Intoduction\_2007

Введение в операционные системы:

*Назначение ОС:*

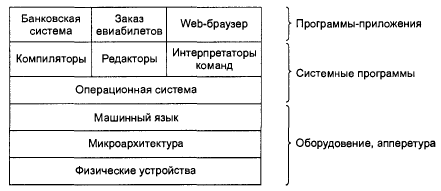
Определение ОС

Операционная система (ОС) – комплекс системных программ, обеспечивающий оптимальное управление ресурсами вычислительной системы в соответствии с некоторым критерием эффективности.

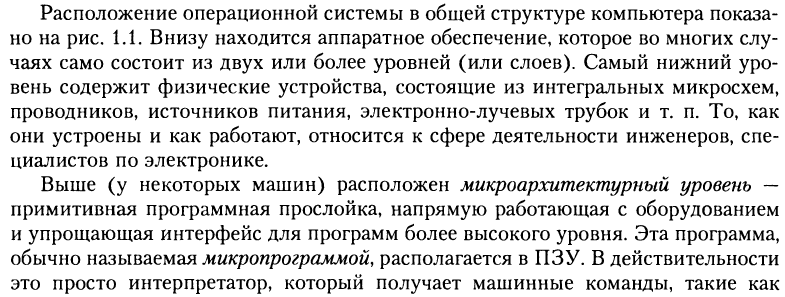
Критерием эффективности ОС может быть, например, пропускная способность (число выполненных задач за единицу времени) или реактивность (время реакции на некоторое событие) системы.

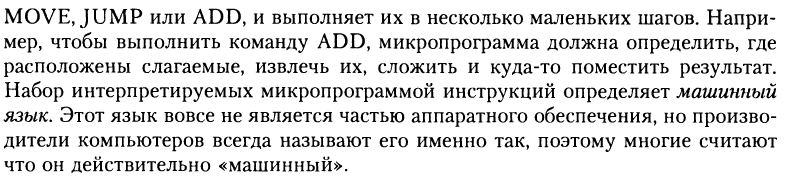
Вычислительная система (ВС) – это взаимосвязанная совокупность аппаратных средств вычислительной техники и программного обеспечения, предназначенная для обработки информации.

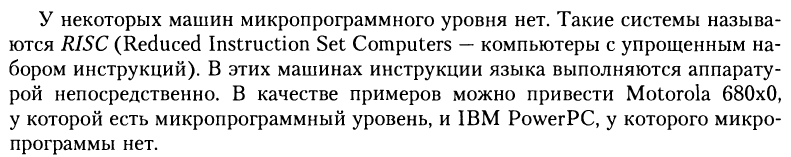
Уровни ВС

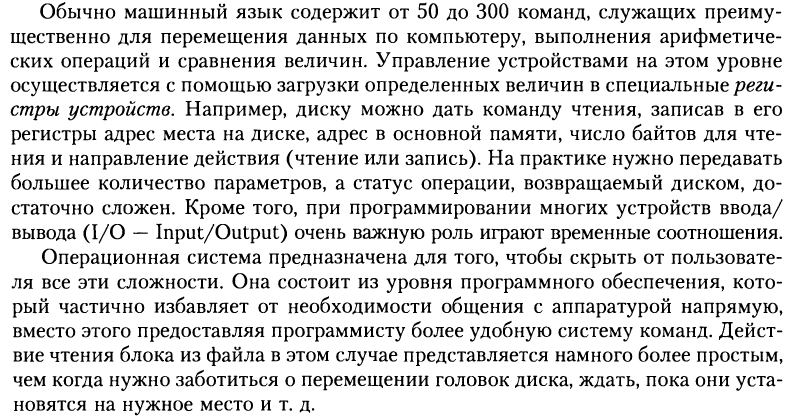


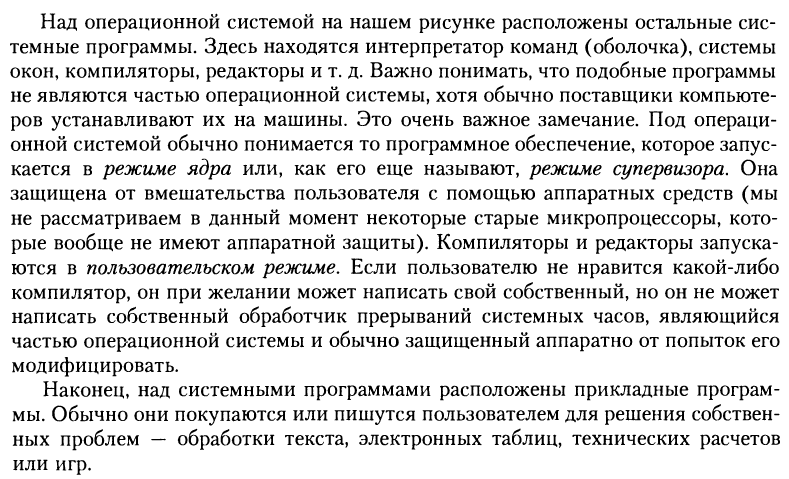
Расположение ОС











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.

