## 🌟 1. ****Single Processor System (Uniprocessor)****

### 🔹 Definition:

A system with **only one CPU (central processing unit)** that executes all instructions, including user and OS processes.

### 🔹 Characteristics:

* One processor handles all tasks.
* Simpler design and easy to manage.
* Low cost, suitable for general-purpose computing.
* Limited processing power.

### 🔹 Example:

* Early personal computers (like old desktops and laptops).
* Embedded systems in microwave ovens, calculators.

## 🌟 2. ****Multiprocessor System (Parallel Systems)****

### 🔹 Definition:

A system with **two or more processors** (CPUs) sharing a **common memory** and clock, working together to execute instructions **simultaneously**.

### 🔹 Characteristics:

* Increased throughput: multiple tasks are processed in parallel.
* Cost-effective in the long run (shared resources).
* Types:
  + **Symmetric Multiprocessing (SMP):** All CPUs are equal (same OS & memory access).
  + **Asymmetric Multiprocessing (AMP):** One master CPU controls others (specialized roles).

### 🔹 Advantages:

* Faster processing.
* Higher reliability (if one fails, others continue).
* Efficient resource utilization.

### 🔹 Example:

* Modern desktops with multi-core CPUs (quad-core, octa-core).
* High-performance servers.

## 🌟 3. ****Clustered Systems****

### 🔹 Definition:

A system where **two or more computers (nodes)** work together, connected via a **local network**, and act as a **single system** to users.

### 🔹 Characteristics:

* Each node is a separate computer with its own memory and CPU.
* Used for **high availability** and **load balancing**.
* Often used in **server farms** or **cloud services**.

### 🔹 Types:

* **Asymmetric Clustering**: One node is active, another on standby.
* **Symmetric Clustering**: All nodes are active and share the workload.

### 🔹 Advantages:

* Redundancy (failure of one node doesn’t bring the system down).
* Scalability (add more nodes as needed).

### 🔹 Example:

* Google Search Server Farms.
* Amazon Web Services (AWS) backend.

## 🌟 4. ****Real-Time Systems****

### 🔹 Definition:

A system designed to **respond to input immediately** (within a guaranteed time), typically used for **critical applications**.

### 🔹 Characteristics:

* Time constraints are strict.
* Used where **timing is crucial**: delays could lead to failure or disaster.

### 🔹 Types:

* **Hard Real-Time**: Missing a deadline = system failure.  
  Example: Airbag deployment system in cars.
* **Soft Real-Time**: Occasional deadline misses are tolerable.  
  Example: Video streaming, online gaming.

### 🔹 Applications:

* Robotics
* Industrial automation
* Flight control systems
* Medical equipment

## 🌟 5. ****Distributed Systems****

### 🔹 Definition:

A system where **multiple independent computers** (often geographically spread out) are connected via a **network**, working together as a single system.

### 🔹 Characteristics:

* Each system (node) has its **own memory and CPU**.
* Communication happens through **message passing**.
* Appears as a single system to users.

### 🔹 Advantages:

* Fault tolerance: if one node fails, the rest keep running.
* Resource sharing across the network.
* Scalability.

### 🔹 Example:

* Internet itself.
* Blockchain networks.
* Distributed databases (like Google Spanner, Cassandra).

## 📊 Comparison Table:

| **Architecture** | **CPUs/Nodes** | **Memory Type** | **Communication Method** | **Use Case Example** |
| --- | --- | --- | --- | --- |
| Single Processor | 1 CPU | Shared | Internal | Simple PCs, calculators |
| Multiprocessor | 2+ CPUs | Shared | Internal | Servers, workstations |
| Clustered | 2+ Computers | Each has own | Network | Cloud systems, web hosting |
| Real-Time | 1+ CPUs | Varies | Internal or Network | Robotics, automotive systems |
| Distributed | 2+ Computers | Each has own | Network (message passing) | Internet, Google systems |