




ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ОС

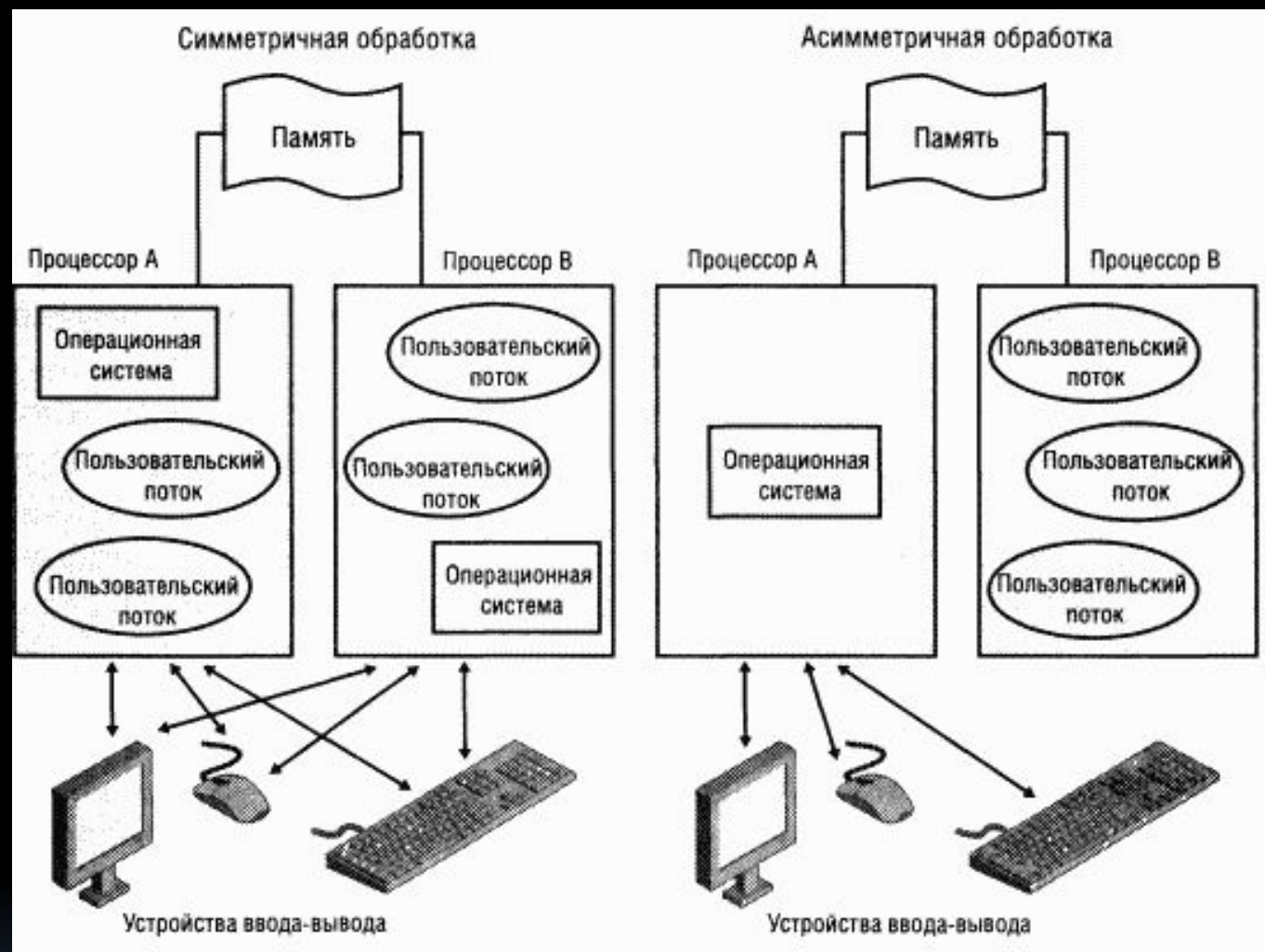


КЛАССИФИКАЦИЯ ОПЕРАЦИОННЫХ СИСТЕМ

1. Назначение (универсальные, специализированные – управление производством, обучение)
 2. Способ загрузки (загружаемые, постоянно находящиеся в памяти)
 3. Особенности алгоритмов управления ресурсами
 - 3.1. Многозадачность:
однозадачные (MS DOS),
невывесняющая многозадачность (Windows 3.x, NewWare),
вывесняющая многозадачность (Windows NT, OS/2, Unix)
- 

3.2. Многопользовательский режим:
отсутствие (MS DOS, Windows 3.x),
имеется (Windows NT, OS/2, Unix)

3.3. Многопроцессорная обработка:
отсутствие, асимметричные ОС,
симметричные ОС



4. По базовой технологии (Юникс-подобные или подобные Windows)

5. По типу лицензии (проприетарная или открытая)

6. Аппаратная платформа

6.1. ОС для смарт-карт (с интерпретатором виртуальной Java-машины)

6.2. Встроенные ОС (Palm OS, Windows CE –Consumer Electronics)

6.3. ОС для ПК (Windows 9.x, Windows 2000, Linux, Mac OS X)

6.4. ОС мэйнфреймов (мэйнфрейм, от англ. mainframe) — большой универсальный высокопроизводительный отказоустойчивый сервер со значительными ресурсами ввода-вывода, большим объёмом оперативной и внешней памяти, предназначенный для использования в критически важных системах (англ. mission-critical) с интенсивной пакетной и оперативной транзакционной обработкой. Основной разработчик мэйнфреймов — корпорация [IBM](#))

6.5. Серверные операционные системы для ЛВС, Интранет и Интернет (UNIX, AIX, Windows 2000/2002, Linux)

6.6. Кластерные операционные системы (Windows 2000 Cluster Server, Sun Cluster (Solaris))

7. Критерии эффективности ОС

Наиболее характерными критериями эффективности вычислительных систем являются:

- *пропускная способность* – количество задач, выполняемых вычислительной системой в единицу времени;
- *удобство работы пользователей*, заключающееся, в частности, в том, что они имеют возможность интерактивно работать одновременно с несколькими приложениями на одной машине;
- *реактивность системы* – способность системы выдерживать заранее заданные (возможно, очень короткие) интервалы времени между запуском программы и получением результата.

