Уровни организации ЭВМ. Операционные системы

Лекция №4

План

- Понятие ОС
- Эволюция ОС
- Классификация ОС
- Архитектура ОС
- Управление памятью
- Управление процессами
- Системный реестр Windows
- Загрузка ОС Windows
- Жесткий диск
- Файловая система

Операционная система как виртуальная машина

Операционная система – совокупность программ, которая скрывает от пользователя все реалии аппаратуры и предоставляет возможность простого, удобного просмотра указанных файлов, чтения или записи.

Операционная система как виртуальная машина

ОС выполняет обработку прерываний, управление таймерами и оперативной памятью, а также другие низкоуровневые проблемы.

С этой точки зрения функцией ОС является предоставление пользователю некоторой расширенной или **виртуальной машины**, которую легче программировать и с которой легче работать, чем непосредственно с аппаратурой, составляющей реальную машину

ОС как система управления ресурсами

Операционная система – совокупность программ, **управляющая** всеми ресурсами вычислительной машины таким образом, чтобы обеспечить максимальную эффективность ее функционирования.

ОС как система управления ресурсами

Управление ресурсами: 2 задачи

планирование ресурса

определение, кому, когда, а для делимых ресурсов и в каком количестве, необходимо выделить данный ресурс

отслеживание состояния ресурса

поддержание оперативной информации о том, занят или не занят ресурс, а для делимых ресурсов - какое количество ресурса уже распределено, а какое свободно

Эволюция ОС. Первый период (1945 -1955)

В середине 40-х были созданы первые ламповые вычислительные устройства. В то время одна и та же группа людей участвовала и в проектировании, и в эксплуатации, и в программировании вычислительной машины.

Программирование осуществлялось исключительно на машинном языке.

Об ОС не было и речи, все задачи организации вычислительного процесса решались вручную каждым программистом с пульта управления. Не было никакого другого системного программного обеспечения, кроме библиотек математических и служебных подпрограмм.

Эволюция ОС. Второй период (1955 -1965)

С середины 50-х годов начался новый период в развитии вычислительной техники, связанный с появлением новой технической базы - полупроводниковых элементов.

В эти годы появились первые алгоритмические языки, а следовательно и первые системные программы - компиляторы.

Появились первые **системы пакетной обработки**, которые просто автоматизировали запуск одной программ за другой и тем самым увеличивали коэффициент загрузки процессора.

Системы пакетной обработки явились **прообразом современных операционных систем**. В этом режиме пользователь не имеет контакта с ЭВМ, получая лишь результаты вычислений.

Эволюция ОС. Третий период (1965 -1980)

В это время в технической базе произошел переход от отдельных полупроводниковых элементов типа транзисторов к малым интегральным микросхемам.

Важнейшим достижением ОС (**OS/360**) данного поколения явилась реализация мультипрограммирования.

Мультипрограммирование - это способ организации вычислительного процесса, при котором на одном процессоре попеременно выполняются несколько программ.

Пока одна программа выполняет операцию ввода-вывода, процессор не простаивает, как это происходило при последовательном выполнении программ (однопрограммный режим), а выполняет другую программу (многопрограммный режим). При этом каждая программа загружается в свой участок оперативной памяти, называемый разделом.

Эволюция ОС. Третий период (1965 -1980)

Наряду с мультипрограммной реализацией систем пакетной обработки появился новый тип ОС - **системы разделения времени**.

Вариант мультипрограммирования, применяемый в системах разделения времени, нацелен на создание для каждого отдельного пользователя иллюзии единоличного использования вычислительной машины.

Эволюция ОС.
4 период (1980-по наст. время)

Следующий период в эволюции операционных систем связан с появлением больших интегральных схем.

На рынке операционных систем доминировали две системы: **MS-DOS** и **UNIX**.

Однопрограммная однопользовательская ОС MS-DOS широко использовалась для компьютеров, построенных на базе микропроцессоров Intel 8088, а затем 80286, 80386 и 80486.

Мультипрограммная многопользовательская ОС UNIX доминировала в среде "не-интеловских" компьютеров, особенно построенных на базе высокопроизводительных RISC-процессоров.

Эволюция ОС.
4 период (1980-по наст.время)

В середине 80-х стали бурно развиваться сети персональных компьютеров, работающие под управлением **сетевых** или **распределенных ОС**.

В сетевых ОС пользователи должны быть осведомлены о наличии других компьютеров и должны делать логический вход в другой компьютер, чтобы воспользоваться его ресурсами, преимущественно файлами.

Сетевая ОС не имеет фундаментальных отличий от ОС однопроцессорного компьютера. Она обязательно содержит программную поддержку для сетевых интерфейсных устройств (драйвер сетевого адаптера), а также средства для удаленного входа в другие компьютеры сети и средства доступа к удаленным файлам, однако эти дополнения существенно не меняют структуру самой операционной системы.

Классификация ОС

От эффективности алгоритмов управления локальными ресурсами компьютера во многом зависит эффективность всей сетевой ОС в целом.

Классификация ОС

Особенности использованного алгоритма управления процессором

- **многозадачные и однозадачные**
- **многопользовательские и однопользовательские**
- **многопроцессорные и однопроцессорные**

Поддержка многозадачности

Число одновременно выполняемых задач

однозадачные
(MS-DOS, MSX, Apple DOS)

1. предоставляют пользователю виртуальную машину,
2. более простой и удобный процесс взаимодействия пользователя с ПК,
3. включают средства управления периферийными устройствами,
4. средства управления файлами,
5. средства общения с пользователем.

многозадачные
(ОС ЕС, OS/2, UNIX, Windows, Mac OS)

управляют разделением совместно используемых ресурсов: процессора, оперативной памяти, файлов и внешних устройств.

Поддержка многопользовательского режима

число одновременно работающих пользователей

Однопользовательские

- MS-DOS
- Windows 3.x
- ранние версии OS/2

многопользовательские
UNIX, Windows, Mac OS

средства защиты информации каждого пользователя от несанкционированного доступа других пользователей.

Многопользовательская система

Однозадачная система

НЕ ВСЕГДА

Многозадачная система

Однопользовательская система

Вытесняющая и невытесняющая многозадачность

Важнейшим разделяемым ресурсом является процессорное время.

Способ распределения процессорного времени между несколькими одновременно существующими в системе процессами (или нитями) во многом определяет специфику ОС.

Вытесняющая и невытесняющая многозадачность

Многозадачность

невытесняющая
(NetWare, Windows 3.x)

активный процесс выполняется до тех пор, пока он сам не отдаст управление ОС, чтобы та выбрала из очереди другой готовый к выполнению процесс.

вытесняющая
(Windows NT, OS/2, UNIX)

решение о переключении процессора с одного процесса на другой принимается ОС, а не активным процессом.

Многопроцессорная обработка

В наши дни становится общепринятым введение в ОС функций поддержки **многопроцессорной обработки данных**.

Примеры подобных ОС:

- Solaris фирмы Sun
- Open Server компании Santa Crus Operations
- OS/2 фирмы IBM
- Windows фирмы Microsoft
- NetWare фирмы Novell

Многопроцессорная обработка

Способ организации вычислительного процесса в системе с многопроцессорной архитектурой

Асимметричная ОС

целиком выполняется только на одном из процессоров системы, распределяя прикладные задачи по остальным процессорам.

Симметричная ОС

полностью децентрализована и использует весь пул процессоров, разделяя их между системными и прикладными задачами.

Классификация ОС

Системы пакетной обработки

Особенности областей использования

Системы разделения времени

Системы реального времени

Системы пакетной обработки

Системы пакетной обработки предназначались для решения задач в основном вычислительного характера, не требующих быстрого получения результатов. В системах пакетной обработки переключение процессора с выполнения одной задачи на выполнение другой происходит только в случае, если активная задача сама отказывается от процессора, например, из-за необходимости выполнить операцию ввода-вывода. Поэтому одна задача может надолго занять процессор, что делает невозможным выполнение интерактивных задач.

