**Synchronization in Distributed Systems**

[Distributed System](https://www.geeksforgeeks.org/design-issues-of-distributed-system/) is a collection of computers connected via the high speed communication network. In the distributed system, the hardware and software components communicate and coordinate their actions by message passing. Each node in distributed systems can share their resources with other nodes. So, there is need of proper allocation of resources to preserve the state of resources and help coordinate between the several processes. To resolve such conflicts, synchronization is used. Synchronization in distributed systems is achieved via clocks.

The physical clocks are used to adjust the time of nodes.Each node in the system can share its local time with other nodes in the system. The time is set based on UTC (Universal Time Coordination). UTC is used as a reference time clock for the nodes in the system.

The clock synchronization can be achieved by 2 ways: External and Internal Clock Synchronization.

1. **External clock synchronization** is the one in which an external reference clock is present. It is used as a reference and the nodes in the system can set and adjust their time accordingly.
2. **Internal clock synchronization** is the one in which each node shares its time with other nodes and all the nodes set and adjust their times accordingly.

There are 2 types of clock synchronization algorithms: Centralized and Distributed.

1. **Centralized** is the one in which a time server is used as a reference. The single time server propagates its time to the nodes and all the nodes adjust the time accordingly. It is dependent on single time server so if that node fails, the whole system will lose synchronization. Examples of centralized are- Berkeley Algorithm, Passive Time Server, Active Time Server etc.
2. **Distributed** is the one in which there is no centralized time server present. Instead the nodes adjust their time by using their local time and then, taking the average of the differences of time with other nodes. Distributed algorithms overcome the issue of centralized algorithms like the scalability and single point failure. Examples of Distributed algorithms are – Global Averaging Algorithm, Localized Averaging Algorithm, NTP (Network time protocol) etc.

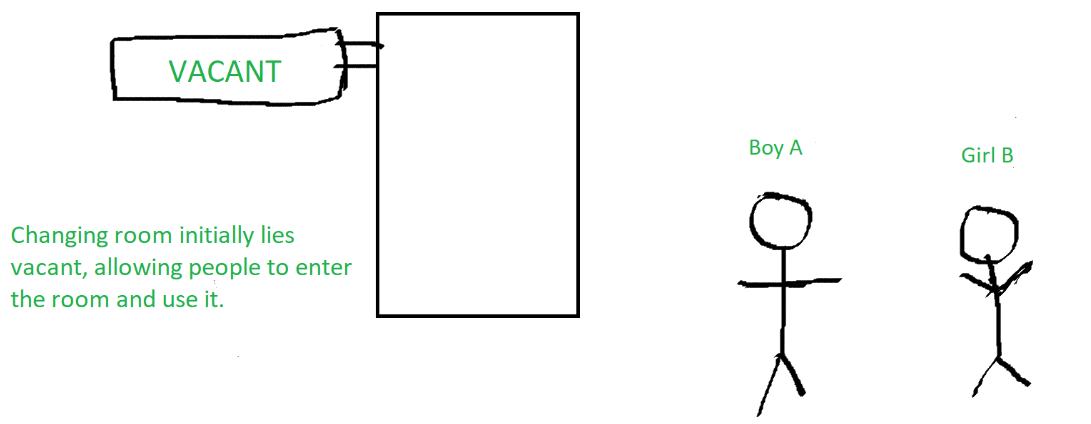
**Mutual Exclusion in Synchronization**

* During concurrent execution of processes, processes need to enter the [critical section](https://www.geeksforgeeks.org/g-fact-70/) (or the section of the program shared across processes) at times for execution. It might so happen that because of the execution of multiple processes at once, the values stored in the critical section become inconsistent. In other words, the values depend on the sequence of execution of instructions – also known as a [race condition](https://practice.geeksforgeeks.org/problems/what-is-race-condition). The primary task of process synchronization is to get rid of race conditions while executing the critical section.   
  This is primarily achieved through [mutual exclusion](https://practice.geeksforgeeks.org/problems/what-is-mutual-exclusion).