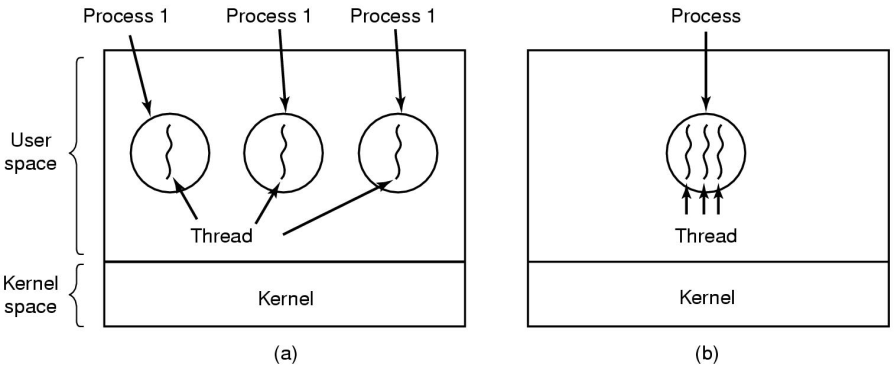
*Базовые концепции и термины*

Мультипрограммирование

* **Мультипрограммирование**, метод одновременно выполнения на одной ЭВМ нескольких программ, относящихся к различным задачам или различным ветвям одной и той же задачи.
* В настоящее время в большинстве ОС определены два типа единиц работы, между которыми разделяется процессор и другие ресурсы компьютера: процесс и поток.

