

КУРС “Распределенные операционные системы”

Глава 2. Архитектуры многопроцессорных вычислительных систем.

Aurelia Prepelita,
conf. univ., dr., Catedra TP, USM

Chisinau, 2010

Оглавление:

Глава 2. Архитектуры многопроцессорных вычислительных систем.

- **Классификация Флинна**
 - **SISD**
 - **SIMD**
 - **MISD**
 - **MIMD**
- **Архитектуры параллельных систем**
 - **SMP**
 - **UMA** (*Uniform Memory Access*)
 - **COMA** (*Cache-Only Memory Access*)
 - **ccNUMA** (*Cache-Coherent Non-Uniform Memory Access*)
 - **NUMA** (*Non-Uniform Memory Access*);
 - **MPP**

Оглавление:

Глава 2. Архитектуры многопроцессорных вычислительных систем.

- **Определение распределенной системы**
 - **Примеры**
- **Важные характеристики распределенных систем**
 - **Достоинства многопроцессорных систем с общей памятью (мультипроцессоров)**
 - **Недостатки распределенных систем**
- **Виды операционных систем (сетевые ОС, распределенные ОС, ОС мультипроцессоров)**
- **Принципы построения распределенных ОС**

Классификация Флинна

По-видимому, самой ранней и наиболее известной является классификация архитектур вычислительных систем, предложенная в 1966 году Майклом Флинном. Классификация базируется на понятии **потока**, под которым понимается последовательность элементов, **команд или данных**, обрабатываемая процессором. На основе числа потоков команд и потоков данных Флинн выделяет четыре класса архитектур: SISD, MISD, SIMD, MIMD.

Классификация Флинна

	Один поток инст- рукций	Несколько пото- ков инструкций
Один поток дан- ных	SISD	MISD
Несколько пото- ков данных	SIMD	MIMD

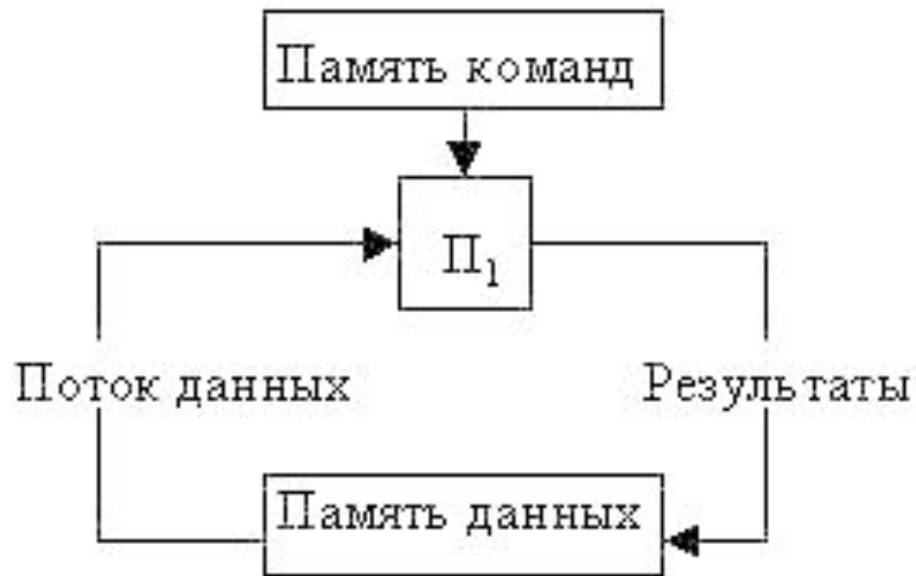
Таблица 1. Классификация Флинна

SISD

SISD (Single Instruction Single Data):

или **ОКОД** (*Одиночный поток Команд, Одиночный поток Данных*) - архитектура компьютера, в которой один процессор выполняет один поток команд, оперируя одним потоком данных. Относится к фон-Неймановской архитектуре. SISD компьютеры это обычные, «традиционные» последовательные компьютеры, в которых в каждый момент времени выполняется лишь одна операция над одним элементом данных (числовым или каким-либо другим значением). Большинство персональных ЭВМ до последнего времени, например, попадает именно в эту категорию.

