Операционные системы

Введение в операционные системы

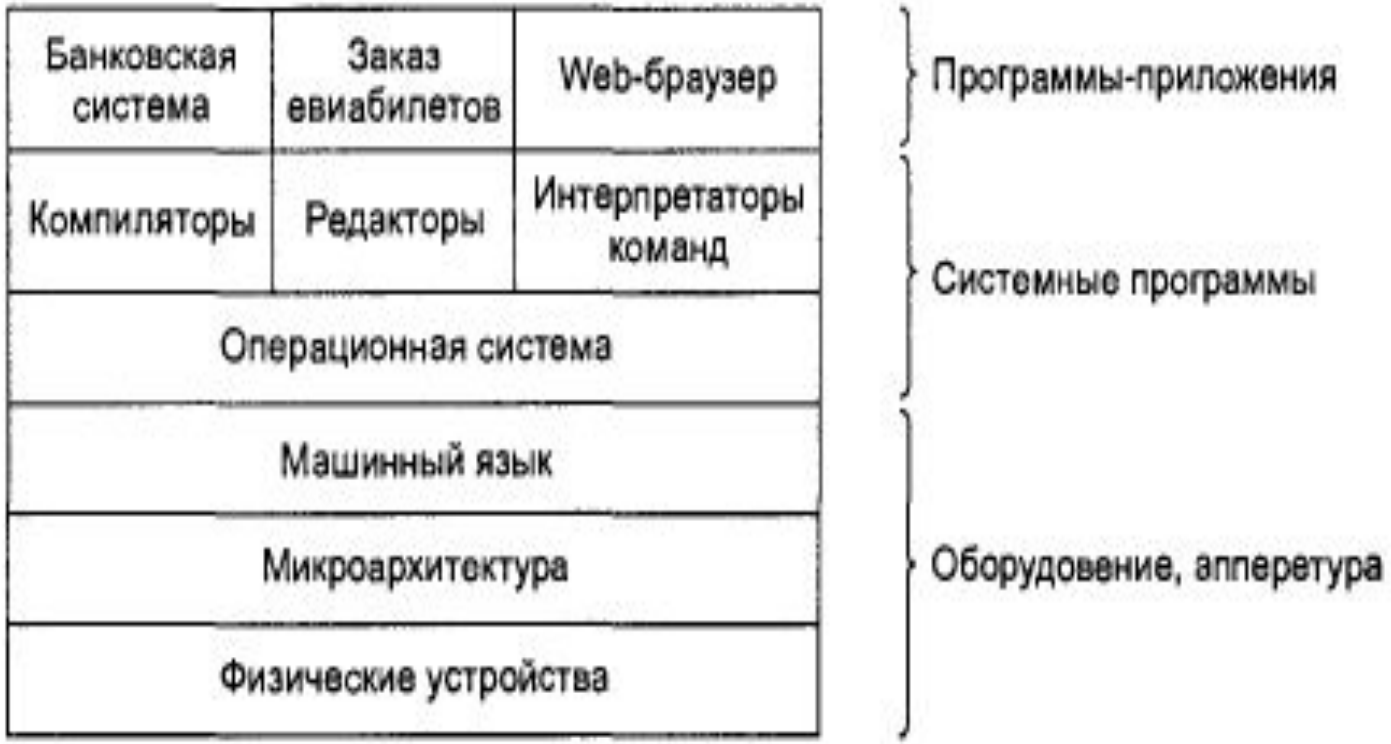
Введение в операционные системы

Назначение ОС

Определение ОС

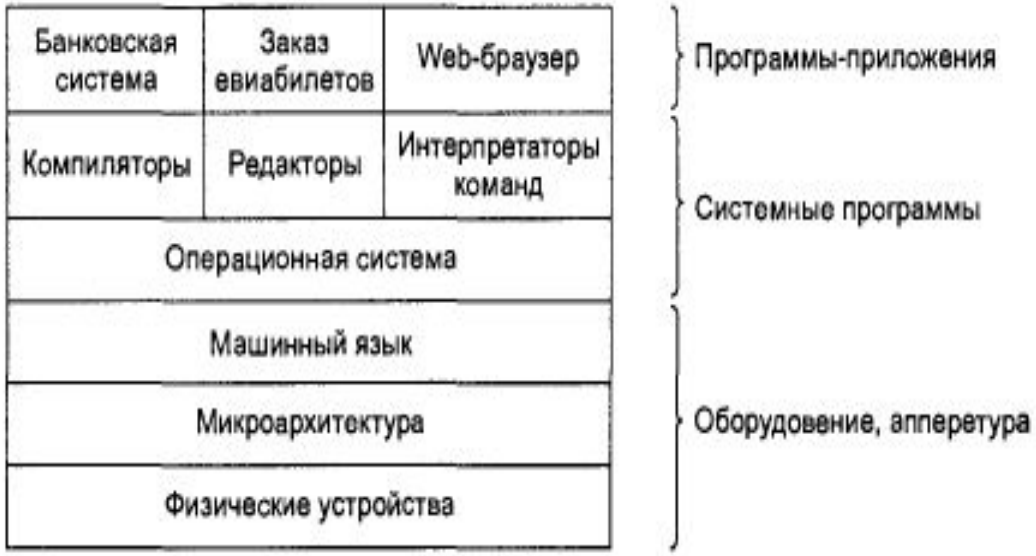
- Операционная система (ОС) – комплекс системных программ, обеспечивающий оптимальное управление ресурсами вычислительной системы в соответствии с некоторым критерием эффективности.
- Критерием эффективности ОС может быть, например, пропускная способность (число выполненных задач за единицу времени) или реактивность (время реакции на некоторое событие) системы.
- Вычислительная система (ВС) – это взаимосвязанная совокупность аппаратных средств вычислительной техники и программного обеспечения, предназначенная для обработки информации.

Уровни ВС



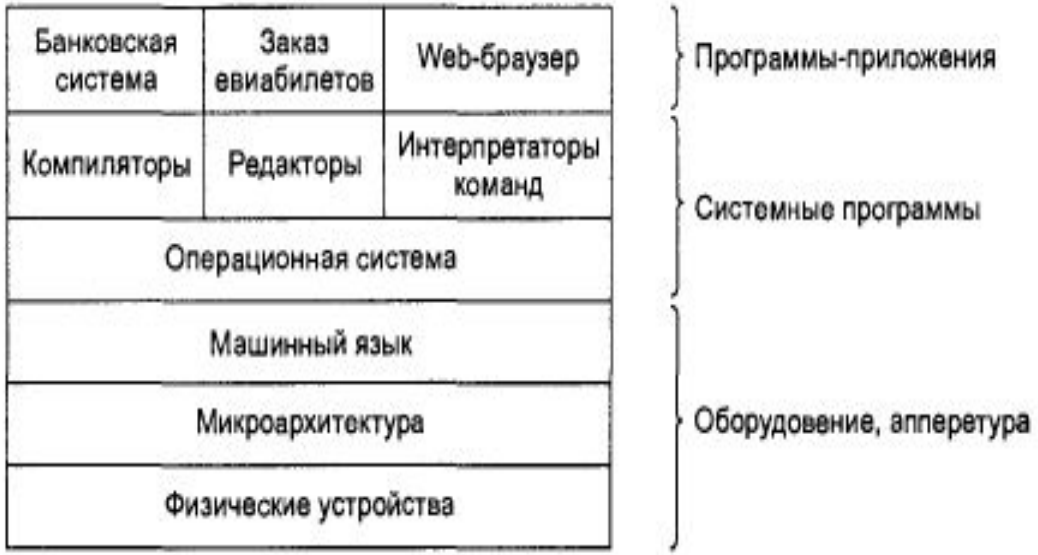
- Расположение ОС в общей структуре компьютера показано на рисунке.

Уровни ВС



- Внизу находится аппаратное обеспечение, которое во многих случаях само состоит из двух или более уровней (или слоев). Самый нижний уровень содержит физические устройства, состоящие из интегральных микросхем, проводников и т. п.

Микроархитектура



- Выше (у некоторых машин) расположен микроархитектурный уровень - примитивная программная прослойка, напрямую работающая с оборудованием и упрощающая интерфейс для программ более высокого уровня. Эта программа, обычно называемая микропрограммой, располагается в ПЗУ.

Машинный язык

- Микропрограмма действует просто как интерпретатор, который получает машинные команды, такие как MOVE, JUMP или ADD, и выполняет их в несколько маленьких шагов. Набор интерпретируемых инструкций определяет машинный язык.
- У некоторых машин микропрограммного уровня нет. Такие системы называются RISC (Reduced Instruction Set Computers - компьютеры с упрощенным набором инструкций). В этих машинах инструкции языка выполняются аппаратурой непосредственно. В качестве примеров можно привести Motorola 680x0, у которой есть микропрограммный уровень, и IBM PowerPC, у которого микропрограммы нет. Обычно машинный язык содержит от 50 до 300 команд.

Системное и прикладное ПО

- Операционная система предназначена для тот, чтобы скрыть от пользователя все эти сложности. Она состоит из уровня программного обеспечения, который частично избавляет от необходимости общения с аппаратурой напрямую, вместо этого предоставляя программисту более удобную систему команд.
- Над ОС на рисунке расположены остальные системные программы. Здесь находятся интерпретатор команд (оболочка), системы окон, компиляторы, редакторы и т. д.
- Наконец, над системными программами расположены прикладные программы – текстовые процессоры, электронные таблицы, пакеты для технических расчетов или игр.

Основная функция ОС

- Основной функцией ОС является управление аппаратными ресурсами ВС и включает решение следующих, не зависящих от типа ресурса задач:
  - планирование и удовлетворение запросов на ресурсы – 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;
  - разрешение конфликтов.

Основные ресурсы ВС

- Процессорное время (процессор)
- Адресное пространство (оперативная память)
- Файлы (накопители данных)
- Внешние устройства ввода/вывода (принтеры, сетевые устройства, …)

Дополнительная функция ОС

- Кроме основной функции управления ресурсами ВС, от ОС зачастую требуется решение еще одной важной задачи – предоставления программного интерфейса доступа к аппаратным ресурсам в виде некоторой виртуальной машины (программного и визуального интерфейсов), которую легче программировать и с которой легче работать, чем непосредственно с аппаратурой, составляющей реальную машину.

Введение в операционные системы

Базовые концепции и термины

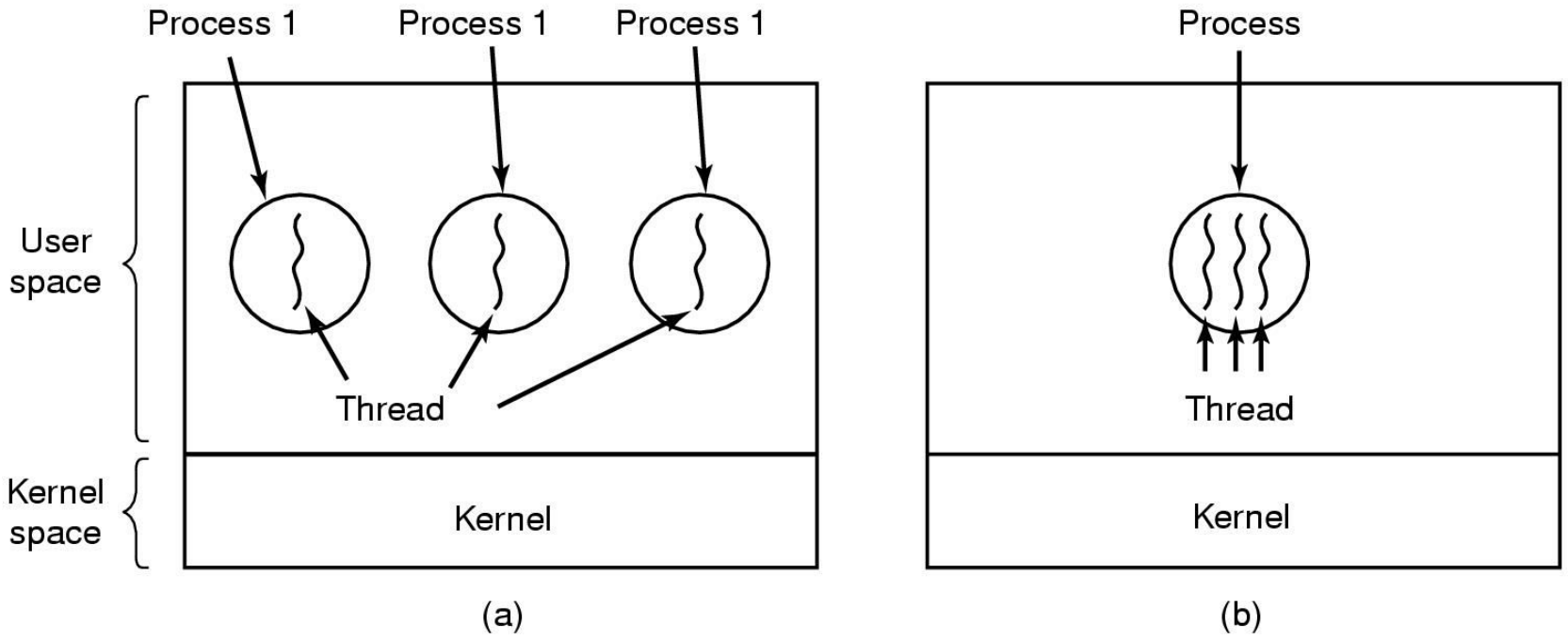
Мультипрограммирование

- **Мультипрограммирование (многозадачность и многопоточность)**, метод “одновременного” выполнения на одной ЭВМ нескольких программ или различных ветвей одной и той же программы.
- В настоящее время в большинстве ОС определены два типа единиц работы, между которыми разделяется процессор и другие ресурсы компьютера: процесс и поток.

Процессы и потоки

- Процесс – абстракция, описывающая выполняющуюся программу. Для ОС процесс представляет собой единицу работы, заявку на потребление системных ресурсов. **Одним из основных ресурсов является адресное пространство процесса.**
- Поток (нить, thread) – последовательность выполнения инструкций процессора. Процесс в этом случае рассматривается как заявка на потребление всех видов ресурсов, кроме одного – процессорного времени, которое ОС распределяет между потоками. Таким образом, **поток представляет собой мини-процесс, который работает в адресном пространстве породившего его процесса**.
- В простейшем случае процесс состоит из одного потока, именно таким образом трактовалось понятие «процесс» до середины 80-х годов (например, в ранних версиях UNIX).

