OPERATING SYSTEMS

Computer System Architecture

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

Course Syllabus - Unit 1

UNIT 1: Introduction and Process Management

Operating-System Structure & Operations, Kernel Data Structures,Computing Environments, Operating-System Services, OperatingSystem Design and Implementation. Process concept: Process in memory, Process State, Process Control Block, Process Creation and Termination, CPU Scheduling and Scheduling Algorithms, IPC - Shared Memory & Message Passing, Pipes - Named and Ordinary. Case Study: Linux/Windows Scheduling Policies.

OPERATING SYSTEMS

Course Outline

Class No.	Chapter Title / Reference Literature	Topics to be covered	% of Portions covered	
			Reference chapter	Cumulative
1	1.1-1.2	What Operating Systems Do, Computer-System Organization?	1	21.4
2	1.3,1.4,1.5	Computer-System Architecture, Operating-System Structure & Operations	1	
3	1.10,1.11	Kernel Data Structures, Computing Environments	1	
4	2.1,2.6	Operating-System Services, Operating System Design and Implementation	2	
5	3.1-3.3	Process concept: Process in memory, Process State, Process Control Block, Process Creation and Termination	3	
6	5.1-5.2	CPU Scheduling: Basic Concepts, Scheduling Criteria	5	
7	5.3	Scheduling Algorithms: First-Come, First-Served Scheduling, Shortest-Job-First Scheduling	5	
8	5.3	Scheduling Algorithms: Shortest-Job-First Scheduling (Pre-emptive), Priority Scheduling	5	
9	5.3	Round-Robin Scheduling, Multi-level Queue, Multi-Level Feedback Queue Scheduling	5	
10	5.5,5.6	Multiple-Processor Scheduling, Real-Time CPU Scheduling	5	
11	5.7	Case Study: Linux/Windows Scheduling Policies	5	
12	3.4,3.6.3	IPC – Shared Memory & Message Passing, Pipes – Named and Ordinary	3,6	

Topics Outline

- Computer System Architecture
  - Single Processor System
  - Multiprocessor Systems
  - Clustered Systems
- Typical Q and As

Computer System Architecture

- A computer system can be organized in a number of different ways, which one can categorize roughly according to the **number** of **general-purpose processors** used.

Single Processor Systems

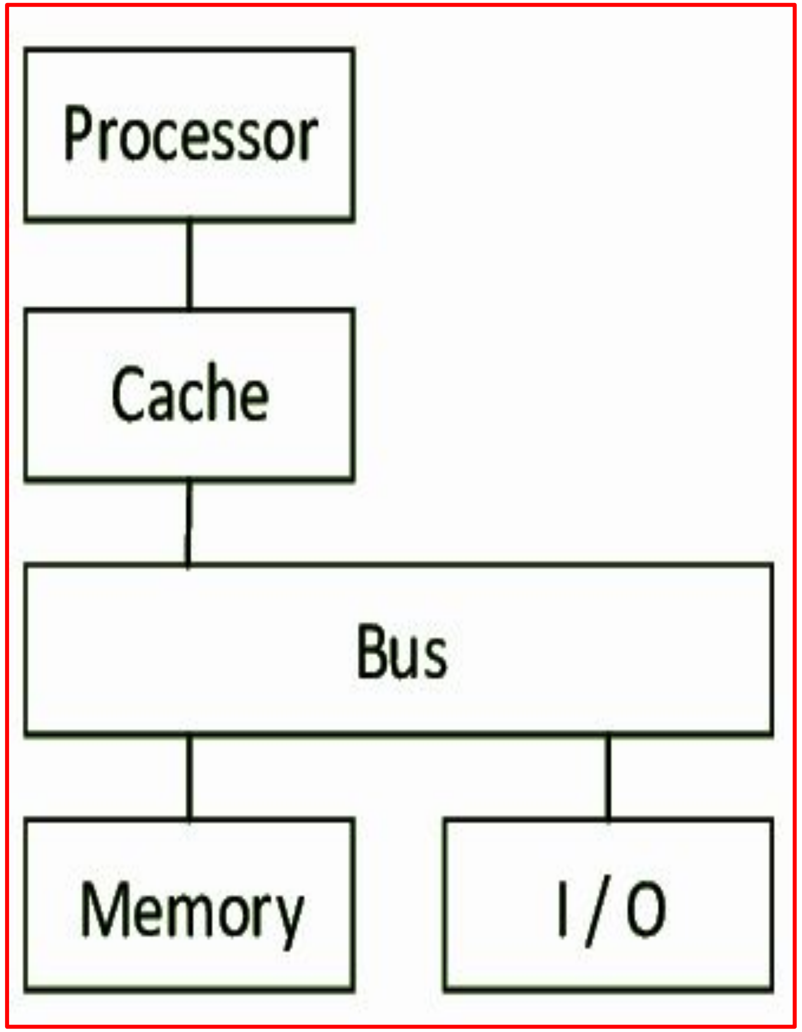
- On a single processor system, there is **One** main **CPU** capable of executing a general-purpose instruction set, including instructions from user processes.

- Almost all single processor systems have other special-purpose processors as well

- They may come in the form of **device-specific processors**, such as disk, keyboard, and graphics controllers;