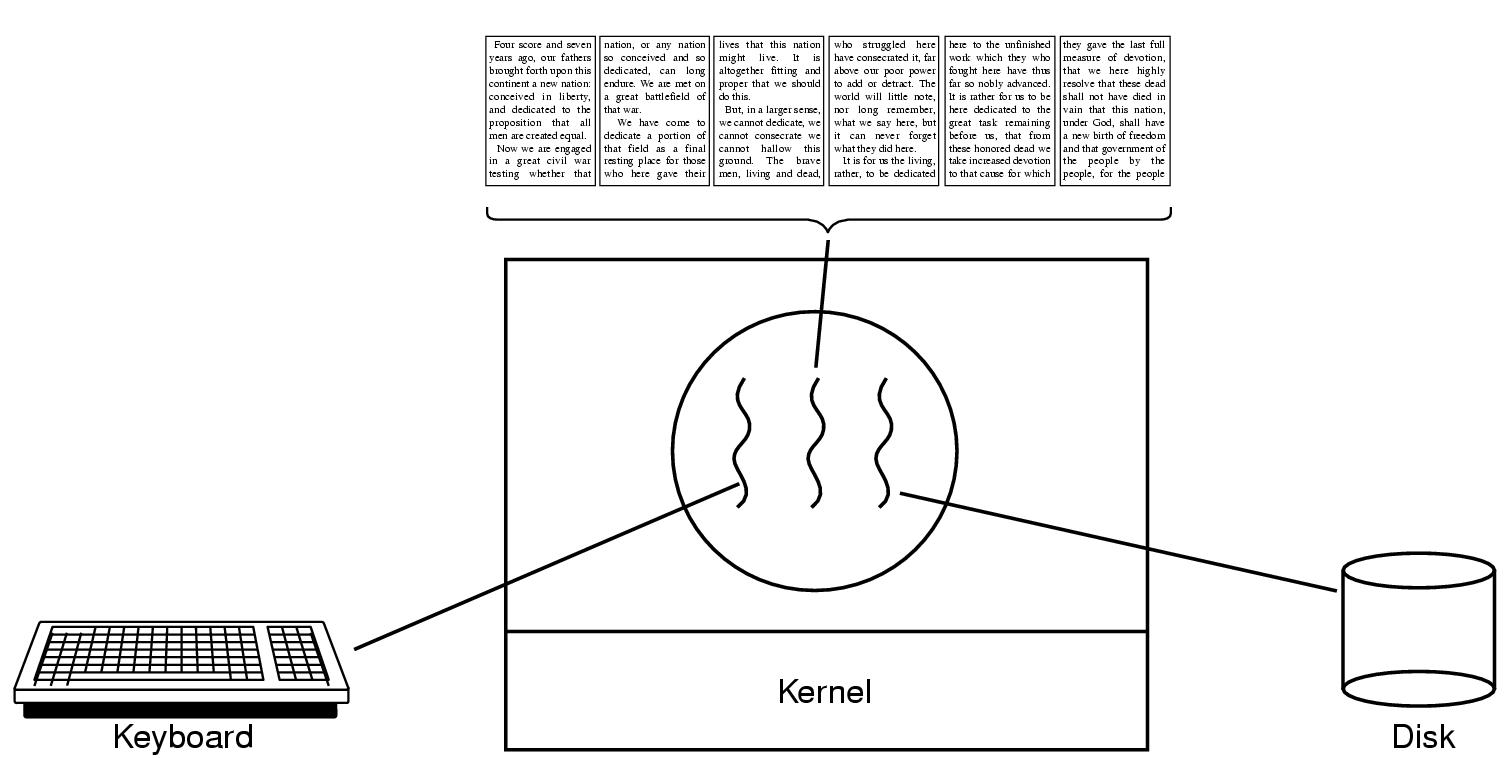
Процессы и потоки

* 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 ОС 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, и именно таким образом трактовалось понятие «процесс» до середины 80-х годов (например, в ранних версиях UNIX).



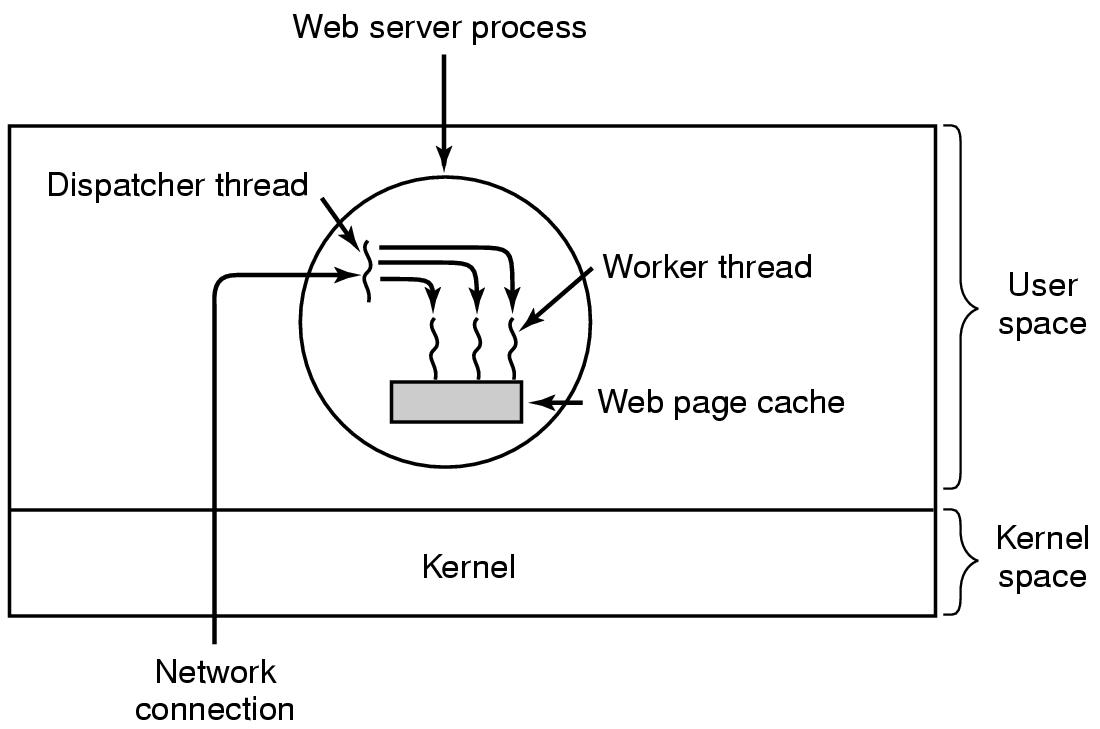
1. Три однопоточных процесса
2. Один процесс с тремя потоками

Пример многопоточного приложения: текстовый процессор



Текстовый процессор с тремя нитями

Пример многопоточного приложения: Web-сервер



*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 аппаратных платформ;
* особенности областей использования;
* структурная организация.

*Структурная организация операционных систем*

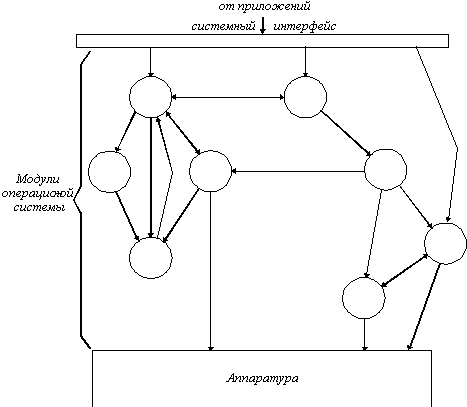
Структурная организация

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

Рассмотрим основные способы структурной организации ОС**.**

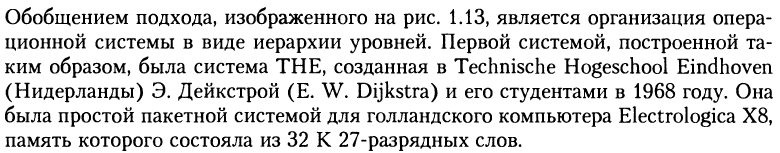
* + Монолитная структура
  + Многоуровневая структура
  + Привилегированное ядро
  + Микроядерная структура
  + Объектно-ориентированный подход