SISD



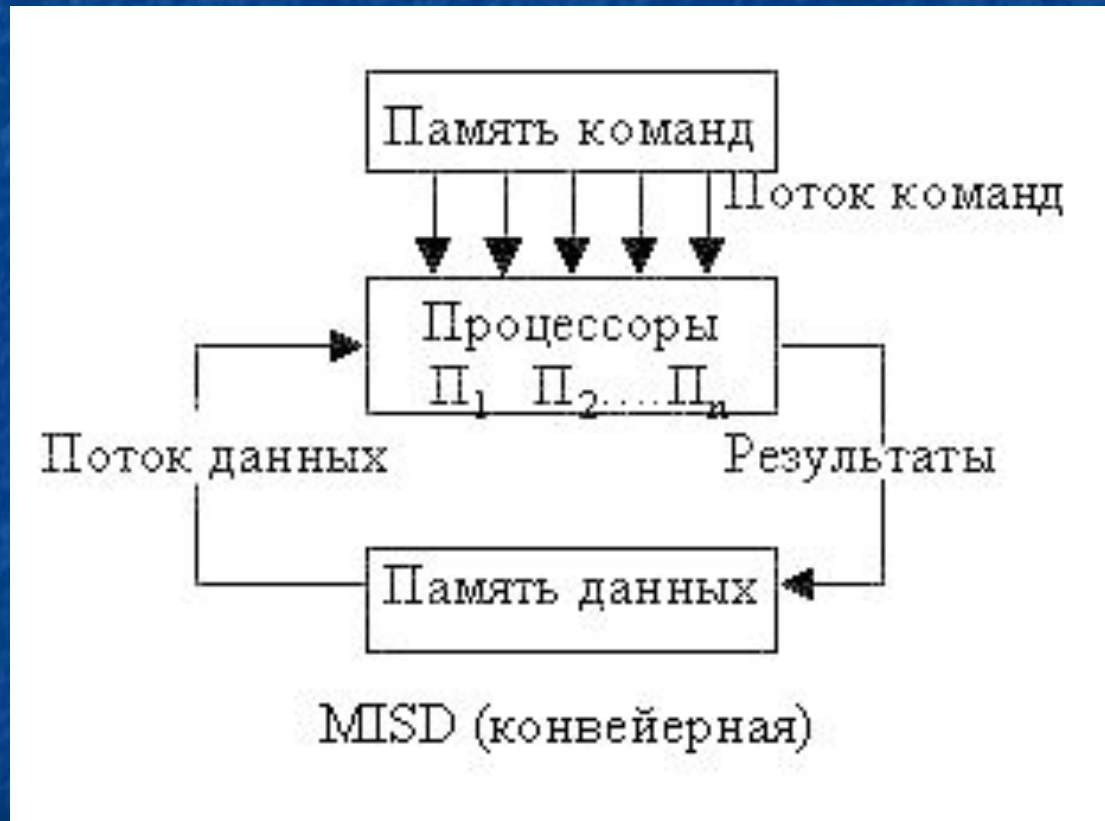
SISD (однопроцессорная)

MISD

MISD (Multiple Instruction Single Data):

Множественный поток Команд, одиночный поток Данных, МКОД) — тип архитектуры параллельных вычислений) — тип архитектуры параллельных вычислений, где несколько функциональных модулей выполняют различные операции над одними данными. К этому типу относят конвейерную архитектуру. Было создано немного ЭВМ с MISD-архитектурой, поскольку MIMD) — тип архитектуры параллельных вычислений, где несколько функциональных модулей выполняют различные операции над одними данными. К этому типу относят конвейерную

MISD



Считается, что впервые конвейерные вычисления были использованы в проекте ILLIAC II

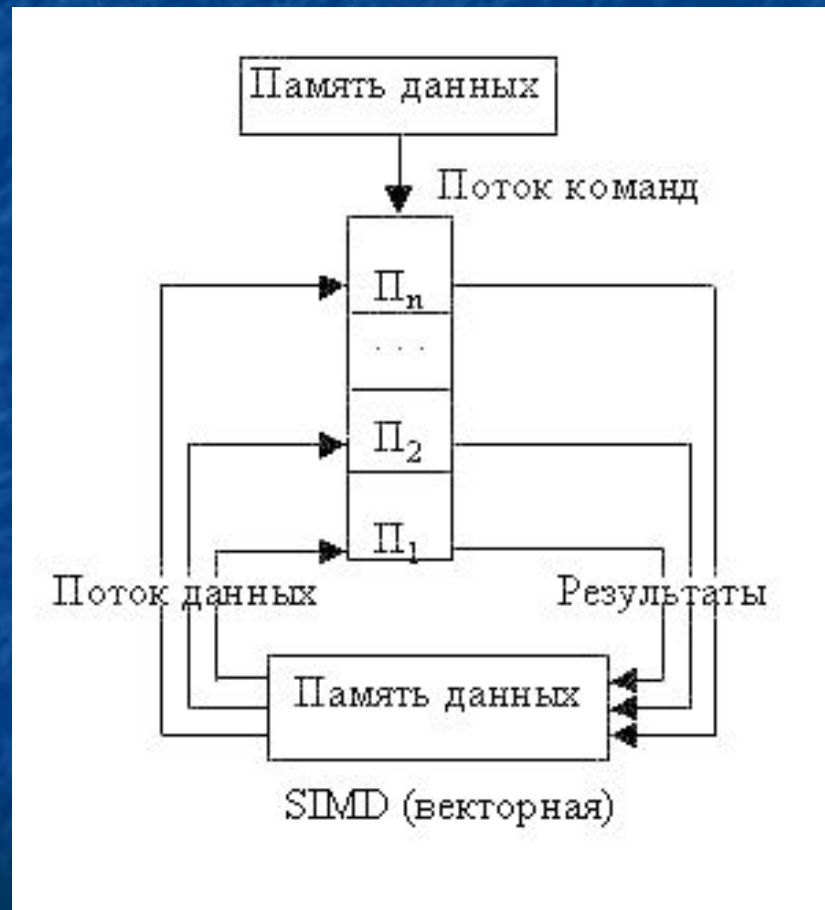
SIMD

SIMD (Single Instruction Multiple Data):

Одиночный поток Команд, Множественный поток Данных, ОКМД) — принцип компьютерных вычислений, позволяющий обеспечить параллелизм на уровне данных.