Таким образом, взаимодействие пользователя с вычислительной машиной, на которой установлена система пакетной обработки, сводится к тому, что он приносит задание, отдает его диспетчеру-оператору, а в конце дня после выполнения всего пакета заданий получает результат. Очевидно, что такой порядок снижает эффективность работы пользователя.

Системы разделения времени

Системы разделения времени призваны исправить основной недостаток систем пакетной обработки - изоляцию пользователя-программиста от процесса выполнения его задач.

Каждому пользователю системы разделения времени предоставляется терминал, с которого он может вести диалог со своей программой. Так как в системах разделения времени каждой задаче выделяется только квант процессорного времени, ни одна задача не занимает процессор надолго, и время ответа оказывается приемлемым. Если квант выбран достаточно небольшим, то у всех пользователей, одновременно работающих на одной и той же машине, складывается впечатление, что каждый из них единолично использует машину.

Системы реального времени

Системы реального времени применяются для управления техническими объектами, такими, например, как станок, спутник, научная экспериментальная установка или технологическими процессами, такими, как гальваническая линия, доменный процесс и т. п. Во всех этих случаях существует предельно допустимое время, в течение которого должна быть выполнена та или иная программа, управляющая объектом, в противном случае может произойти авария: спутник выйдет из зоны видимости, экспериментальные данные, поступающие с датчиков, будут потеряны, толщина гальванического покрытия не будет соответствовать норме.

Критерием эффективности для систем реального времени является их способность выдерживать заранее заданные интервалы времени между запуском программы и получением результата (управляющего воздействия). Это время называется **временем реакции системы**, а соответствующее свойство системы - **реактивностью**.

Архитектура операционной системы

Архитектура ОС – структурная организация ОС на основе программных модулей.

Модуль – программная единица, которая имеет вполне законченное функциональное назначение.

Современные ОС представляют собой структурированные модульные системы, способные к развитию и расширению.

Архитектура операционной системы

Модули ОС

ЯДРО

Модули, выполняющие вспомогательные функции

утилиты, библиотеки процедур различного назначения

Монолитное ядро

Монолитное ядро – схема ОС, при которой все ее компоненты являются составными частями одной программы.

ЯДРО = ОС

Так как ядро является единой программой, перекомпиляция – это единственный способ добавить в него новые компоненты или исключить неиспользуемые.

Монолитное ядро

Присутствие в ядре лишних компонентов крайне нежелательно, так как ядро всегда полностью располагается в оперативной памяти.

Монолитное ядро – старейший способ организации операционных систем.

Примером систем с монолитным ядром является большинство Unix-систем.

Микроядерная архитектура

Современная тенденция в разработке ОС: перенесение значительной части системного кода в большое количество самостоятельных программам.

Взаимодействие между ними обеспечивает специальный модуль ядра, называемый **микроядром**.

Микроядро работает в привилегированном режиме и обеспечивает взаимодействие между программами, планирование использования процессора, операции ввода-вывода и базовое управление памятью.

Микроядерная архитектура

Микроядерная архитектура

Основное достоинство:

1. Высокая степень модульности ядра операционной системы.
2. Упрощенное добавление в ядро новых компонентов.
3. Возможность, не прерывая работы ОС, загружать и выгружать новые драйверы, файловые системы и т. д.
4. Повышенная надежность системы, т.к. ошибка на уровне непривилегированной программы менее опасна, чем отказ на уровне режима ядра.

Структура сетевой ОС

Сетевая ОС

В широком смысле

совокупность ОС отдельных компьютеров, взаимодействующих с целью обмена сообщениями и разделения ресурсов по единым правилам - протоколам.

В узком смысле:

ОС отдельного компьютера, обеспечивающая ему возможность работать в сети.

Структура сетевой ОС

Управление памятью

Память – важнейший ресурс, требующий тщательного управления со стороны мультипрограммной ОС.

Распределению подлежит вся оперативная память, не занятая ОС.

Управление памятью

Функции ОС по управлению памятью:

- отслеживание свободной и занятой памяти,
- выделение памяти процессам и освобождение памяти при завершении процессов,
- вытеснение процессов из оперативной памяти на диск, когда размеры основной памяти не достаточны для размещения в ней всех процессов, и возвращение их в оперативную память, когда в ней освобождается место, а также настройка адресов программы на конкретную область физической памяти.

Методы распределения памяти с использованием дискового пространства. Понятие виртуальной памяти

Проблема: как разместить в памяти программу, размер которой превышает имеющуюся в наличии свободную память?

Виртуальная память - совокупность программно-аппаратных средств, позволяющих пользователям писать программы, размер которых превосходит имеющуюся оперативную память;

Методы распределения памяти с использованием дискового пространства. Понятие виртуальной памяти

Виртуальная память решает задачи:

- размещения данных в ЗУ разного типа, например, часть программы в оперативной памяти, а часть на диске;
- перемещения по мере необходимости данных между ЗУ разного типа, например, подгружает нужную часть программы с диска в оперативную память;
- преобразования виртуальных адресов в физические.

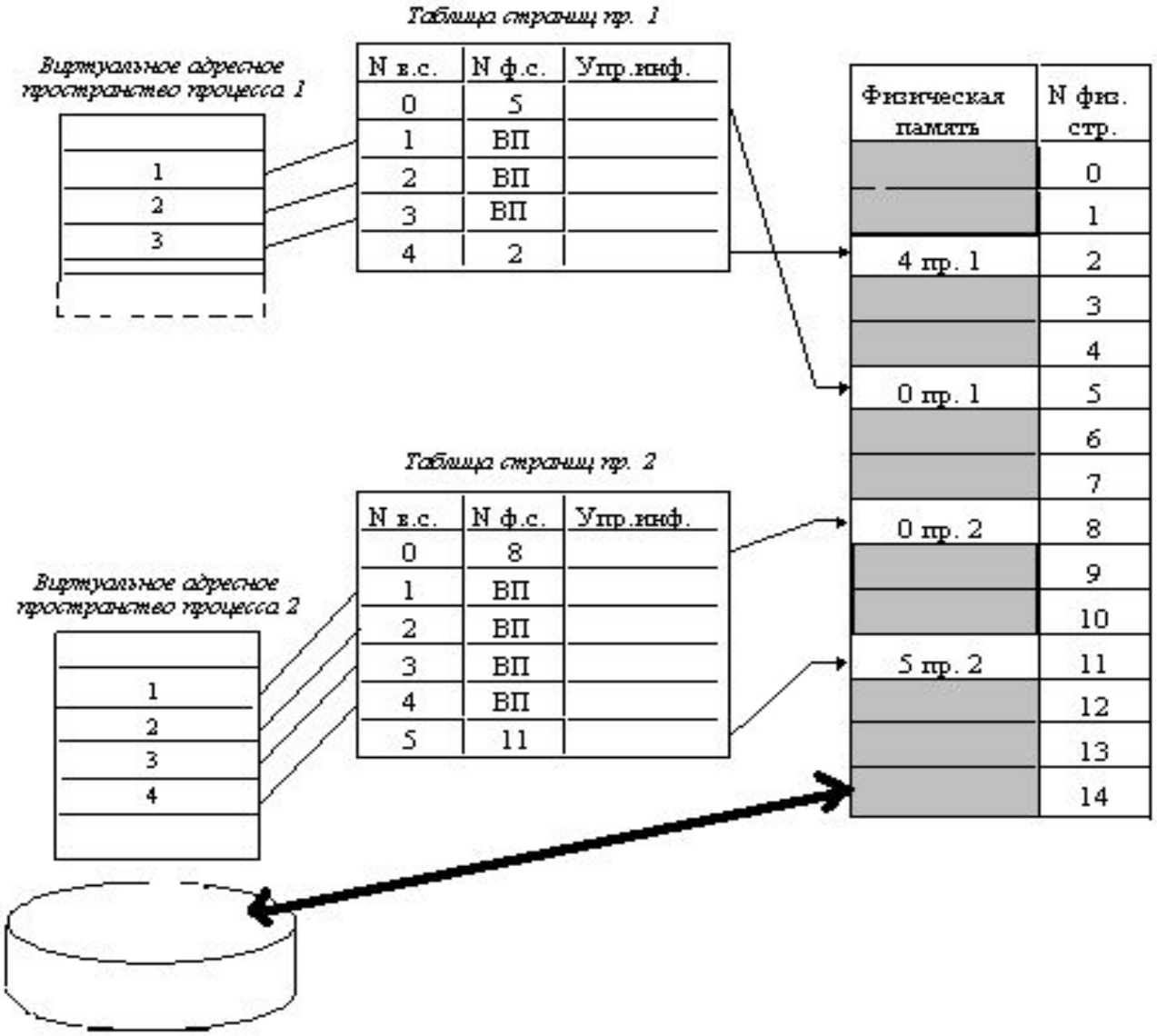
Методы распределения памяти с использованием дискового пространства. Понятие виртуальной памяти