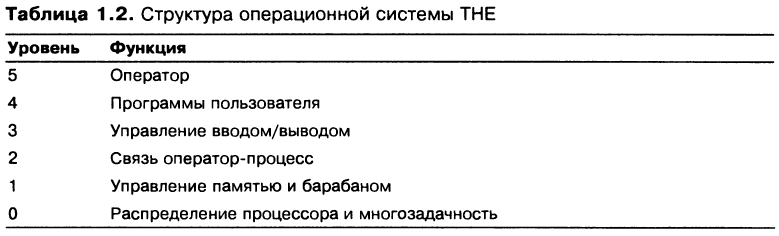
Монолитная структура

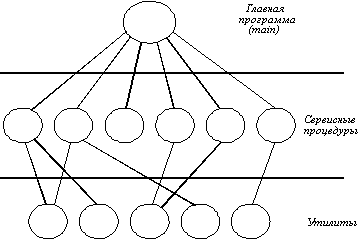


Наиболее простым и распространенным способом построения ОС является монолитная структура, когда ОС компонуется как одна программа Для построения монолитной системы необходимо скомпилировать все отдельные процедуры, а затем связать их вместе в единый объектный файл с помощью компоновщика (примерами могут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

Многоуровневая структураэ

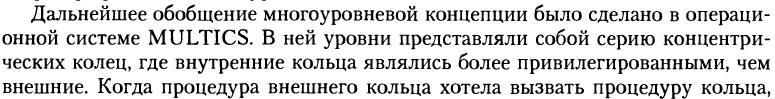


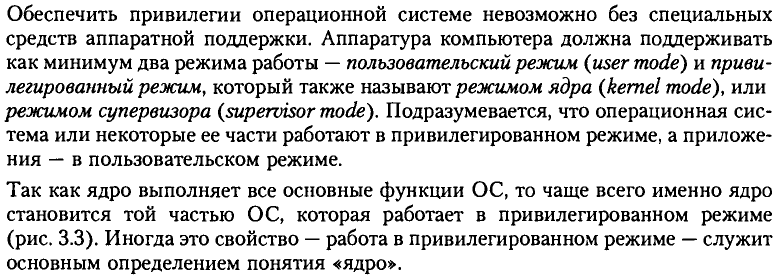


Развитием монолитного подхода является многоуровневый, когда ОС реализуется как иерархии уровней.

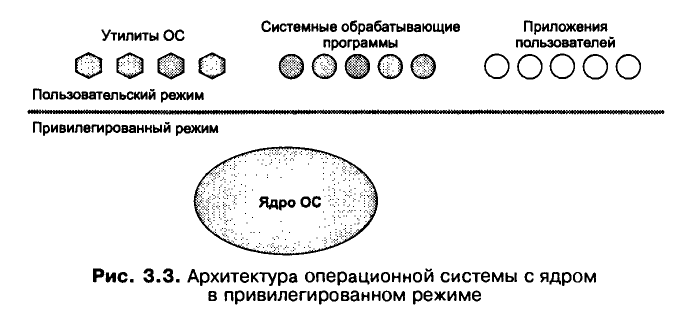
Уровни образуются группами функций ОС – файловая система, управление процессами и устройствами и т.п. Каждый уровень может взаимодействовать только со своим непосредственным соседом – выше- или нижележащим уровнем.