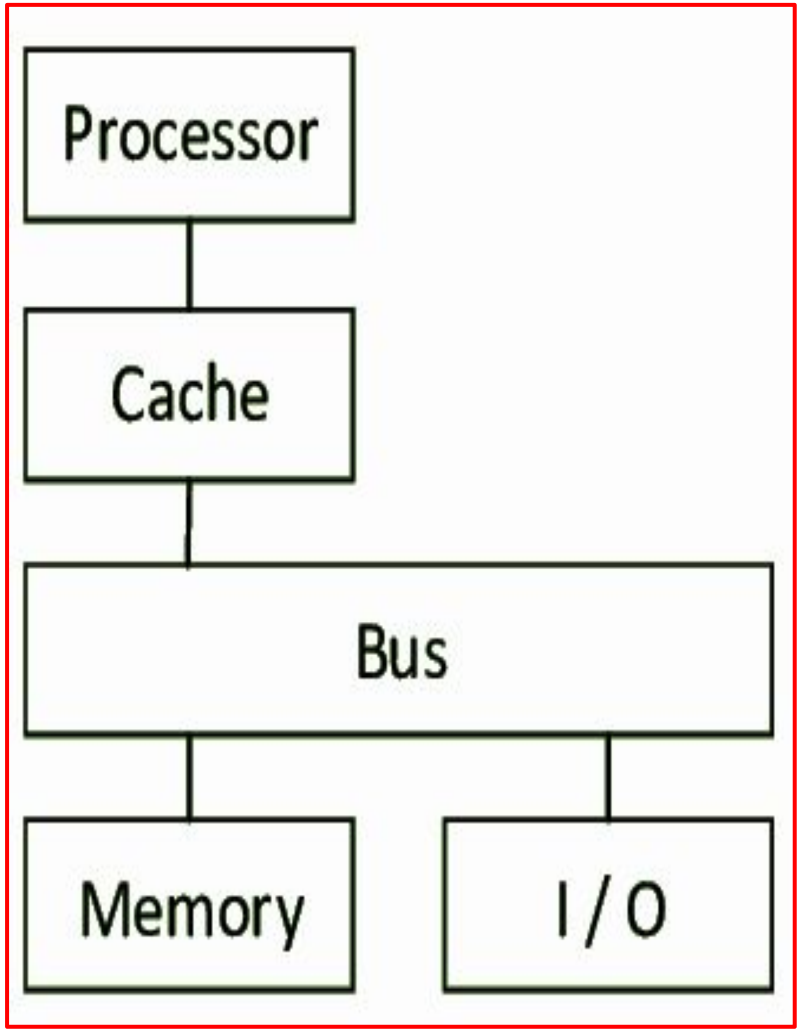
- On mainframes, they may come in the form of more general-purpose processors, such as I/O processors that move data rapidly among the components of the system

- All of these **special-purpose processors** run a limited instruction set and **do not run user processes.**



Single Processor Systems

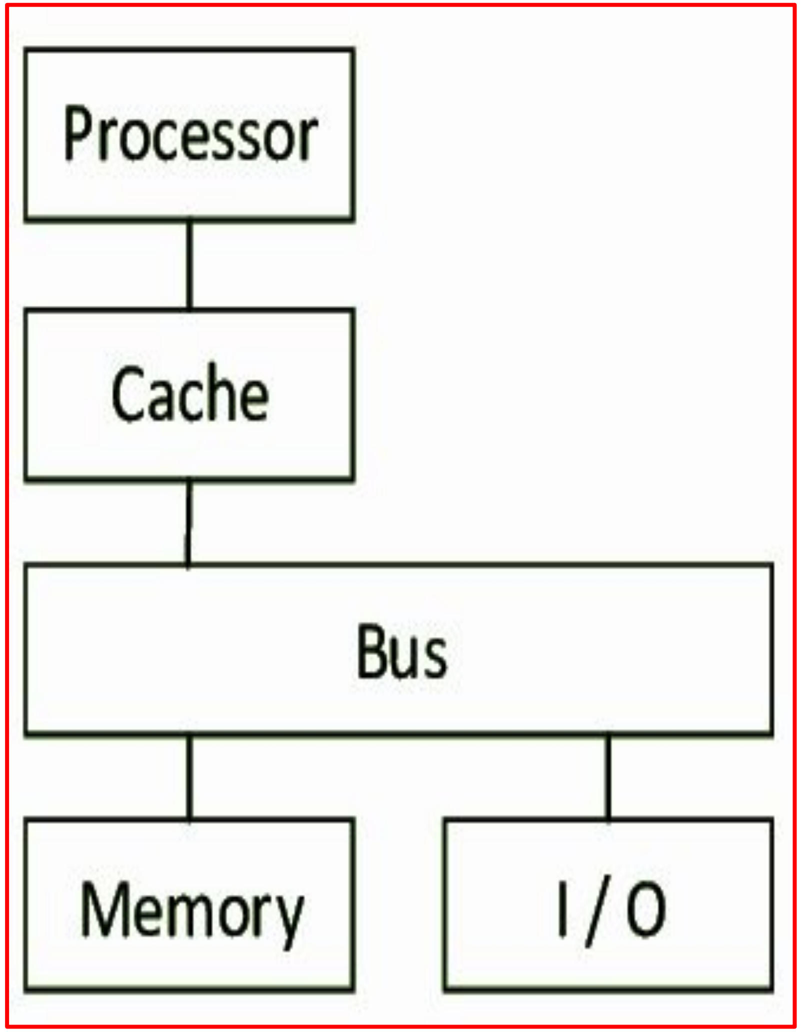
- A **disk-controller** microprocessor receives a sequence of requests from the main CPU and implements its own disk queue and scheduling algorithm.
- This arrangement **relieves** the **main CPU** of the overhead of disk scheduling.
- PCs contain a **microprocessor** in the **keyboard** to convert the keystrokes into codes to be sent to the CPU .
- In other systems or circumstances, special-purpose processors are low-level components built into the hardware.
- The operating system **cannot communicate** with these processors; they do their jobs autonomously.



Single Processor Systems

- The use of special-purpose microprocessors is common and **does not turn** a single-processor system into a multiprocessor.