Действия выполняются **автоматически**, без участия программиста.

Наиболее распространенные реализации виртуальной памяти:

1. Страничное
2. Сегментное
3. Странично-сегментное распределение памяти
4. Свопинг

Страничное распределение (1)



Страничное распределение (2)

Виртуальное адресное пространство каждого процесса делится на части одинакового, фиксированного для данной системы размера, называемые виртуальными страницами. В общем случае размер виртуального адресного пространства не является кратным размеру страницы, поэтому последняя страница каждого процесса дополняется фиктивной областью.

Вся оперативная память машины также делится на части такого же размера, называемые физическими страницами (или блоками).

Размер страницы обычно выбирается равным степени двойки: 512, 1024 и т.д., это позволяет упростить механизм преобразования адресов.

Страничное распределение (3)

При загрузке ОС создает для каждого процесса информационную структуру - таблицу страниц, в которой устанавливается соответствие между номерами виртуальных и физических страниц для страниц, загруженных в оперативную память, или делается отметка о том, что виртуальная страница выгружена на диск.

При загрузке процесса часть его виртуальных страниц помещается в оперативную память, а остальные - на диск.

Страничное распределение (4)

При каждом обращении к памяти происходит чтение из таблицы страниц информации о виртуальной странице, к которой произошло обращение.

Если данная виртуальная страница находится в оперативной памяти, то выполняется преобразование виртуального адреса в физический.

Если же нужная виртуальная страница в данный момент выгружена на диск, то происходит так называемое страничное прерывание. Выполняющийся процесс переводится в состояние ожидания, и активизируется другой процесс из очереди готовых. Параллельно программа обработки страничного прерывания находит на диске требуемую виртуальную страницу и пытается загрузить ее в оперативную память. Если в памяти имеется свободная физическая страница, то загрузка выполняется немедленно, если же свободных страниц нет, то решается вопрос, какую страницу следует выгрузить из оперативной памяти.

Страничное распределение (5)

В данной ситуации может быть использовано много разных критериев выбора, наиболее популярные из них следующие:

- дольше всего не использовавшаяся страница
- первая попавшаяся страница
- страница, к которой в последнее время было меньше всего обращений

Сегментное распределение

Организация памяти

Страничная

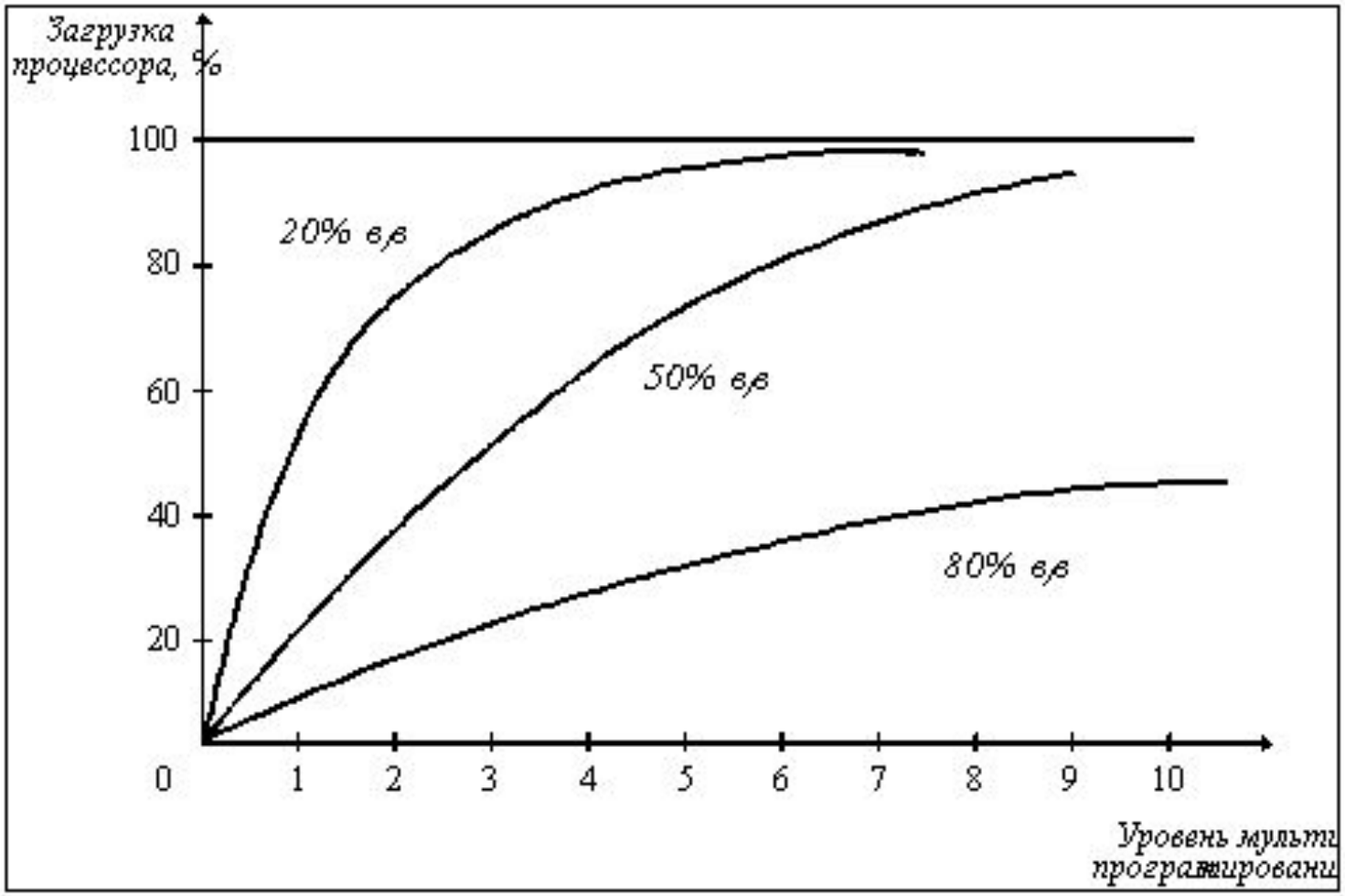
виртуальное адресное пространство процесса делится механически на равные части

Сегментная

виртуальное адресное пространство процесса делится на сегменты произвольной длины

Свопинг

Свопинг – разновидность виртуальной памяти.



Свопинг

Из рисунка видно, что для загрузки процессора на 90% достаточно всего трех счетных задач. Однако для того, чтобы обеспечить такую же загрузку интерактивными задачами, выполняющими интенсивный ввод-вывод, потребуются десятки таких задач. Необходимым условием для выполнения задачи является загрузка ее в оперативную память, объем которой ограничен. В этих условиях был предложен метод организации вычислительного процесса, называемый ***свопингом***.