Понятие ядра

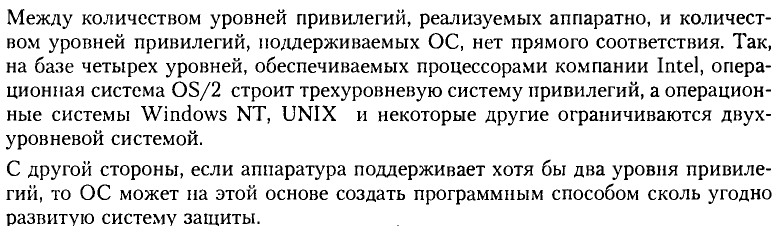


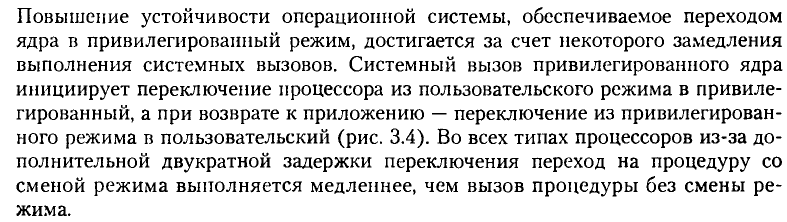


Ядро в привилегированном режиме

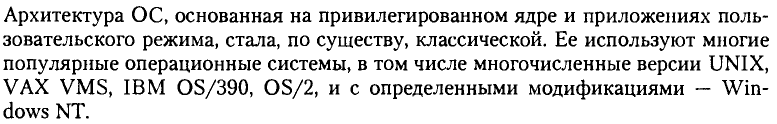


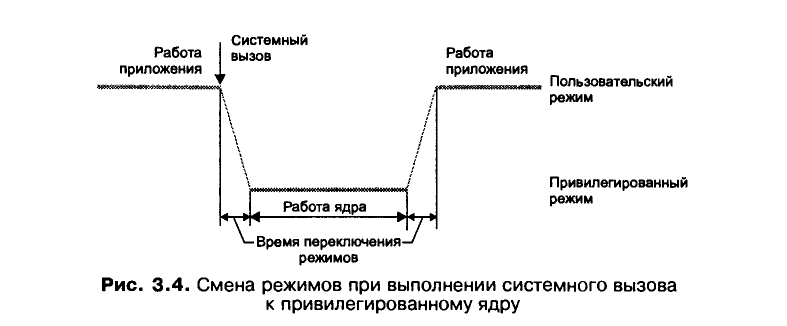
Уровни привилегий



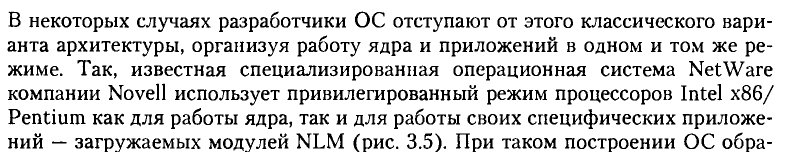


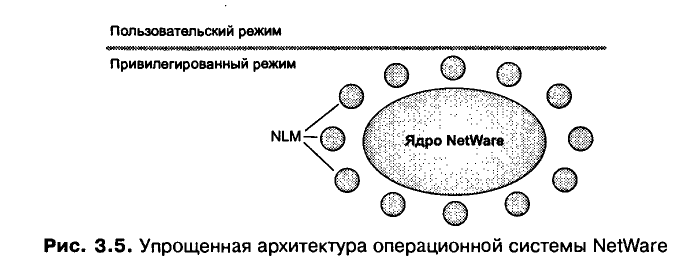
Смена режимов





Ядро в непривилегированном режиме



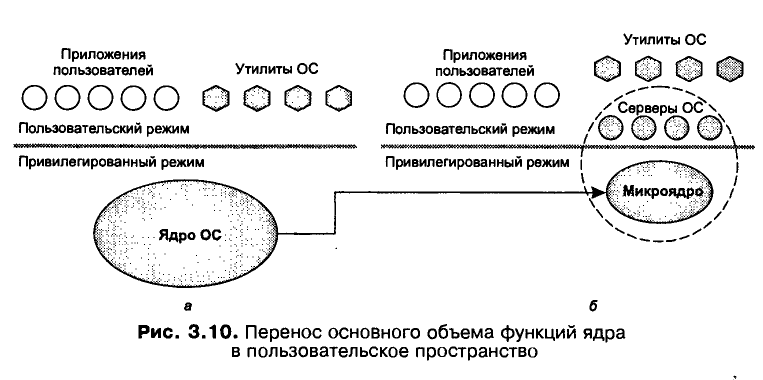


Клиент-серверная и микроядерная структура

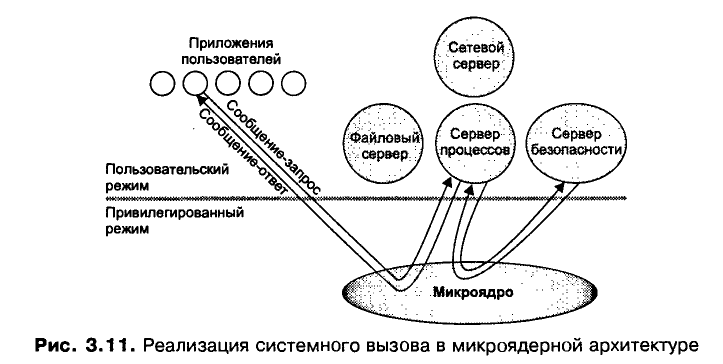
Альтернативой является построение ОС **на базе модели клиент-сервер и тесно связанной с ней концепции микроядра**.

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.