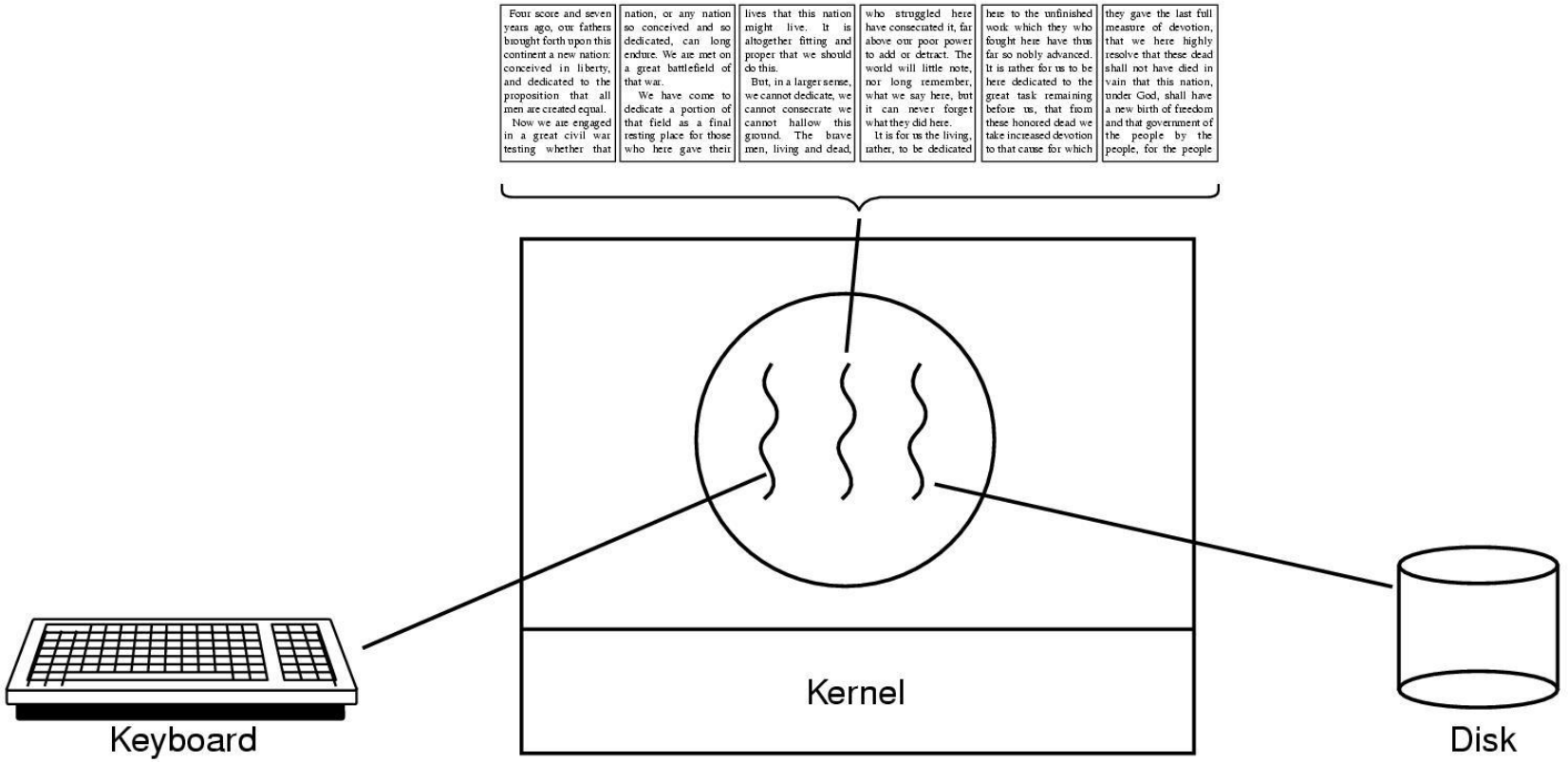
Варианты мультипрограммирования



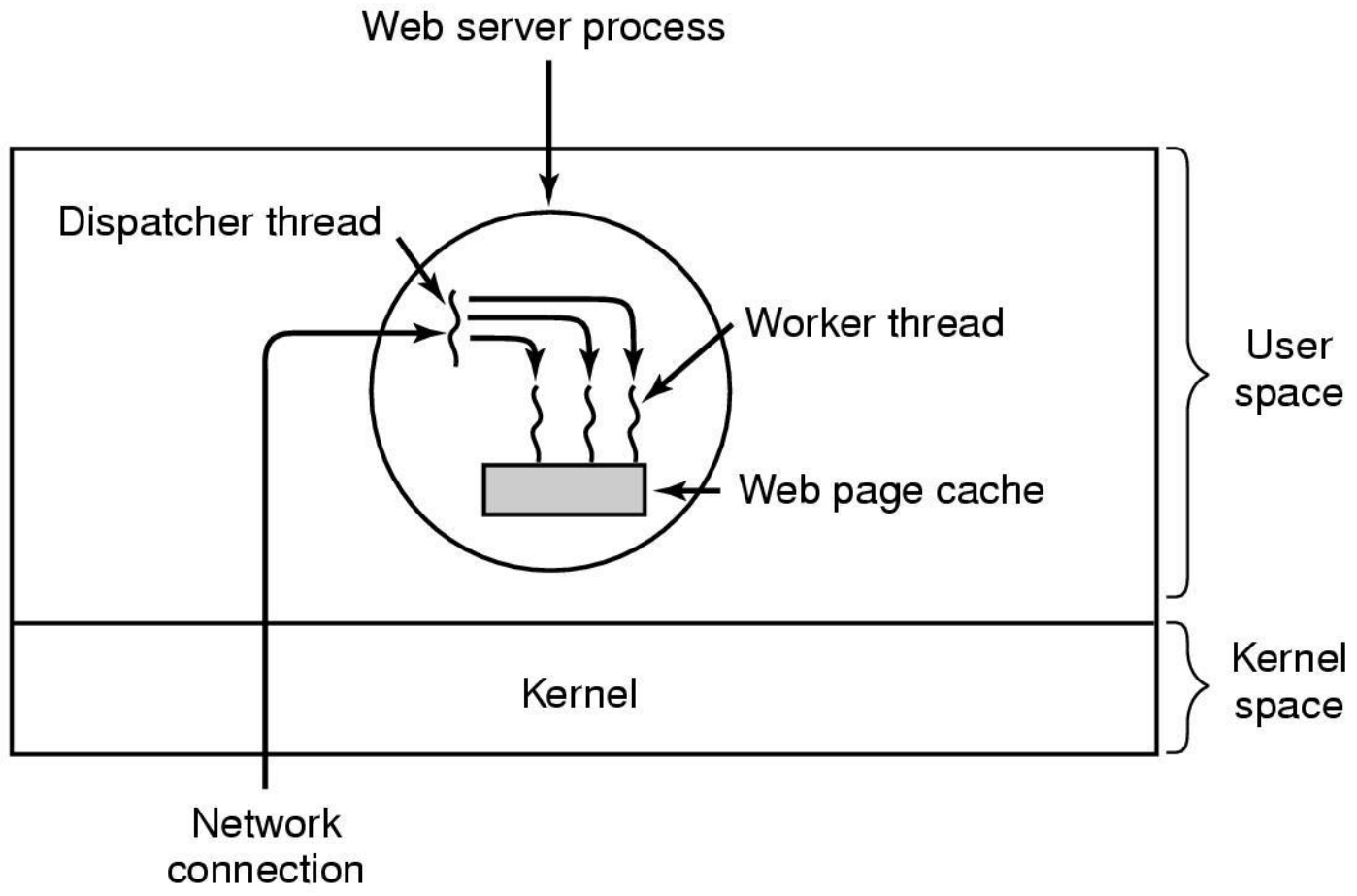
(a) Три однопоточных процесса

(b) Один процесс с тремя потоками

Пример многопоточного приложения: текстовый процессор с 3-мя нитями



Пример многопоточного приложения: Web-сервер

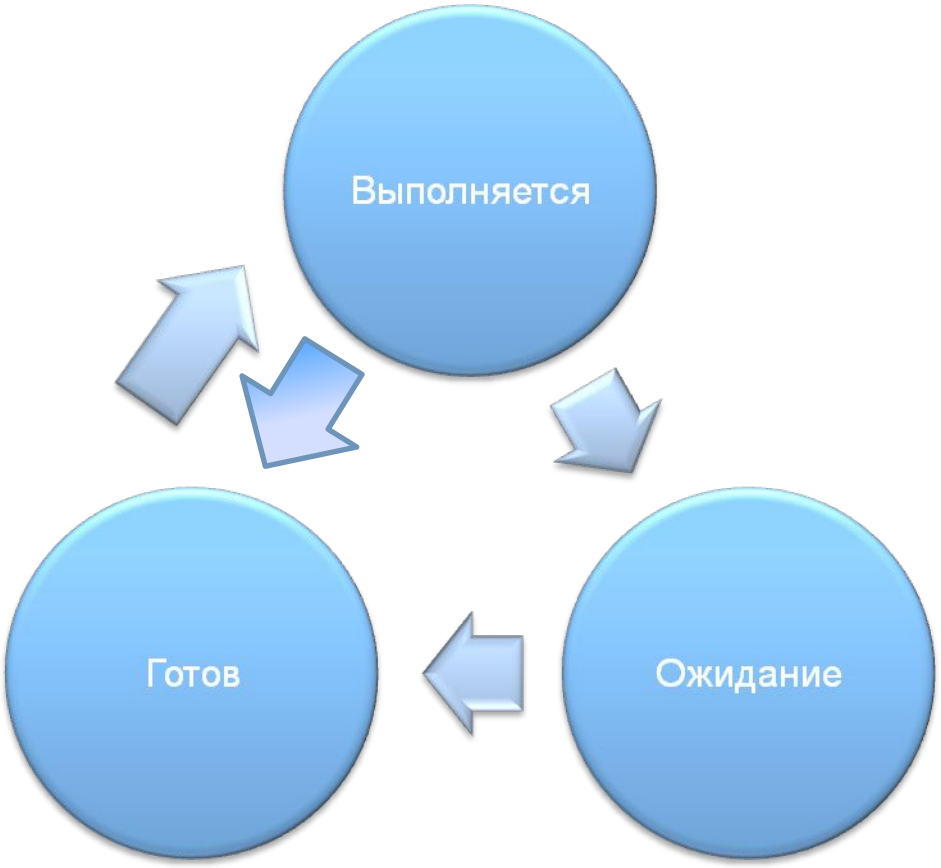


Состояния процессов и потоков

- Выделяют 3 основных дискретных состояния процесса (потока):
  - Готов к выполнению – ждет ЦП
  - Выполняется – выделен ЦП
  - Приостановлен (блокирован) – ждет некоторого события (например, окончания ввода-вывода)

Состояния процессов и потоков

- Вытеснение и постановка на выполнение происходит на основе выбранной дисциплины обслуживания.



Введение в операционные системы

Классификация ОС

Признаки классификации

- ОС могут различаться особенностями реализации внутренних алгоритмов управления основными ресурсами компьютера, особенностями использованных методов проектирования, типами аппаратных платформ, областями использования и многими другими свойствами.
- Рассмотрим подробнее классификацию ОС по нескольким наиболее основным признакам:
  - особенности алгоритмов управления ресурсами;
  - особенности областей использования;
  - особенности аппаратных платформ;
  - структурная организация.

Классификация ОС

Особенности алгоритмов управления ресурсами

Поддержка многозадачности

- По числу одновременно выполняемых задач ОС могут быть разделены на два класса:
  - однозадачные (например, MS-DOS, MSX);
  - многозадачные (ОС ЕС, UNIX, Windows 9x, NT и выше).
- Однозадачные ОС в основном выполняют функцию предоставления пользователю виртуальной машины.
- Многозадачные ОС поддерживают в том или ином виде **мультипрограммирование** и управляют разделением совместно используемых ресурсов (процессор, оперативная память, файлы и пр.).

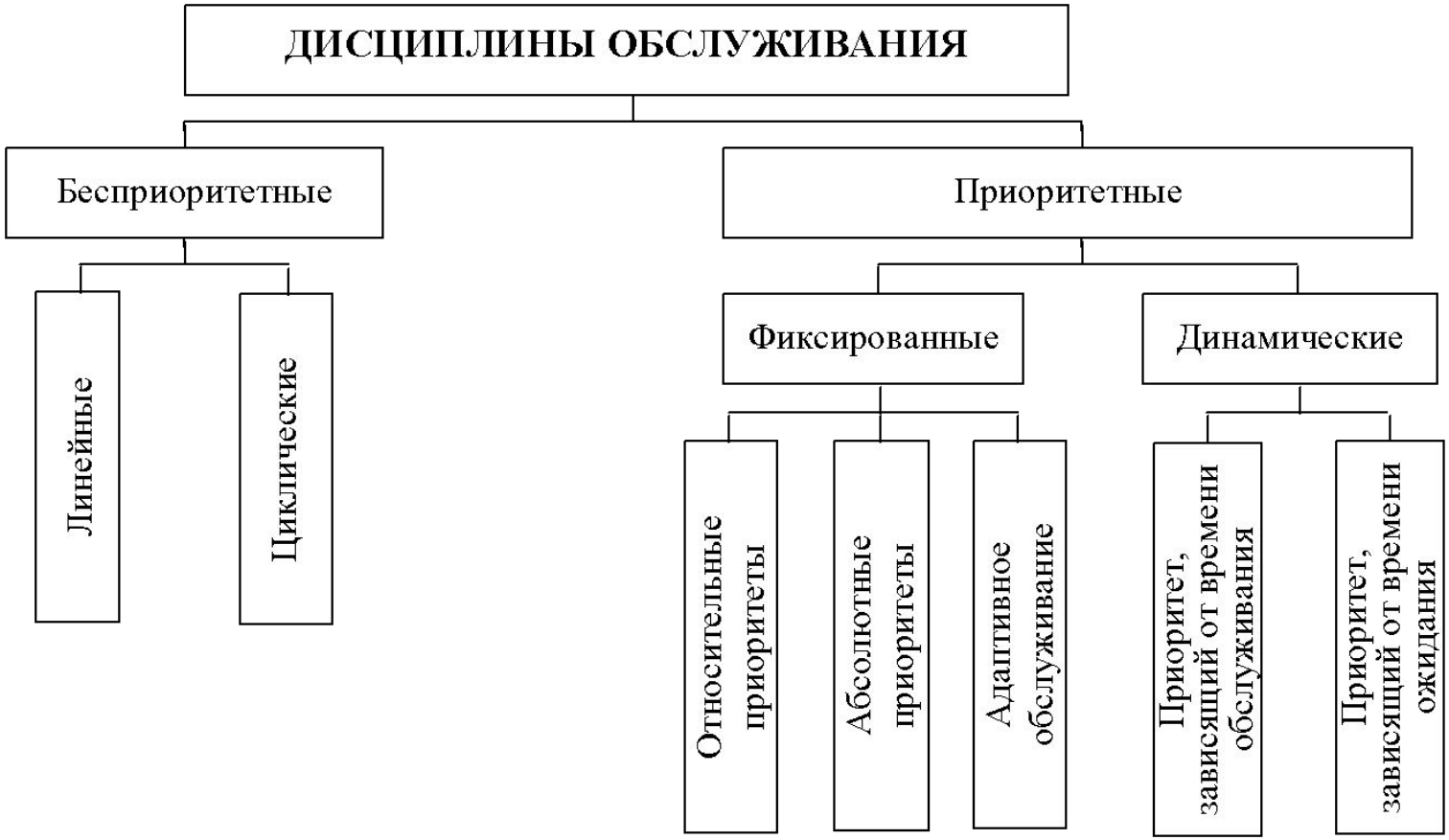
Многозадачность

- Способ распределения процессорного времени между несколькими одновременно существующими в системе задачами (процессами или потоками) в режиме мультипрограммирования во многом определяет специфику ОС.
- Среди множества существующих вариантов реализации многозадачности можно выделить две группы алгоритмов:
  - невытесняющая (корпоративная) многозадачность (NetWare, Windows 3.x);
  - вытесняющая многозадачность (OS/2, UNIX, Win’95 и выше).