- If there is only one **general-purpose CPU** , then the system is a **single-processor system**

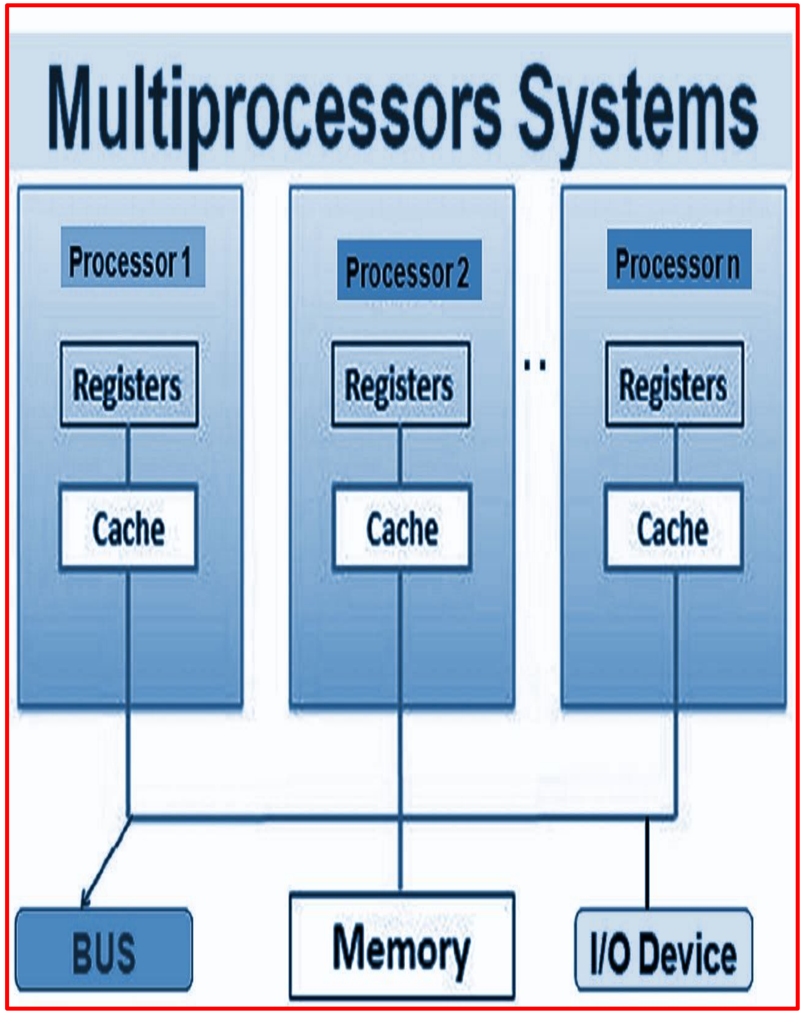


Multi Processor Systems

- Within the past several years, **multiprocessor systems** aka as **parallel systems** or **multicore systems** have begun to dominate the landscape of computing

- Multiprocessor systems first appeared prominently appeared in **servers** and have since **migrated** to **desktop** and **laptop** systems mostly as **multicore**.

- Multiple processors have appeared on mobile devices such as smartphones and tablet computers.



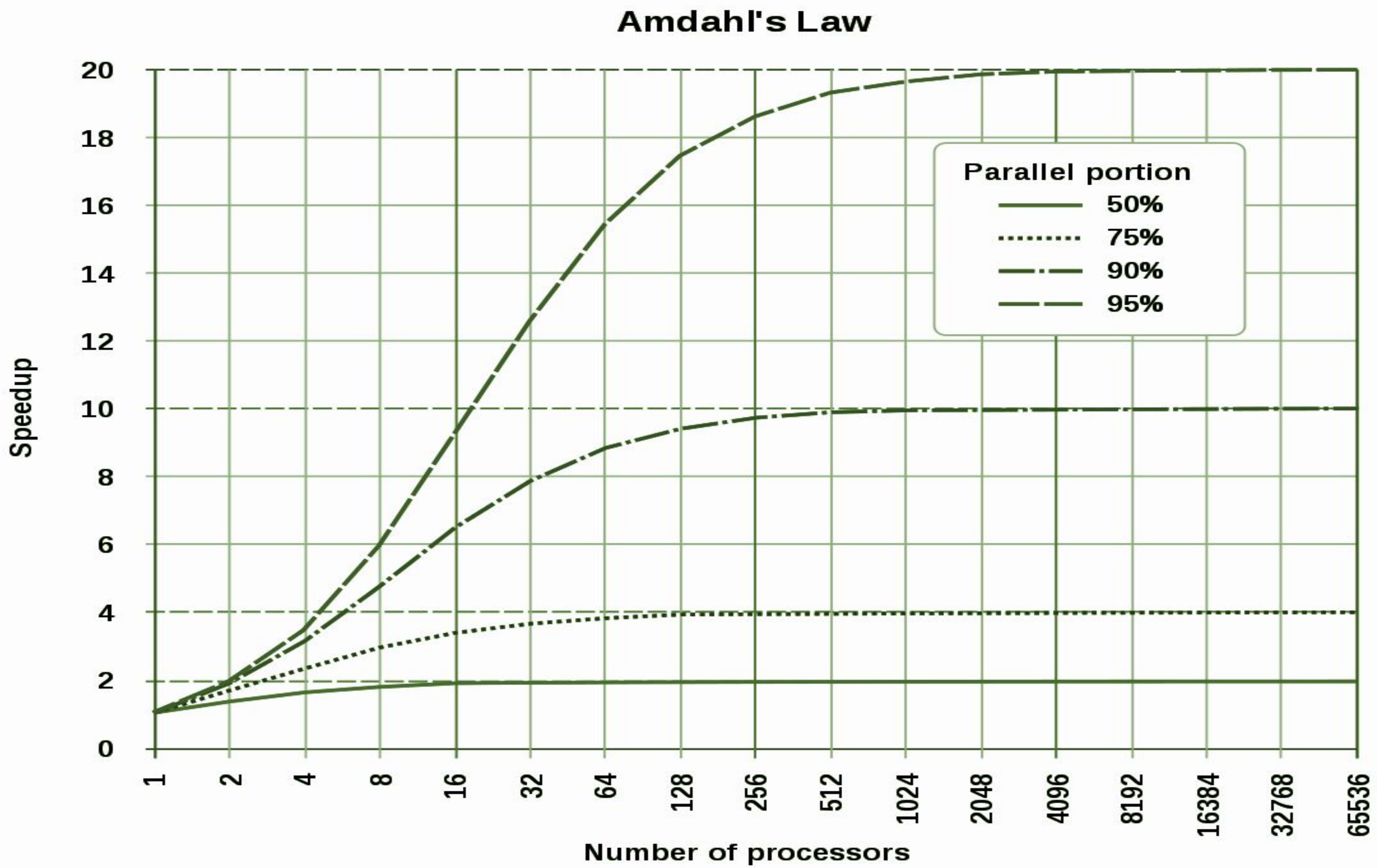
Multi Processor Systems

Multiprocessor systems have three main advantages

1. Increased throughput: The speed-up ratio with N processors is not N , however; rather, it is less than N .

- When multiple processors cooperate on a task, a certain amount of **overhead** is incurred in keeping all the parts working correctly.
- This **overhead**, plus **contention** for **shared resources**, lowers the expected gain from additional processors.