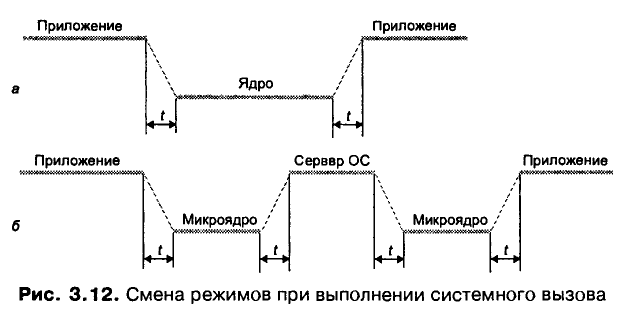
Формирование микроядра



Реализация системного вызова



Смена режимов



Достоинства и недостатки

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

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.

*Особенности алгоритмов управления ресурсами*

Поддержка многозадачности

По числу одновременно выполняемых задач ОС могут быть разделены на два класса:

* однозадачные (например, MS-DOS, MSX);
* многозадачные (OC EC, UNIX, Windows 9х, NT).

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.).

Поддержка многопользовательского режима

По числу одновременно работающих пользователей ОС делятся на:

* однопользовательские (MS-DOS, Windows 3.x);
* многопользовательские (UNIX, Windows NT).

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.

Вытесняющая и не вытесняющая 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 (Windows NT, OS/2, UNIX).
* 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 ДО* – 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 статический.
* 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 многопоточности

Важным свойством операционных систем является возможность распараллеливания вычислений в рамках одного процесса.

Многопоточная ОС разделяет процессорное время не между процессами, а между их отдельными нитями (потоками).

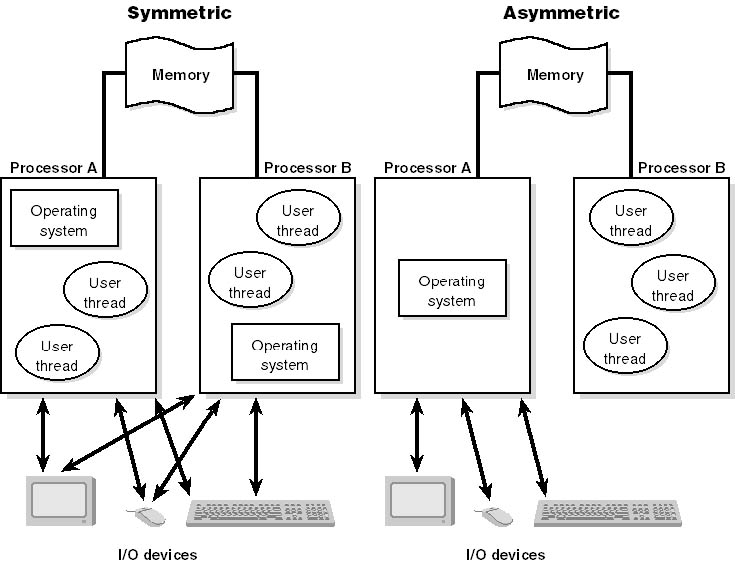
Многопроцессорная обработка

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-2000 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.

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.

*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.

Специализированные системы:

* 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

SMP-системы (оперативная память физически представляет последовательное адресное пространство, доступ к которому имеют одновременно все процессоры системы по единой шине).

Логические процессоры (hyperthreading) –

многопоточность в рамках одного ядра

Многоядерные процессоры

NUMA (Non-Uniform Memory Architecture)

Планы Intel

* В начале 21-века компания Intel прогнозировала появление к 2010 году процессоров с частотой 20 ГГц.
* Теперь на 2009-2010 год запланирован проект Keifer. Пока в качестве ориентира звучит количество ядер равное 32, каждое из которых способно обрабатывать одновременно до четырех потоков.

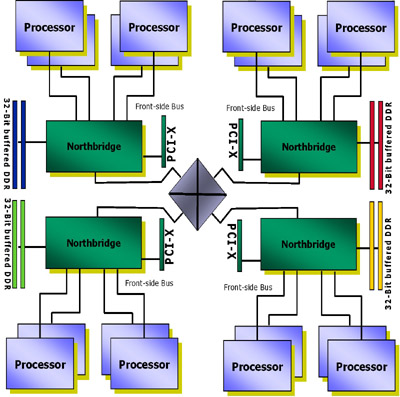
Многоядерные процессоры

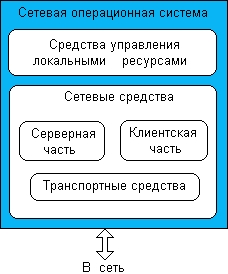
* SUN (Niagara – 8 ядер с 4-мя потоками каждое, FPU общий)
* Intel (Quad-Core Intel Xeon Processor 7300 Series – 4 ядра, реально 2 кристалла по 2 ядра)
* AMD (AMD Quad FX)

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 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 системы, и реализует эффективный способ разделения ресурсов между множеством выполняемых в сети процессов.

Сетевые средства ОС

* 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.