Вытесняющая и не вытесняющая многозадачность

- При невытесняющей многозадачности активный процесс (поток) выполняется до тех пор, пока он сам, по собственной инициативе, не отдаст управление ОС для того, чтобы та выбрала из очереди другой готовый к выполнению процесс (поток).
- При вытесняющей многозадачности решение о переключении процессора с одного процесса (потока) на другой принимается ОС.

Классификация дисциплин обслуживания



Дисциплины обслуживания

- *Бесприоритетные дисциплины* – выбор из очереди производится без учета относительной важности задач и времени их обслуживания.
- *Приоритетное обслуживание* – отдельным задачам предоставляется преимущественное право перейти в состояние ВЫПОЛНЕНИЯ.
- *Фиксированные приоритеты* – являются величиной постоянной на всем жизненном цикле процесса.
- *Динамические приоритеты* – изменяются в зависимости от некоторых условий в соответствии с определенными правилами. Для реализации динамических приоритетов необходимы дополнительные затраты, но их использование предполагает более справедливое распределение процессорного времени между процессами.

Приоритетное обслуживание

- Каждому процессу присваивается **приоритет**, и управление передается процессу с самым высоким приоритетом.
- Приоритетное обслуживание может использовать относительные и абсолютные приоритеты.
- Приоритет может быть динамический и фиксированный.
- Часто процессы объединяют по приоритетам в группы, и используют приоритетное планирование среди групп, но внутри группы используют циклическое планирование.

Динамический приоритет

- Динамический приоритет может устанавливаться так:
  - П=1/Т, где Т- часть использованного в последний раз кванта
  - Если использовано 1/50 кванта, то приоритет 50.
  - Если использован весь квант, то приоритет 1.
- Т.е. процессы, ограниченные вводом/вывода, будут иметь приоритет над процессами ограниченными процессором.

Поддержка многопользовательского режима

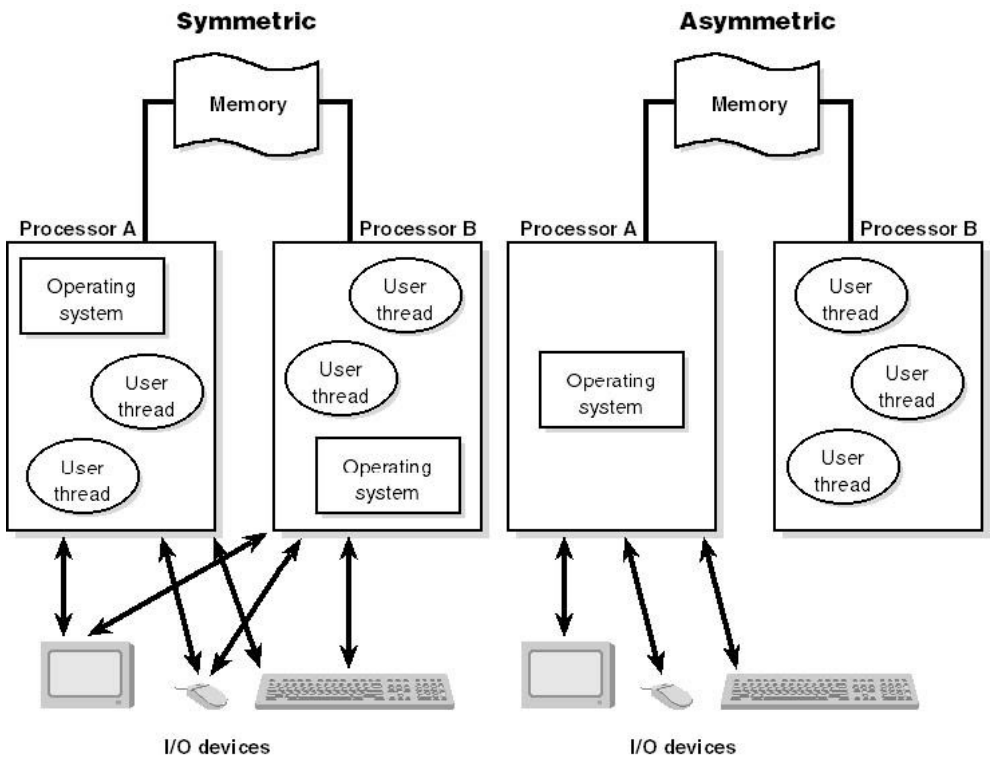
- По числу “одновременно” работающих пользователей ОС делятся на:
  - однопользовательские (MS-DOS, Windows 3.x, Windows 9x);
  - многопользовательские (UNIX, Windows NT, 2000-2007).
- Главным отличием многопользовательских систем от однопользовательских является **наличие средств защиты информации** каждого пользователя от несанкционированного доступа других пользователей.
- Следует заметить, что не всякая многозадачная система является многопользовательской, и не всякая однопользовательская ОС является однозадачной.

Многопроцессорная обработка

- Другим важным свойством ОС является отсутствие или наличие в ней средств поддержки многопроцессорной обработки. Мультипроцессирование приводит к усложнению всех алгоритмов управления ресурсами.
- В наши дни становится общепринятым введение в ОС функций поддержки многопроцессорной обработки данных. Такие функции имеются в операционных системах Solaris фирмы Sun, Windows NT-2007 фирмы Microsoft и NetWare фирмы Novell.
- Многопроцессорные ОС могут классифицироваться по способу организации вычислительного процесса: асимметричные ОС и симметричные ОС.

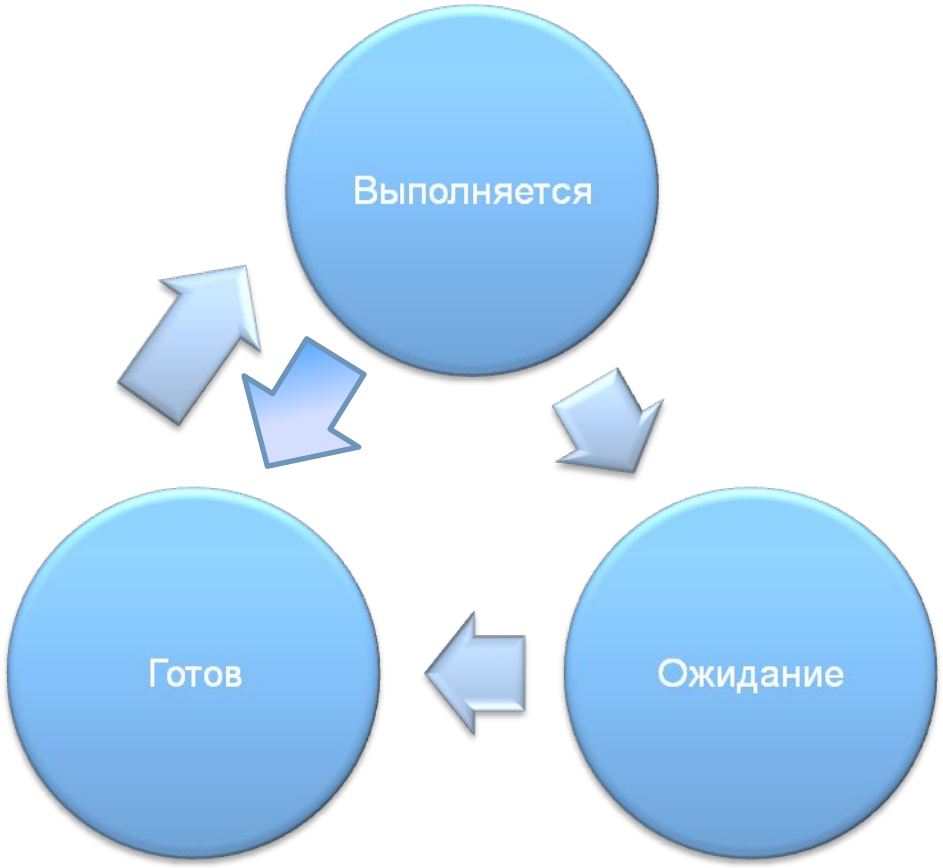
Виды мультипроцессирования

- 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.



Состояния процессов и потоков при мультипроцессировании

- Несколько состояний “Выполняется” (по одному на каждый CPU)
- Одно или несколько состояний “Готов” (общая очередь или отдельная на каждый CPU)



Особенности алгоритмов управления ресурсами

- Выше были рассмотрены характеристики ОС, связанные с управлением только одним типом ресурсов – процессором. Важное влияние на облик операционной системы в целом, на возможности ее использования в той или иной области оказывают особенности и других подсистем управления локальными ресурсами - подсистем управления памятью, файлами, устройствами ввода-вывода.

Классификация ОС

Особенности областей использования

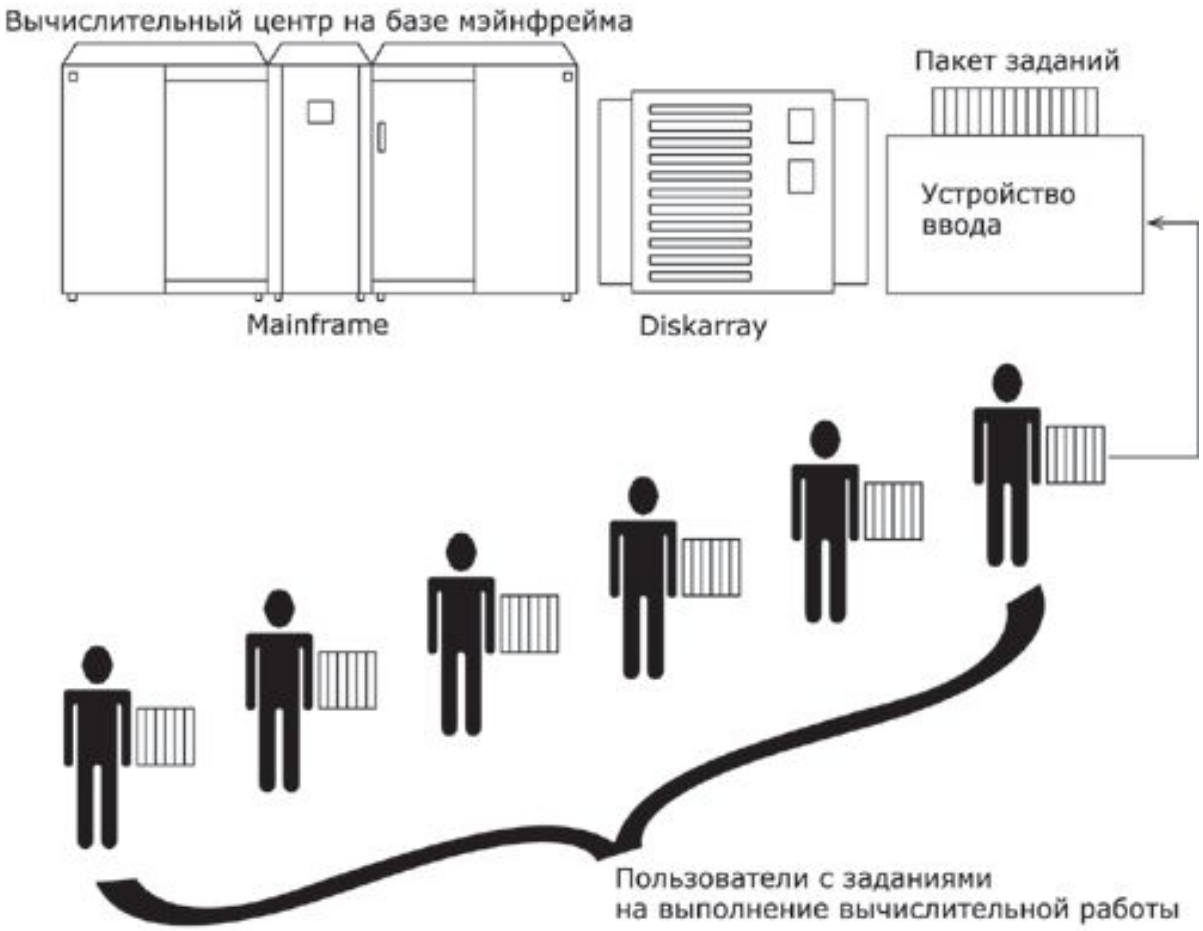
Типы многозадачных ОС

- Многозадачные ОС подразделяются на три типа в соответствии с использованными при их разработке критериями эффективности:
  - системы пакетной обработки (например, ОС ЕС);
  - системы разделения времени (UNIX, MS Windows);
  - системы реального времени (QNX, RT/11).

Системы пакетной обработки

- Системы пакетной обработки (batch processing) предназначались для решения задач в основном вычислительного характера, не требующих быстрого получения результатов.
- Главной целью и критерием эффективности систем пакетной обработки является максимальная пропускная способность.
- В системах пакетной обработки используются следующая схема функционирования: в начале работы формируется пакет заданий, каждое задание содержит требование к системным ресурсам; из этого пакета заданий формируется множество одновременно выполняемых задач.

Системы пакетной обработки



Системы пакетной обработки

- Для одновременного выполнения выбираются задачи, предъявляющие отличающиеся требования к ресурсам, так, чтобы обеспечивалась сбалансированная загрузка всех устройств ВС; так, например, желательно одновременное присутствие вычислительных задач и задач с интенсивным вводом-выводом.
- 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 в системах пакетной обработки невозможно гарантировать выполнение интерактивных задач.