SIMD-компьютеры состоят из одного командного процессора (управляющего модуля), называемого **контроллером**, и нескольких модулей обработки данных, называемых **процессорными элементами**. Управляющий модуль принимает, анализирует и выполняет команды. Если в команде встречаются данные, контроллер рассылает на все процессорные элементы команду, и эта команда выполняется на нескольких или на всех процессорных элементах. В SIMD компьютере управление выполняется контроллером, а «арифметика» отдана процессорным элементам. SIMD-процессоры называются также векторными.

SIMD

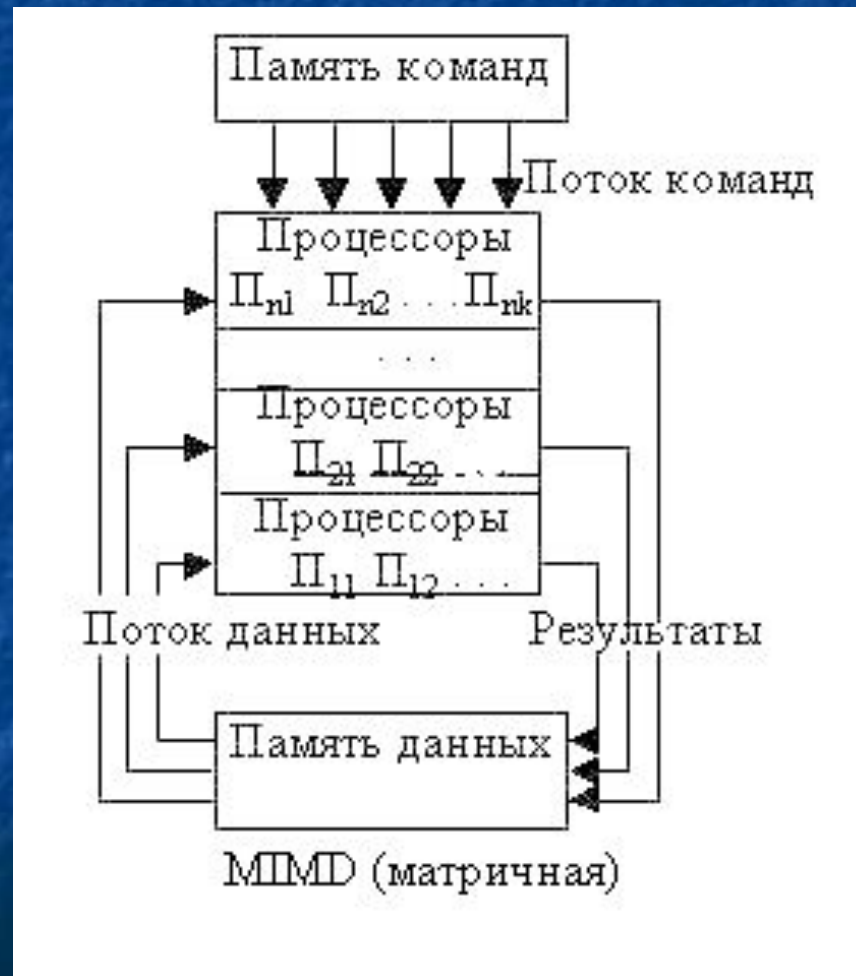


MIMD

MIMD (Multiple Instruction Multiple Data): разные потоки инструкций оперируют различными данными. Это системы наиболее общего вида, поэтому их проще всего использовать для решения различных параллельных задач.

MIMD-системы, в свою очередь, принято разделять (**классификация Джонсона**) на **системы с общей памятью** (несколько вычислителей имеют общую память) и **системы с распределенной памятью** (каждый вычислитель имеет свою память; вычислители могут обмениваться данными). Кроме того, существуют **системы с неоднородным доступом к памяти (NUMA)** — в которых доступ к памяти других вычислителей существует, но он значительно медленнее, чем доступ к «своей» памяти.