*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, OC EC);
* системы разделения времени (UNIX, MS Windows);
* системы реального времени (QNX, RT/11).

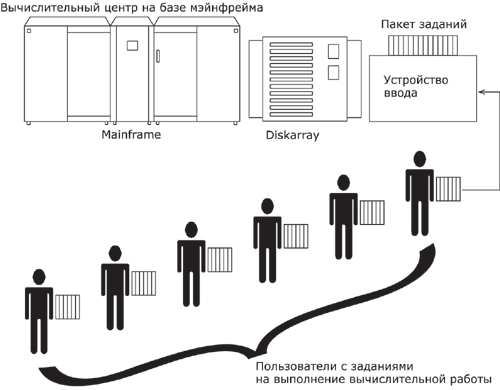
Системы пакетной обработки

**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

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 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.

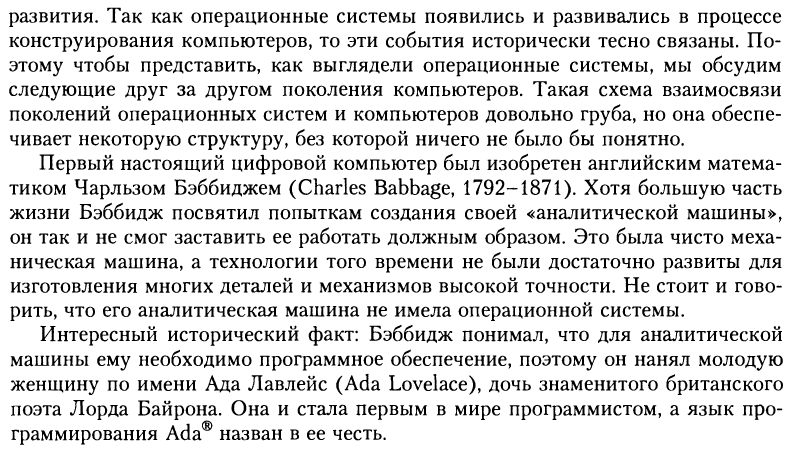
Таким образом, критерием эффективности для систем реального времени временя реакции системы на события от объекта управления, это свойство системы называется реактивностью.

Другие 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.

*Эволюция операционных систем*

Появление ОС



Этапы эволюции

* 1 этап (1940-60)

системный монитор, ранние пакетные системы

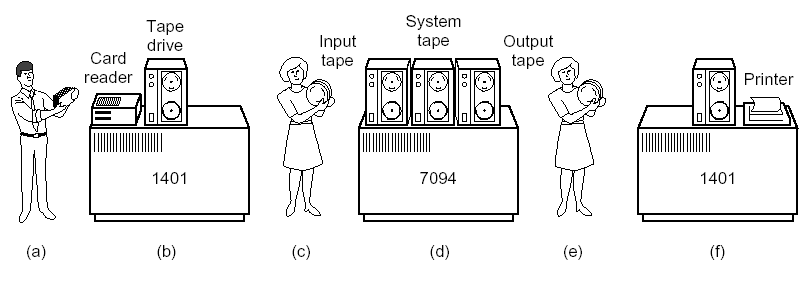
* 2 этап (1965-75)

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.
* 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 ЭВМ. 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.
* Ранние системы пакетной обработки **значительно сократили затраты времени** на вспомогательные действия по организации вычислительного процесса и способствовали **повышению эффективности** использования компьютеров.
* 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 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.

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.

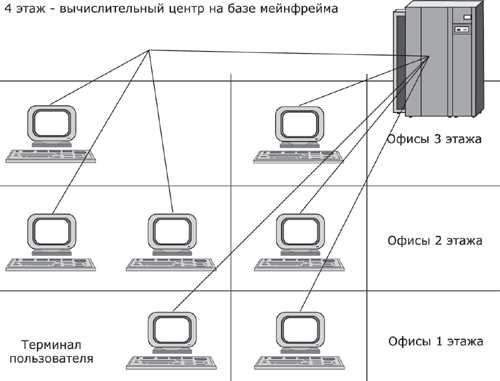
(e) Оператор переносит выходную ленту на устройство 1401.

(f) Устройство 1401 выполняет печать результатов.