Системы разделения времени

- **Системы разделения времени** (time sharing) призваны исправить основной недостаток систем пакетной обработки – отсутствие интерактивности.
- Первоначально системы разделения времени представляли собой многотерминальные ВС на базе мэйнфреймов. Каждому пользователю предоставлялся удаленный терминал, с которого он вел диалог со своей программой. Каждой программе (пользователю) выделяется квант процессорного времени, за счет чего достигалась иллюзия “персональной” ЭВМ.
- В настоящее время все многозадачные ОС, которые предоставляют пользователю интерактивный режим работы, считаются системами разделения времени.

Системы разделения времени

- Системы разделения времени обладают меньшей пропускной способностью, чем системы пакетной обработки, так как на выполнение принимается каждая запущенная пользователем задача, а не та, которая "выгодна" системе, и, кроме того, имеются накладные расходы вычислительной мощности на более частое переключение процессора с задачи на задачу.
- Критерием эффективности систем разделения времени является не максимальная пропускная способность, а удобство и эффективность работы пользователя.

Системы реального времени

- **Системы реального времени** применяются для управления различными техническими объектами, такими, (станок, научная экспериментальная установка) или технологическими процессами (гальваническая линия, доменный процесс). 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.
- Таким образом, критерием эффективности для систем реального времени временя реакции системы на события от объекта управления, это свойство системы называется реактивностью.

Гибридные системы

- Некоторые операционные системы могут совмещать в себе свойства систем разных типов, например, часть задач может выполняться в режиме пакетной обработки, а часть – в режиме реального времени или в режиме разделения времени. В таких случаях режим пакетной обработки часто называют фоновым режимом.

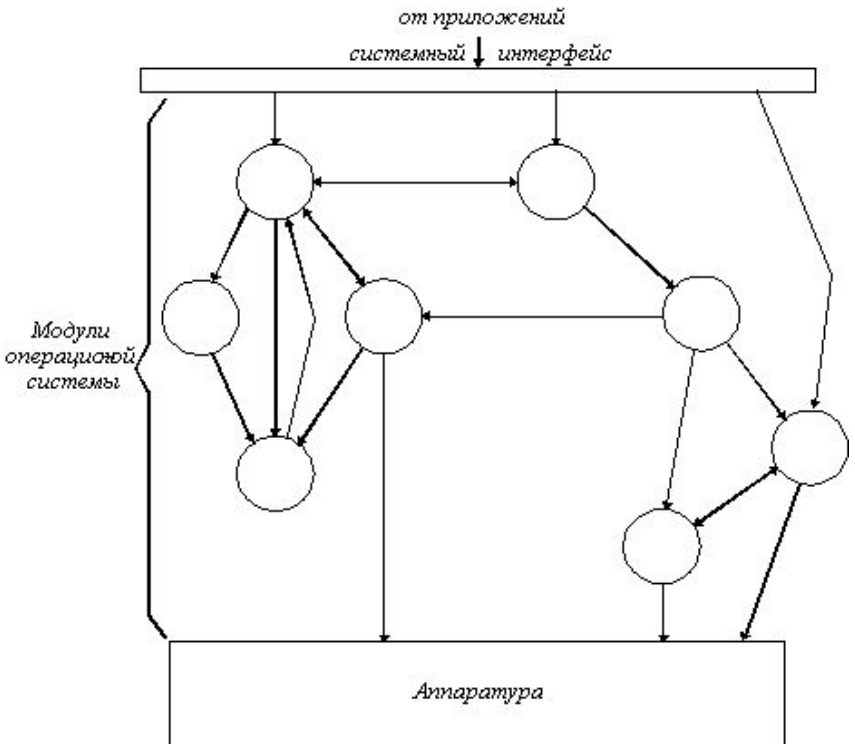
Операционные системы

Структурная организация операционных систем

Структурная организация

- При описании операционной системы часто указываются особенности ее структурной организации и основные концепции, положенные в ее основу.
- Рассмотрим основные способы структурной организации ОС:
  - монолитную структуру;
  - многоуровневую структуру;
  - ядерную структуру;
  - микроядерную структуру;
  - объектно-ориентированный подход.

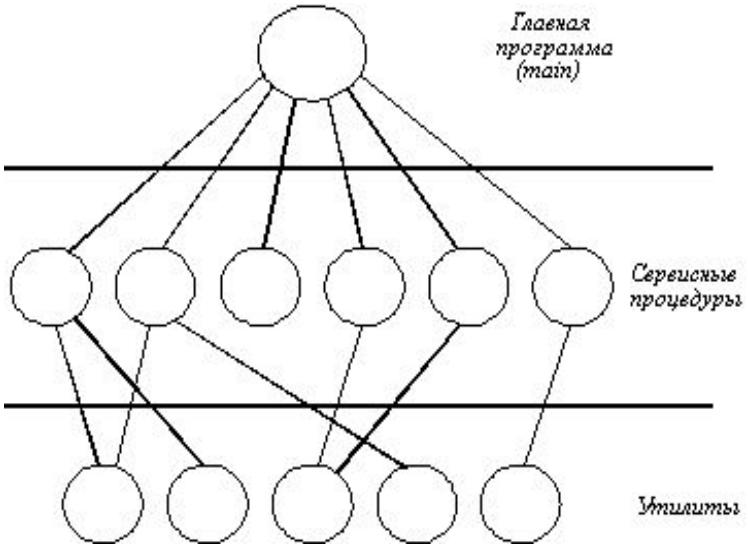
Монолитная структура



Наиболее простым и распространенным способом построения ОС является **монолитная структура**, когда ОС компонуется как одна программа Для построения монолитной системы необходимо скомпилировать все отдельные процедуры, а затем связать их вместе в единый объектный файл с помощью компоновщика (примерами могут служить ранние версии ядра UNIX или Novell NetWare).

Многоуровневая структура

- Развитием монолитного подхода является многоуровневый, когда ОС реализуется как иерархии уровней.
- Уровни образуются группами функций ОС – файловая система, управление процессами и устройствами и т.п.



- Каждый уровень может взаимодействовать только со своим непосредственным соседом – выше- или нижележащим уровнем.

Многоуровневая структура

- Первой многоуровневой ОС считают систему THE.
- ОС THE была создана в Technische Hogeschool Eindhoven (Нидерланды) Э. Дейкстрой (E. W. Dijkstra) и его студентами в 1968 году.
- Она была простой пакетной системой для голландского компьютера Electrologica X8, память которого состояла из 32 К 27-разрядных слов.

Уровень	Функция
5	Оператор
4	Программы пользователя
3	Управление вводом/выводом
2	Связь оператор-процесс
1	Управление памятью и барабаном
0	Распределение процессора и многозадачность

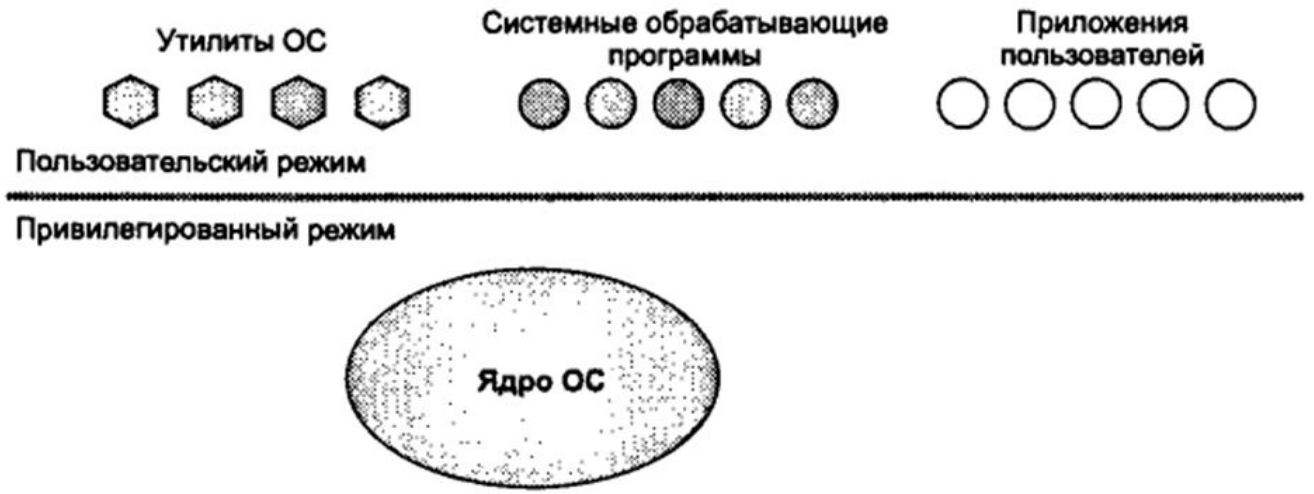
Понятие ядра

- Развитием многоуровневой концепции стала ядерная архитектура. В общем случае уровни ОС представляют собой серию концентрических колец, где внутренние кольца являлись более привилегированными, чем внешние.
- Ядро – центральная часть ОС, выполняющая основные функции.
- Ядро системы MULTICS, находящееся постоянно в памяти компьютера занимало всего 135 Килобайт кода.

Уровни привилегий (защиты)

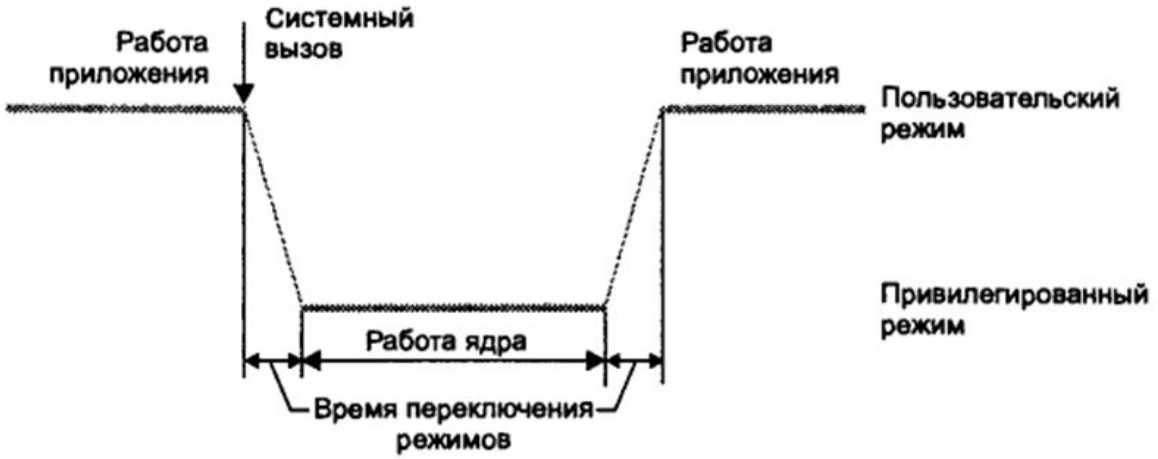
- Для обеспечения привилегий ОС необходима соответствующая аппаратная поддержка.
- Между числом уровней привилегий, поддерживаемых аппаратно, и числом уровней привилегий ОС нет прямого соответствия.
- Для реализации ядра необходимо хотя бы два уровня: основные процедуры ОС выполняются в привилегированном режиме, тогда как пользовательские программы – в непривилегированном.

Ядро в привилегированном (защищенном) режиме



- Повышение устойчивости ОС обеспечиваемое рабой ядра в привилегированном режиме достигается за счет некоторого замедления выполнения системных вызовов.

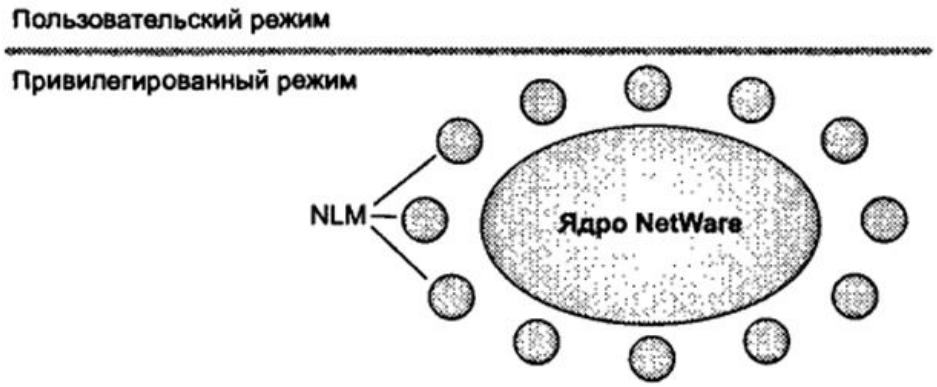
Ядро в привилегированном (защищенном) режиме



- Системный вызов привилегированного ядра инициирует переключение процессора из пользовательского режима в защищенный, а при возврате к приложению – обратно. В результате вызов выполняется медленнее.

Пример ядра в непривилегированном режиме

- В некоторых случаях разработчики ОС отступают от этого классического варианта архитектуры, организуя работу ядра и приложений в одном и том же режиме.
- Так, сетевая ОС Novell NetWare использует привилегированный режим процессоров Intel x86/Pentium как для работы ядра, так и для работы своих специфических приложений – загружаемых модулей NLM.



Монолитное ядро

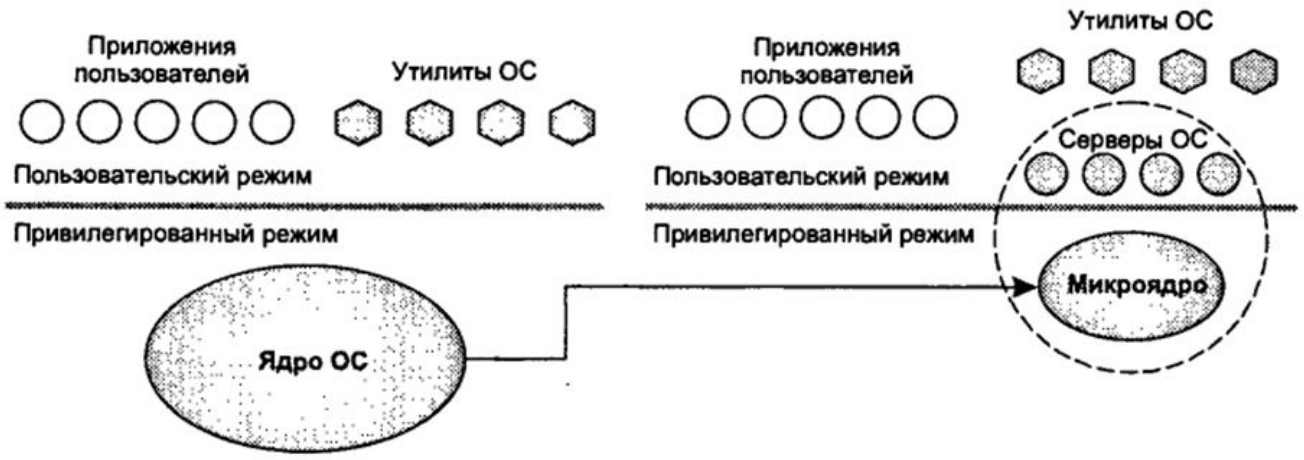
- Наиболее распространенным и классическим вариантом реализации ядерного подхода является **моноли́тное ядро́**.
- Монолитность ядер усложняет их отладку, понимание кода ядра, добавление новых функций и возможностей, удаление «мёртвого», ненужного, унаследованного от предыдущих версий, кода.
- «Разбухание» кода монолитных ядер также повышает требования к объёму оперативной памяти, требуемому для функционирования ядра ОС.
- Это делает монолитные ядерные архитектуры мало пригодными к эксплуатации в системах, сильно ограниченных по объёму ОЗУ, например, встраиваемых системах, производственных микроконтроллерах и т. д.

