* Concurrent processes:

Independent processes are easier to manage, since they do not share any resources among them. The problem arises when the processes are interacting, that is, when they share some data structures or need to communicate. The root problem here is the concurrent processing.

In a single-processor multiprogramming system, processes are interleaved in time to yield the appearance of simultaneous execution. In a multi-processor system, it is possible not only to interleave the execution of multiple processes but also to overlap them. The following difficulties arise:

- The sharing of global resources is fraught with peril.
- It is difficult for the OS to manage the allocation of resources optimaly.
- It becomes very difficult to locate a programming error because result are typically not deterministic and reproducible.

* Race Condition:

A race condition occurs when multiple processes or threads read and write data items so that the final result depends on the order of execution of instructions in the multiple processes.

For eg: The two processes P1 and P2, share the global variable 'a'. At some point in its execution, P1 updates a to the value of 1, and at some point in its execution, P2 updates a to the value of 2. Thus, the two tasks are in a race to write variable 'a'. In this example the 'loser' of the race (the process that updates last) determines the final value of 'a'.

→ Operating System Concerns:

What design & management issues are raised by the existence of concurrency? We can list the following concerns:

1. The OS must be able to keep track of the various processes. This is done with the use of process control blocks.
2. The OS must allocate & deallocate various resources for each active process. These resources include Processor time, Memory, Files, & I/O devices.
3. The OS must protect the data & physical resources of each process against unintended interference by the other processes. This involves techniques that relate to memory, files, & I/O devices.
4. The functioning of a process, & the output it produces, must be independent of the speed at which its execution is carried out relative to the speed of other concurrent processes.

- To understand how the issue of speed independence can be addressed, we need to look at the ways in which processes can interact.

→ Process Interaction :- We can classify the ways in which processes interact on the basis of the degree to which they are aware of each other's existence.

- Processes unaware of each other.
- Processes indirectly aware of each other
- Processes directly aware of each other.

⇒ Competition among Processes for Resources:

Concurrent processes come into conflict with each other when they are competing for the use of the same resources. In the case of competing processes three control problems must be faced.

- Mutual Exclusion
- Deadlock
- Starvation.