Multi Processor Systems



Multi Processor Systems

Multiprocessor systems have three main advantages

2. Economy of scale: Multiprocessor systems can cost less than equivalent multiple single-processor systems,

- Because they can **share peripherals, mass storage, and power supplies**.

- If several programs operate on the same set of data, it is **cheaper** to store those data on **one disk** and to have all the processors **share** them than to have many computers with local disks and many copies of the data.

Multi Processor Systems

Multiprocessor systems have three main advantages

3. Increased reliability. If functions can be distributed properly among several processors, then the failure of one processor will not halt the system, only slow it down.

- If we have ten processors and one fails, then each of the remaining nine processors can pick up a share of the work of the failed processor. Thus, the entire system runs only 10 percent slower, rather than failing altogether.

Multi Processor Systems

- The **ability** to continue providing service proportional to the level of **surviving hardware** is called **graceful degradation.**

- Some systems go beyond graceful degradation and are called **fault tolerant**, because they **can suffer a failure** of any single component and still **continue** operation.

- The **HP NonStop** (formerly Tandem) system uses both hardware and software duplication to ensure continued operation despite faults.

- The system consists of multiple pairs of CPUs, working in lockstep.

- As an example, if there are n CPUs in a multiprocessor system, each pair will execute each instruction and the results are compared

- If the results differ, then one CPU of the pair is at fault, and both are halted.

- The process that was being executed is then moved to another pair of CPUs, and the instruction that failed is restarted.

- This solution is expensive, since it involves special hardware and considerable hardware duplication

Multi Processor Systems

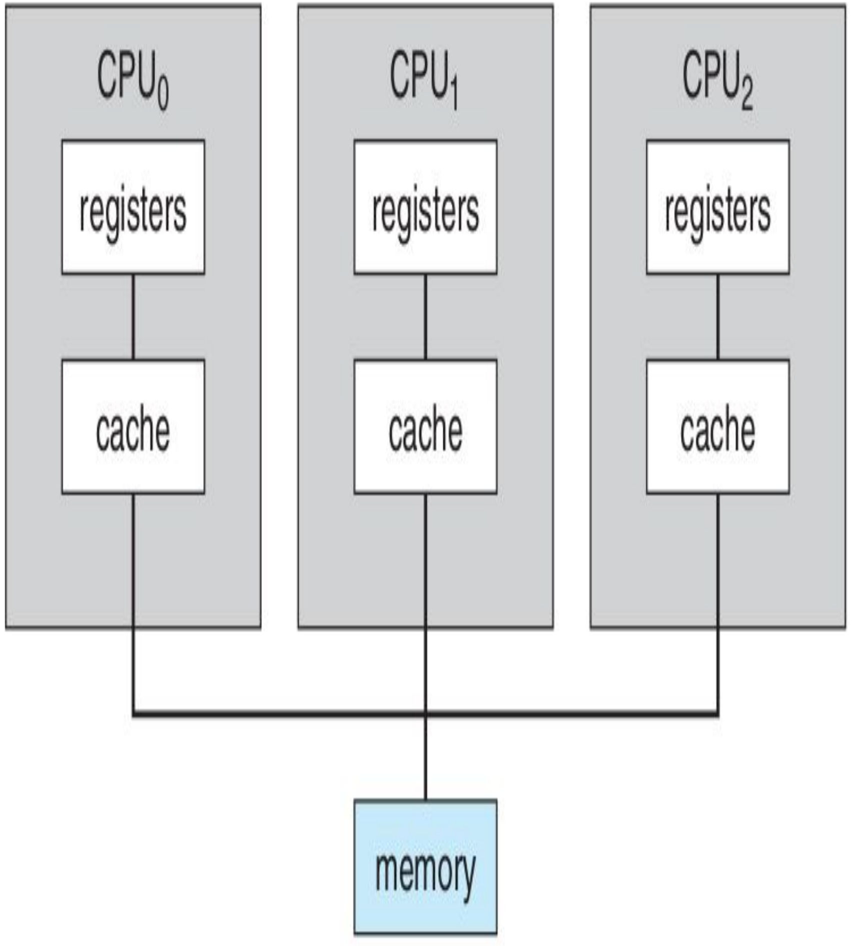
- The multiple-processor systems are of two types.
  - **Asymmetric Multiprocessing**
    - Each processor is assigned a specific task.
    - A boss processor controls the system.
    - The other processors either look to the boss for instruction or have predefined tasks.
    - This scheme defines a **boss–worker relationship**.
    - The boss processor schedules and allocates work to the worker processors.

Multi Processor Systems

- The multiple-processor systems are of two types.
  - **Symmetric Multiprocessing - SMP**
    - Each processor performs all tasks within the operating system.
    - SMP means that all processors are peers
    - **No boss–worker relationship** exists between processors.