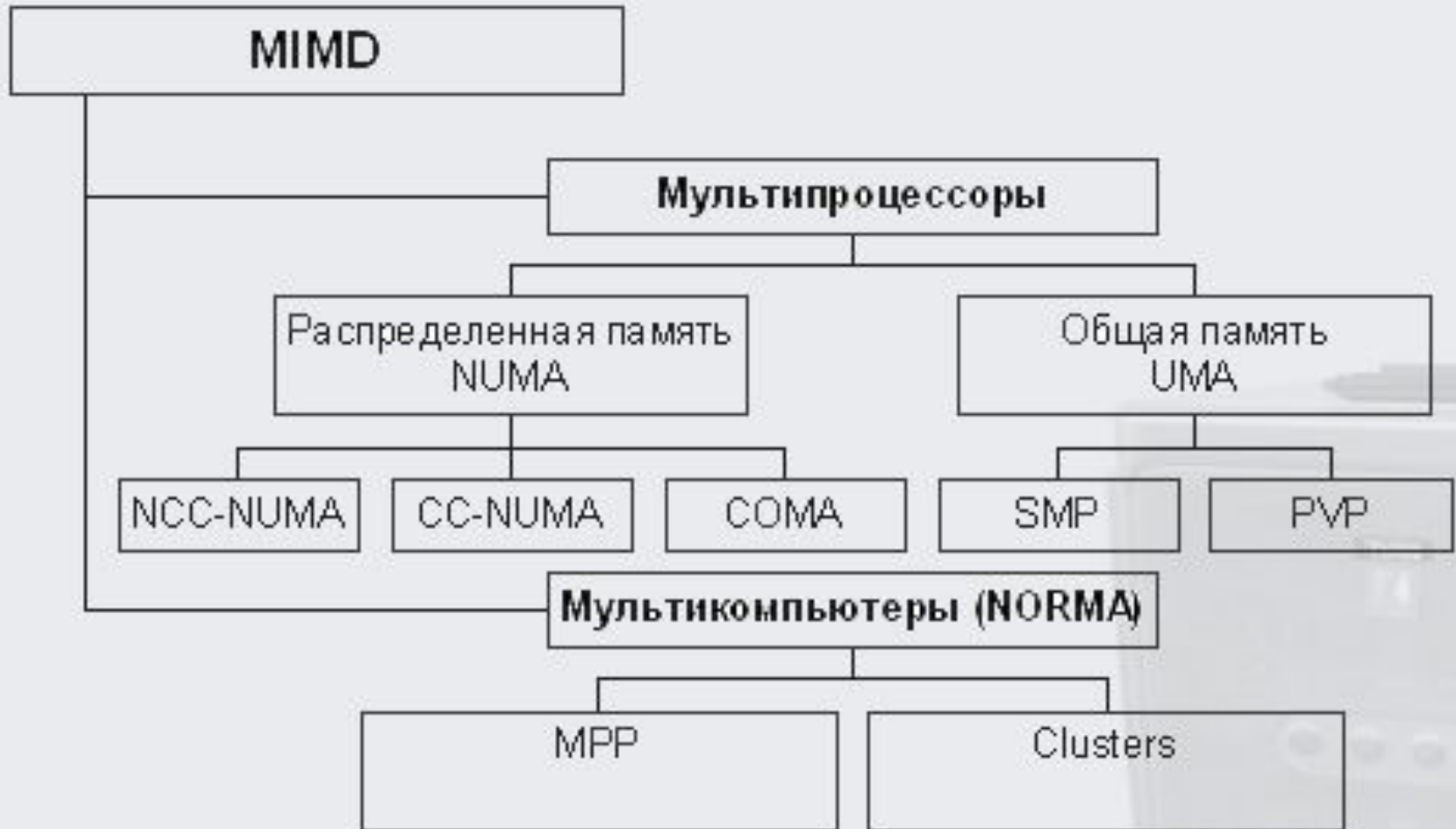
MIMD



MIMD

Следует отметить, что хотя систематика Флинна широко используется при конкретизации типов компьютерных систем, такая классификация приводит к тому, что практически все виды параллельных систем (несмотря на их существенную разнородность) относятся к одной группе MIMD. Как результат, многими исследователями предпринимались неоднократные попытки детализации систематики Флинна. Так, например, для класса MIMD предложена практически общепризнанная структурная схема, в которой дальнейшее разделение типов многопроцессорных систем основывается на используемых способах организации оперативной памяти в этих системах.

MIMD



Данный подход позволяет различать два важных типа многопроцессорных систем – multiprocessors (**мультипроцессоры** или системы с общей разделяемой памятью) и multicomputers (**мультикомпьютеры** или системы с распределенной памятью).

