M. Maheeth Reddy
1801CS31

Answer for 11th question

Ans 11 We could use an EDGE LOCK PROTOCOL to improve the concurrency of tree based protocol.

Transforming a Graph Protocol to Edge Protocol:

Consider a tree T on which transaction T_i operates only through exclusive locks (X-locks), through following protocol

T_i can request a lock on item A only when,

1. A is the first item locked by T_i or the parent of A is locked in X-mode by T_i
2. A is not yet locked by T_i
3. A vertex can be unlocked at any time.

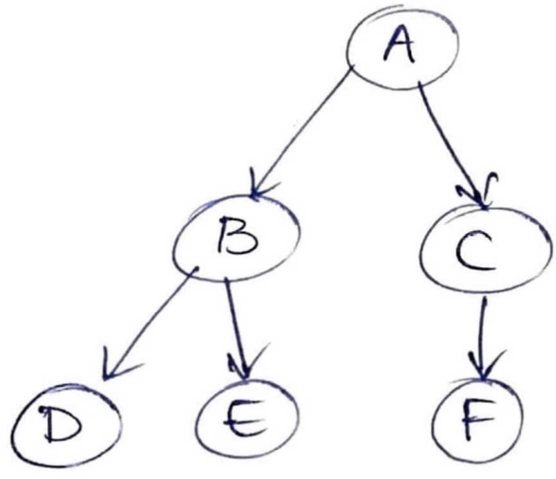
The edge lock protocol for the above tree protocol is

T_i can lock edge (A,B) in exclusive (EX-mode) lock only when,
1. (A,B) is the first edge to be locked by T_i or T_i has locked parent of (A,B)
2. (A,B) has not been previously locked by T_i

T_i can lock item A in X-mode if it holds a lock on the incoming edge (Q, A) where Q is parent of A.

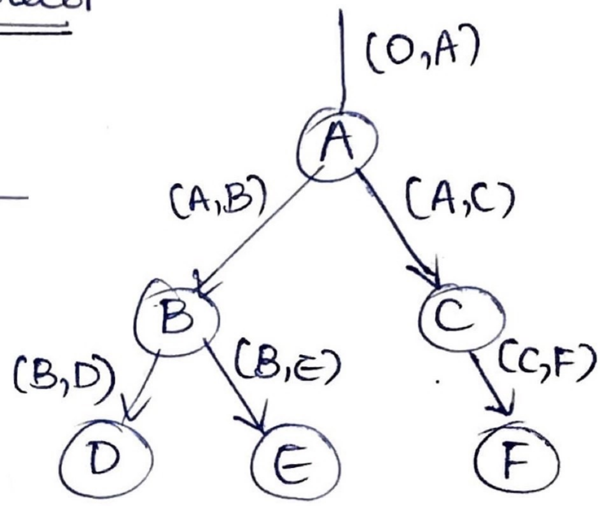
Example Schedule: Tree Protocol

T_1	T_2	T_3
Lock-X(A)		
	Lock-X(B)	
	Unlock(B)	
		Lock-X(C)
		Unlock(C)
Lock-X(B)		
Lock-X(D)		
Lock-X(C)		
Lock-X(F)		
Unlock(F)		
		Lock-X(F)
		Unlock(F)
Unlock(C)		
Unlock(D)		
Unlock(B)		



Transaction T_1 needs to work with only D and F. For that purpose, it not only needs to lock D, F but also B, C as per tree protocol. But B, C have been locked by T_2, T_3 respectively. Hence T_1 has to wait for B, C though it doesn't necessarily need them. This increases the waiting time and hence decreases concurrency. This drawback of Tree protocol is overcome in Edge Lock Protocol.

Example Schedule : Edge Lock Protocol



T_1	T_2	T_3
EX(O,A)		
	EX(A,B)	
X(A)		
	X(B)	
	UE(A,B)	
		EX(A,C)
		X(C)
		UE(A,C)
EX(A,B)		
EX(B,D)		
X(D)		
EX(A,C)		
EX(C,F)		
X(F)		
UE(C,F)		
U(F)		
		X(F)
		U(F)
U(A,B)		
U(D)		
U(A)		
U(O,A)		
	U(B)	
		U(C)

X → exclusive lock

EX → edge exclusive lock

U → unlock

UE → unlock edge

In this protocol, T_1 doesn't require to lock B, C to work with D, F. The constraints are actually on Edges (A, B) and (B, D) in case of D; (A, C) and (C, F) in case of F. Notice that, no matter how long T_2 locks B and T_3 locks C, they release the locks on (A, B) and (A, C) edges just in time for T_1 to take over.