Multi Processor Systems

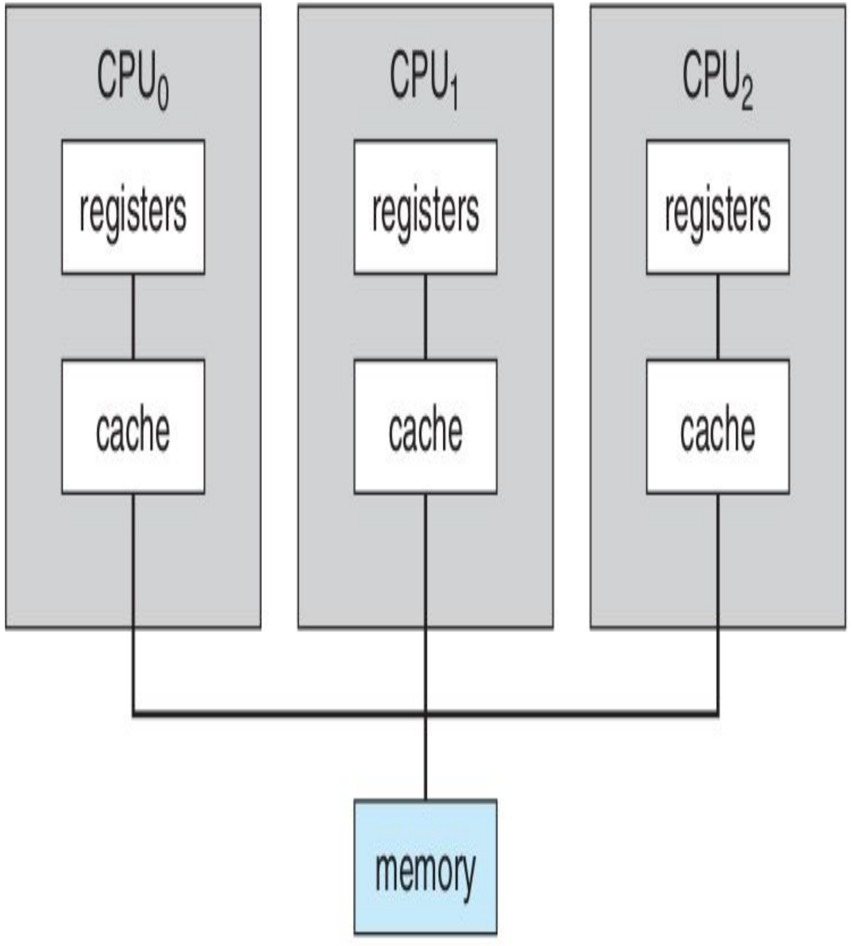
- **Symmetric Multiprocessing - SMP**
  - Each processor has its own set of registers, as well as a private or local cache.
  - **All** processors **share** physical memory.
  - N processes can run if there are N CPUs without causing performance to deteriorate significantly.
  - One must carefully control I/O to ensure that the data reach the appropriate processor.
  - Since the CPUs are separate, one may be sitting idle while another is overloaded, resulting in inefficiencies which can be mitigated by applying appropriate **load balancing algorithms**



Symmetric multiprocessing architecture.

Multi Processor Systems

- **Symmetric Multiprocessing - SMP**
    - Such a system must be written carefully
    - Virtually all modern operating systems including Windows, Mac OS X , and Linux, now provide support for SMP.



Symmetric multiprocessing architecture.

Multi Processor Systems

- **Symmetric versus Asymmetric Multiprocessing**

  - May result from either hardware or software.

  - Special hardware can differentiate the multiple processors, or the software can be written to allow only one boss and multiple workers.

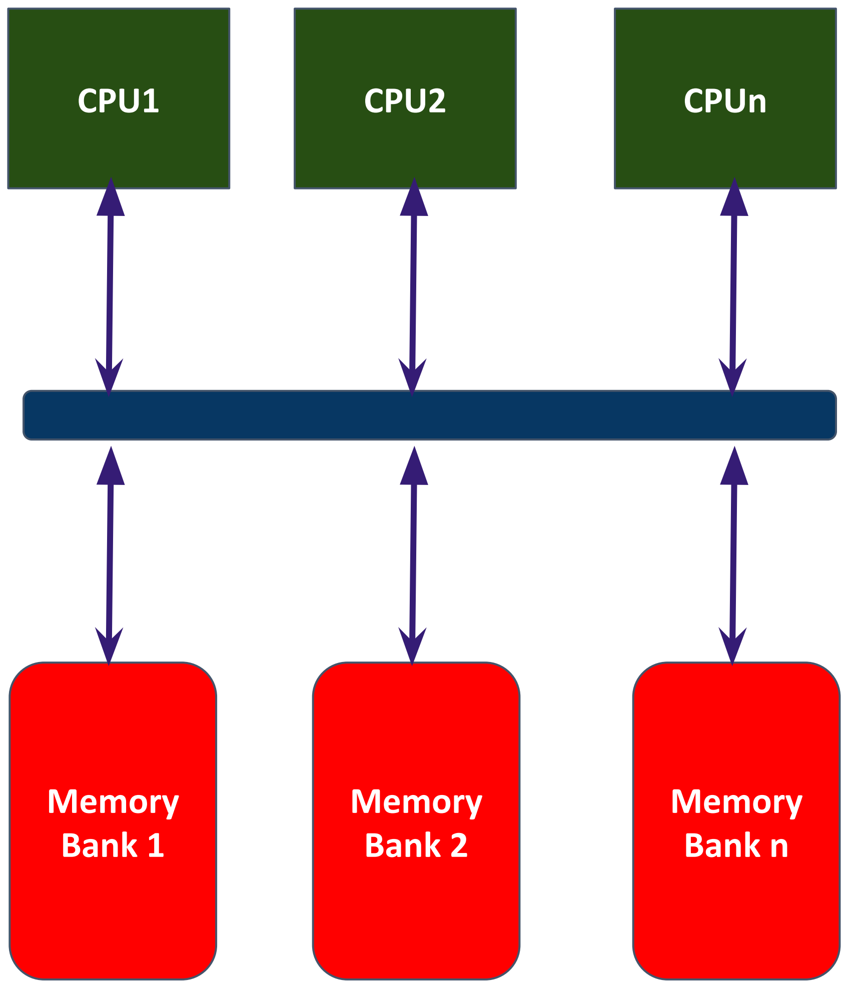
  - As an example for instance, Sun Microsystems' operating system **Sun OS Version 4** provided **asymmetric** multiprocessing, whereas **Solaris Version 5** is **symmetric** on the same hardware.