7.1 Системы пакетной обработки

- Главной целью ОС пакетной обработки является максимальная пропускная способность или решение максимального числа задач в единицу времени. Эти системы обеспечивают высокую производительность при обработке больших объемов информации, но снижают эффективность работы пользователя в интерактивном режиме. Пакеты могут автоматически запускаться в запланированное время, а также выполняться в зависимости от доступности ресурсов компьютера.
- Современные пакетные приложения используют современные пакетные фреймворки, такие как Jem The Bee, Spring Batch, написанные для Java, и другие фреймворки для других языков программирования, чтобы обеспечить отказоустойчивость и масштабируемость, необходимые для обработки больших объемов. Применяются в big data

Основные методы группировки информации по разным пакетам:

- По времени создания. Например, в пакет поступают все файлы, поступившие на сервер компании за последние 30 минут (или все сигналы с сенсоров в турбине самолета за последние четыре полета).
- По типу данных
- По источнику. Записи о перемещениях грузов на складе - в один пакет, записи о перемещении грузов внутри магазина — в другой. Каждая строка в этих данных может выглядеть одинаково и представлять собой запись из базы данных, но источники происхождения у них разные.

Обработка данных в пакетном режиме означает, что все файлы пакета будут обработаны определенным способом:

- Последовательно. В таком случае обработка выглядит как конвейерная лента, по которой перемещается пакет записей. На каждом этапе к данным применяется какая-то одна логическая операция: фильтрация или трансформация
- Параллельно. На вход системы идут несколько пакетов, все они проходят через стадии обработки сразу на нескольких компьютерах (серверах) или несколькими копиями программ.

7.2 Системы с разделением времени

- В системах с разделением времени для выполнения каждой задачи выделяется небольшой промежуток времени, и ни одна задача не занимает процессор надолго. Если этот промежуток времени выбран минимальным, то создается видимость одновременного выполнения нескольких задач. Эти системы обладают меньшей пропускной способностью, но обеспечивают высокую эффективность работы пользователя в интерактивном режиме.

7.2 Системы реального времени

Операционная система реального времени, ОС РВ (англ. Real-Time Operating System) — тип операционной системы, как правило, специального назначения.

- **Стандарт POSIX 1003.1 , определение: «Реальное время в операционных системах — это способность операционной системы обеспечить требуемый уровень сервиса в определённый промежуток времени»**
- **ОС, реагирующая в предсказуемое время на непредсказуемое появление внешних событий**




Для *систем реального времени* характерно следующее:


- гарантированное время реакции на внешние события (например на прерывания от оборудования);
- жёсткая подсистема планирования процессов (высокоприоритетные задачи не должны вытесняться низкоприоритетными);
- повышенные требования к времени реакции на внешние события или реактивности (задержка вызова обработчика прерывания не более десятков микросекунд, задержка при переключении задач не более сотен микросекунд)

- Примером такого процесса является срабатывание стандартной подушки безопасности в современных автомобилях. В какой-то момент датчик сообщает о столкновении и отправляет прерывание центральному процессору, и тогда ОС должна быстро выделить задачу, которая управляет работой подушки безопасности, причем ее выполнению не должны мешать другие задачи, которые не связаны с требованиями реального времени. Если подушка безопасности срабатывает на секунду позднее, чем это требуется, то лучше совсем отказаться от ее использования.





Для обеспечения стабильной работы системы необходимо, чтобы определенная задача выполнялась через некоторый период времени (p).



Например, рассмотрим работу антиблокировочной системы (АБС) в автомобиле. Эта система опрашивает скорость каждого колеса автомобиля и контролирует давление в тормозном устройстве каждого колеса (для предотвращения его блокировки) с частотой до 100 раз в секунду. Для обеспечения нормальной работы системы опрос датчика и управление должно выполняться через периодические интервалы времени. Это означает, что через определенный период времени выполнение других задач прерывается и выполняется задача, связанная с работой АБС.



Планирование периодических задач

Windows Embedded — это семейство операционных систем реального времени, было специально разработано для применения в различных встраиваемых системах. Ядро системы имеет общее с семейством ОС Windows CE и поддерживает процессоры ARM, MIPS, SuperH и x86.

Windows Embedded включает дополнительные функции по встраиванию, среди которых фильтр защиты от записи (EWF и FBWF), загрузка с флеш-памяти, CD-ROM, сети, использование собственной оболочки системы и т. п.

ОС этого семейства продаются только в составе готовых устройств, таких как: банкоматы, медицинские приборы, навигационное оборудование, «тонкие» клиенты, VoIP-терминалы, медиапроигрыватели, цифровые рамки (альбомы), кассовые терминалы, платёжные терминалы, роботы, игровые автоматы, музыкальные автоматы и другие.

Виды ОС РВ

Динамические свойства программ реального времени принято характеризовать тремя определениями: программы «жесткого» (hard), «мягкого» (soft) и интерактивного («условного») реального времени.

- **Жесткое реальное время.** Предусматривает наличие гарантированного времени отклика системы на конкретное событие, например, аппаратное прерывание, выдачу команды управления и т.п. Абсолютная величина времени отклика большого значения не имеет. Так, если необходимо, чтобы программа отработала некоторую команду за 1 миллисекунду, но она справляется с этим заданием лишь в 95% случаев, а в 5% не укладывается в норматив, такую систему нельзя охарактеризовать как работающую в жестком реальном времени. Если же команду нужно отработать в течение часа, что и

- **Мягкое реальное время.** В этом случае ожидающееся время отклика системы является величиной скорее индикативной, нежели директивной. Конечно, предполагается что в большинстве случаев (процентов 80 — 90) отклик уложится в заданные пределы. Однако и остальные варианты – в том числе полное отсутствие реакции системы – не должны приводить к плачевным результатам. Обычно считается, что если временной норматив превышен на один порядок, то это еще терпимо .
- **Интерактивное реальное время.** Является скорее психологической, нежели технической характеристикой. Определяет время, в течение которого оператор – человек – способен спокойно, без нервозности, ожидать реакции системы на данные им указания. В качестве примера можно привести весьма популярные сегодня игры из категории «стратегии реального времени» (real-time strategy, см. например квазар на основе Warhammer).

Системные вызовы в ОС

- В любой *операционной системе* поддерживается механизм, который позволяет пользовательским программам обращаться к услугам ядра ОС.
- В большинстве *операционных систем* системный вызов осуществляется командой программного *прерывания* (INT). Программное *прерывание* – это синхронное событие, которое может быть повторено при выполнении одного и того же программного кода.