При **свопинге**, в отличие от рассмотренных ранее методов реализации виртуальной памяти, процесс перемещается между памятью и диском целиком, то есть в течение некоторого времени процесс может полностью отсутствовать в оперативной памяти. Существуют различные алгоритмы выбора процессов на загрузку и выгрузку, а также различные способы выделения оперативной и дисковой памяти загружаемому процессу.

Управление процессами

Процесс – это программный код, загруженный в оперативную память с возможностью потребления всех видов ресурсов (кроме процессора).

Управление процессами ОС – распределение процессорного времени между несколькими одновременно существующими в системе процессами, создание и уничтожение процессов, обеспечение процессов необходимыми системными ресурсами, поддержка взаимодействия между процессами.

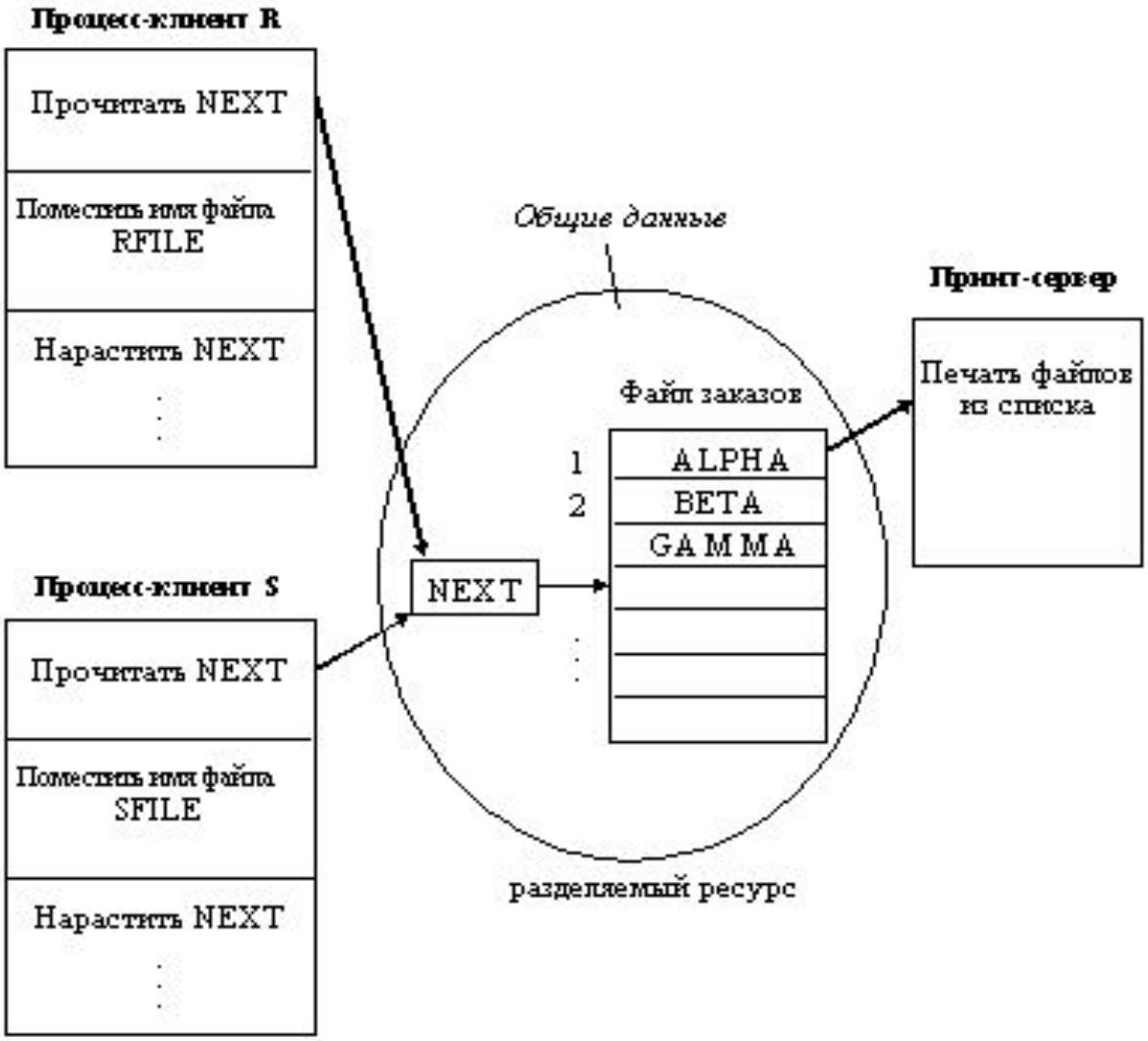
Проблема синхронизации при управлении процессами

Процессам часто нужно взаимодействовать друг с другом, например, один процесс может передавать данные другому процессу, или несколько процессов могут обрабатывать данные из общего файла.

Во всех этих случаях возникает **проблема синхронизации процессов**, которая может решаться приостановкой и активизацией процессов, организацией очередей, блокированием и освобождением ресурсов.

Пренебрежение вопросами синхронизации процессов, выполняющихся в режиме мультипрограммирования, может привести к их неправильной работе или даже к краху системы.

Проблема синхронизации при управлении процессами



Проблема синхронизации при управлении процессами

Рассмотрим программу печати файлов (принт-сервер). Эта программа печатает по очереди все файлы, имена которых последовательно в порядке поступления записывают в специальный общедоступный файл "заказов" другие программы. Особая переменная NEXT, также доступная всем процессам-клиентам, содержит номер первой свободной для записи имени файла позиции файла "заказов". Процессы-клиенты читают эту переменную, записывают в соответствующую позицию файла "заказов" имя своего файла и наращивают значение NEXT на единицу.

Проблема синхронизации при управлении процессами

Предположим, что в некоторый момент процесс R решил распечатать свой файл, для этого он прочитал значение переменной NEXT, значение которой для определенности предположим равным 4. Процесс запомнил это значение, но поместить имя файла не успел, так как его выполнение было прервано (например, в следствие исчерпания кванта). Очередной процесс S, желающий распечатать файл, прочитал то же самое значение переменной NEXT, поместил в четвертую позицию имя своего файла и нарастил значение переменной на единицу. Когда в очередной раз управление будет передано процессу R, то он, продолжая свое выполнение, в полном соответствии со значением текущей свободной позиции, полученным во время предыдущей итерации, запишет имя файла также в позицию 4, поверх имени файла процесса S.

Таким образом, ***процесс S никогда не увидит свой файл распечатанным***.

Системный реестр Windows

Реестр - база данных операционной системы, в которой хранятся настройки аппаратных и программных средств ОС.

По замыслу Microsoft реестр должен был полностью заменить файлы **ini**, которые были оставлены только для совместимости со старыми программами, ориентированными на более ранние версии операционной системы.

Почему произошел переход от ini файлов к реестру? Дело в том, что на эти файлы накладывается ряд серьезных ограничений, и главное из них состоит в том, что предельный размер такого файла составляет 64Кб.