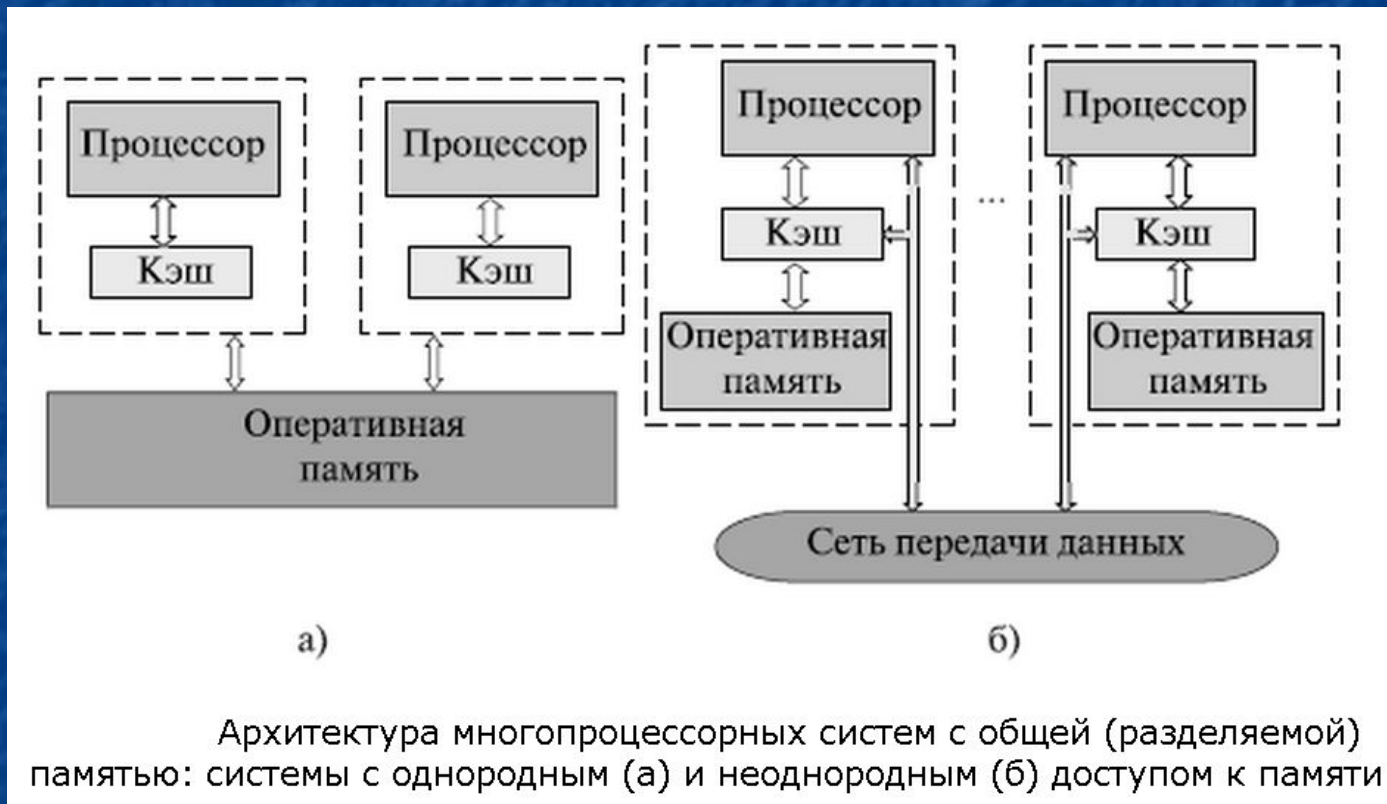
SMP-архитектура



SMP (symmetric multiprocessing) – **симметричная многопроцессорная архитектура**. Главной особенностью систем с архитектурой SMP является наличие **общей физической памяти**, разделяемой всеми процессорами.

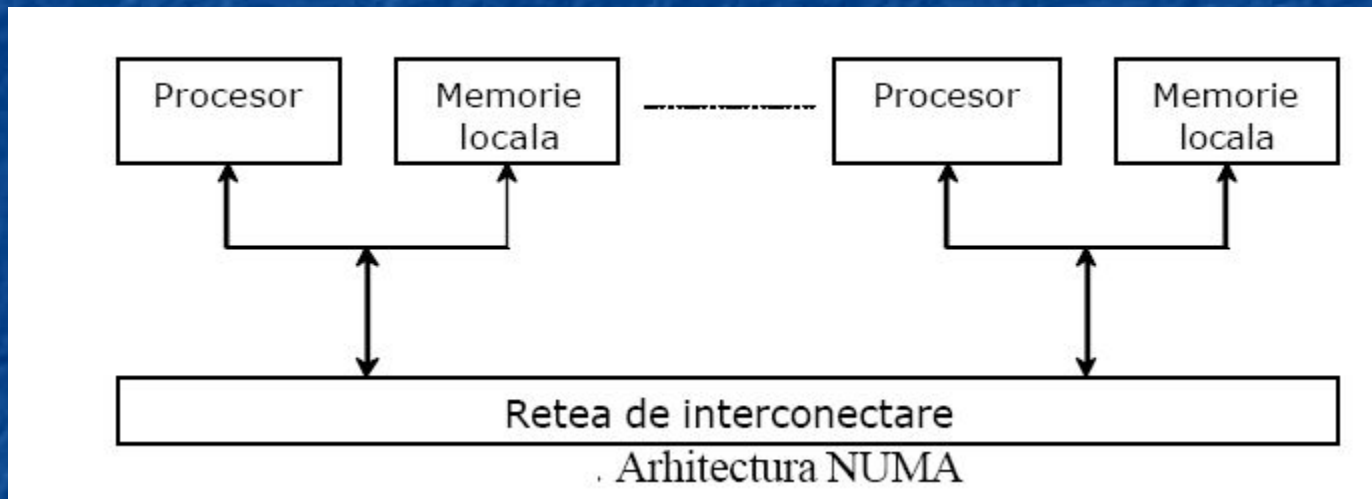
Наиболее известными SMP-системами являются SMP-серверы и рабочие станции на базе процессоров Intel (IBM, HP, Compaq, Dell, ALR, Unisys, DG, Fujitsu и др.) Вся система работает под управлением единой ОС (обычно UNIX-подобной, но для Intel-платформ поддерживается Windows NT). ОС автоматически (в процессе работы) распределяет процессы по процессорам, но иногда возможна и явная привязка.