Механизм прерываний


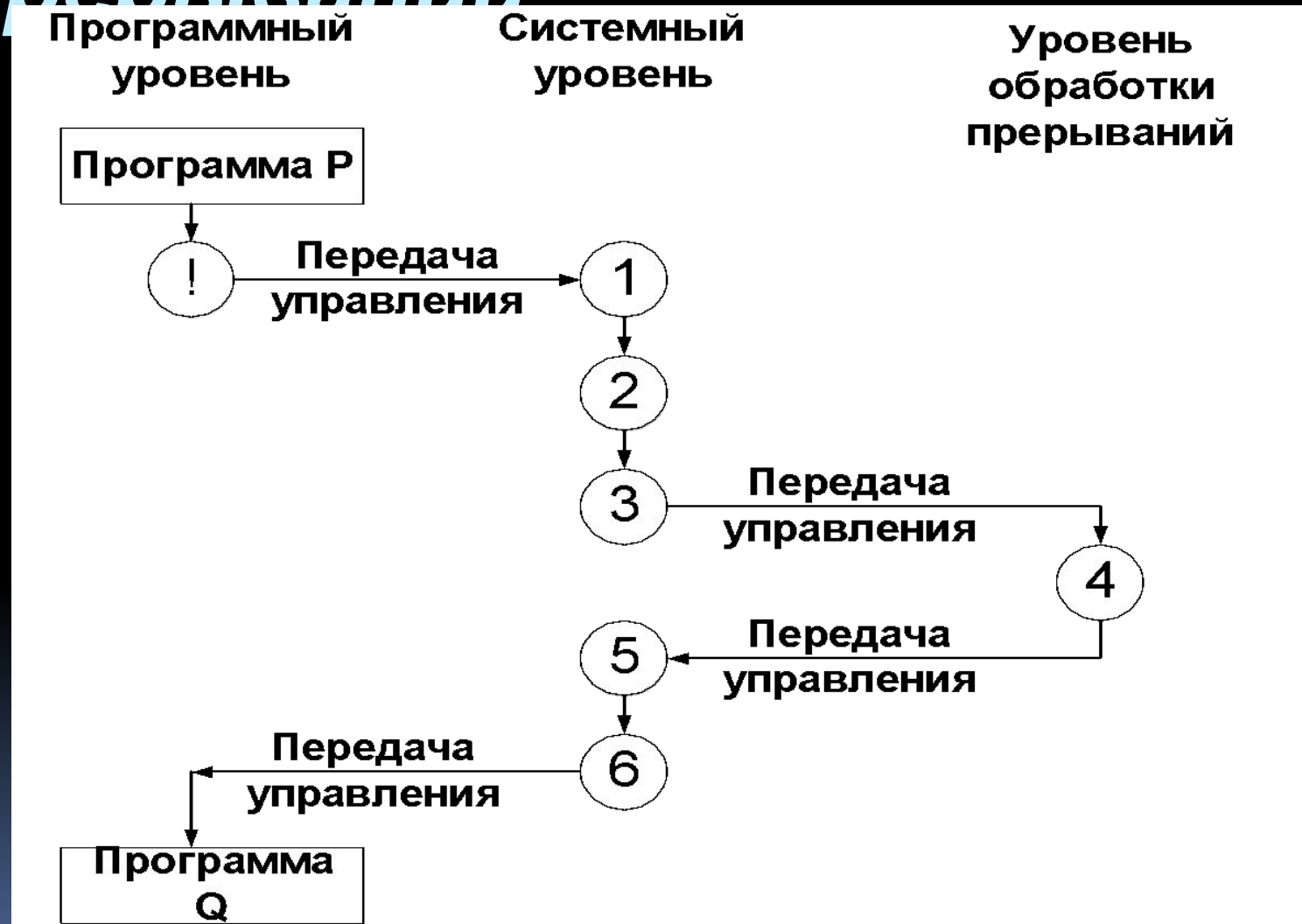
- *Прерывание* принудительная передача управления от выполняемой программы к системе, происходящая при возникновении определенного события.
 - Термин «ловушка» (англ. trap) иногда используется как синоним термина «прерывание» или «внутреннее прерывание».
- 

Схема обработки прерываний





Обозначения:


- ! Прерывание (сигнал – установление факта прерывания)
- 1 Идентификация прерывания
- 2 Отключение всех других прерываний
- 3 Смена контекста_1 (сохранение состояния прерванного процесса из системных регистров, загрузка в системные регистры контекста соответствующего обработчика прерываний)
- 4 Обработка прерывания, включающая определение программы Q, которую следует запустить;
- 5 Смена контекста_2 (загрузка в системные регистры контекста определенной на предыдущем шаге программы Q)
- 6 Установка прежнего режима системы прерываний.

Назначение и типы прерываний

В зависимости от источника прерывания делятся на три больших класса:


- внешние;
- внутренние;
- программные.
- *Внешние* прерывания могут возникать в результате действий пользователя или оператора за терминалом, или же в результате поступления сигналов от аппаратных устройств — сигналов завершения операций ввода-вывода, вырабатываемых контроллерами внешних устройств компьютера. Внешние прерывания называют также *аппаратными*.
- Данный класс прерываний является *асинхронным* по отношению к потоку инструкций прерываемой программы.

- 
- *Внутренние прерывания, называемые также исключениями (exception), происходят синхронно выполнению программы при появлении аварийной ситуации в ходе исполнения некоторой инструкции программы.*
 - *Примерами исключений являются деление на нуль, ошибки защиты памяти, обращения по несуществующему адресу, попытка выполнить привилегированную инструкцию в пользовательском режиме и т. п.*
- 







Программные (частный случай внутреннего прерывания) по своей сути не являются «истинными» .прерываниями

Программное прерывание инициируется выполнением особой команды процессора. Программные прерывания, как правило, используются для обращения к функциям встроенного программного обеспечения .(firmware), драйверов и операционной системы