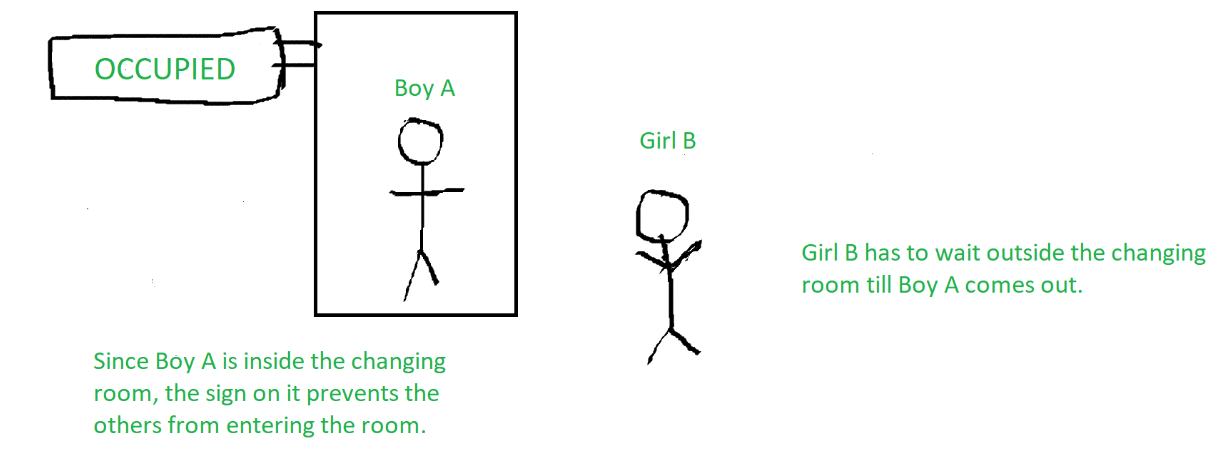
**Mutual exclusion** is a property of [process synchronization](https://www.geeksforgeeks.org/introduction-of-process-synchronization/) which states that “no two processes can exist in the critical section at any given point of time”. The term was first coined by Dijkstra. Any process synchronization technique being used must satisfy the property of mutual exclusion, without which it would not be possible to get rid of a race condition.

To understand mutual exclusion, let’s take an example.

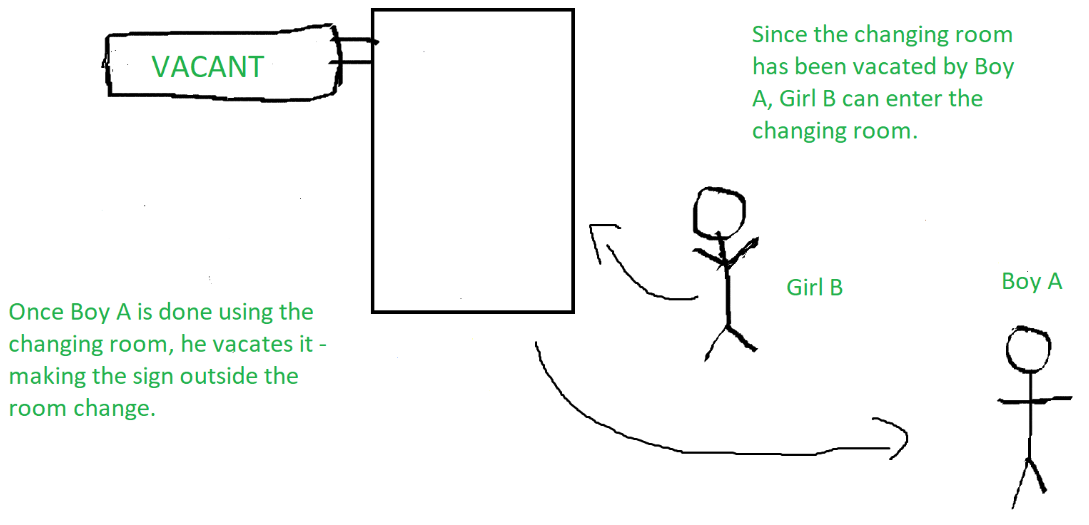
**Example:**   
In the clothes section of a supermarket, two people are shopping for clothes.



Boy A decides upon some clothes to buy and heads to the changing room to try them out. Now, while boy A is inside the changing room, there is an ‘occupied’ sign on it – indicating that no one else can come in. Girl B has to use the changing room too, so she has to wait till boy A is done using the changing room.



Once boy A comes out of the changing room, the sign on it changes from ‘occupied’ to ‘vacant’ – indicating that another person can use it. Hence, girl B proceeds to use the changing room, while the sign displays ‘occupied’ again.



The changing room is nothing but the critical section, boy A and girl B are two different processes, while the sign outside the changing room indicates the process synchronization mechanism being used.