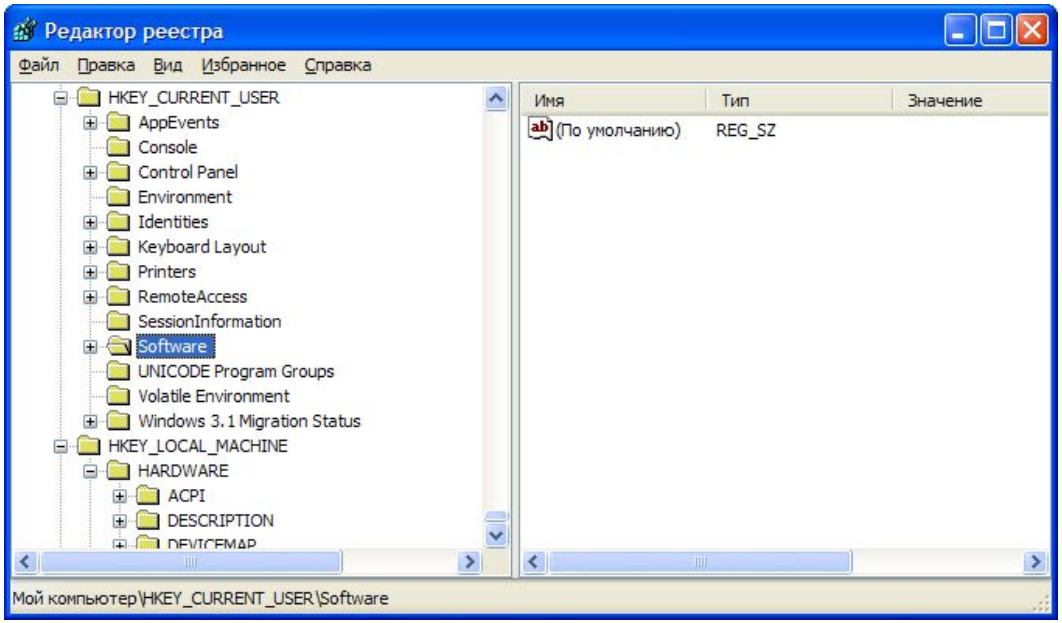
Системный реестр Windows

ПРЕДУПРЕЖДЕНИЕ:

НИКОГДА не удаляйте или не меняйте информацию в реестре, если Вы не уверены что это именно то, что нужно. В противном случае некорректное изменение данных может привести к сбоям в работе Windows и, в лучшем случае, информацию придется восстанавливать из резервной копии

Системный реестр Windows

Основным средством для просмотра и редактирования записей реестра служит специализированная утилита **"Редактор реестра".** Для ее запуска наберите в запуске программы (Пуск->Выполнить) команду **regedit**



Системный реестр Windows

Реестр содержит шесть **корневых разделов** (ветвей), каждый из разделов включает подразделы, отображаемые в левой части окна в виде значка папки.

Конечным элементом дерева реестра являются **ключи** или параметры, делящиеся на три типа:

- строковые (напр. "C:\Windows");
- двоичные (напр. 10 82 A0 8F). Максимальная длина такого ключа 16Кб;
- DWORD. Этот тип ключа занимает 4 байта и отображается в шестнадцатеричном и в десятичном виде (напр. 0x00000020 (32) - в скобках указано десятичное значение ключа).

Системный реестр Windows

Корневые разделы реестра:

HKEY_CLASSES_ROOT. В этом разделе содержится информация о зарегистрированных в Windows типах файлов, что позволяет открывать их по двойному щелчку мыши, а также информация для OLE и операций drag-and-drop.

HKEY_CURRENT_USER. Здесь содержатся настройки оболочки пользователя (например, Рабочего стола, меню "Пуск", ...), вошедшего в Windows.

HKEY_LOCAL_MACHINE. Этот раздел содержит информацию, относящуюся к компьютеру: драйверы, установленное программное обеспечение и его настройки

HKEY_USERS. Содержит настройки оболочки Windows для всех пользователей.

HKEY_CURRENT_CONFIG. В этом разделе содержится информация о конфигурации устройств Plug&Play.

HKEY_DYN_DATA. Здесь хранятся динамические данные о состоянии различных устройств, установленных на компьютере пользователя. Именно сведения этой ветви отображаются в окне "Свойства: Система" на вкладке "Устройства", вызываемого из Панели управления. Данные этого раздела изменяются самой операционной системой, так что редактировать что-либо вручную не рекомендуется.

Загрузка ОС Windows

Большинство компьютерных систем могут исполнять только команды, находящиеся в оперативной памяти компьютера, в то время как современные операционные системы в большинстве случаев хранятся на жёстких дисках, загрузочных CDROM-ах, USB дисках или в локальной сети.

Загрузка ОС Windows

После включения компьютера в его оперативной памяти нет операционной системы. Само по себе, без операционной системы, аппаратное обеспечение компьютера не может выполнять сложные действия, такие как, например, загрузку программы в память. Таким образом мы сталкиваемся с парадоксом, который кажется неразрешимым: для того, чтобы загрузить операционную систему в память, мы уже должны иметь операционную систему в памяти.

Загрузка ОС Windows

Решением данного парадокса является использование специальной маленькой компьютерной программы, называемой **начальным загрузчиком**, или **BIOS**. Эта программа не обладает всей функциональностью операционной системы, но ее достаточно для того, чтобы загрузить другую программу, которая будет загружать операционную систему.

Загрузка ОС Windows
1 этап

1. После включения персонального компьютера его процессор начинает работу. Первая выполняемая команда расположена по адресу FFFF0h и принадлежит пространству адресов BIOS. Как правило, данная команда просто передает управление программе инициализации BIOS.

Загрузка ОС Windows
2 этап

2. Программа инициализации BIOS с помощью программы POST проверяет, что устройства компьютера работают корректно и инициализирует их.

Загрузка ОС Windows
3 этап

3. Затем BIOS опрашивает устройства, перечисляемые в заранее созданном списке, пока не найдет загрузочное устройство. Если такое устройство найдено не будет, будет выведено сообщение об ошибке, а процесс загрузки будет остановлен. Если BIOS обнаружит загрузочное устройство, он считает с него начальный загрузчик и передаст ему управление.

Загрузка ОС Windows
4 этап

4. В случае жесткого диска, начальный загрузчик называется **главной загрузочной записью** (Master Boot Record - MBR) и часто не зависит от операционной системы. Обычно он ищет активный разделы жесткого диска, загружает **загрузочный сектор** данного раздела и передает ему управление.

Загрузка ОС Windows
5 этап

5. Программный код из загрузочного сектора анализирует файл boot.ini на наличие установленных ОС, в случае наличия двух и более ОС предлагает выбрать, какую загрузить, иначе приступает к загрузке ОС.

Загрузка ОС Windows
6 этап

6. После выбора ОС загрузчик считывает из CMOS системную дату и время, определяет тип компьютера, системной шины, клавиатуры, видеоадаптеров и т.д. Список всего найденного оборудования размещается в разделах реестра **HKEY_LOCAL_MACHINE/HARDWARE.**

   Процедура поиска оборудования выполняется в момент, когда на экране появляется надпись «Запуск Windows».

Загрузка ОС Windows
7 этап

7. Начинается загрузка ОС: в оперативную память помещается ядро, базовые драйвера устройств. Базовые драйвера устройств указываются в ключе реестра **HKEY_LOCAL_MACHINE/SYSTEM.**

Загрузка ОС Windows
8 этап

8. Выдается приглашение на вход в систему, загружаются сетевые драйвера. Пользователь может подключиться к системе.

Устройство жесткого диска

Треки и сектора жесткого диска

Пластины в жестких дисках имеют специальную структуру, для обеспечения упорядоченной записи и хранения информации.

Каждая пластина разбита на **треки**, которые представляют собой концентрические окружности. На одной пластине помещаются десятки тысяч треков.