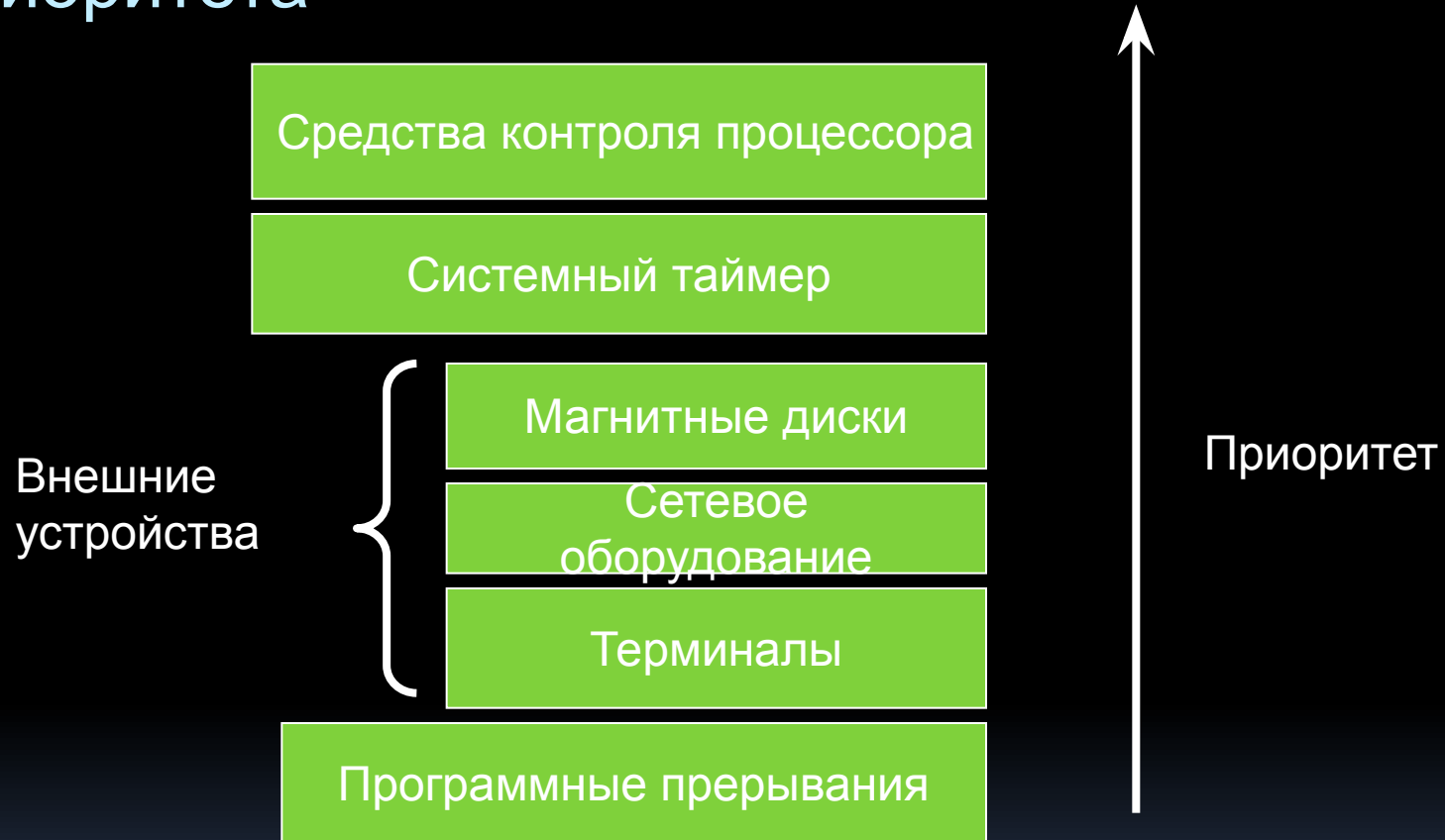




В процессорах архитектуры x86 для явного вызова синхронного прерывания имеется инструкция `Int`, аргументом которой является номер прерывания (от 0 до 255). В IBM PC-совместимых компьютерах обработку некоторых прерываний осуществляют подпрограммы BIOS, хранящиеся в ПЗУ, и прерывание служит интерфейсом для доступа к сервису, .предоставляемому BIOS



- 
- Прерываниям присписывается приоритет, с помощью которого они ранжируются по степени важности и срочности. О прерываниях, имеющих одинаковое значение приоритета, говорят, что они относятся к одному *уровню приоритета* прерываний.
- 


- 
- 
- Процедуры, вызываемые по прерываниям, обычно называют *обработчиками прерываний*, или *процедурами обслуживания прерываний* (*Interrupt Service Routine, ISR*). Аппаратные прерывания обрабатываются драйверами соответствующих внешних устройств, исключения — специальными модулями ядра, а программные прерывания — процедурами ОС, обслуживающими системные вызовы.
 - В операционной системе может находиться так называемый *диспетчер прерываний*, который координирует работу отдельных обработчиков прерываний.

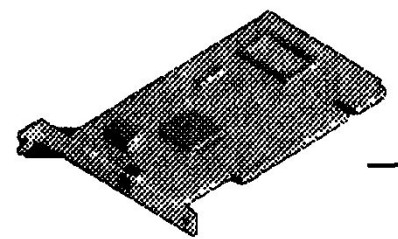
Распределение прерываний по уровням приоритета



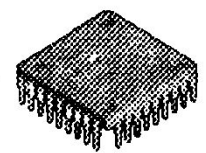
- 
- 
- Два основных способа, с помощью которых шины выполняют прерывания: *векторный (vectored)* и *опрашиваемый (polled)*. В обоих способах процессору предоставляется информация об уровне приоритета прерывания.
 - В случае векторных прерываний в процессор передается также информация о начальном адресе программы - *обработчика прерываний*.

- 
- Вектор прерываний, передаваемый в процессор, представляет собой целое число в диапазоне от 0 до 255, указывающее на одну из 256 программ обработки прерываний, адреса которых хранятся в таблице обработчиков прерываний.
- 

- 
- При использовании опрашиваемых прерываний процессор получает от устройства только информацию об уровне приоритета прерывания.
 - С каждым уровнем прерываний может быть связано несколько устройств и соответственно несколько программ — обработчиков прерываний. Процессор должен определить, какое устройство запросило прерывание. Это достигается вызовом всех обработчиков прерываний для данного уровня приоритета, пока один из обработчиков не подтвердит, что прерывание пришло от обслуживаемого им устройства.



Контроллер периферийного устройства



Контроллер прерываний процессора

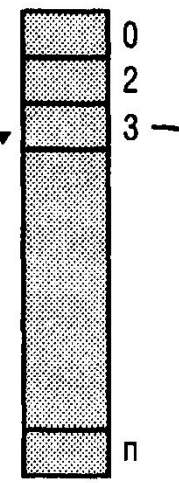
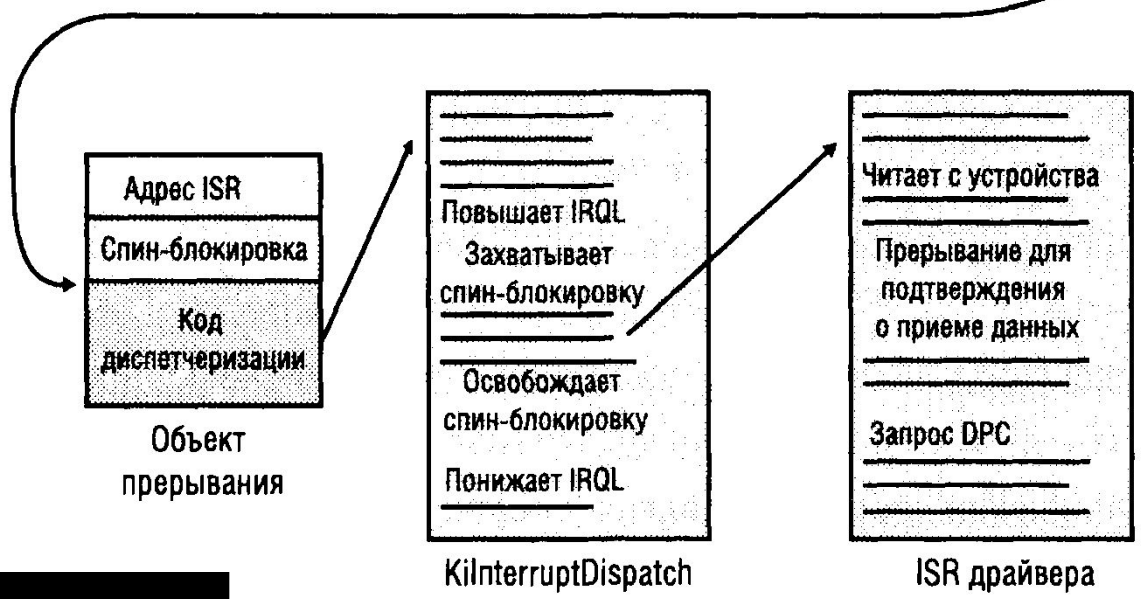