SMP-архитектура UMA



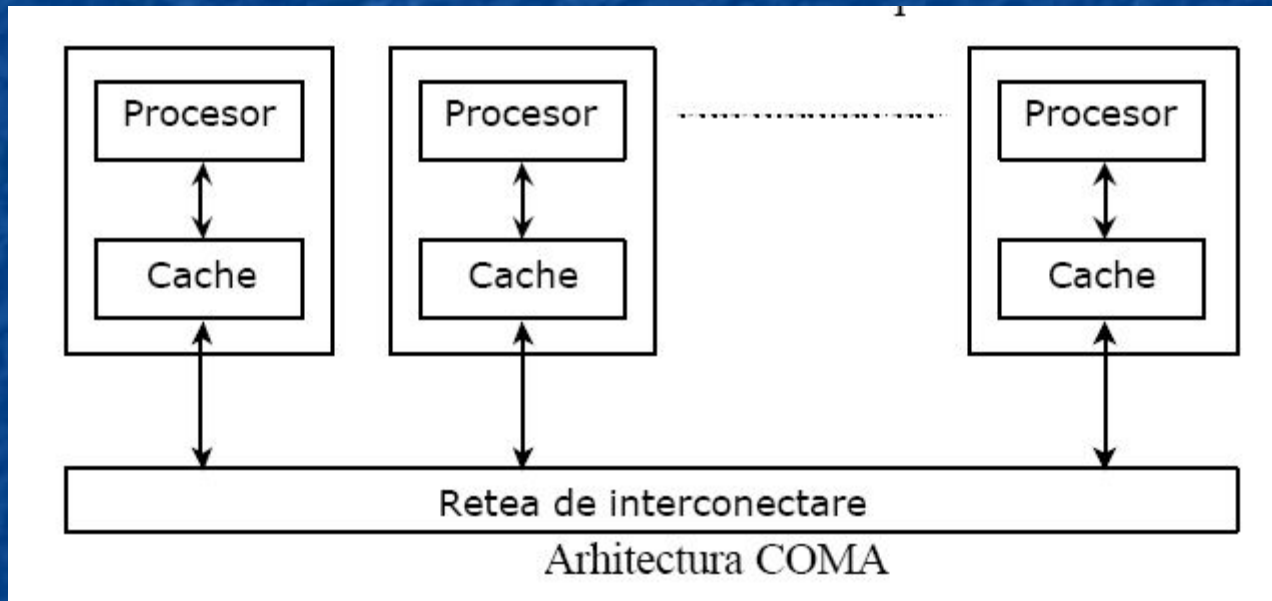
а) UMA с локальной cache-памятью

NUMA



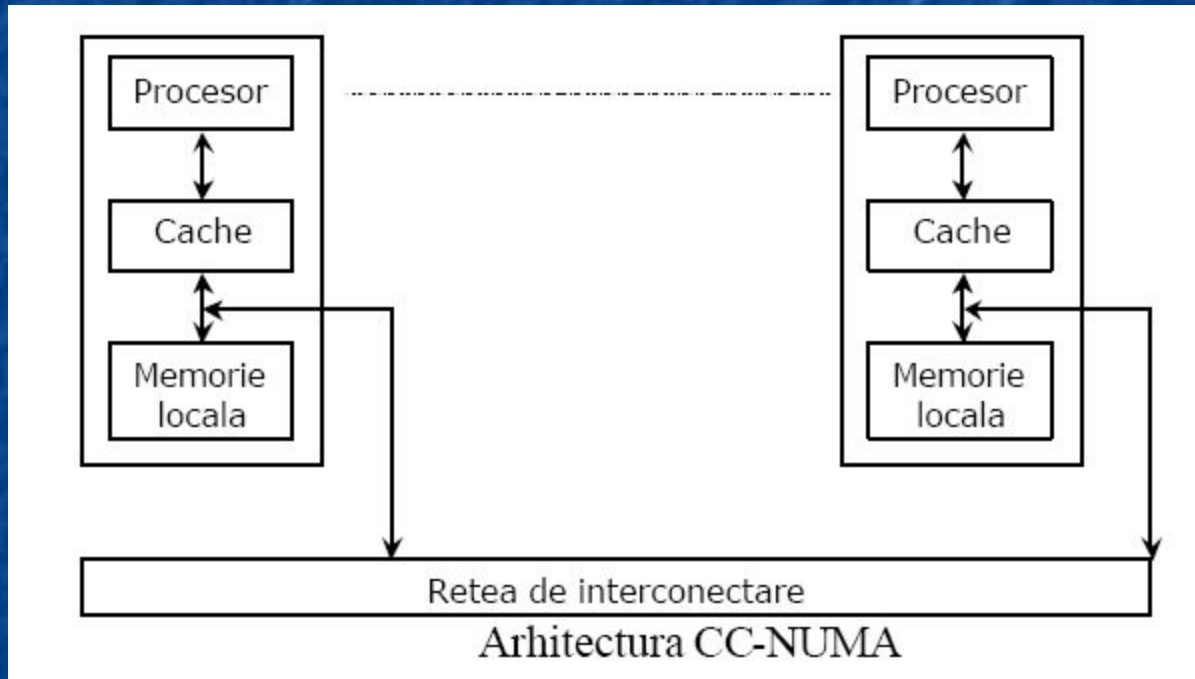
Общий доступ к данным обеспечен при физически распределенной памяти (при этом, длительность доступа уже не будет одинаковой для всех элементов памяти)

COMA



системы, в которых для представления данных используется только локальная кэш-память имеющихся процессоров (*cache-only memory architecture* или *COMA*)

CC-NUMA



системы, в которых обеспечивается *когерентность* локальных кэшей разных процессоров (**cache-coherent NUMA** или *CC-NUMA*)

Определение распределенной системы

Распределенная система - совокупность независимых компьютеров, которая представляется пользователю единым компьютером (metacomputer), использование которого не намного сложнее, чем использование персональной ЭВМ.

Определение распределенной системы

В этом определении оговариваются два момента. Первый относится к аппаратуре: **все машины автономны**. Второй касается программного обеспечения: **пользователи думают, что имеют дело с единой системой**. Важны оба момента.

Определение распределенной системы

Распределённая ОС, динамически и автоматически распределяя задания по различным машинам системы для обработки, заставляет набор сетевых машин работать как виртуальный унипроцессор. Пользователь распределённой ОС, вообще говоря, не имеет сведений о том, на какой машине выполняется его программа (задание).

Распределённая ОС существует как единая операционная система в масштабах вычислительной системы. Каждый компьютер сети, работающей под управлением распределённой ОС, выполняет часть функций этой глобальной ОС. Распределённая ОС объединяет все компьютеры сети в том смысле, что они работают в тесной кооперации друг с другом для эффективного использования всех ресурсов компьютерной сети.

Примеры