Multi Processor Systems

- **Multiprocessing**
  - Adds CPUs to increase computing power.
  - If the CPU has an integrated memory controller, then adding CPUs can also increase the amount of memory addressable in the system.
  - Can cause a system to change its **memory access model**

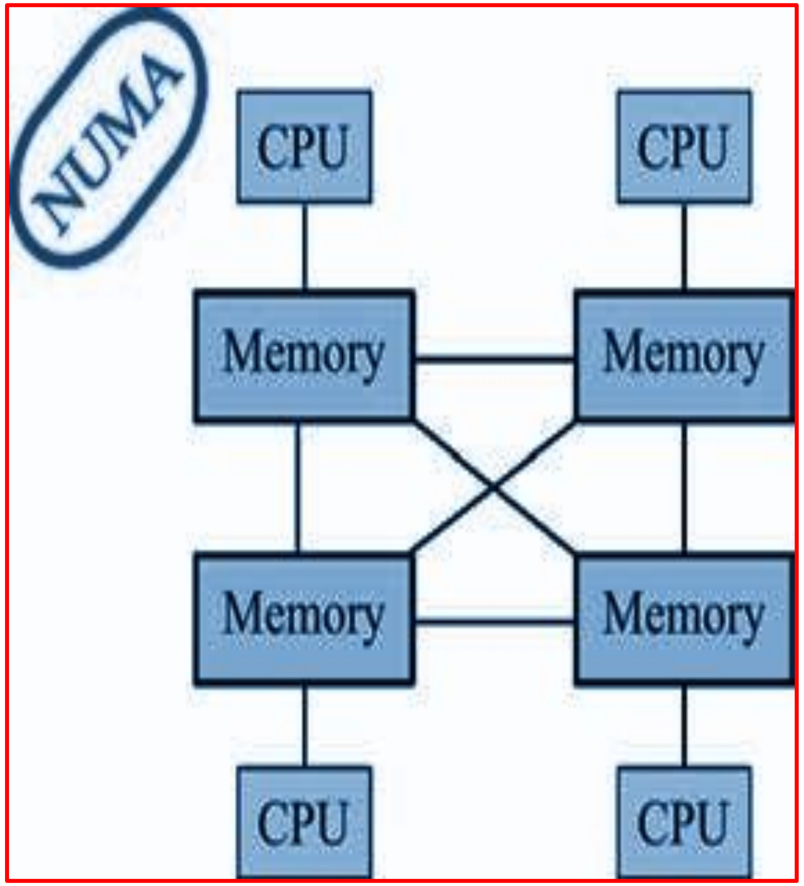
Multi Processor Systems

- **Multiprocessing**
  - **Memory access model**
    - **Uniform memory access - UMA**
      - Is defined as the situation in which access to any RAM from any CPU takes the same amount of time



Multi Processor Systems

- **Multiprocessing**
  - **Memory access model**
    - **Non Uniform memory access - NUMA**
      - Some parts of memory may take longer to access than other parts, creating a performance penalty.
      - Operating systems can minimize the **NUMA** penalty through efficient resource management



Multi Processor Systems

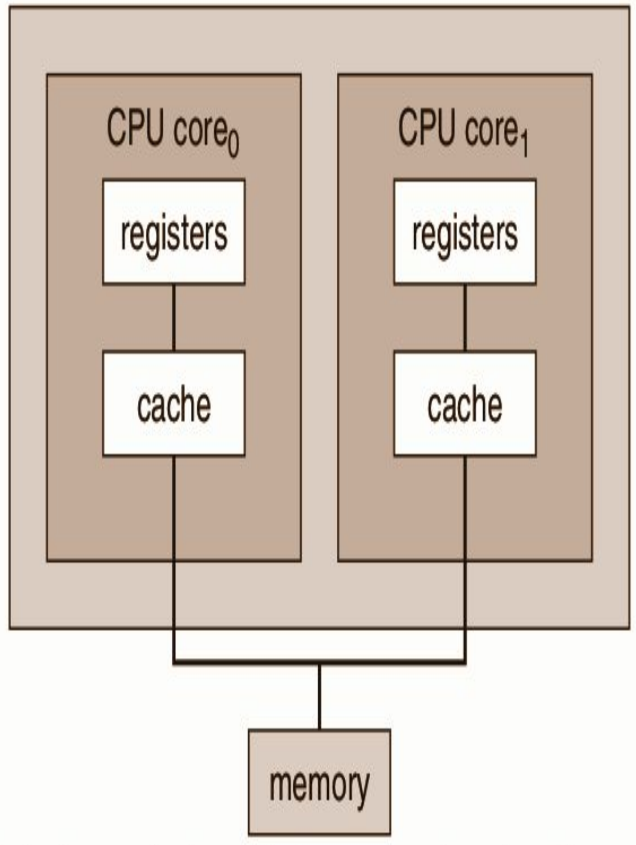
- **Multicore**

  - A recent trend in CPU design is to include multiple computing **cores** on a single chip.

  - Such multiprocessor systems are termed **multicore**.

  - They can be more efficient than multiple chips with single cores because on-chip communication is faster than between-chip communication.

  - In addition, one chip with multiple cores uses significantly less power than multiple single-core chips.