Таблица IDT для процессора

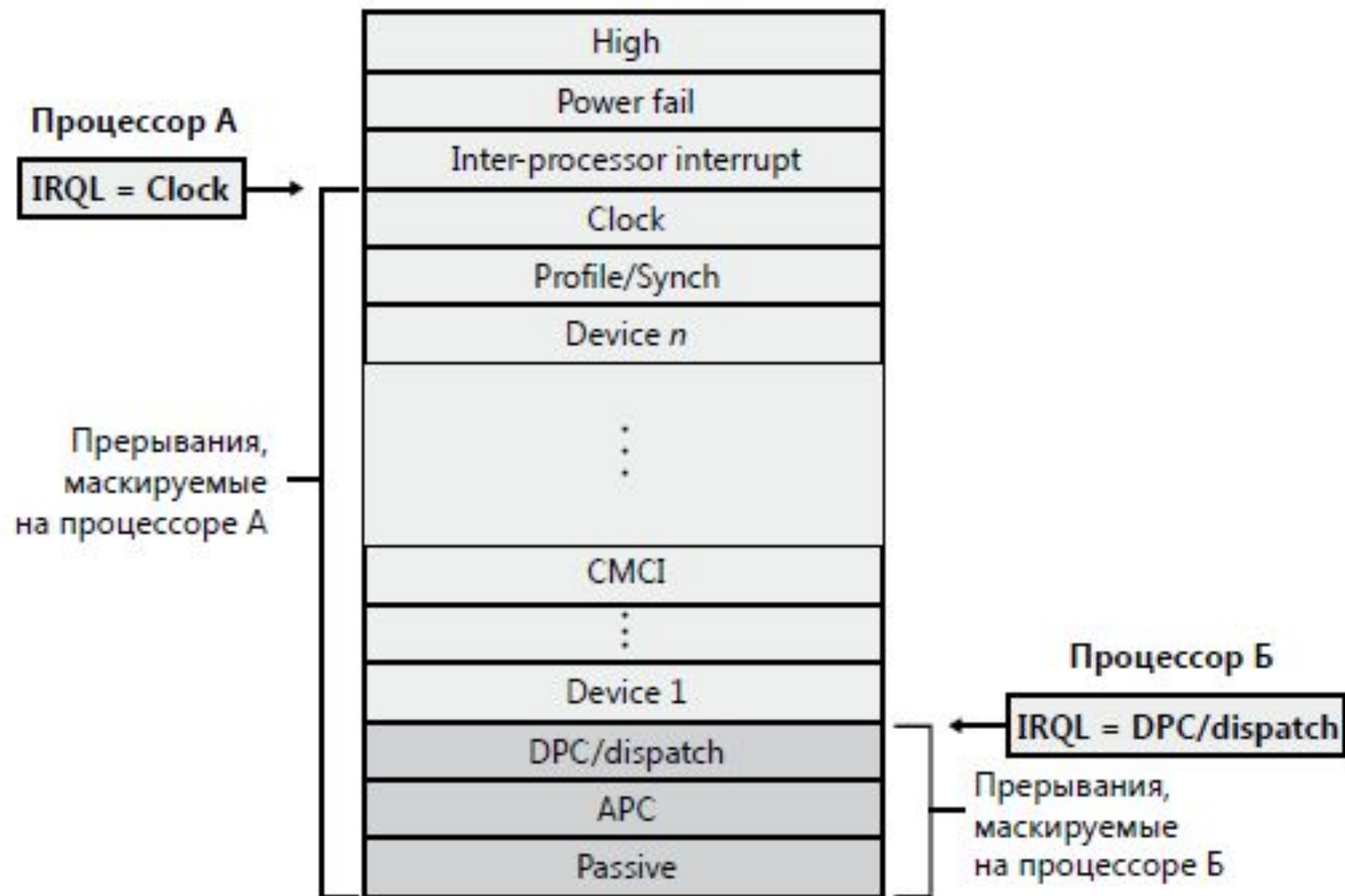


Типичная схема обслуживания прерываний

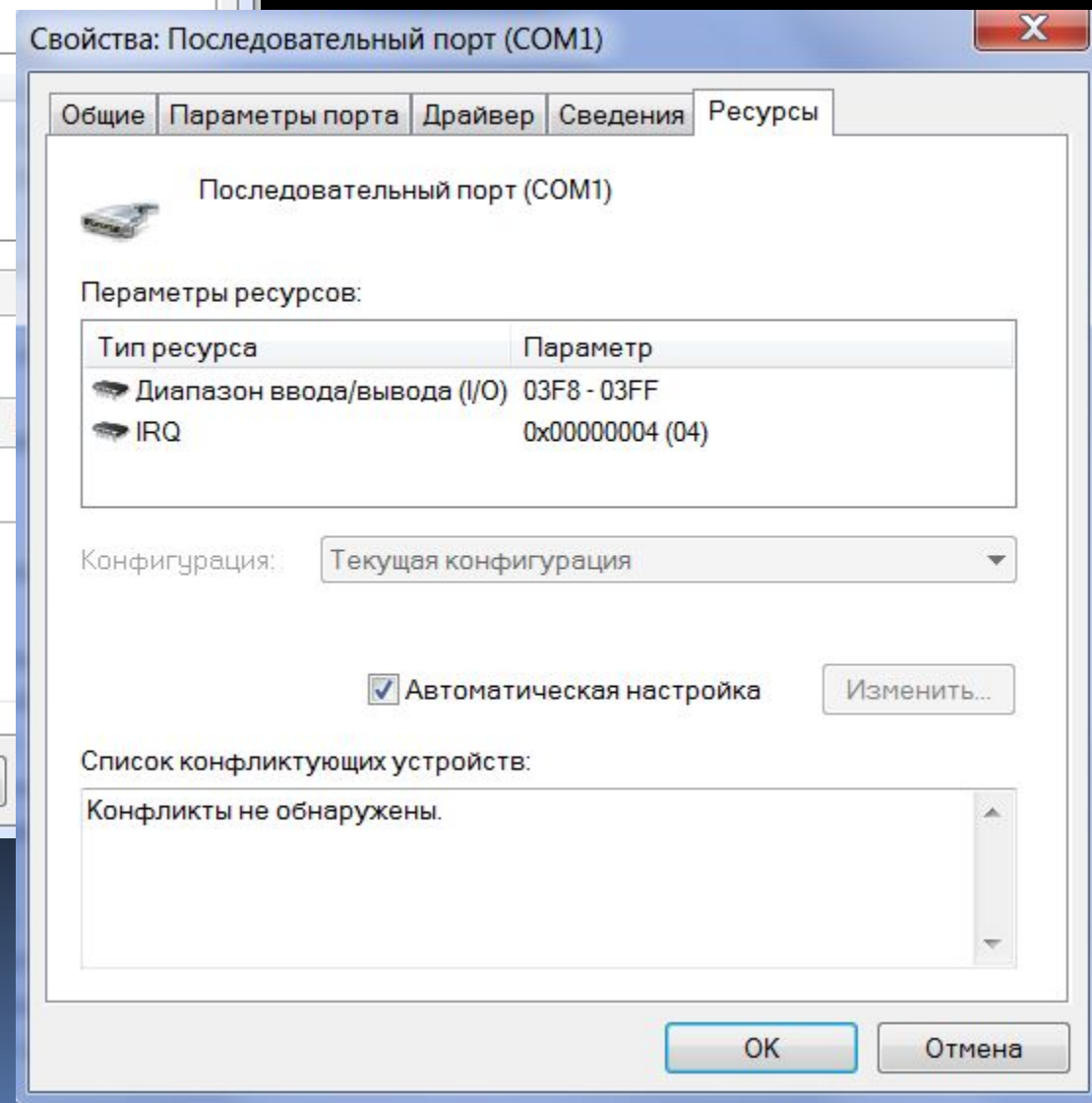
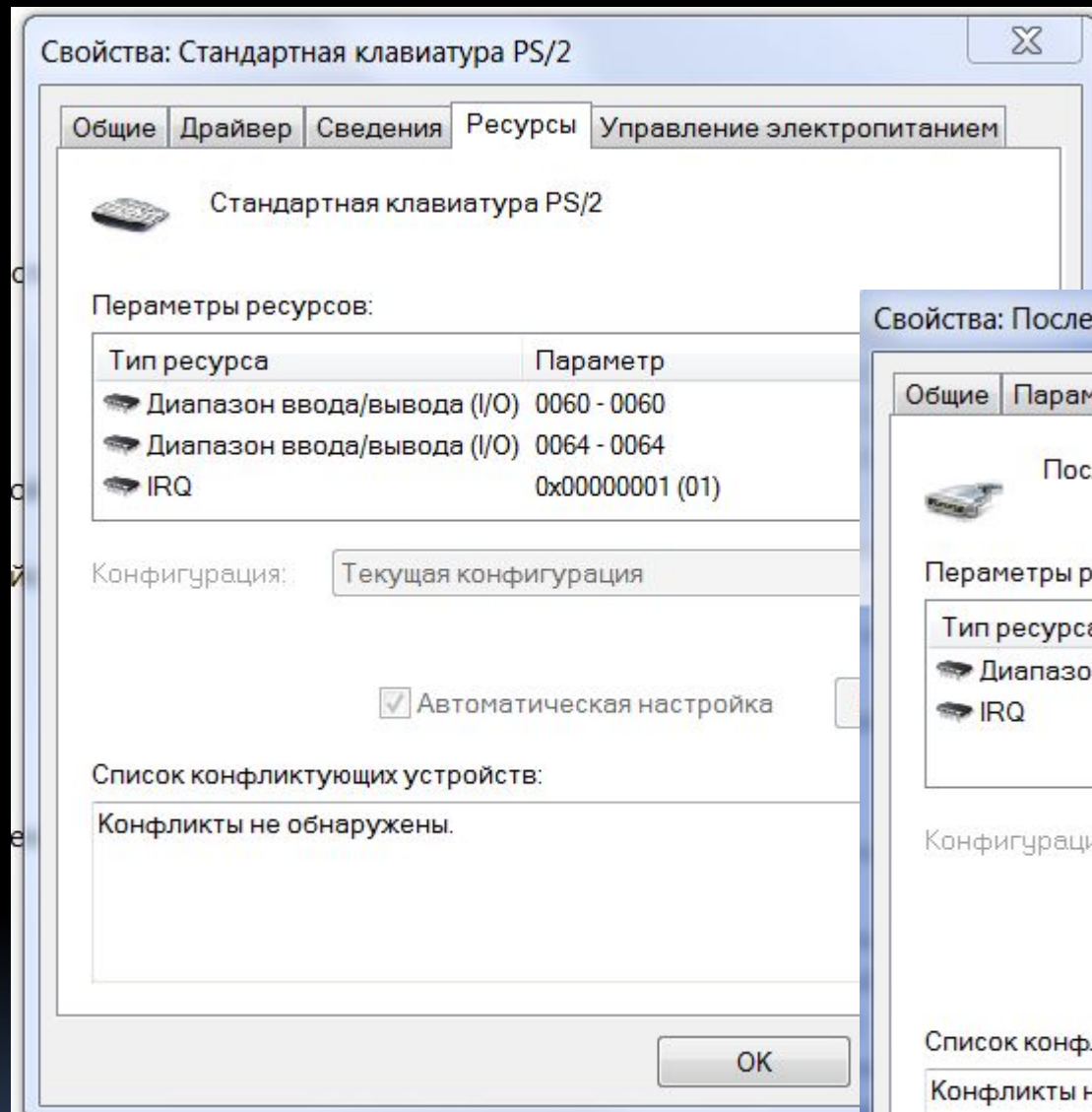
Процессор, работающий по схеме с абсолютными приоритетами, поддерживает в одном из своих внутренних регистров переменную (IRQL - *Interrupt Request Level* — уровень запроса прерывания), фиксирующую уровень приоритета обслуживаемого в данный момент прерывания. При поступлении запроса его приоритет сравнивается с текущим приоритетом процессора, и если приоритет запроса выше, то текущая процедура обработки прерываний вытесняется, а по завершении обслуживания нового прерывания происходит возврат к прерванной процедуре.

