелчком). При этом открывается приложение-сервер, обслуживающее данный объект.

**II** – связь между документом-клиентом и документом-сервером.

При существовании этой связи изменение данных в исходном документе (сервере) автоматически отражается на объекте документа-клиента. Это - связанные объекты.

Связанный объект хранит ссылки на приложение-сервер, документ сервер и связанный объект в документе-сервере.

Приложение-сервер модернизирует документ-клиент всякий раз, когда изменяется информация в документе-сервере.

Реализуется путем помещения данных из приложения-сервера в буфер обмена, к которым подсоединяется маркерный файл – link. В приложении-клиенте из меню правка выбирается команда **Специальная вставка** (или **Связать с ...**). Она вызывает библиотеки OLE, которые ищут файл link. При его обнаружении вызываются специальные библиотеки OLE, создающие связь с приложением-сервером.

Основные библиотеки OLE – в каталоге System основного каталога Windows:

OLE2.DLL – поддержка основных функций.

OLE32.DLL – поддержка 32-разрядных приложений.

OLECLI.DLL – библиотека клиента; создание, воспроизведение, загрузка и сохранение объектов клиента.

OLESRV.DLL – библиотека сервера.

OLE2CONV.DLL – преобразование объекта в формат программы-клиента.

OLE2DISP.DLL – отображение объекта в документе-клиенте.

SHELL.DLL – регистрация W-приложения, информация о поддержке OLE.

DDEML.DDL – поддержка DDE.

Усовершенствования OLE – OLE2

- 1) **Наглядное редактирование.** При выполнении двойного щелчка на объекте, объект редактируется на месте его расположения в документе-клиенте. Панель инструментов и структура меню при этом заменяются на присущие приложению-серверу. Для пользователя это выглядит как смена инструментов, а не смена приложения.
- 2) **Технология drag & drop** (перетаскивание). Новый способ установления связей – объект захватывается мышью и перетаскивается в нужное место, приложение и др.
- 3) **Независимость связей от места расположения объекта** (в случае, если объект расположен не на локальном диске).
- 4) **Адаптируемые связи, т.е. при перемещении** (изменении местоположения) **файла составного документа связи сохраняются.**
- 5) **Оптимизированное хранение объектов** – связанный документ хранится на диске до тех пор, пока не будет вызван в приложении-клиенте. Т.о. не загружаются в память сразу все приложения и данные, входящие в составной документ.
- 6) **Преобразование объектов.** Если приложение-сервер отсутствует на данном ПК, то можно воспользоваться командой Преобразовать из контекстного меню объекта для вызова сходного по функциям приложения.
- 7) **Автоматизация OLE.** Использование языка VBA (Visual Basic Application), который определяет стандартный интерфейс для общения с приложением-сервером.