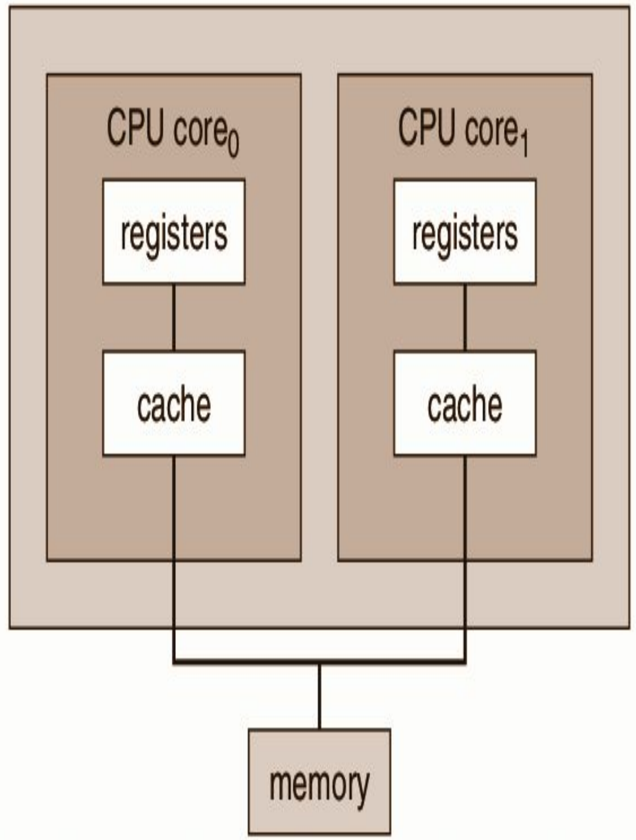


A dual-core design with two cores placed on the same chip

Multi Processor Systems

- **Multicore**

  - It is important to note that while **Multicore systems** are **multiprocessor systems**
  - **Not** all **multiprocessor** systems are **multicore**
  - In the figure on the side , each core has its own register set as well as its own local cache.
  - Other multicore designs might use a shared cache or a combination of local and shared caches.



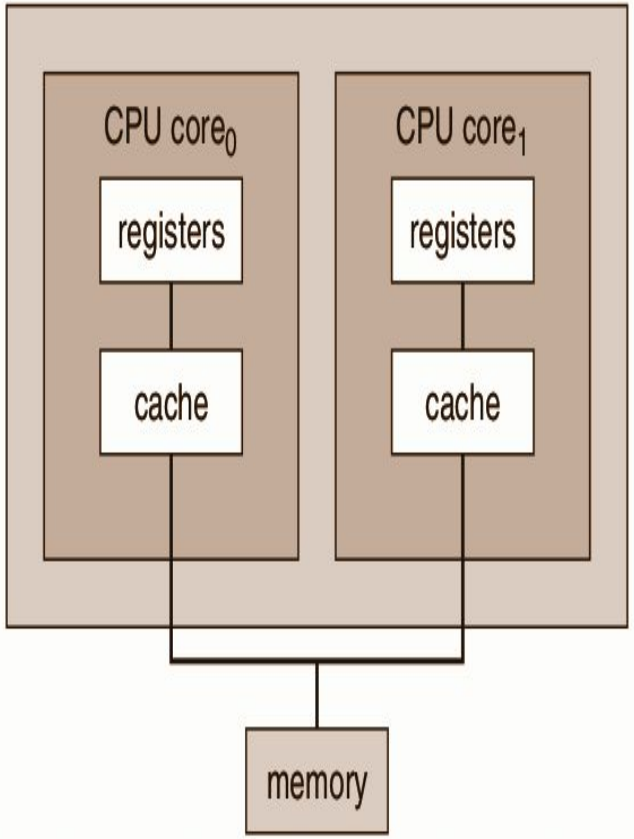
A dual-core design with two cores placed on the same chip

Multi Processor Systems

- **Multicore**

  - Besides architectural considerations, such as cache, memory, and bus contention, these multicore CPUs appear to the operating system as N standard processors

  - This characteristic puts pressure on operating system designers and application programmers to make use of those processing cores.



A dual-core design with two cores placed on the same chip

Multi Processor Systems

- **Blade Servers**

    - Are a relatively recent development in which multiple processor boards, I/O boards, and networking boards are placed in the same chassis.

    - The difference between these and traditional multiprocessor systems is that each blade-processor board boots independently and runs its own operating system.

    - Some blade-server boards are multiprocessor as well, which blurs the lines between types of computers.

    - These servers consist of multiple independent multiprocessor systems as summarisation

Clustered Systems

- **Clustered systems** differ from the multiprocessor systems described so far
- **Clustered systems** are composed of two or more individual systems or nodes joined together.
- **Clustered systems** are considered **loosely coupled.**
- Each node may be a single processor system or a multicore system.
- The general accepted definition is that clustered computers share storage and are closely linked via a local-area network LAN or a faster interconnect, such as InfiniBand
- **Clustering** is usually used to provide **high-availability service** that is, service will continue even if one or more systems in the cluster fail.

Clustered Systems

- **Clustering** can be structured **asymmetrically** or **symmetrically**.

- In **asymmetric clustering**, one machine is in **hot-standby** mode while the other is running the applications.

    - The **hot-standby** host machine does nothing but **monitor** the active server.

    - If that server fails, the hot-standby **host** becomes the active server.

Clustered Systems

- **Clustering** can be structured **asymmetrically** or **symmetrically**.

  - In **symmetric clustering**, two or more hosts are running applications and are monitoring each other.

  - This structure is obviously more efficient, as it uses all of the available hardware.

  - However it does require that more than one application be available to run.

Clustered Systems

- Systems in **High-performance computing** environments can supply significantly greater computational power than single-processor or even SMP systems because they can run an application concurrently on all computers in the cluster.

- The application must have been written specifically to take advantage of the cluster.

- This involves a technique known as **parallelization**, which divides a program into separate components that run in parallel on individual computers in the cluster.