Маскирование — метод обработки прерывания, когда при обслуживании некоторого запроса все запросы с равным или более низким приоритетом маскируются, то есть не обслуживаются.

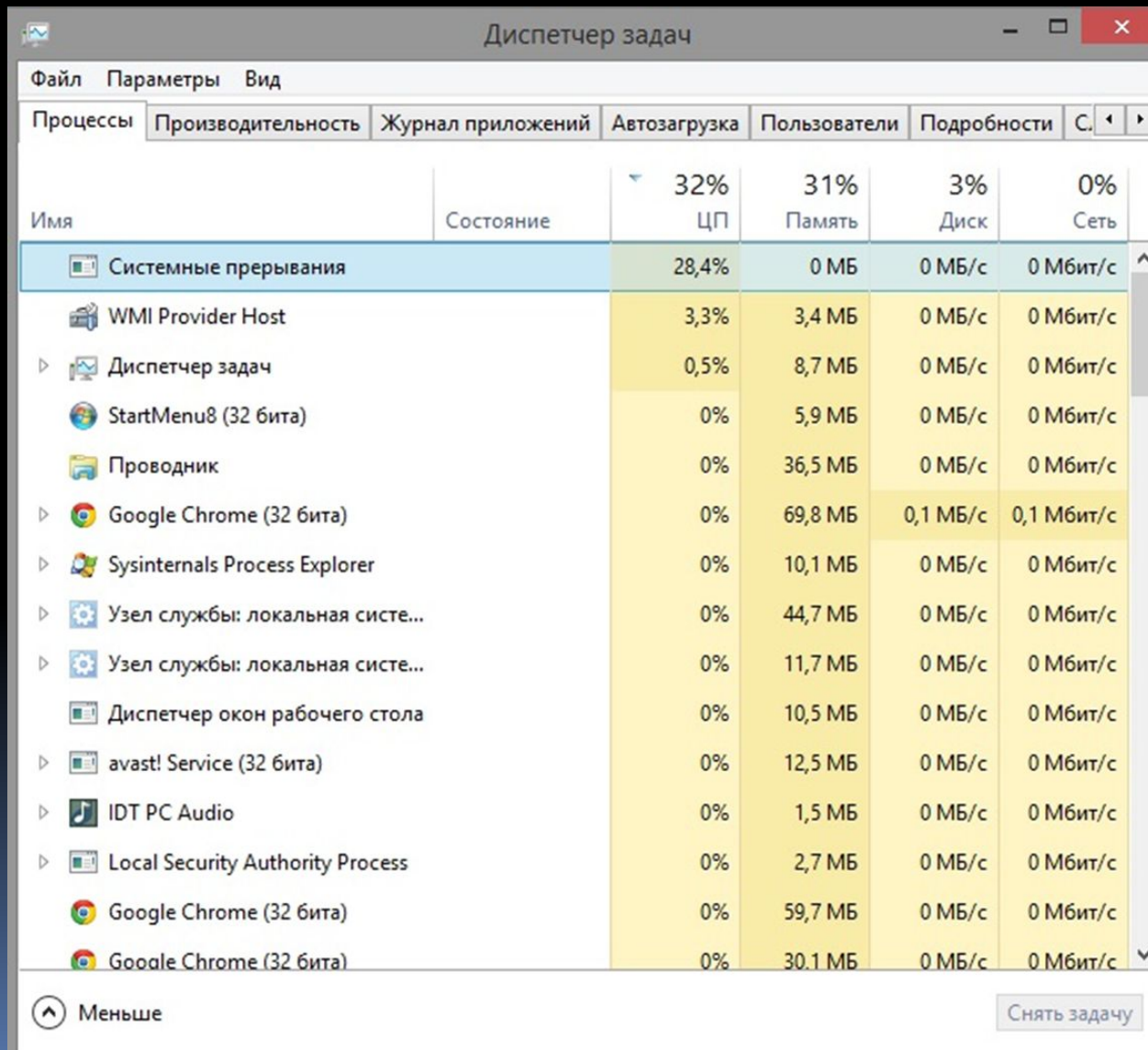
Установка IRQL



Маскирование прерываний



Системные прерывания – стандартный процесс Windows, сигнализирующий о нарушенной работе одного из устройств компьютера. Т.е. это не отдельное приложение, а лишь индикатор, указывающий на аппаратную проблему. Нормальное значение не более 5%.







Имя	Состояние	32% ЦП	31% Память	3% Диск	0% Сеть
Системные прерывания		28,4%	0 МБ	0 МБ/с	0 Мбит/с
WMI Provider Host		3,3%	3,4 МБ	0 МБ/с	0 Мбит/с
Диспетчер задач		0,5%	8,7 МБ	0 МБ/с	0 Мбит/с
StartMenu8 (32 бита)		0%	5,9 МБ	0 МБ/с	0 Мбит/с
Проводник		0%	36,5 МБ	0 МБ/с	0 Мбит/с
Google Chrome (32 бита)		0%	69,8 МБ	0,1 МБ/с	0,1 Мбит/с
Sysinternals Process Explorer		0%	10,1 МБ	0 МБ/с	0 Мбит/с
Узел службы: локальная систе...		0%	44,7 МБ	0 МБ/с	0 Мбит/с
Узел службы: локальная систе...		0%	11,7 МБ	0 МБ/с	0 Мбит/с
Диспетчер окон рабочего стола		0%	10,5 МБ	0 МБ/с	0 Мбит/с
avast! Service (32 бита)		0%	12,5 МБ	0 МБ/с	0 Мбит/с
IDT PC Audio		0%	1,5 МБ	0 МБ/с	0 Мбит/с
Local Security Authority Process		0%	2,7 МБ	0 МБ/с	0 Мбит/с
Google Chrome (32 бита)		0%	59,7 МБ	0 МБ/с	0 Мбит/с
Google Chrome (32 бита)		0%	30,1 МБ	0 МБ/с	0 Мбит/с

Файлы

Файловая система – это часть ОС, включающая:

- совокупность всех файлов на различных носителях информации (магнитные диски, облачные хранилища и т.п.);
- наборы структур данных, используемых для управления файлами (каталоги и дескрипторы файлов, таблицы распределения свободного и занятого пространства носителей информации);
- комплекс системных программных средств, реализующих различные операции над файлами (создание, чтение, запись, уничтожение, изменение свойств и др.).

