

- ⑥ Timestamp Ordering :- It is used to order the transactions based on their timestamps. The order of transaction is nothing but ascending order of the transaction creation.
- The priority of the order transaction is higher that's why it executes first. To determine the timestamp of the transaction, this protocol uses system time or logical counter.
 - The lock-based protocol is used to manage the order between conflicting pairs among transactions at the execution time. But timestamp based protocol start working as soon as a transaction is created.
 - Let's assume there are two transactions T_1 & T_2 . Suppose T_1 has entered the system at 007 times and transaction T_2 has entered the system at 009 times. T_1 has the higher priority, so it executes first as it is entered the system first.
 - Timestamp Ordering protocol also maintain the timestamp of last read and write operation on a data.

Protocol :- Check the following condition whenever a transaction T_i issues a Read(x) operation:

- If $W_TS(x) > TS(T_i)$ then operation is rejected.
- If $W_TS(x) \leq TS(T_i)$ then operation is executed.
- Timestamps of all the data items are updated.

- ii) Check the following condition whenever a transaction T_i issues a $Write(x)$ operation
- If $TS(T_i) < R_TS(x)$ then the operation is rejected
 - If $TS(T_i) < W_TS(x)$ then the operation is rejected
- And T_i is rolled back otherwise the operation is executed.

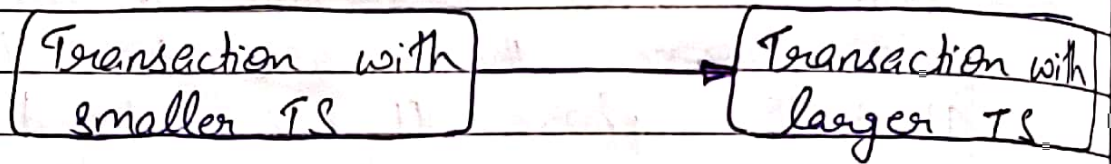
Where, $TS(T_i)$ denotes timestamp of T_i .

$R_TS(x)$ denotes read time stamp of data x

$W_TS(x)$ denotes write time-stamp of data x

Advantages And Disadvantages

- It ensures serializability since the precedence graph is given as



- It ensures freedom from deadlock that means no transaction ever waits.
- But the schedule may not be recoverable and may not even be cascade free.