The edges are basically acting like dummy data items, here.

Since, T_1, T_2, T_3 can be performed without any dependence, concurrency has improved.

This is the strategy I propose to improve the concurrency degree of tree based protocol.

Ans 1: Answer for 1^{st} question in End Term Quiz

Justification in quiz was brief. This is more detailed justification.

Average Time needed to find the block

= Access Time + Block Transfer Time

Access Time = Seek Time + Rotational Latency Time

Given, $\boxed{\text{Seek Time} = 5ms}$

Rotational Latency Time = Time taken to position head at right sector

= Time required to traverse half of track.

Given, Speed of rotation of disk = 10,000 rotations per minute

$\therefore$ Time for half revolution $= \dfrac{\frac{1}{2}}{10{,}000}$ minutes

$$= \frac{1}{\cancel{20{,}000}_{1000}} \times \cancel{60}^{3}\, s = \underline{3ms}$$

$\therefore \boxed{\text{Rotational Latency Time} = 3ms}$

Block Transfer Time

= Time for transferring block of data

= Time to transfer 2 sectors of data

Time taken for 1 rotation = $\frac{1}{10{,}000}$ minutes = $\frac{60}{10{,}000}$ s

$\left.\begin{array}{l}\text{Time taken for rotation} \\ \text{through 120 sectors}\end{array}\right\} = \frac{60}{10000}$ s

$\Rightarrow$ $\left.\begin{array}{l}\text{Time taken for rotation} \\ \text{through 2 sectors}\end{array}\right\} = \frac{1}{10{,}000}$ s = 0.1 ms

$\Rightarrow$ $\boxed{\text{Block Transfer Time} = 0.1 \text{ ms}}$

Hence, Average Time needed to find a block given the block address is $5+3+0.1$ ms = $\boxed{8.1 \text{ ms}}$

which is $\boxed{(\geq 8 \text{ and } < 10) \text{ ms}}$

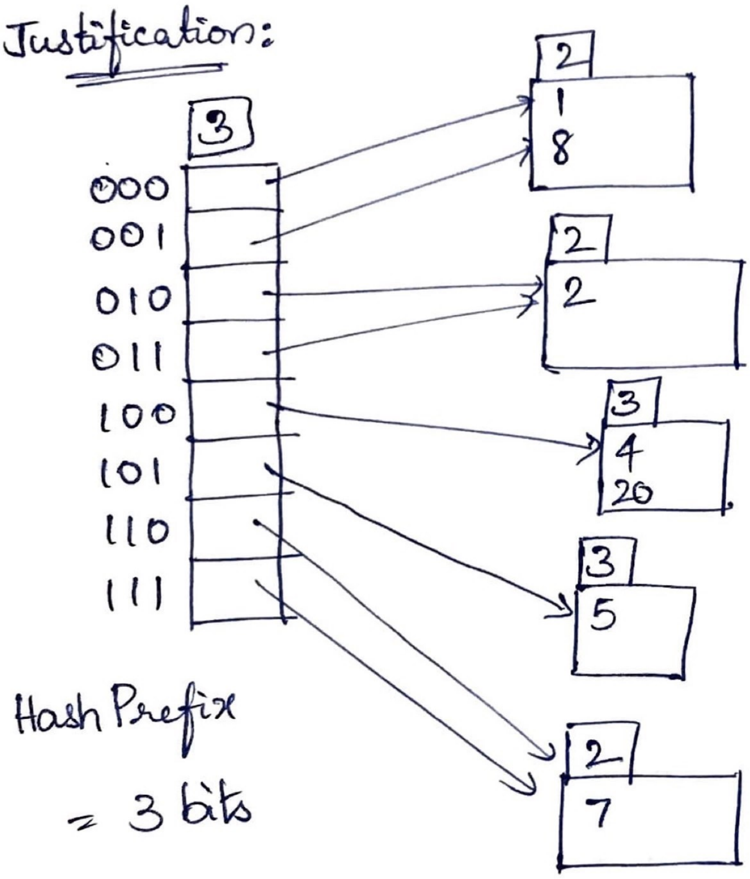
option (B)

Ans5: Answer for 5th question in EndTermQuiz

Justification given in Quiz

Schematic attached along with 11th question

Justification:



These are the buckets after the final insertion.

∴ $\boxed{\text{There are 5 buckets — option (E)}}$