Пример 1. Рассмотрим **сеть рабочих станций в университете или отделе компании**. Вдобавок к персональной рабочей станции каждого из пользователей имеется **пул процессоров машинного зала, не назначенных заранее ни одному из пользователей**, но динамически выделяемых им при необходимости. Эта распределенная система может обладать: **единой файловой системой**, в которой все файлы одинаково доступны со всех машин с использованием постоянного пути доступа

Примеры

когда пользователь набирает команду, система может найти наилучшее место для выполнения запрашиваемого действия, возможно, на собственной рабочей станции пользователя, возможно, на простаивающей рабочей станции, принадлежащей кому-то другому, а может быть, и на одном из свободных процессоров машинного зала.

Если система в целом **выглядит и ведет себя как классическая однопроцессорная система с разделением времени** (т.е. многопользовательская), она считается распределенной системой.

Примеры

Пример 2. Рассмотрим работу информационной системы, которая поддерживает **автоматическую обработку заказов**. Обычно подобные системы используются сотрудниками нескольких отделов, возможно в разных местах. Так, сотрудники отдела продаж могут быть разбросаны по обширному региону или даже по всей стране. Заказы передаются с переносных компьютеров, соединяемых с системой при помощи телефонной сети, а возможно, и при помощи сотовых телефонов. Приходящие заказы автоматически передаются в отдел планирования, превращаясь там во внутренние заказы на поставку, которые поступают в отдел доставки, и в заявки на оплату, поступающие в бухгалтерию.

Примеры

Система автоматически пересылает эти документы имеющимся на месте сотрудникам, отвечающим за их обработку. Пользователи остаются в полном неведении о том, как заказы на самом деле курсируют внутри системы, для них все это представляется так, будто вся работа происходит в централизованной базе данных.