Модульное ядро

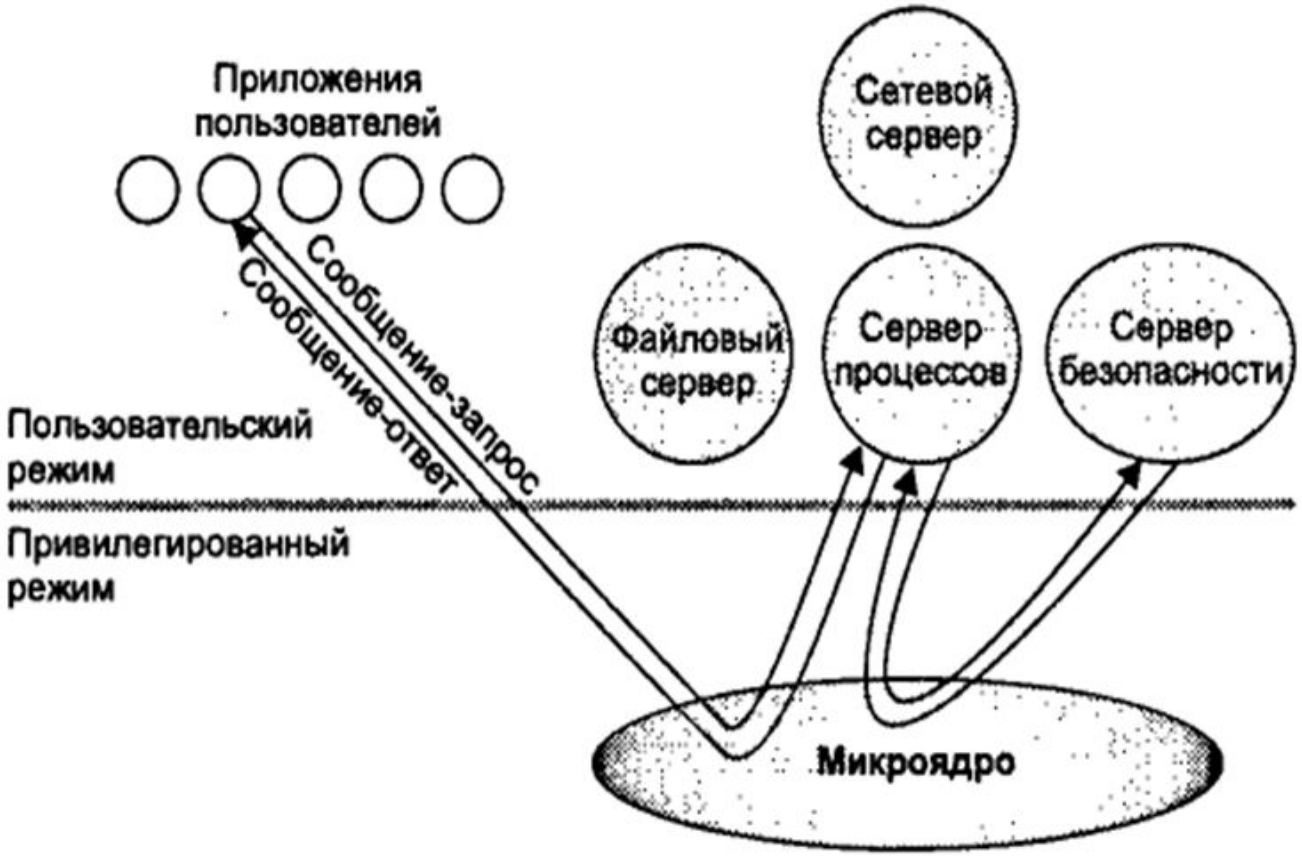
- Современная, усовершенствованная модификация архитектуры монолитных ядер ОС.
- В отличие от «классических» монолитных ядер, считающихся ныне устаревшими, модульные ядра, как правило, не требуют полной перекомпиляцииВ отличие от «классических» монолитных ядер, считающихся ныне устаревшими, модульные ядра, как правило, не требуют полной перекомпиляции ядра при изменении состава аппаратного обеспечения компьютера.
- Модульные ядра предоставляют тот или иной механизм подгрузки модулей ядра, поддерживающих то или иное аппаратное обеспечение (например, драйверов). При этом подгрузка модулей может быть как динамической (выполняемой «на лету», без перезагрузки ОС, в работающей системе), так и статической (выполняемой при перезагрузке ОС после переконфигурирования системы на загрузку тех или иных модулей).

Микроядро

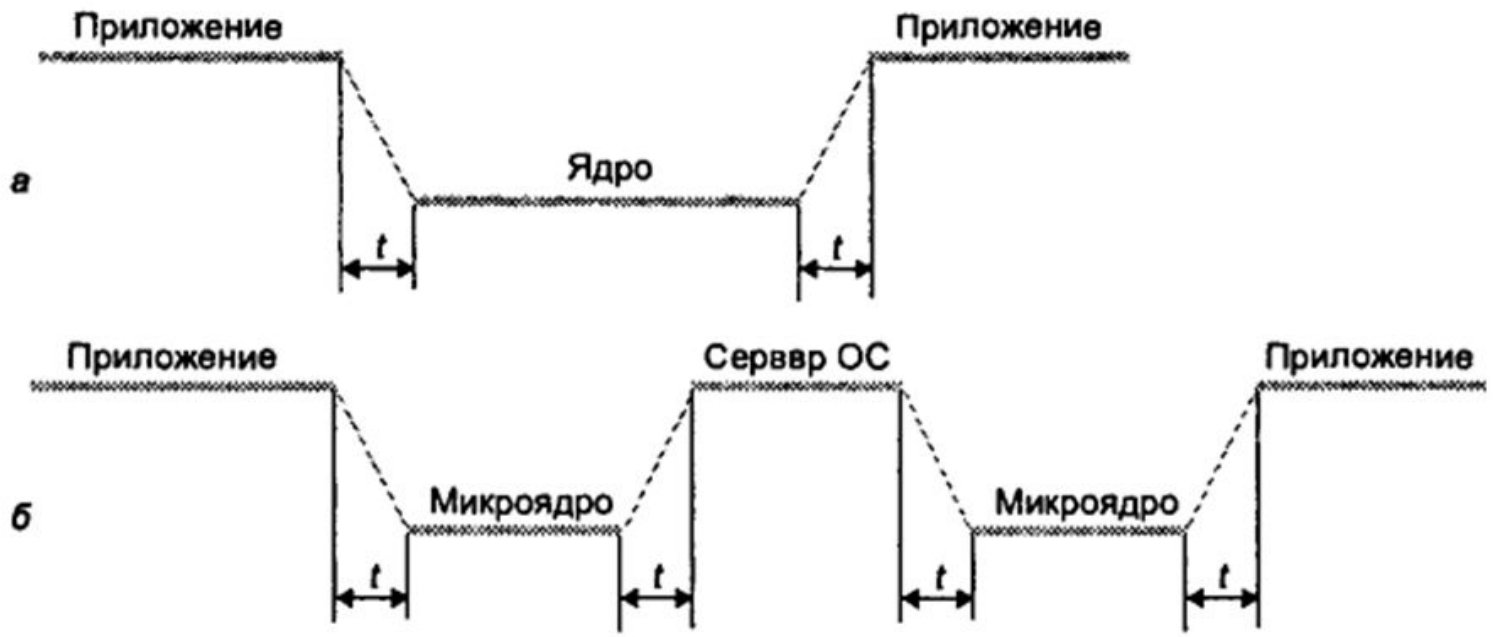
- Развитием ядерного подхода явилась архитектура на основе микроядра.
- Микроядро работает в привилегированном режиме и выполняет только минимум функций по управлению аппаратурой, в то время как функции ОС более высокого уровня выполняют специализированные компоненты ОС – серверы, работающие в пользовательском режиме.



Реализация системного вызова в микроядерной архитектуре



Смена режимов при выполнении системного вызова к микроядру



Достоинства и недостатки микроядра

- При микроядерном построении ОС работает более медленно, так как часто выполняются переходы между привилегированным и пользовательским режимом.
- Систему проще функционально развивать, добавляя, модифицируя или исключая серверы пользовательского режима.
- Серверы хорошо защищены друг от друга.

Микроядро Mach

- **Mach** – это микроядро ОС, разработанное в Carnegie Mellon University в исследовательских целях для решения задач с использованием распределенных вычислений. Это одно из микроядер первого поколения.
- Проект разрабатывался с 1985 по 1994 год, закончился на Mach 3.0. Сравнения проведенные в 1997 году показали, что Unix построенный на Mach 3.0 на 50 % медленнее чем традиционный Unix.
- Некоторое число разработчиков продолжило Mach исследования, например, микроядро второго поколения L4.
- Наиболее удачным примером коммерческой реализации можно считать Mac OS X, которая использует сильно модифицированный Mach 3.0 .

Модификации ядерного и микроядерного подходов

- Большинство современных проектов коммерческих ОС используют различные комбинации подходов на основе ядра и микроядра, например:
  - гибридное ядро;
  - наноядро.

Гибридное ядро

- **Гибридное ядро** (*Hybrid kernel*) — модифицированные микроядра, позволяющие для ускорения работы запускать «несущественные» части ОС в пространстве ядра.
- Примером реализации гибридного ядра можно считать ОС Linux и Windows 2000-2008.

Наноядро

- **Наноядро** – архитектура ядра ОС компьютеров, в рамках которой крайне упрощённое и минималистичное ядро выполняет лишь одну задачу – обработку аппаратных прерываний, генерируемых устройствами компьютера. После обработки прерываний от аппаратуры наноядро, в свою очередь, посылает информацию о результатах обработки вышележащему программному обеспечению при помощи того же механизма прерываний.

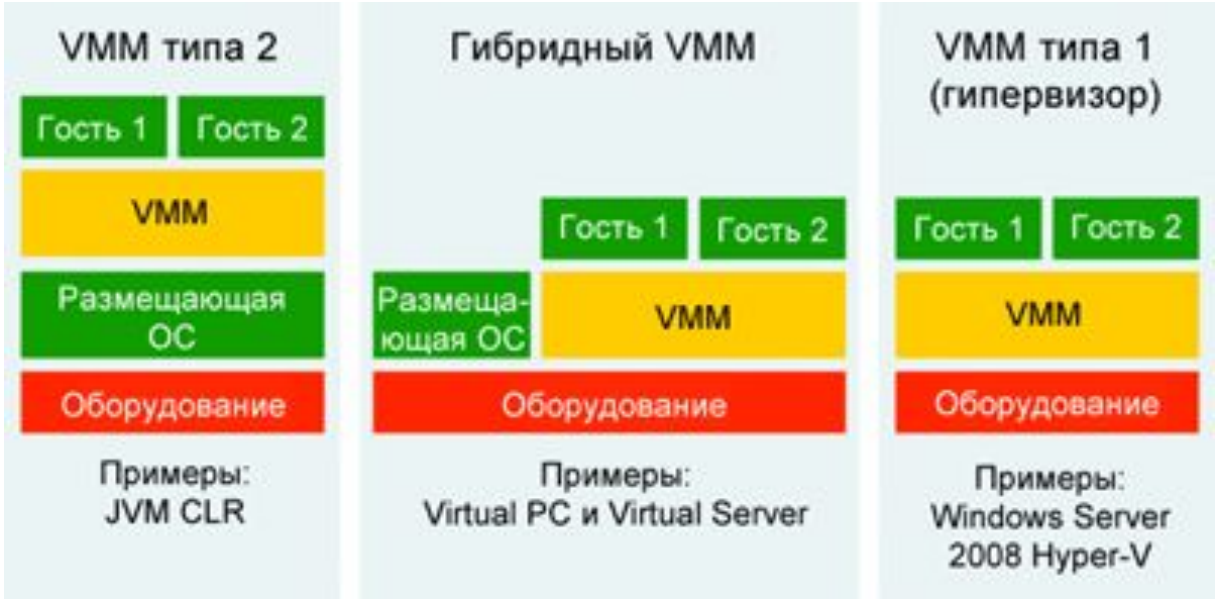
Наноядро

- Наиболее часто в современных компьютерах наноядра используются для виртуализации аппаратного обеспечения реальных компьютеров или для реализации механизма гипервизора, с целью позволить нескольким или многим различным ОС работать одновременно и параллельно на одном и том же компьютере.
- Наноядра также могут использоваться для обеспечения переносимости (портабельности) ОС на разное аппаратное обеспечение или для обеспечения возможности запуска «старой» ОС на новом аппаратном обеспечении без ее полного переписывания и портирования.

Поддержка виртуализации

- Виртуализация серверов, говоря обобщенно, позволяет взять одно физическое устройство и установить на нем (и использовать одновременно) две или более среды ОС.
- Все это направлено на уменьшение затрат, улучшение использования ресурсов, оптимизацию инфраструктуры.