Clustered Systems

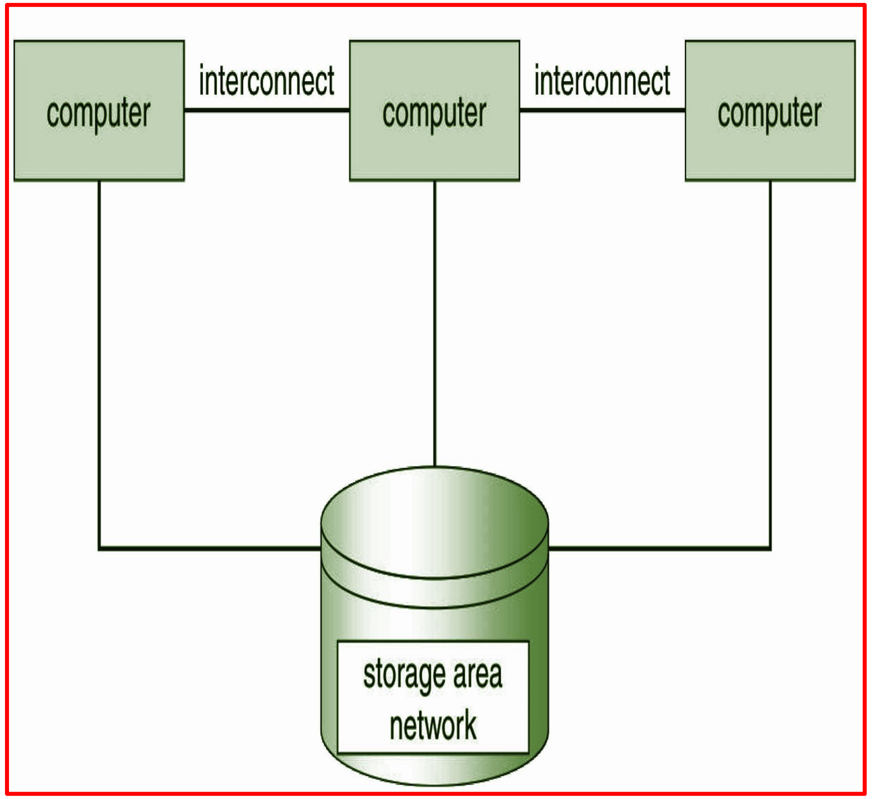
- Other forms of clusters include parallel clusters and clustering over a wide-area network (WAN)

- **Parallel clusters** allow multiple hosts to access the same data on shared storage.

- Oracle Real Application Cluster is a version of Oracle's database that has been designed to run on a parallel cluster.

- Each machine runs Oracle, and a layer of software tracks access to the shared disk.

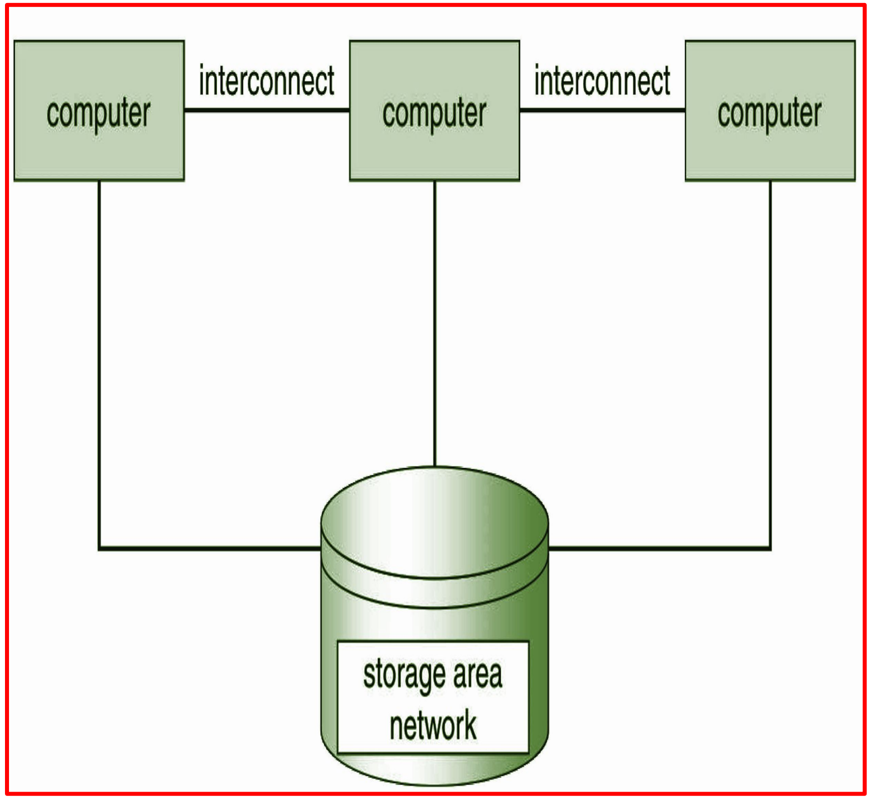
- Each machine has full access to all data in the database.



Clustered Systems

- To provide this shared access, the system must also supply access control and locking to ensure that no conflicting operations occur.

- This function, commonly known as a **distributed lock manager (DLM)**, is included in some cluster technology.



Typical Q and As

Single-Processor System	systems that have only one main CPU
Multi-processor system	parallel systems that have more than one processor in close communication
Graceful Degradation	Ability to continue providing service proportional to the level of surviving hardware
Fault Tolerant	can suffer a failure of any single component and still continue operation
Asymmetric Multiprocessing	each processor is assigned specific tasks by a "Master processor"
Symmetric Multiprocessing (SMP)	each processor performs all tasks within the OS, processors are peers, there are no peers
Uniform Memory Access (UMA)	uniform memory access, in which access to any RAM from any CPU takes the same amount of time
Non-Uniform Memory Access (NUMA)	non uniform memory access, in which some parts of memory take longer to access than other parts

Typical Q and As

Computer Core	multiprocessor chips, can be more efficient than multiple chips with single cores because on-chip communication is faster than between-chip communication
Clustered System	composed of two or more individual systems or odes joined together over a LAN or faster interconnect (Infiniband)
High-Availability Service	service continues even if one or more systems in the cluster fail
Asymmetric Clustering	one machine is in hot-standby mode while the other is running applications
Symmetric Mode (in clustering)	two ore more hosts are running applications and monitoring each other
High-Performance computing	supply significantly greater computational power than a single-processor or even SMP systems because they are capable of running an application concurrently on all computers in the cluster
Parallelization	consist of dividing a program into separate components that run in parallel on individual computers in the cluster

Typical Q and As

Distributed Lock Manager	to provide shared access to data, system must supply access control and locking to ensure that no conflicting operations occur
Multiprogramming	increases CPU utilization by organizing jobs (code and data) so that the cpu always has one to execute
Job Pool	consists of all processes residing on disk awaiting allocation of main memory
Time Sharing	cpu executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with each program while its occuring
Interactive Computer System	provides direct communication between the user and the system
Response Time	time it takes for the computer to give results

Topics Outline

- **Computer System Architecture**
  - **Single Processor System**
  - **Multiprocessor Systems**
  - **Clustered Systems**
- **Typical Q and As**

THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com