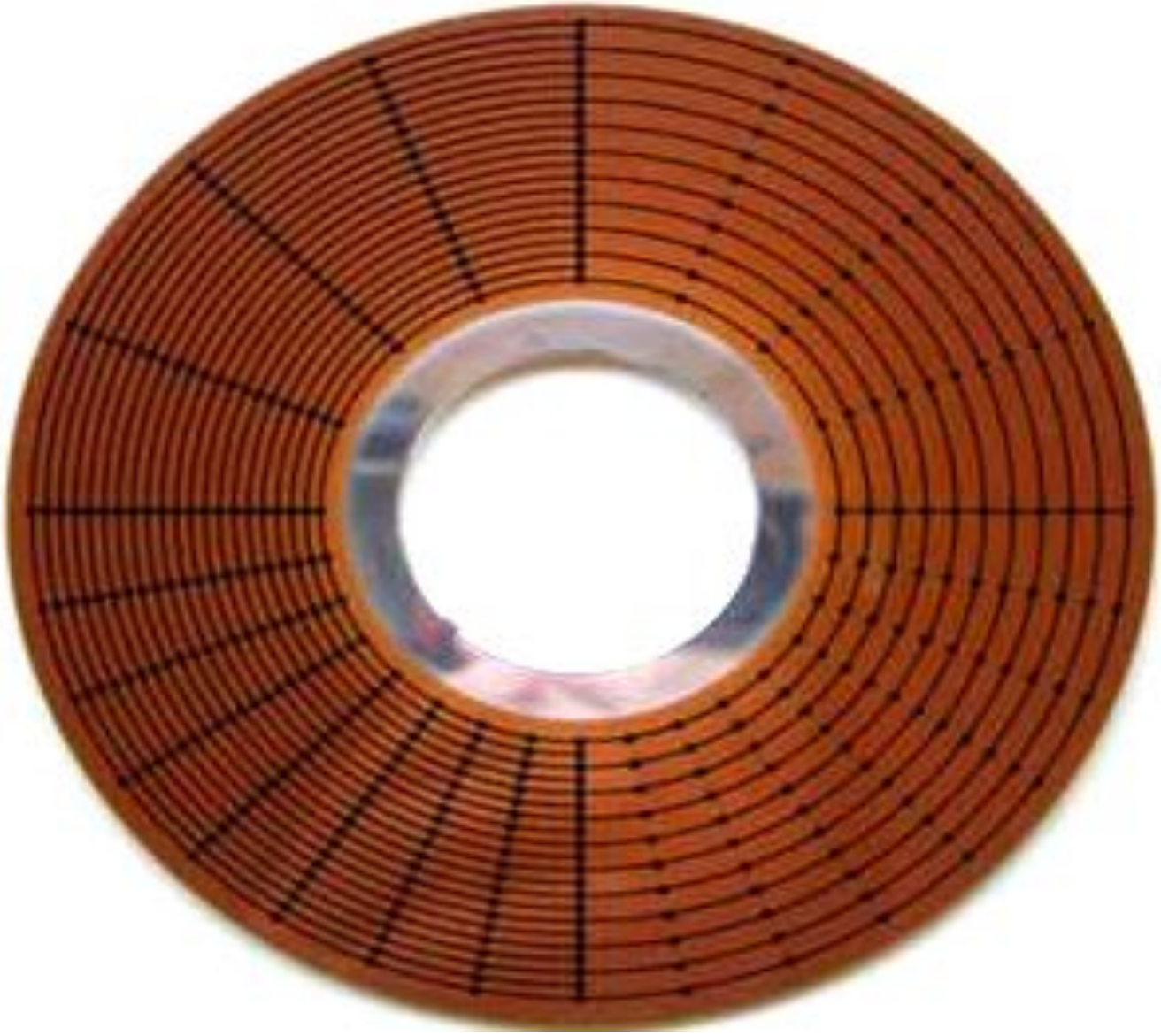
Треки и сектора жесткого диска

На треке помещается слишком много информации, если использовать трек как меньшую единицу информации. Поэтому треки разбиваются на **сектора**.

Сектор хранит 512 байт информации.

Жесткий диск способен выполнять лишь команды чтения/записи в определенный сектор на диске.

Треки и сектора жесткого диска



Логическая структура жесткого диска

Раздел (англ. partition) — часть жёсткого диска, выделенная для удобства работы.

Выделение на одном жёстком диске нескольких разделов даёт следующие преимущества:

- на одном физическом жёстком диске можно хранить информацию в разных файловых системах;
- можно отделить информацию пользователя от файлов операционной системы (для безопасности последней);
- на одном жёстком диске можно установить несколько операционных систем;
- форматирование и дефрагментация каждого раздела не затрагивает другие.

Логическая структура жесткого диска

<table>
<tr><td rowspan="3">Физический диск</td><td colspan="2">Первичный раздел 1 (Логический раздел 1): ФС</td></tr>
<tr><td rowspan="2">Расширенный раздел (Первичный раздел 2, Логический раздел 2)</td><td>Логический раздел 4: ФС</td></tr>
<tr><td>Логический раздел 5: ФС</td></tr>
<tr><td colspan="2">Первичный раздел 3 (Логический раздел 3): ФС</td></tr>
</table>

Файловые системы

В отличие от жесткого диска файловая система позволяет пользователю оперировать с более удобным для него понятием - файл.

Файловая система – часть ОС, определяющая способ организации, хранения и именования данных на носителях информации.

Файловая система определяет формат физического хранения информации, которую принято группировать в виде **файлов**.

Задачи файловой системы

- Определение размера имени файла, набора атрибутов файла, максимального возможного размера файла.
- Отображение логической модели файловой системы на физическую организацию хранилища данных.
- Поддержка устойчивости файловой системы к сбоям питания, ошибкам аппаратных и программных средств.

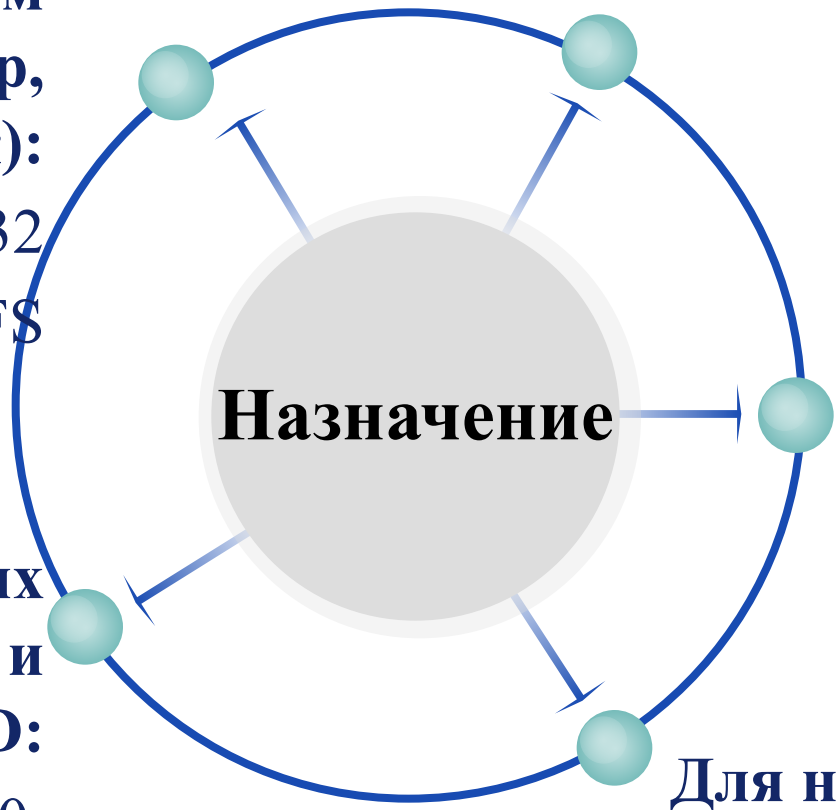
Задачи файловой системы

- Содержание параметров файла необходимых для правильного его взаимодействия с другими объектами системы (ядро, приложения и пр.).
- В многопользовательских системах появляется еще одна задача: защита файлов одного пользователя от несанкционированного доступа другого пользователя.