- 
- Файл – это именованная последовательность байтов произвольной длины, имеющая уникальное собственное имя.
- 

- 
- 
- Минимальный набор требований к файлам со стороны пользователя диалоговой *системы общего назначения* :
 - создание, удаление, чтение и изменение файлов;
 - контролируемый доступ к файлам других пользователей;
 - управление доступом к своим файлам;
 - реструктурирование файлов в соответствии с решаемой задачей;
 - перемещение данных между файлами;
 - резервирование и восстановление файлов в случае повреждения;
 - доступ к файлам по символьным именам.

Типы файлов

- **обычные файлы**

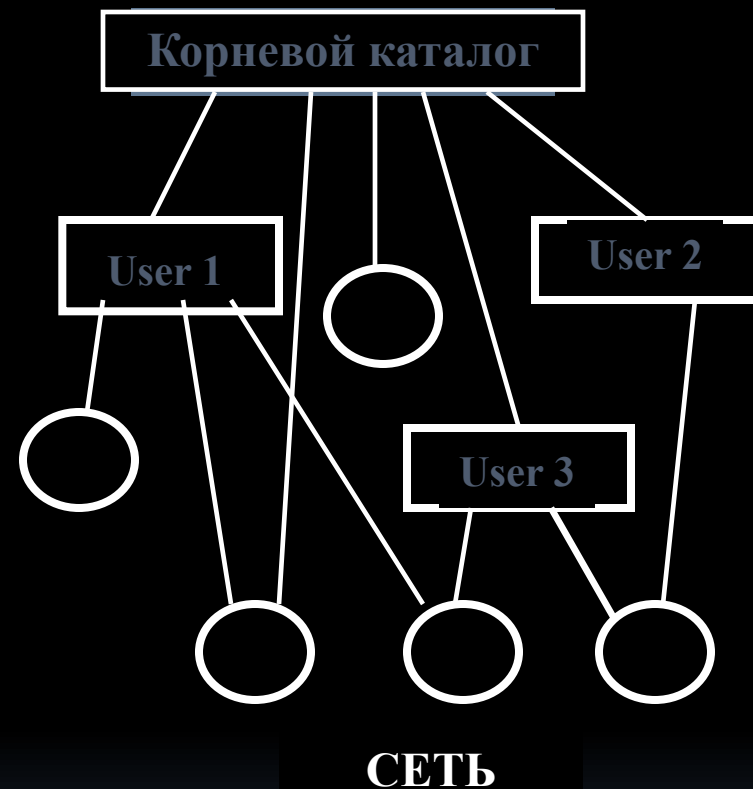
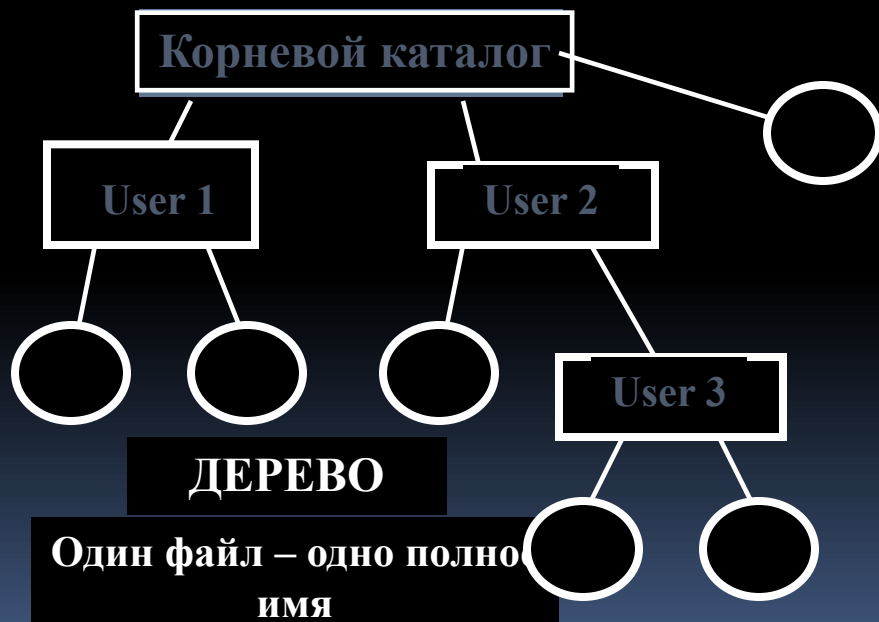
текстовые и двоичные. Все ОС должны уметь распознавать хотя бы один тип файлов - их собственные исполняемые файлы

- **файлы-каталоги**

Каталог - это, с одной стороны, группа файлов, объединенных пользователем исходя из некоторых соображений, а с другой стороны - это файл, содержащий системную информацию о группе файлов, его составляющих. В каталоге содержится список файлов, входящих в него, и устанавливается соответствие между файлами и их характеристиками (атрибутами).

- **специальные файлы**

Это фиктивные файлы, ассоциированные с устройствами ввода-вывода, которые используются для унификации механизма доступа к последовательным устройствам ввода-вывода, таким как терминалы, принтеры и др. (например, монитор и клавиатуру ОС Windows рассматривает как файл со стандартным именем con - консоль). Блочные специальные файлы используются для моделирования дисков.



Один файл – много полных имен

Файловый каталог является связующим звеном между системой управления файлами и набором файлов

Типы имен файлов

В иерархически организованных ФС используются типы имен:

- Простое (короткое) символьное имя идентифицирует файл в пределах одного каталога.
- Составное (полное) символьное имя представляет собой цепочку, содержащую имя диска и имена всех каталогов, через которые проходит путь от корневого каталога до данного файла.
- Относительное имя файла определяется через текущий каталог, т. е. каталог, в котором в данный момент времени работает пользователь.

Атрибут	Значение
Тип файла	Обычный, каталог, специальный и т. д.
Владелец файла	Текущий владелец
Создатель файла	Идентификатор пользователя, создавшего файл
Пароль	Пароль для получения доступа к файлу
Время	Создания, последнего доступа, последнего изменения
Текущий размер файла	Количество байтов в записи
Максимальный размер	Количество байтов, до которого можно увеличивать размер
Флаг «только чтение»	0 – чтение-запись, 1 – только чтение
Флаг «скрытый»	0 – нормальный, 1 – не показывать в перечне файлов каталога
Флаг «системный»	0 – нормальный, 1 – системный
Флаг «архивный»	0 – заархивирован, 1- требуется архивация
Флаг ASCII/двоичный	0 – ASCII, 1 – двоичный
Флаг произвольного доступа	0 – только последовательный доступ, 1 – произвольный доступ
Флаг «временный» процесса	0 – нормальный, 1 – удаление после окончания работы процесса

- 
- **Способ преобразования длинных имен файлов в короткие**

Обычно от имени остаются первые шесть символов, затем идет тильда (~) и после нее число. Т.е. вы видите только первые 6 символов имени (например, regist~1.txt).