Типы виртуализации

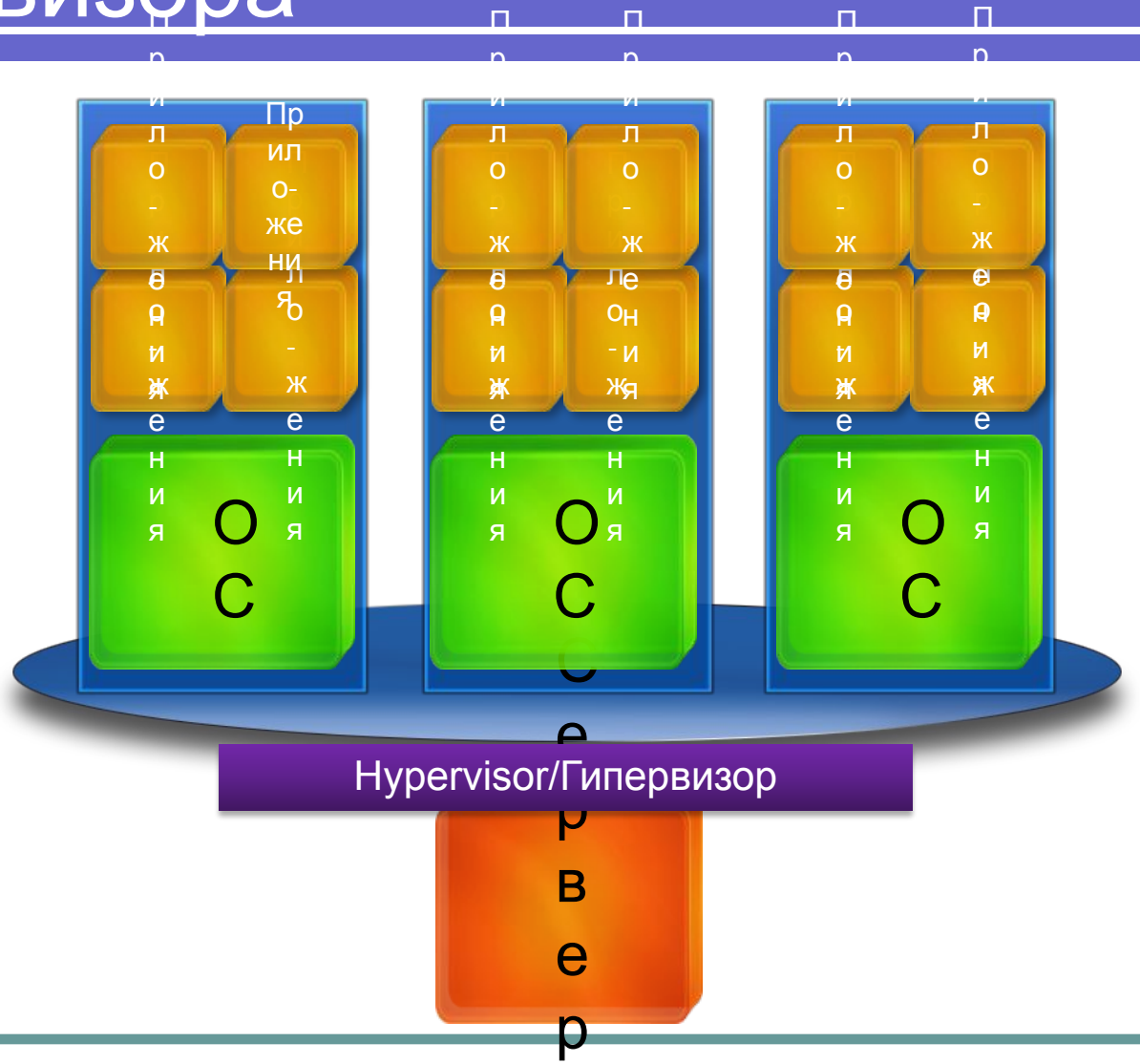


- VMM – монитор виртуальных машин
- В архитектуре VMM типа 1, уровень VMM работает прямо над оборудованием. Это часто называется уровнем гипервизора. Эта архитектура была первоначально разработана IBM в 1960-е годы для мэйнфреймов и недавно стала доступной на платформах x86/x64.

Гипервизор

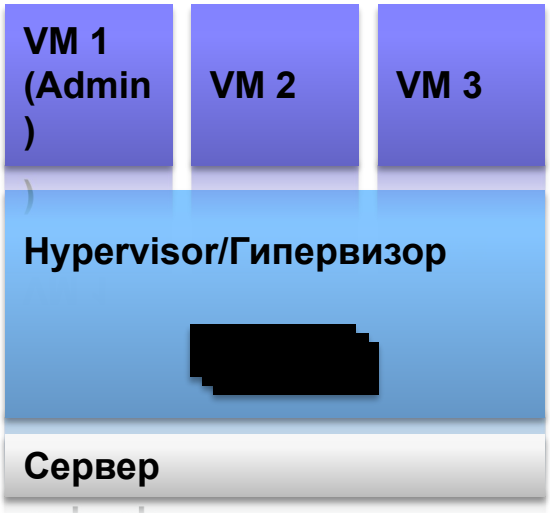
- **Гипервизор** — программа или аппаратная схема, обеспечивающая или позволяющая одновременное, параллельное выполнение нескольких операционных систем на одном и том же хост-компьютере.
- Гипервизор также обеспечивает изоляцию операционных систем друг от друга, защиту и безопасность, разделение ресурсов между различными запущенными ОС и управление ресурсами.

Виртуализация на базе гипервизора

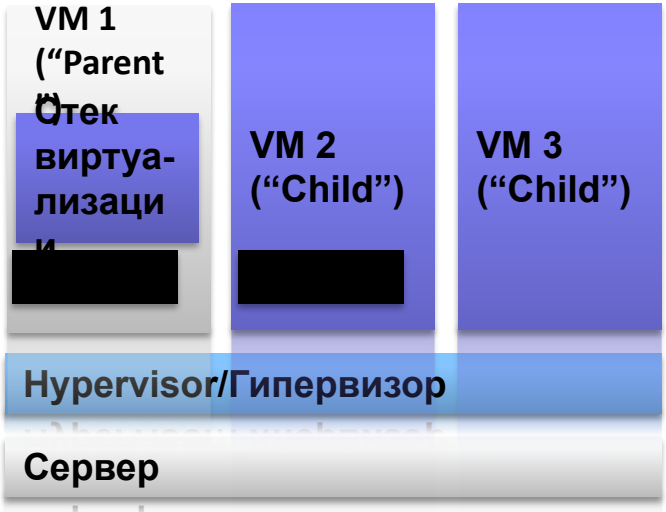


Типы архитектуры гипервизора

Монолитный



Микроядерный



Монолитная модель гипервизора

- Монолитный подход размещает гипервизор/VMM в едином уровне, который также включает большинство требуемых компонентов, таких как ядро, драйверы устройств и стек ввода/вывода. Это подход, используемый такими решениями, как VMware ESX и традиционные системы мэйнфреймов.

Микроядерная модель гипервизора

- Микроядерный подход использует очень тонкий, специализированный гипервизор, выполняющий лишь основные задачи обеспечения изоляции разделов и управления памятью. Этот уровень не включает стека ввода/вывода или драйверов устройств. В этой архитектуре стек виртуализации и драйверы конкретных устройств расположены в специальном разделе, именуемом родительским разделом.

Объектно-ориентированный подход

- Развитием технологии расширяемых модульных систем является **объектно-ориентированный подход**, при котором каждый программный компонент ОС является функционально изолированным от других. Основным понятием этого подхода является “объект”.
- *Объект* – это единица программ и данных, взаимодействующая с другими объектам посредством приема и передачи сообщений. Объект может быть представлением как некоторых конкретных вещей – прикладной программы или документа, так и некоторых абстракций – процесса, события.
- Программы (функции) объекта определяют перечень действий, которые могут быть выполнены над данными этого объекта. Объект-клиент может обратиться к другому объекту, послав сообщение с запросом на выполнение какой-либо функции объекта-сервера.

ООП: достоинства и недостатки

- Построение ОС на базе объектно-ориентированного подхода имеет следующие достоинства:
  - аккумуляция удачных решений в форме стандартных объектов и создание новых объектов на их базе с помощью механизма наследования;
  - предотвращение несанкционированного доступа к данным за счет их инкапсуляции во внутренние структуры объекта;
  - структурированность системы, состоящей из набора хорошо определенных объектов.
- В качестве основных недостатков объектно-ориентированного похода следует выделить сложность управления объектами и как следствие более медленную работу системы.

Классификация ОС

Особенности аппаратных платформ

Особенности аппаратных платформ

- На свойства ОС непосредственное влияние оказывают аппаратные средства, на которые она ориентирована. По типу аппаратуры различают ОС персональных компьютеров, мини-компьютеров, мэйнфреймов, кластеров и сетей ЭВМ.
- Наряду с ОС, ориентированными на совершенно определенный тип аппаратной платформы, существуют системы, специально разработанные таким образом, чтобы они могли быть легко перенесены с компьютера одного типа на компьютер другого типа.
- В этих системах аппаратно-зависимые места тщательно локализованы, так что при переносе системы на новую платформу переписываются только они. Средством, облегчающем перенос остальной части ОС, является написание ее на машинно-независимом языке, например, на Си, который и был разработан для программирования ОС.
- Наиболее ярким примером такой ОС является популярная система UNIX.

Примеры специализированных систем

Обработка цифровых изображений, устройства печати

Устройства розничной торговли, банкоматы, кассовые аппараты

Игровые автоматы

Мобильные телефоны, автомобильные системы

Телевизионные приставки, цифровые видеомагнитофоны, бытовая автоматизация, медиаплееры

Промышленная автоматизация

Медицинские системы

Измерительные приборы

Высокопроизводительные системы

- Логические процессоры (hyperthreading) – многопоточность в рамках одного ядра
- Многоядерные процессоры
- SMP-системы (оперативная память физически представляет последовательное адресное пространство, доступ к которому имеют одновременно все процессоры системы по единой шине)
- NUMA (Non-Uniform Memory Architecture)

Планы Intel

- В начале 21-века компания Intel прогнозировала появление к 2010 году процессоров с частотой 20 ГГц.
- На 2009-2010 год был запланирован проект Keifer (в настоящее время закрыт). В качестве ориентира было установлено количество ядер равное 32 в 8 узлах, каждое из которых способно обрабатывать одновременно до четырех потоков.

Многоядерные процессоры

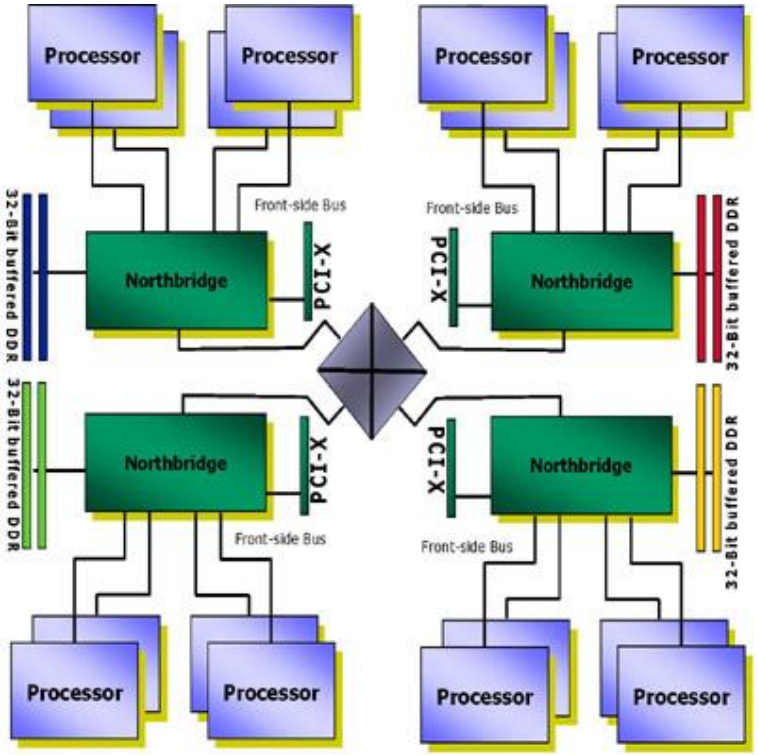
- SUN (Niagara 2 – 8 ядер с 8-мя потоками каждое, Sun Rock – 4 модуля по 4 ядра)
- Intel (Xeon 7500 – до 8 ядер и 16 потоков)
- AMD (Opteron – 6 ядер)

Перспективы многоядерных процессоров

- Intel – гомогенные структуры
- AMD – гетерогенные структуры

NUMA

- Процессоры группируются в узлы (Nodes).
- В каждом узле несколько CPU и память (SMP-система, но за счет минимальной компоновки элементов достигается высокая пропускная способность между процессором и локальной памятью модуля).
- Узлы объединяются высокопроизводительной шиной.

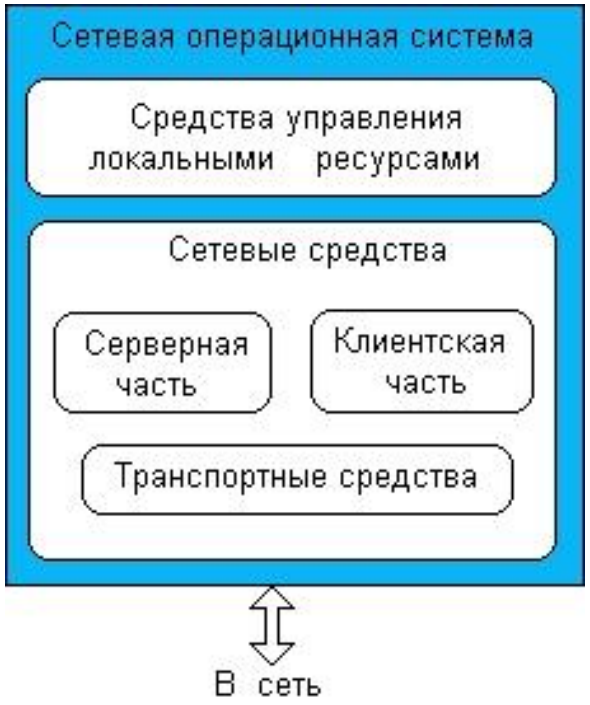


Сетевые операционные системы

- Операционная система компьютерной сети во многом аналогична ОС автономного компьютера – она также представляет собой комплекс взаимосвязанных программ, который обеспечивает удобство работы пользователям и программистам путем предоставления им некоторой виртуальной вычислительной машины, и реализует эффективный способ разделения ресурсов между множеством выполняемых в сети процессов.

Сетевые операционные системы

- сетевые средства, в свою очередь, можно разделить на три компонента:
  - средства предоставления локальных ресурсов и услуг в общее пользование – серверная часть ОС;
  - средства запроса доступа к удаленным ресурсам и услугам – клиентская часть ОС;
  - транспортные средства ОС, которые совместно с коммуникационной системой обеспечивают передачу сообщений между компьютерами сети.