Классификация файловых систем

Для носителей с произвольным доступом (например, жёсткий диск):
FAT, VFAT, FAT32
NTFS, ReFS

Виртуальные файловые системы: AEFS



Сетевые файловые системы:
NFS, SMBFS, SSHFS

Для оптических носителей CD и DVD:
ISO9660, ISO9690, HFS, UDF

Для носителей с последовательным доступом (например, магнитные ленты):
QIC

FAT

File Allocation Table

Разработана Биллом Гейтсом и Марком МакДональдом в 1977 году.

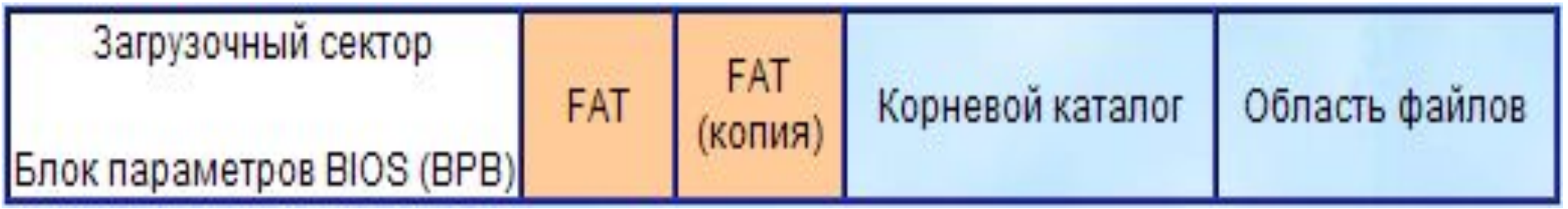
Поддерживает файлы и разделы размером до 2 Гбайт.

FAT

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 FAT



В блоке параметров BIOS содержится необходимая ему информация о физических характеристиках жесткого диска.

Файловая система FAT не может контролировать отдельно каждый сектор, поэтому она объединяет смежные сектора в кластеры (clusters).

Структура FAT

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.

Недостатки 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.

Недостатки FAT

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.

Недостатки FAT

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

FAT не предотвращает порчи файлов из-за ненормального завершения работы компьютера.

VFAT

Virtual FAT, реализованная в Windows NT 3.5, Windows 95 - это FAT, включающая поддержку длинных имен файлов в кодировке UNICODE.

VFAT использует ту же самую схему распределения дискового пространства, что и файловая система FAT.

VFAT

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

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

VFAT

Основной задачей при разработке VFAT была необходимость корректной работы старых программ, не поддерживающих длинные имена файлов.

В итоге для каждого файла и подкаталога в VFAT хранится два имени: длинное и короткое в формате 8.3 для совместимости со старыми программами.

FAT 32

FAT32 - усовершенствованная версия VFAT, поддерживающая жесткие диски объемом до 2 терабайт.

В FAT32 были расширены атрибуты файлов, позволяющие теперь хранить время и дату создания, модификации и последнего доступа к файлу или каталогу.

FAT 32

Из-за требования совместимости с ранее созданными программами структура FAT32 содержит минимальные изменения. Блок начальной загрузки на разделах с FAT32 был увеличен до 2 секторов и включает в себя резервную копию загрузочного сектора, что позволяет системе быть более устойчивой к возможным сбоям на диске.

FAT 32

Корневой каталог в FAT32 больше не располагается в определенном месте, вместо этого в блоке BPB хранится указатель на начальный кластер корневого каталога. В результате снимается ранее существовавшее ограничение на число записей в корневом каталоге.

FAT 32

Для учета свободных кластеров, в зарезервированной области на разделе FAT32 имеется сектор, содержащий число свободных кластеров и номер самого последнего использованного кластера. Это позволяет системе при выделении следующего кластера не перечитывать заново всю таблицу размещения файла.

FAT 32

В данный момент FAT32 поддерживается в следующих ОС:

- Windows 95 OSR2
- Windows 98
- Windows ME
- Windows 2000
- Windows XP

NTFS

NTFS (New Technology File System) - наиболее предпочтительная файловая система при работе с ОС Windows NT (Windows 2000 и XP также являются NT системами).

Структура раздела NTFS

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. Первые 16 файлов (метафайлы) носят служебный характер и недоступны операционной системе. Эти первые 16 элементов MFT - единственная часть диска, имеющая фиксированное положение. Интересно, что вторая копия первых трех записей, для надежности хранится ровно посередине диска.

Структура раздела NTFS

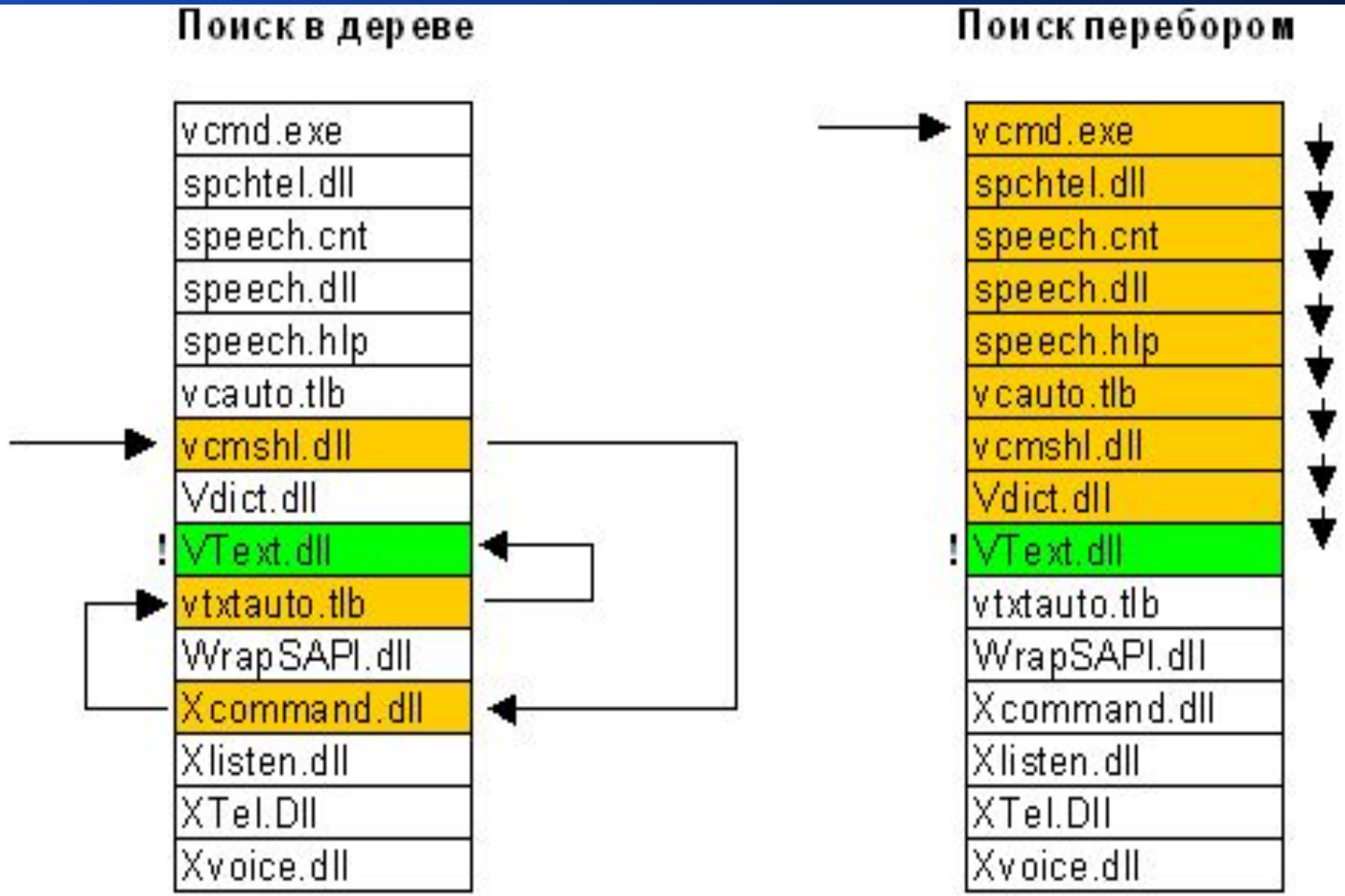
Каталоги NTFS

Каталог на NTFS представляет собой специфический файл, хранящий ссылки на другие файлы и каталоги, создавая иерархическое строение данных на диске.

