Transaction concurreny control can be done by scheduling.

Schedules:-

🡪A series of operation performed from one transaction to another

🡪Schedule is used to preserve order of operations in each individual transactions.

Type of scheduling

1.Serial Schedule

2.Non-Serial Schedule

3.Serializable Schedule

🡪Serial Scheduling

One transaction is executed completely before starting another transaction

So order of two transaction T1 and T2,T1 executes first then,

T1 🡪 T2

**For set of n transaction n! schedules can be possible**

🡪Non Serial Scheduling

Here instruction of transaction will be executed concurrently

**So total no of schedules can be found be how many process in each transaction done ,**

**Suppose T1 has 4 ,T2 has 2 and T3 has 3 process**

**So total schedules can be**

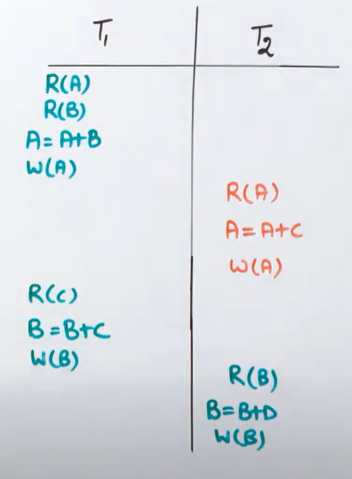
**((4+2+3)! )/(4!+2!+3!)**

**Serial is 3! So non serial can be total\_schedules-serial\_schedules**

**🡪Serializable Scheduling**

**When non serial schedule matche with serial schedule then it is called as Serializable Scheduling.**

**Eg**

** suppose A=500,B=600,C=700,D=800**

**In serial Scheduling**

|  |  |
| --- | --- |
| **T1** | **T2** |
| **500** | **1100** |
| **600** | **1800** |
| **1100** | **1800** |
| **1100** |  |
|  | **1300** |
| **700** | **2100** |
| **1300** | **2100** |
| **1300** |  |

**T1🡪T2**

**So T2 values are 1800 and 2100**

**In non serial Scheduling**

|  |  |
| --- | --- |
| **T1** | **T2** |
| **500** |  |
| **600** |  |
| **1100** |  |
| **1100** | **1100** |
|  | **1800** |
|  | **1800** |
| **700** |  |
| **1300** |  |
| **1300** |  |
|  | **1300** |
|  | **2100** |
|  | **2100** |
|  |  |

**T1🡪T2🡪T1🡪T2**

**SO T2 values are 1800 and 2100**

Both serial and non serial values matches so this is a serializable schedules.

Serializable scheduling can be tested whether non serial = serial schedule value by precedence graph.

If graph is not cyclic it mean it is serializable .

To test using precedence graph some rules r there as follows

2 transaction atleast required for scheduling

Get an edge if

R(Q) = W(Q)

W(Q) = R(Q)

W(Q) = W(Q)

**Note:- only one time check allowed**

Else if

R(Q) = R(Q) then no edge

**View and conflict scheduling link:**

**https://www.geeksforgeeks.org/difference-between-conflict-and-view-serializability/**