Операционные системы

Эволюция операционных систем

Появление ОС

- Так как ОС появились и развивались в процессе конструирования компьютеров, то эти события исторически тесно связаны. Поэтому чтобы представить, как выглядели ОС, мы кратко рассмотрим следующие друг за другом поколения компьютеров.
- Первый настоящий цифровой компьютер был изобретен английским математиком Чарльзом Бэббиджем (Charles Babbage. 1792-1871). Хотя большую часть жизни Бэббидж посвятил попыткам создания своей «аналитической машины», он так и не смог заставить ее работать должным образом.

Появление ОС

- Это была чисто механическая машина, а технологии того времени не были достаточно развиты для изготовления многих деталей и механизмов высокой точности. Не стоит и говорить, что его аналитическая машина не имела операционной системы.
- Интересный исторический факт: Бэббидж понимал, что для аналитической машины ему необходимо программное обеспечение, поэтому он нанял молодую женщину по имени Ада Лавлейс (Ada Lovelace), дочь знаменитого британского поэта Лорда Байрона. Она и стала первым в мире программистом, а язык программирования Ada назван в ее честь.

Первый “баг”

- По легенде, **9 сентября** 1945 года учёные Гарвардского университета, тестировавшие вычислительную машину Mark II Aiken Relay Calculator, нашли мотылька, застрявшего между контактами электромеханического реле и Грейс Хоппер произнесла этот термин. Извлечённое насекомое было вклеено в технический дневник, с сопроводительной надписью: «First actual case of bug being found» (англ. «первый случай в практике, когда был обнаружен жучок»). Этот забавный факт положил начало использованию слова «баг» в значении «ошибка».

Этапы эволюции

- 1 этап (1940-60)

  системный монитор, ранние пакетные системы
- 2 этап (1965-75)

  мультипрограммирование, пакетные ОС и ОС разделения времени
- 3 этап (1970-80) ОС мини-ЭВМ
- 4 этап (1980-90) ОС ПК
- 5 этап (1990-наст.вр.) корпоративные ОС

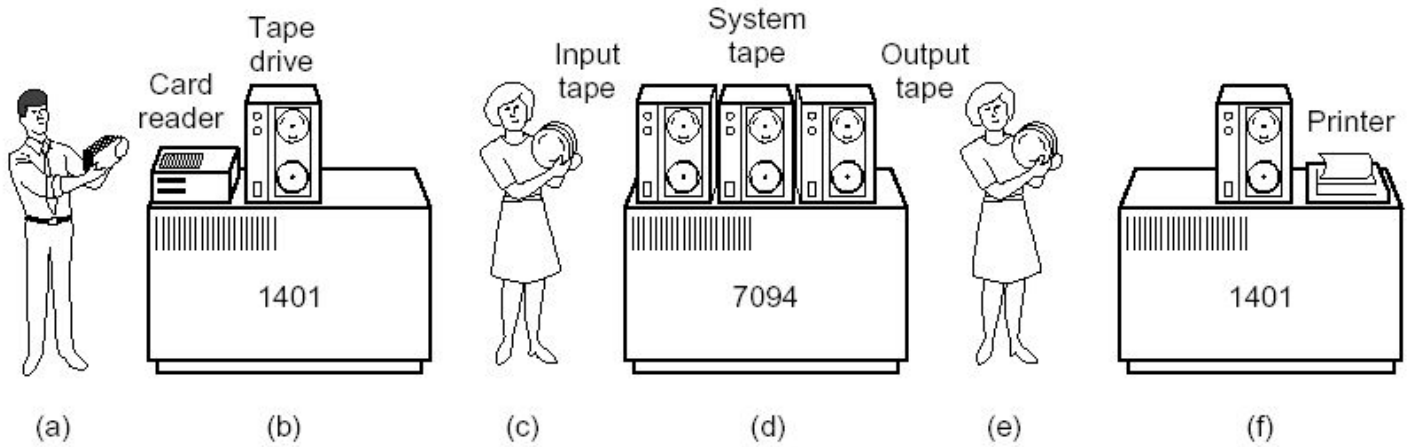
1 этап (1940-60)

- Середина 40-х XX-века – первые ламповые вычислительные устройства. ОС еще не появились, все задачи организации вычислительного процесса решались программистом вручную с пульта управления.
- С середины 50-х годов – новая техническая база – полупроводниковые элементы:
  - выросли технические характеристики ЭВМ: быстродействие процессоров, объемы оперативной и внешней памяти, надежность;
  - появились первые **алгоритмические языки**, и появился новый тип системного программного обеспечения – трансляторы;
  - были разработаны первые системные управляющие программы – **мониторы**.
- **Программные мониторы** – прообраз современных ОС, первые системные программы, предназначенные для управления вычислительным процессом.

1 этап (1940-60)

- Программные мониторы предоставляли пакетный режим обслуживания на базе **язык управления заданиями**, с помощью которого программист сообщал системе и оператору, какие действия и в какой последовательности он хотел бы выполнить на вычислительной машине. Типовой набор директив обычно включал признак начала отдельной работы, вызов транслятора, вызов загрузчика, признаки начала и конца исходных данных. Оператор составлял **пакет заданий**, которые в дальнейшем без его участия последовательно запускались на выполнение монитором. Кроме того, монитор был способен самостоятельно обрабатывать наиболее распространенные аварийные ситуации, возникающие при работе пользовательских программ, такие как отсутствие исходных данных, переполнение регистров, деление на ноль, обращение к несуществующей области памяти и т. д.

Ранние системы пакетной обработки (1 этап)



(a) Программист приносит перфокарты к устройству ввода 1401.
(b) Устройство 1401 считывает пакет заданий на ленточный накопитель.
(c) Оператор переносит входную ленту на устройство 7094.
(d) Устройство 7094 выполняет вычисления.
(e) Оператор переносит выходную ленту на устройство 1401.
(f) Устройство 1401 выполняет печать результатов.

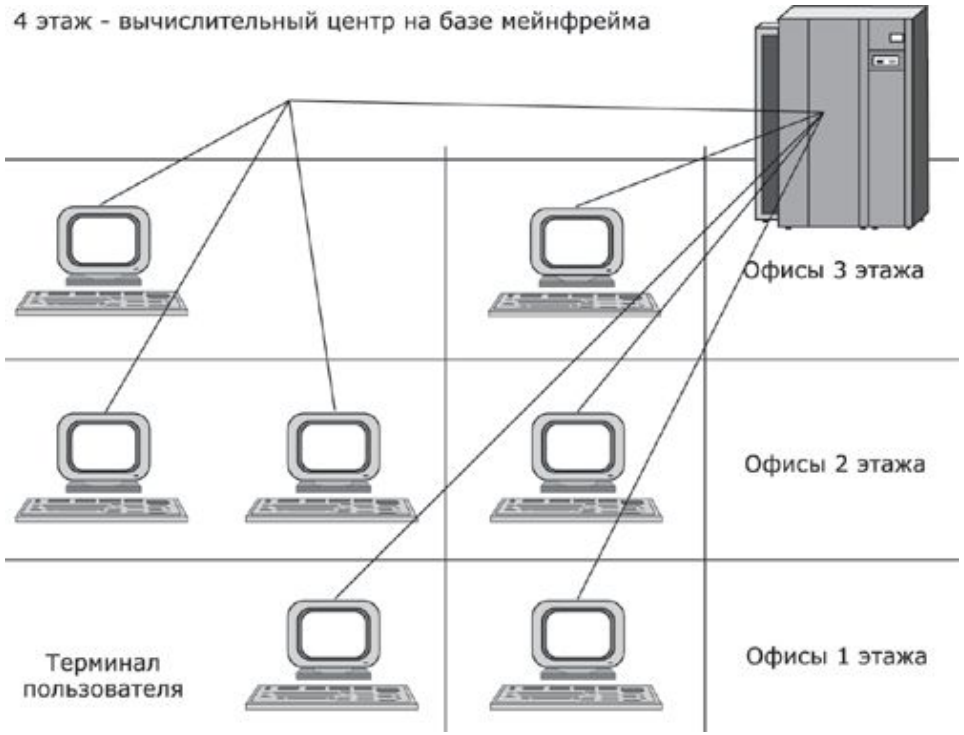
Ранние системы пакетной обработки (1 этап)

- Ранние системы пакетной обработки значительно сократили затраты времени на вспомогательные действия по организации вычислительного процесса, а значит, был сделан еще один шаг по повышению эффективности использования компьютеров. Однако при этом программисты-пользователи лишились непосредственного доступа к компьютеру, что снижало эффективность их работы — внесение любого исправления требовало значительно больше времени, чем при интерактивной работе за пультом машины.

2 этап (1965-75)

- 1965-1975 годы переход к ИС, новое поколение ЭВМ – IBM/360, многопроцессорная ЭВМ для централизованных вычислений.
- Реализованы основные концепции, присущие современным ОС:
  - мультипрограммирование,
  - мультипроцессирование,
  - многотерминальный режим,
  - виртуальная память,
  - файловые системы,
  - разграничение доступа и сетевая работа.
- Мультипрограммирование было реализовано в двух вариантах – пакетная обработка и разделение времени.
- Для поддержания удаленной работы терминалов в ОС появились специальные программные модули, реализующие различные (в то время, как правило, нестандартные) протоколы связи. Поэтому эти ОС можно считать прообразом современных сетевых ОС.
- 1965-69 годы – разработка фирмами Bell Telephone Lab., General Electric и Массачусетским технологическим институтом новой многозадачной ОС – **Multics** (MULTiplexed Information and Computing Service), которая была потом переименована на UNIX.

2 этап – многотерминальные системы



- Терминалы, выйдя за пределы вычислительного центра, рассредоточились по всему предприятию.

2 этап – разделение времени

- Желание сократить время ожидания ответа привело к разработке режима разделения времени, варианту многозадачности, при котором у каждого пользователя есть свой диалоговый терминал. Так как люди, отлаживая программы, обычно выдают короткие команды чаще, чем длинные, то компьютер может обеспечивать быстрое интерактивное обслуживание нескольких пользователей. При этом он может работать над большими пакетами в фоновом режиме, когда центральный процессор не занят другими заданиями.
- Первая серьезная система с режимом разделения времени **CTSS (Compatible Time Sharing System —** совместимая система разделения времени) была разработана в Массачусетском технологическом институте (M.I.T.) на специально переделанном компьютере IBM 7094. Однако режим разделения времени не стал действительно популярным до тех пор. пока не получили широкого распространения необходимые технические средства защиты.

2 этап – многотерминальные системы

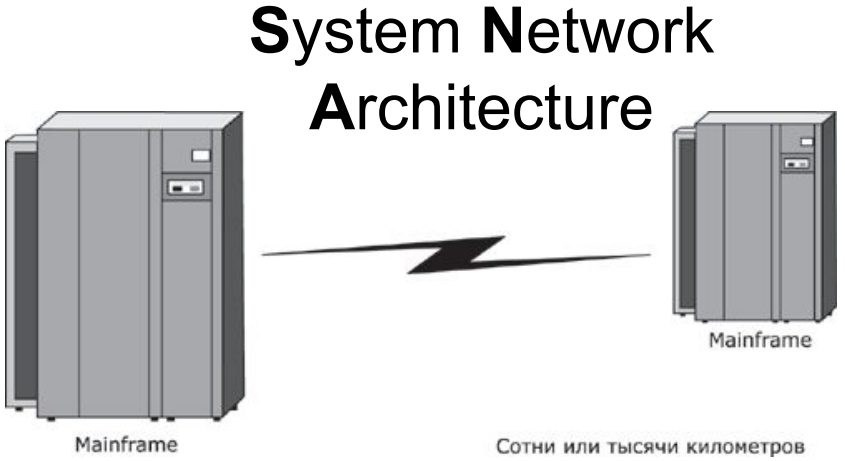
- Многотерминальный режим использовался не только в системах разделения времени, но и в системах пакетной обработки. При этом не только оператор, но и все пользователи получали возможность формировать свои задания и управлять их выполнением со своего терминала. Такие ОС получили название **систем удаленного ввода заданий**.
- Для поддержки удаленной работы терминалов в ОС появились специальные программные модули, реализующие различные (как правило, нестандартные) протоколы связи. Такие ВС с **удаленными терминалами**, сохраняя централизованный характер обработки данных, в какой-то степени являлись прообразом современных компьютерных сетей, а соответствующее системное ПО – прообразом сетевых ОС.

3 этап (1970-80)

- Начало 70-х годов – первые сетевые ОС, которые в отличие от многотерминальных ОС позволяли не только рассредоточить пользователей, но и организовать распределенное хранение и обработку данных между несколькими компьютерами, связанными сетью.
- 1969 год – начало работ Министерства обороны США по объединению суперкомпьютеров оборонных и научно-исследовательских центров в единую сеть ARPANET, которая явилась отправной точкой для создания глобальной сети Интернет.
- Середина 70-х годов – широкое распространение получили мини-ЭВМ (PDP-11, Nova, HP) на базе технологии БИС, которая позволила реализовать достаточно мощные функции при сравнительно невысокой стоимости компьютера. Архитектура мини-ЭВМ была значительно упрощена по сравнению с мэйнфреймами, что нашло отражение и в их ОС. Многие функции мультипрограммных многопользовательских ОС мэйнфреймов были усечены, учитывая ограниченность ресурсов мини-компьютеров.
- ОС мини-компьютеров часто стали делать специализированными, например, только для управления в реальном времени (ОС RT-11 для PDP-11) или только для поддержания режима разделения времени (RSX-11M для PDP-11). Эти ОС не всегда были многопользовательскими, что во многих случаях оправдывалось невысокой стоимостью компьютеров.

3 этап – объединение удаленных мэйнфреймов с помощью SNA

- 1974 год – создание компанией IBM сетевой архитектуры для своих мэйнфреймов.