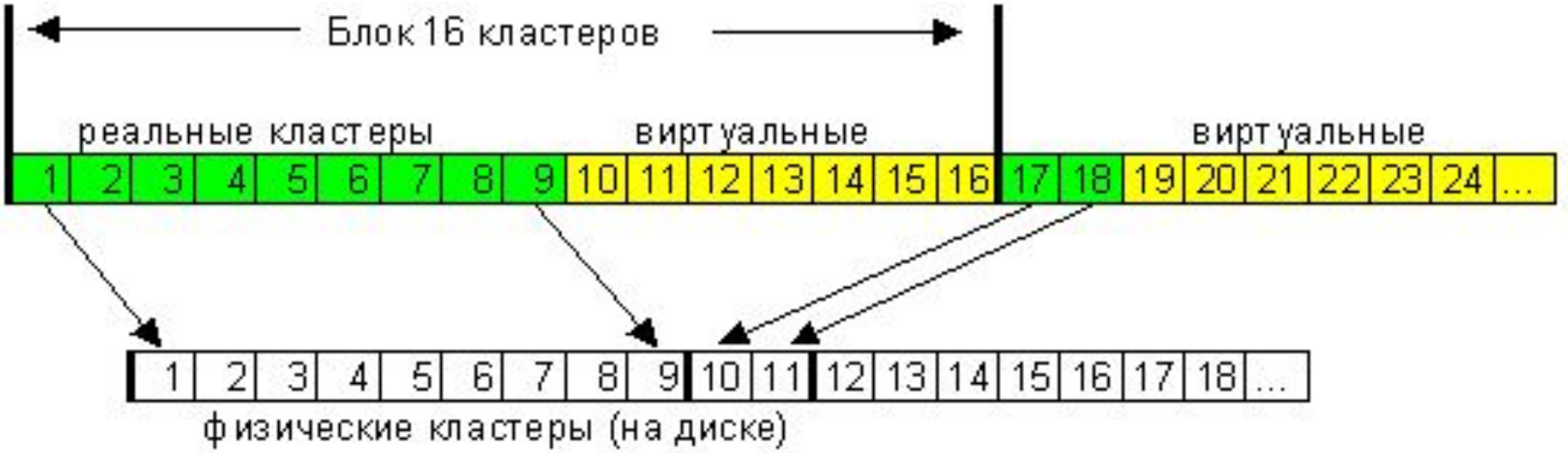
Файл каталога поделен на блоки, каждый из которых содержит имя файла, базовые атрибуты и ссылку на элемент MFT, который уже предоставляет полную информацию об элементе каталога.

Внутренняя структура каталога представляет собой бинарное дерево. Вот что это означает: для поиска файла с данным именем в линейном каталоге, таком, например, как у FAT, ОС приходится просматривать все элементы каталога, пока она не найдет нужный. Бинарное же дерево располагает имена файлов таким образом, чтобы поиск файла осуществлялся более быстрым способом - с помощью получения двухзначных ответов на вопросы о положении файла.

Каталоги NTFS



Сжатие в NTFS



NTFS имеет встроенную поддержку сжатия дисков.

Файлы NTFS имеют один довольно полезный атрибут - "сжатый".

Журналирование NTFS

NTFS - отказоустойчивая система, которая вполне может привести себя в корректное состояние при практически любых реальных сбоях. Любая современная файловая система основана на таком понятии, как транзакция - действие, совершаемое целиком и корректно или не совершаемое вообще.

Журналирование NTFS

Пример 1:

Осуществляется запись данных на диск. Вдруг выясняется, что в то место, куда мы только что решили записать очередную порцию данных, писать не удалось - физическое повреждение поверхности. Поведение NTFS в этом случае довольно логично: транзакция записи откатывается целиком - система осознает, что запись не произведена. Место помечается как сбойное, а данные записываются в другое место - начинается новая транзакция.

Журналирование NTFS

Пример 2:

Более сложный случай - идет запись данных на диск. Вдруг отключается питание и система перезагружается. На какой фазе остановилась запись? На помощь приходит журнал транзакций. Дело в том, что система, осознав свое желание писать на диск, пометила в метафайле $LogFile это свое состояние. При перезагрузке это файл изучается на предмет наличия незавершенных транзакций, которые были прерваны аварией и результат которых непредсказуем - все эти транзакции отменяются: место, в которое осуществлялась запись, помечается снова как свободное, элементы MFT приводятся в с состояние, в котором они были до сбоя, и система в целом остается стабильна.

exFAT

exFAT (от англ. Extended FAT — «расширенная FAT»), иногда называется FAT64 — проприетарная файловая система, предназначенная главным образом для флэш-накопителей. Впервые представлена фирмой Microsoft для встроенных устройств в Windows Embedded CE 6.0. Размер кластера по умолчанию для файловой системы exFAT составляет от 4 КБ до 256 КБ в зависимости от размера тома

ReFS

ReFS (Resilient file system, дословно — устройчивая файловая система) — новая ФС, изначально появилась в Windows Server 2012, а начиная с Windows 10 Creators Update в нее можно отформатировать любой несистемный накопитель.

Одна из главных особенностей файловой системы REFS — защита от потери данных: по умолчанию, на дисках хранятся контрольные суммы для метаданных или файлов. При операциях чтения-записи данные файлов сверяются с хранимыми для них контрольными суммами, таким образом, в случае повреждения данных есть возможность сразу «обратить на это внимание».

ReFS

Плюсов у ReFS в сравнении с NTFS много:

- Максимальная длина пути к файлу 32768 символов (в NTFS — 255).
- Максимальный размер тома 262144 экзабайта (в NTFS — 16).
- Более быстрый поиск по системе, а также лучшая надежность благодаря использованию B+-деревьев для хранения данных.
- Отсутствие поддержки имен файлов DOS (с тильдой ~, были в NTFS для совместимости со старыми программами, сейчас это не актуально)
- Улучшенная защита от потери данных

Форматы файловой системы, доступные в Mac

Apple File System (APFS)

Apple File System (APFS) — стандартная файловая система для компьютеров Mac с SSD-дисками, которая обеспечивает надежное шифрование, совместное использование пространства, получение моментальных снимков, быстрое изменение размеров каталогов и улучшенные принципы файловой системы.

Mac OS Extended, MS-DOS (FAT) и ExFAT

Можно также отформатировать тома как Mac OS Extended (эта файловая система поставляется с macOS 10.12 и более ранними версиями), MS-DOS (FAT) или ExFAT.

Форматы файловой системы, доступные в Linux

Файловые системы в Linux используются не только для работы с файлами на диске, но и для хранения данных в оперативной памяти или доступа к конфигурации ядра во время работы системы.

Ext2, Ext3, Ext4 или **Extended Filesystem** - это стандартная файловая система для Linux. Она была разработана еще для Minix. Она самая стабильная из всех существующих, кодовая база изменяется очень редко и эта файловая система содержит больше всего функций. Версия ext2 была разработана уже именно для Linux и получила много улучшений.

Общие выводы по файловым системам

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.

Общие выводы по файловым системам

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 почти 40 летнюю историю файловая система прошла путь от простой системы, взявшей на себя функции управления файлами, до системы, представляющей собой полноценную СУБД, обладающую встроенным механизмом транзакций и восстановления данных.

Вопросы для самостоятельного изучения

1. Функции ОС
2. Сетевые файловые системы
3. Файловые системы для оптических носителей
4. Виртуальные файловые системы