Примеры

Другие примеры:

сеть рабочих станций (выбор процессора для выполнения программы, единая файловая система)

роботизированный завод (роботы связаны с разными компьютерами, но действуют как внешние устройства единого компьютера)

банк с множеством филиалов

система резервирования авиабилетов

важные характеристики распределенных систем

- от пользователей скрыты различия между компьютерами и способы связи между ними
- то же самое относится и к внешней организации распределенных систем
- пользователи и приложения единообразно работают в распределенных системах, независимо от того, где и когда происходит их взаимодействие.

важные характеристики распределенных систем

Распределенные системы обычно существуют постоянно, однако некоторые их части могут временно выходить из строя. Пользователи и приложения не должны уведомляться о том, что эти части заменены или починены или что добавлены новые части для поддержки дополнительных пользователей или приложений.

Достоинства многопроцессорных систем с общей памятью (мультипроцессоров)

1. Производительность
2. Надежность

Недостатки (мультипроцессоров)

1. ПО (приложения, языки, ОС) сложнее, чем для однопроцессорных ЭВМ
2. Ограниченность при наращивании (физ. размеры - близость к памяти, когерентность КЭШей, 64 процессора - максимально достигнутое).

Необходимость построения РОС

Основная задача распределенных систем — облегчить пользователям доступ к удаленным ресурсам и обеспечить их совместное использование, регулируя этот процесс. Ресурсы могут быть виртуальными, однако традиционно они включают в себя **принтеры, компьютеры, устройства хранения данных, файлы и данные**. Web-страницы и сети также входят в этот список.

Необходимость построения РОС

Например, гораздо дешевле разрешить совместную работу с принтером нескольких пользователей, чем покупать и обслуживать отдельный принтер для каждого пользователя. Точно так же имеет смысл совместно использовать дорогие ресурсы, такие как суперкомпьютеры или высокопроизводительные хранилища данных.

Почему создаются распределенные системы? В чем их преимущества перед централизованными ЭВМ?

наращиваемость высокой производительности.

путем объединения микропроцессоров, которая недостижима в централизованном компьютере.

естественная распределенность (банк, поддержка совместной работы группы пользователей).

надежность (выход из строя нескольких узлов незначительно снизит производительность).

Почему создаются распределенные системы? В чем их преимущества перед централизованными ЭВМ?

В будущем главной причиной будет наличие огромного количества персональных компьютеров и необходимость совместной работы без ощущения неудобства от географического и физического распределения людей, данных и машин.

Почему нужно объединять РС в сети?

*Существует два основных подхода к организации операционных систем для вычислительных комплексов, связанных в сеть, – это **сетевые** и **распределенные операционные системы**.*

Недостатки распределенных систем

- 1. Проблемы ПО (приложения, языки , ОС).*
- 2. Проблемы коммуникационной сети (потери информации, перегрузка, развитие и замена).*
- 3. Секретность.*

Виды операционных систем (сетевые ОС, распределенные ОС, ОС мультипроцессоров)

Сетевые ОС - машины обладают высокой степенью автономности, общесистемных требований мало. Можно вести диалог с другой ЭВМ, вводить задания в ее очередь пакетных заданий, иметь доступ к удаленным файлам, хотя иерархия директорий может быть разной для разных клиентов. Пример - серверы файлов (многие WS могут не иметь дисков вообще).

Распределенные ОС - единый глобальный межпроцессный коммуникационный механизм, глобальная схема контроля доступа, одинаковое видение файловой системы. Вообще - иллюзия единой ЭВМ.

ОС мультипроцессоров - единая очередь процессов, ожидающих выполнения, одна файловая система.

Виды операционных систем

	Сетевая ОС	Распределенная ОС	ОС мультипроцессора
Компьютерная система выглядит как виртуальная однопроцессорная ЭВМ	НЕТ	ДА	ДА
Одна и та же ОС выполняется на всех процессорах	НЕТ	ДА	ДА
Сколько копий ОС имеется в памяти	N	N	1
Как осуществляются коммуникации	Разделяемые файлы	Сообщения	Разделяемая память
Требуется ли согласованный сетевой протокол	ДА	ДА	НЕТ
Имеется ли единая очередь выполняющихся процессов	НЕТ	НЕТ	ДА
Имеется хорошо определенная семантика разделения файлов	Обычно НЕТ	ДА	ДА

Принципы построения распределенных ОС

Прозрачность расположения	Пользователь не должен знать, где расположены ресурсы
Прозрачность миграции	Ресурсы могут перемещаться без изменения их имен
Прозрачность размножения	Пользователь не должен знать, сколько копий существует
Прозрачность конкуренции	Множество пользователей разделяет ресурсы автоматически
Прозрачность параллелизма	Работа может выполняться параллельно без участия пользователя

Принципы построения распределенных ОС

Централизованные (плохие) решения

Децентрализованные алгоритмы со следующими чертами (хорошие) решения :

- ни одна машина не имеет полной информации о состоянии системы;
- машины принимают решения на основе только локальной информации;
- выход из строя одной машины не должен приводить к отказу алгоритма;
- не должно быть неявного предположения о существовании глобальных часов.