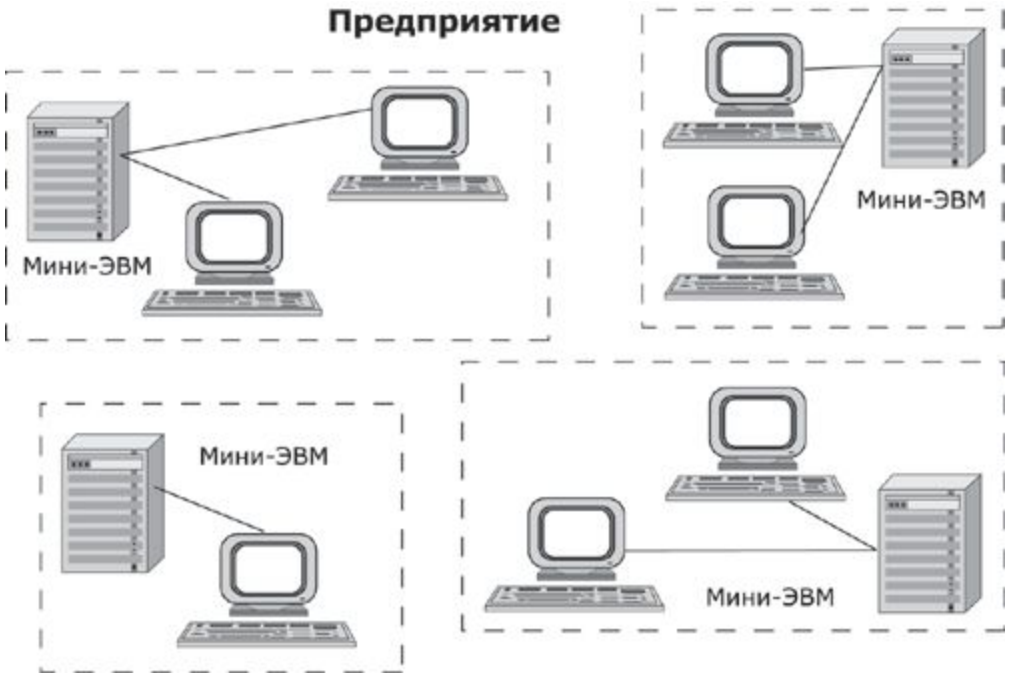


- Эта многоуровневая архитектура (во многом прообраз модели OSI) обеспечивала взаимодействие типа «терминал-терминал», «терминал-компьютер» и «компьютер-компьютер» по глобальным связям.

4 этап (1980-90)

- Постоянное развитие версий ОС UNIX для ЭВМ различных архитектур.
- Начало 80-х годов – появление **персональных компьютеров** (ПК), которые стали мощным катализатором для бурного роста ЛВС, в результате чего поддержка сетевых функций стала для ОС ПК необходимым условием.
- Также в 80-е годы – приняты основные стандарты на коммуникационные технологии для ЛВС (например, Ethernet). Это позволило обеспечить совместимость сетевых ОС на нижних уровнях, а также стандартизовать интерфейс ОС с драйверами сетевых адаптеров.
- 1981 год – первая ОС компании Microsoft для ПК. **MS-DOS** было однопрограммной однопользовательской ОС с интерфейсом командной строки. Недостающие функции MS-DOS (например, интерфейсные и сетевые) компенсировались внешними программами. Начиная с MS-DOS v3.1 к файловой системе добавились необходимые для сетевой работы средства блокировки файлов и записей (совместная работа пользователей).
- 1983 год – первая сетевая ОС компании **Novell OS-Net** для сетей со звездообразной топологией. После выпуска фирмой IBM ПК типа PC XT, компания Novell разработала сетевую ОС **NetWare 86** для ПК.
- 1987 год – Microsoft и IBM выпустили первую многозадачную ОС **OS/2** для ПК на базе МП Intel 80286. Эта ОС поддерживала вытесняющую многозадачность, многопоточность, виртуальную память, графический пользовательский интерфейс и виртуальную машину для выполнения DOS-приложений.
- Начиная с МП Intel 80286 с поддержкой мультипрограммирования, перенос ОС UNIX на ПК, например, версия UNIX компании Santa Cruz Operation (**SCO UNIX**).

4 этап – мини-ЭВМ и ЛВС



5 этап (1990 – …)

- 90-е годы – практически все ОС стали сетевыми. Сетевые функции встраиваются в ядро ОС, являясь ее неотъемлемой частью.
- Появились специализированные ОС, которые предназначены исключительно для выполнения коммуникационных задач. Например, сетевая ОС IOS компании Cisco Systems, работающая в маршрутизаторах, организует в мультипрограммном режиме выполнение набора программ, каждая из которых реализует один из коммуникационных протоколов.
- Вторая половина 90-х годов – особая поддержка со стороны ОС средств работы с Интернетом.
- Появилось понятие корпоративной сетевой ОС. Корпоративная ОС отличается способностью хорошо и устойчиво работать в крупных сетях. Таким сетям органически присуща высокая степень гетерогенности программных и аппаратных средств, поэтому корпоративная ОС должна взаимодействовать с ОС разных типов и работать на различных аппаратных платформах.

5 этап – сетевые и распределенные ОС

- Сетевые ОС несущественно отличаются от однопроцессорных операционных систем. Ясно, что они нуждаются в сетевом интерфейсном контроллере и специальном низкоуровневом программном обеспечении, поддерживающем работу контроллера, а также в программах, разрешающих пользователям удаленную регистрацию в системе И доступ к удаленным файлам. Но эти дополнения по сути не изменяют структуры операционной системы. Каждый компьютер работает под управлением локальной ОС и имеет своего собственного локального пользователя (или пользователей).

5 этап – сетевые и распределенные ОС

- Распределенная операционная система, напротив, представляется пользователям традиционной однопроцессорной системой, хотя она и составлена из множества процессоров. При этом пользователи не должны беспокоиться о том, где работают их программы или где расположены файлы; все это должно автоматически и эффективно обрабатываться самой операционной системой.

5 этап – сетевые и распределенные ОС

- Чтобы создать настоящую распределенную операционную систему, недостаточно просто добавить несколько страниц кода к однопроцессорной ОС, так как распределенные и централизованные системы имеют существенные различия. Поэтому требуется более сложный алгоритм загрузки процессоров для оптимизации распараллеливания.
- Наличие задержек при передаче данных в сетях означает, что эти алгоритмы должны работать с неполной, устаревшей или даже неправильной информацией. Эта ситуация радикально отличается от однопроцессорной системы, в которой ОС обладает полной информацией относительно состояния системы.

5 этап – сетевые и распределенные ОС

- К настоящему времени достаточно явно определилась тройка лидеров в классе корпоративных ОС – это Novell NetWare 6 (перестала обновляться), Microsoft Windows 2000-2008, а также Linux и UNIX-системы различных производителей аппаратных платформ.

Три эпохи программирования

Годы	Языки программирования	ОС
1952 – 1966	**Фортран**, Алгол-60, **Кобол**, Лисп, Бейсик, APL, **PL/I**	IBM OS/360, IBM OS/370
1967 – 1987	Simula-67, Алгол-68, Форт, **Пролог**, **Паскаль**, **Си**, Ada, Modula-2, Smalltalk, **C++**, Eiffel	DEC RT-11, DEC RSX-11, DEC VAX/VMS, UNIX, CP/M, MS-DOS
1988 – 2007	Oberon, **Visual Basic, Java, C#, Perl, PHP**, Python	Windows, OS/2, Mac OS, Linux, Palm OS, Pocket PC

Операционные системы

История Unix и Linux

Предшественники

- В 1957 году в Bell Labs была начата работа по созданию операционной системы для собственных нужд. Под руководством Виктора Высотского (русского по происхождению) была создана система BESYS. Впоследствии он возглавил проект Multics, а затем стал главой информационного подразделения Bell Labs.
- В 1964 году появились компьютеры третьего поколения, для которых возможности BESYS уже не подходили. Высотский и его коллеги приняли решение не разрабатывать новую собственную ОС, а подключиться к совместному проекту General Electric и Массачусетского технологического института Multics. Существенную поддержку проекту оказала корпорация AT&T, но в 1969 году она вышла из проекта, поскольку он не приносил никаких финансовых выгод.

Разработчики

- Первоначально UNIX была разработана в конце 1960-х годов сотрудниками Bell Labs, в первую очередь Кеном Томпсоном, Денисом Ритчи и Дугласом МакИлроем.
  - **Кен Томпсон** (англ. Kenneth Thompson; род. 4 февраля 1943) — пионер компьютерной науки, известен за свой вклад в создание языка программирования Си и операционной системы UNIX
  - **Денис Ритчи** (Dennis MacAlistair Ritchie; род. 9 сентября 1941) — компьютерный специалист, известен по участию в создании ALTRAN, B, BCPL, Си, Multics, и Unix.

Unix

- *«UNIX прост. Но надо быть гением, чтобы понять его простоту»* Деннис Ритчи
- *«UNIX не был разработан так, чтобы отгораживать своих пользователей от глупостей, поскольку это отгородило бы их от делания умных вещей»* Дуг Гвин
- *«UNIX никогда не скажет “пожалуйста”»* Роб Пайк

Первые версии UNIX

- В **1969** году Кен Томпсон, стремясь реализовать идеи, что были положены в основу MULTICS, но на более скромном аппаратном обеспечении (DEC PDP-7), написал первую версию новой операционной системы, а Брайан Керниган придумал для неё название — UNICS (UNIplexed Information and Computing System) — в противовес MULTICS (MULTIplexed Information and Computing Service). Позже это название сократилось до UNIX.
- В **1970** г. вышла версия для PDP-11, наиболее успешного семейства миникомпьютеров 1970-х (в СССР оно известно как СМ ЭВМ).
- В **1973** г. было принято решение переписать ядро системы на языке Си. UNIX стал первой ОС, практически полностью написанной на языке высокого уровня, что существенно упростило перенос системы на другие архитектуры.
- 15 октября 1973 г. была представлена четвёртая версия UNIX. Вскоре появилась UNIX Version 5, с 1974 года начавшая распространяться бесплатно среди университетов и академических учреждений.
- К 1975 году вышла UNIX Version 6. К 1978 г. система была установлена более чем на 600 машинах, прежде всего, в университетах.
- Версия 7 была последней единой версией UNIX.

Раскол

- В начале 1980-х компания AT&T, которой принадлежали Bell Labs, осознала ценность UNIX и начала создание коммерческой версии UNIX. Эта версия, поступившая в продажу в 1982 году, носила название UNIX System III и была основана на седьмой версии системы. Несколько раньше Билл Джой из университета Беркли создал собственный дистрибутив, основанный на UNIX Version 7. Этот дистрибутив получил название BSD (англ. Berkeley Software Distribution).
- Поворотным моментом в истории UNIX стала реализация в 1980 г. стека протоколов TCP/IP. Было предложено два интерфейса программирования сетевых приложений: Berkley sockets и интерфейс транспортного уровня TLI (англ. Transport Layer Interface).
- Интерфейс Berkley sockets был разработан в университете Беркли и использовал стек протоколов TCP/IP, разработанный там же. TLI был создан AT&T в соответствии с определением транспортного уровня модели OSI и впервые появился в системе System V версии 3. Это, как и другие соображения, вызвало окончательное размежевание между двумя ветвями UNIX – BSD (университета Беркли) и System V (коммерческая версия от AT&T). Впоследствии, многие компании, лицензировав System V у AT&T, разработали собственные коммерческие разновидности UNIX, такие, как AIX, HP-UX, IRIX, Solaris.

Современность

- После разделения компании AT&TПосле разделения компании AT&T, товарный знакПосле разделения компании AT&T, товарный знак UNIX и права на оригинальный исходный кодПосле разделения компании AT&T, товарный знак UNIX и права на оригинальный исходный код неоднократно меняли владельцев, в частности, длительное время принадлежали компании Novell.
- В 1993 годуВ 1993 году Novell передала права на товарный знак и на сертификацию программного обеспечения на соответствие этому знаку консорциуму X/Open, который затем объединился с Open Software Foundation, образовав консорциум The Open GroupВ 1993 году Novell передала права на товарный знак и на сертификацию программного обеспечения на соответствие этому знаку консорциуму X/Open, который затем объединился с Open Software Foundation, образовав консорциум The Open Group. Он объединяет ведущие компьютерные корпорации и государственные организации, в том числе IBMВ 1993 году Novell передала права на товарный знак и на сертификацию программного обеспечения на соответствие этому знаку консорциуму X/Open, который затем объединился с Open Software Foundation, образовав консорциум The Open Group. Он объединяет ведущие компьютерные корпорации и государственные организации, в том числе IBM, Hewlett-PackardВ 1993 году Novell передала права на товарный знак и на сертификацию программного обеспечения на

Свободные UNIX системы

- GNU/Hurd
- GNU/Linux
- BSD
- Open Solaris

Свободные UNIX системы

- В результате урегулирования юридического дела, возбуждённого UNIX Systems Laboratories против университета Беркли и Berkeley Software Design Inc., было установлено, что университет может распространять BSD UNIX, в том числе и бесплатно. После этого были возобновлены эксперименты, связанные с BSD-версией UNIX.
- Вскоре разработка дистрибутива BSD была продолжена в нескольких направлениях одновременно, что привело к появлению проектов, известных как FreeBSD, NetBSD, OpenBSD, TrustedBSD и DragonFlyBSD.

Свободные UNIX системы

- В 1983 году Ричард Столлмэн объявил о создании проекта GNU — попытки создания свободной UNIX-подобной ОС с нуля, без использования оригинального исходного кода.
- В настоящее время существует два направления **GNU/Hurd** и **GNU/Linux.**
- **GNU/Hurd** – попытка создать современное ядро на основе микроядерной архитектуры Mach, проект разрабатывается с 1990 и в настоящее время не завершен. Наиболее актуальная версия на сегодняшний день – Debian GNU/Hurd K16.
- **GNU/Linux** – ядро ОС было опубликовано в 1991 году Линусом Товальдсом. В отличие от большинства других операционных систем, GNU/Linux не имеет единой «официальной» комплектации. Вместо этого GNU/Linux поставляется в большом количестве так называемых дистрибутивов– ядро ОС было опубликовано в 1991 году Линусом Товальдсом. В отличие от большинства других операционных систем, GNU/Linux не имеет единой «официальной» комплектации. Вместо этого GNU/Linux поставляется в большом количестве так называемых дистрибутивов, наиболее известными дистрибутивами GNU/Linux являются Slackware , ядро ОС было

Свободные UNIX системы

- **Open Solaris** – операционная система, ставшая продолжением развития Sun Solaris и на сегодня являющаяся основой Solaris.
- http://www.opensolaris.org/os/

Mac OS X

- Mac OS X значительно отличается от предыдущих версий Mac OS. Основа системы – ОС DarwinMac OS X значительно отличается от предыдущих версий Mac OS. Основа системы – ОС Darwin. Darwin — свободное программное обеспечение.
- Ядром ОС DarwinЯдром ОС Darwin является XNUЯдром ОС Darwin является XNU (рекурсивный акронимЯдром ОС Darwin является XNU (рекурсивный акроним от «Xnu Not Unix» — «Xnu — не Юникс»), в котором используется ядроЯдром ОС

Mac OS X

- Mac OS X включается в цену для новых компьютеров Apple Macintosh и официально не совместимо с остальными ПК, основанными на x86.
- Однако существуют взломанные версии Mac OS X, которые запускаются на x86 оборудовании близком по конфигурации к Mac. Они разрабатываются сообществом OSx86.