

Chapter 21

Transaction Processing

References/Resources:

R. Elmasri, S.B. Navathe, “Fundamentals of Database Systems”,
6th Edition, Pearson Education

Transaction processing systems are systems with large databases and hundreds of concurrent users executing database transactions.

Examples: airline reservations, banking, online retail purchase

Classify the database systems according to the no. of users who can use the system concurrently

Single User

At most one user at a time can use the system.
Eg: personal computer

Multi User

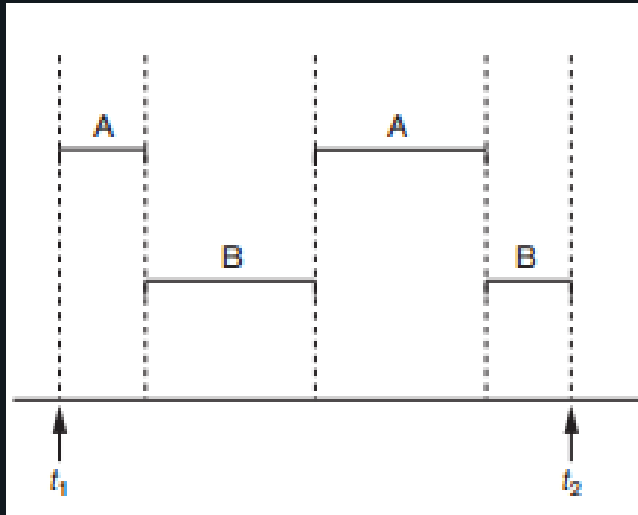
Many users access the system concurrently
Eg: banks

Multiple users can access the database because of the concept of multiprogramming.

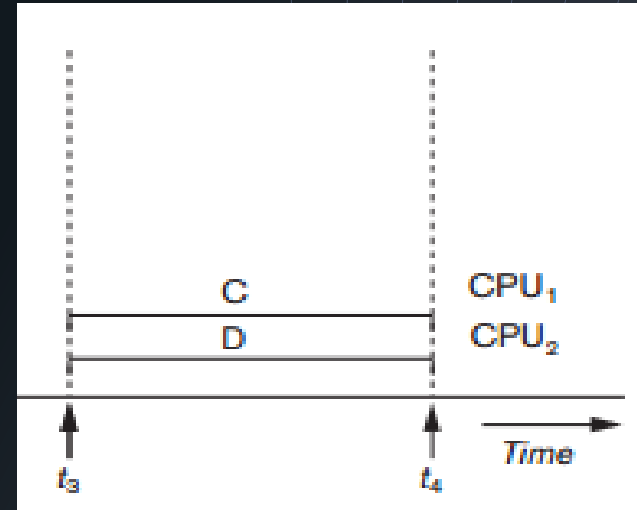
Two Types of processing of concurrent transactions

**Interleaved
processing**

**Parallel
processing**



Eg: With a single processor, process A and B executing concurrently in an interleaved fashion

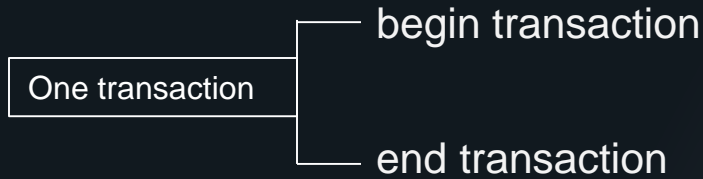


Eg: With two processor, process C and D executing concurrently in a parallel fashion

Concurrency control in database is developed in terms of interleaved concurrency

Transaction: is an executing program that forms a logical unit of database processing. A transaction includes one or more database access operations—these can include insertion, deletion, modification, or retrieval operations.

Transaction boundaries: explicit **begin transaction** and **end transaction** statements in an application program.



Two types of transactions: Read-only Transaction and Read-Write Transaction

Database: collection of named data items.

Granularity: size of data item is called its granularity

A transaction can include:

`read_item(X)` : Reads a database item named X into a program variable also named as X.

`write_item(X)` : Write the value of program variable X into the database item named X.

Steps for executing a read_item(X)

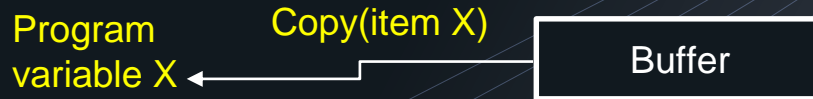
1. Find the address of the disk block that contains item X.



2. Copy that disk block into a buffer in main memory (if that disk block is not already in some main memory buffer).

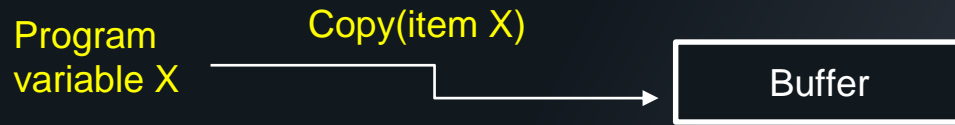


3. Copy item X from the buffer to the program variable named X.



Steps for executing a write_item(X)

1. Find the address of the disk block that contains item X .
2. Copy that disk block into a buffer in main memory (if that disk block is not already in some main memory buffer).
3. Copy item X from the program variable named X into its correct location in the buffer.

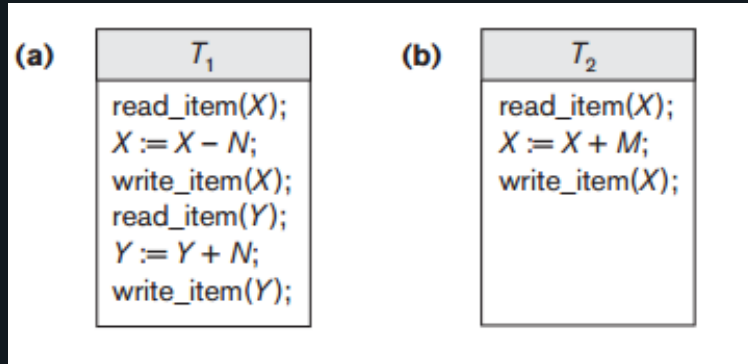


4. Store the updated block from the buffer back to disk (either immediately or at some later point in time).



Why concurrency control is needed with an airline reservation database example

A record is stored for each airline flight. Each record includes the no. of reserved seats on that flight as a named data item.



T_1 and T_2 are two transactions.

X: no. of reserved seats (first flight)

Y: no. of reserved seats (second flight)

T_1 transfers N reservation from first flight to second flight

T_2 reserves M seats on the first flight

Eg: X = 500, Y = 700, N = 100 and M = 200

T_1 transaction
read_item(500)
X = 500 - 100
X = 400
write_item(400)
read_item(700)
Y = 700 + 100
Y = 800
write_item(800)

T_2 transaction
read_item(400)
X = 400 + 200
X = 600
write_item(600)

Types of problems arises due to concurrent transaction

