

1. feladat

Tegyük fel, hogy az ütemezőbe a következő sorrendben érkeznek az írási és olvasási kérések:

$$r_1(A), r_2(B), w_3(A), w_1(A), w_2(B), r_1(B), r_3(B).$$

Az ütemező egyszerű zárolási modellben zárkérésekkel biztosítja a konzisztenciát, úgy hogy a zárat mindig közvetlenül a kért művelet előtt adja ki. Ha egy zárkérést nem tud ekkor teljesíteni, akkor azt a tranzakciót várakoztatja, amíg felszabadul a kívánt adatelem, ekkor viszont azonnal folytatódik az a tranzakció. A zárat csak a tranzakció befejezése után engedi fel, de akkor azonnal.

Milyen zárolási ütemezést eredményez ez? (Azaz hogyan követik egymást a zárkiadások és zárelengedések?) Sorosítható-e az így kapott ütemezés, ha a sorosíthatóságról a zárolási sorozat alapján döntünk? Ha igen, adj egy ekvivalens soros ütemezést, ha nem, akkor bizonyítsd ezt be.

1. feladat megoldása

$$r_1(A), r_2(B), w_3(A), w_1(A), w_2(B), r_1(B), r_3(B).$$

A zárolási ütemezés:

$$l_1(A), l_2(B), u_2(B), l_1(B), u_1(A), u_1(B), l_3(A), l_3(B), u_3(A), u_3(B)$$

A sorosítási gráf:



Mivel ez a gráf DAG, ezért az ütemezés sorosítható. Ekvivalens soros ütemezés csak egy van: T_2, T_1, T_3

2. feladat

Tekintsük az alábbi (csak olvasásokból és írásokból álló) ütemezést:

$$r_2(A), w_3(B), r_1(A), w_2(B), w_1(C).$$

Az egyszerű tranzakciómodellt használva illesz be zárkéréseket a fenti ütemezésbe oly módon, hogy legális zárolást kapjunk és

- (a) ne kövesse mindegyik tranzakció a 2PL-t, de (a zárkérések alapján döntve) az ütemezés sorosítható legyen,
 - (b) mindegyik tranzakció kövesse a 2PL-t, de (a zárkérések alapján döntve) ne legyen sorosítható az ütemezés,
 - (c) mindegyik tranzakció kövesse a 2PL-t, és (a zárkérések alapján döntve) legyen sorosítható az ütemezés.
- (Természetesen három külön zárolást kell adni, mindegyik esetre egyet.)

2. feladat megoldása

(a) Ha mindegyik tranzakció minden művelet előtt közvetlenül kéri a zárat és rögtön utána el is engedi, akkor se T_1 , se T_2 nem követi a 2PL-t, de az ütemezés sorosítható lesz, mert a sorosítási gráfban nem lesz kör. (A sorosítási gráfban összesen két él lesz, T_3 -ból T_2 -be B miatt és T_2 -ből T_1 -be A miatt.)

(b) Ilyen nincs, mert ha minden tranzakció követi a 2PL-t, akkor az ütemezés sorosítható.

(c)

$$r_2(A), w_3(B), r_1(A), w_2(B), w_1(C).$$

Az alábbi zárolási sorozat jó lesz:

$$l_2(A), r_2(A), l_3(B), w_3(B), u_3(B), l_2(B), u_2(A), l_1(A), r_1(A), w_2(B), \\ u_2(B), l_1(C), w_1(C), u_1(A), u_1(C)$$

Itt mindhárom tranzakció követi a 2PL-t, így az ütemezés biztos sorosítható.

3. feladat

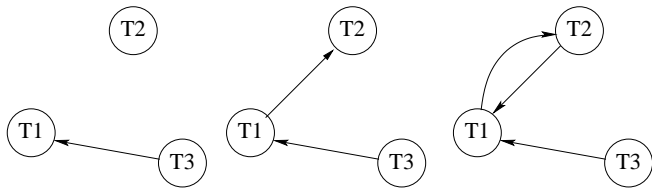
Tekintsük a következő olvasásokból és írásokból álló kérésorozatot:

$$r_1(A), r_3(A), w_2(B), w_3(C), w_1(B), r_2(A), r_3(C).$$

Hogy alakul a várakozási gráf az egyszerű zármódelben, ha a zárkérések és elengedések úgy mennek, hogy mindig a művelet előtt közvetlenül kéri a zárat a tranzakciók és a végén egyszerre engedik el?