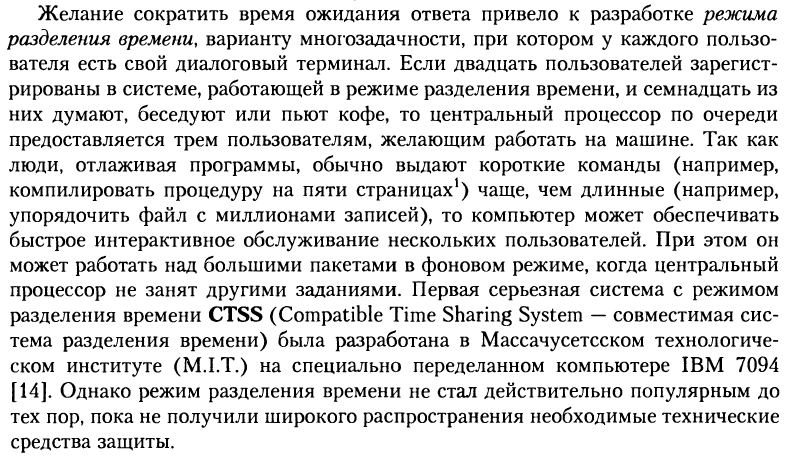
2 этап (1965-75)

* 1965-1975 годы переход к интегральным микросхемам, 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 наUNIX.

2 этап – многотерминальные системы

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

2 этап – разделение времени



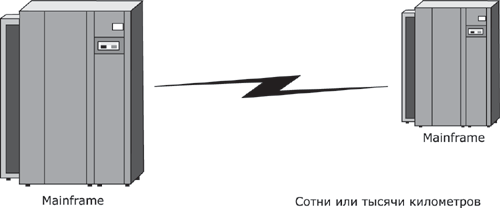
2 этап – многотерминальные системы

* 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.

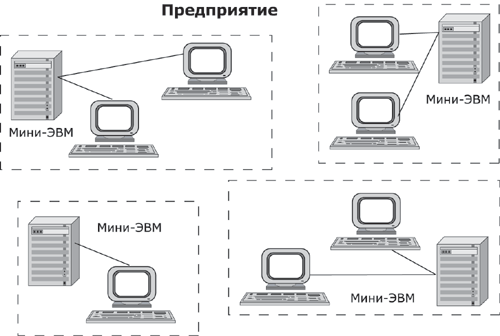
Эта многоуровневая архитектура (во многом прообраз модели OSI) обеспечивала взаимодействие типа «терминал-терминал», «терминал-компьютер» и «компьютер-компьютер» по глобальным связям.

**S**ystem **N**etwork **A**rchitecture

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**).

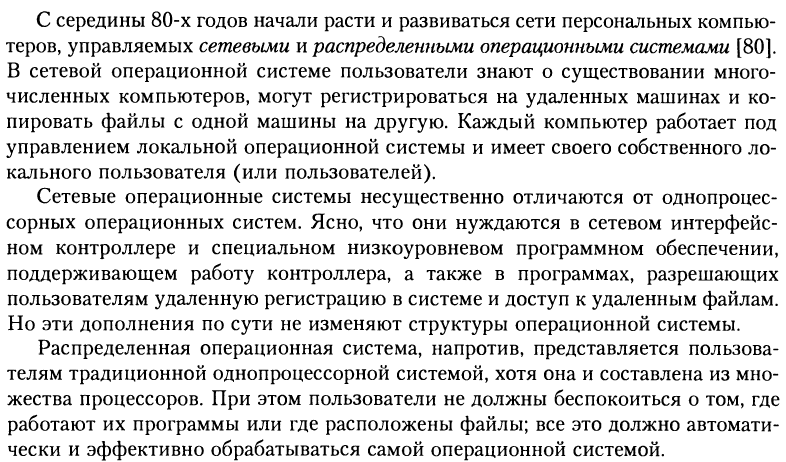
4 этап – мини-ЭВМ и ЛВС



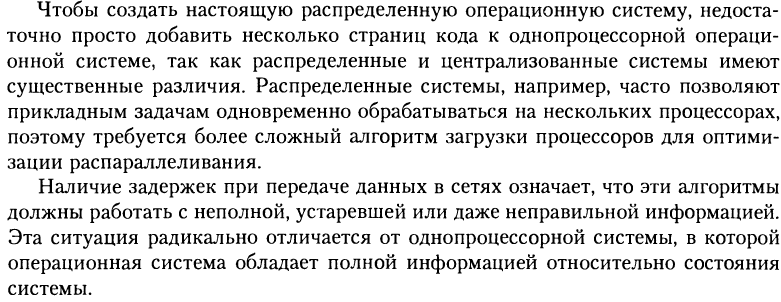
5 этап (1990 – …)

* 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.
* Понятие корпоративная сетевая ОС. Корпоративная ОС отличается способностью хорошо и устойчиво работать в крупных сетях, которые характерны для больших предприятий, имеющих отделения в десятках городов и, возможно, в разных странах. 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. К настоящему времени достаточно явно определилась тройка лидеров в классе корпоративных ОС – это Novell NetWare 4-6, Microsoft Windows NT-2000, а также UNIX-системы различных производителей аппаратных платформ.

5 этап – сетевые и распределенные ОС



5 этап – сетевые и распределенные ОС



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 –  2003 | Oberon, **Visual Basic, Java, C#, Perl, PHP**, Python | Windows, OS/2, Mac OS, Linux, Palm OS, Pocket PC |