

Bài 1: Cho lịch S1 như sau:

	T1	T2	T3	T4
1	RLock(A)			
2		RLock(B)		
3	Read(A)			
4			WLock(A)	
5		WLock(A)		
6				RLock(B)
7		Read(B)		
8	WLock(C)			
9		Write(A)		
10			Write(A)	
11				Read(B)
12	UnLock(A)			
13				RLock(C)
14	Write(C)			
15		UnLock(A)		
16			UnLock(A)	
17				Read(C)
18				UnLock(C)
19	UnLock(C)			
20		UnLock(B)		
21				UnLock(B)

a) Lịch S1 có khả năng tuần tự không? Nếu có thì tương đương với lịch tuần tự nào?

b) Trong lịch S1 trên, bỏ các RLock, WLock và UnLock. Biết các timestamp của các giao tác là $t(T1) = 100$, $t(T2) = 200$, $t(T3) = 300$, $t(T4) = 400$. Hãy điều khiển việc truy xuất đồng thời của các giao tác dùng kỹ thuật timestamp từng phần.

c) Trong lịch S1 trên, bỏ các RLock, WLock và UnLock. Biết các timestamp của các giao tác là $t(T1) = 100$, $t(T2) = 200$, $t(T3) = 300$, $t(T4) = 400$. Hãy điều khiển việc truy xuất đồng thời của các giao tác dùng kỹ thuật timestamp nhiều phiên bản.

3 đơn vị dữ liệu chứa A, B, C.

Bài 2: Cho lịch thao tác S2 như sau:

$r1(A)$; $r2(C)$; $w1(B)$; $r3(D)$; $r4(E)$; $w3(C)$, $w2(B)$; $w4(A)$; $w1(D)$

Giả sử rằng shared locks được yêu cầu ngay lập tức trước mỗi thao tác đọc, exclusive locks được yêu cầu ngay lập tức trước mỗi thao tác ghi và UnLock xảy ra ngay lập tức sau thao tác cuối của giao tác.

- a)** Dùng đồ thị chờ (waits-for graph) để đánh giá lịch S2 có xảy ra deadlock hay không ?
- b)** Nếu có deadlock, hãy đưa ra 1 giải pháp cụ thể để tránh và 1 giải pháp để giải quyết.