3. feladat megoldása

$r_1(A)$, $r_3(A)$, $w_2(B)$, $w_3(C)$, $w_1(B)$, $r_2(A)$, $r_3(C)$.



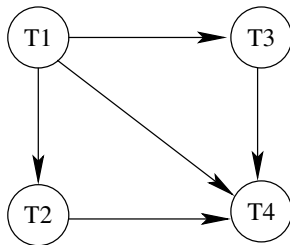
Az első várakozás A miatt következik be $r_3(A)$ -nál, a második B miatt $w_1(B)$ -nél, a harmadik A miatt $r_2(A)$ -nál. Ekkor holtpont alakul ki.

4. feladat

Az alábbi legális ütemezés négy tranzakció zárjait tartalmazza az RLOCK/WLOCK modellben. Rajzoljuk fel a sorosítási gráfot és jelöljük azt is, hogy melyik nyíl melyik adatelemek miatt került behúzásra! Sorosítható-e az ütemezés? Ha igen, milyen soros ütemezések ekvivalensek az eredeti ütemezéssel?

T_1	T_2	T_3	T_4
	WLOCK D		
RLOCK A			
WLOCK C			
UNLOCK C			
		RLOCK C	
WLOCK B			
UNLOCK B			
			RLOCK B
UNLOCK A			
	UNLOCK D		
		WLOCK A	
			RLOCK C
	WLOCK D		
			UNLOCK B
		UNLOCK C	
	RLOCK B		
		UNLOCK A	
			WLOCK A
	UNLOCK B		
			WLOCK B
			UNLOCK B
	UNLOCK D		
			UNLOCK C
			UNLOCK A

4. feladat megoldása



A gráf élei a következők miatt kerültek behúzásra: T_1 és T_3 közé A és C miatt, T_1 és T_2 közé B miatt, T_1 és T_4 közé A , B és C miatt, T_3 és T_4 közé A miatt és T_2 és T_4 közé B miatt.

Mivel ez a gráf DAG, ezért sorosítható az ütemezés. Ekvivalens ütemezés minden topologikus rendezés, azaz: T_1, T_2, T_3, T_4 és T_1, T_3, T_2, T_4 .

5. feladat

Tekintsük a T_1 , T_2 , T_3 és T_4 tranzakciók írási és olvasási kéréseiből álló

$$r_1(A), w_2(B), r_3(A), w_1(B), r_2(A), r_4(B), w_4(A), w_3(B)$$

sorozatot. Időbélyeges tranzakciókezeléssel akarjuk a sorosítható ütemezést kikényszeríteni, a tranzakciók időbélyegei: $t(T_1) = 1$, $t(T_2) = 2$, $t(T_3) = 3$, $t(T_4) = 4$. Írd le, hogy mi történik a fenti sorozat esetén, azaz mely kéréseket teljesíti az ütemező, melyeket nem, mely kérések vezetnek ABORT-hoz és add meg azt is, hogy hogyan változnak az egyes műveletek után az adategységek írási és olvasási idejei (ezek kezdetben mind nullák).

5. feladat megoldása

$$r_1(A), w_2(B), r_3(A), w_1(B), r_2(A), r_4(B), w_4(A), w_3(B)$$

Vegyük sorra a kéréseket:

$r_1(A)$ után: A kérés teljesíthető, $r(A) = 1$, minden más írási és olvasási idő marad 0.

$w_2(B)$ után: A kérés teljesíthető, $w(B) = 2$, minden más írási és olvasási marad ami volt.

$r_3(A)$ után: A kérés teljesíthető, $r(A) = 3$, minden más írási és olvasási marad ami volt.

$w_1(B)$ után: Kisebb időbélyegű tranzakció akarja írni B -t, mint $w(B)$ aktuális értéke, így ez a kérés nem teljesíthető, de ABORT nincs, egyszerűen csak T_1 nem írhat. Az írási és olvasási idők maradnak, amik voltak.

$r_2(A)$ után: A kérés teljesíthető, de $r(A)$ marad 3 és minden más írási és olvasási is marad ami volt.

$r_4(B)$ után: A kérés teljesíthető, $r(B) = 4$, minden más írási és olvasási marad ami volt.

$w_4(A)$ után: A kérés teljesíthető, $w(A) = 4$, minden más írási és olvasási marad ami volt.

$w_3(B)$ után: A kérés nem teljesíthető, mert nagyobb időbélyegű tranzakció már olvasta B -t, az ütemező ABORT T_3 -t rendel el. Az írási és olvasási idők maradnak, amik voltak.